

# Semesterarbeit

## Wireless Weather Station for High-Alpine Locations

---

*Autoren:* Pascal Häfliger  
Lino Schmid

*Betreuung:* Matthias Keller  
Dr. Jan Beutel

*Professor:* Prof. Dr. Lothar Thiele

Zürich, 11. Juni 2009



# Abstract

Within the PermaSense project, two wireless sensor networks have been deployed on two sites in Switzerland (Matterhorn since 2008, Jungfrauoch since 2009). Both installations are located on a height of over 3.500 m and are exposed to extremely high alpine weather conditions.

For supporting system operation and maintenance, knowledge of the local weather conditions on site is desirable.

This thesis presents the system integration of a commercial Vaisala WXT520 weather station into the existing PermaSense architecture. This is shown in two variants. Firstly, a solution for operating the weather station at the existing Linux-based base station is described. Second, an approach for operating the sensor with a TinyNode wireless sensor network node is also presented.

Both variants include a solution for storing the gathered data in the existing data backend. Here, Global Sensor Network (GSN) is used.

Additionally, an analysis for dimensioning a power supply based on solar power for feeding the weather station is presented.





# Kurzfassung

Im Rahmen des PermaSense-Projektes, wurden an zwei Standorten in der Schweiz funkbetriebene Sensornetzwerke installiert (Matterhorn seit 2008, Jungfraujoch seit 2009). Beide Installationen befinden sich auf einer Höhe, von über 3.000 m und sind dabei extremen, hochalpinen Wetterbedingungen ausgesetzt.

Im Hinblick auf den Betrieb und die Instandhaltung des Systems sind Kenntnisse der lokalen Wetterbedingungen erstrebenswert.

Diese Semesterarbeit beschreibt die Systemintegration einer handelsüblichen Wetterstation (Vaisala WXT520) in die bestehende Infrastruktur von PermaSense. Es werden zwei Betriebsarten aufgezeigt. Bei der ersten Lösung wird die Wetterstation mit der bereits bestehenden Basisstation betrieben. In einer weiteren Variante wird ein funkbetriebener Sensorknoten vom Typ TinyNode eingesetzt.

Beide Varianten beinhalten eine Lösung zur Speicherung der erfassten Daten in der bereits bestehenden Datenbank. In diesem Fall wird ein Global Sensor Network (GSN) eingesetzt.

Zusätzlich liegt eine Auswertung zur Dimensionierung einer Solaranlage für die Wetterstation vor.



# Plagiatserklärung

## Beiblatt zu an der ETH Zürich verfassten schriftlichen Arbeiten

Ich erkläre mit meiner Unterschrift, das Merkblatt Plagiat zur Kenntnis genommen, die vorliegende Arbeit selbständig verfasst und die im betroffenen Fachgebiet üblichen Zitiervorschriften eingehalten zu haben.

Merkblatt Plagiat: [http://www.ethz.ch/students/semester/plagiarism\\_s.de.pdf](http://www.ethz.ch/students/semester/plagiarism_s.de.pdf)

---

Ort, Datum

---

Lino Schmid

---

Ort, Datum

---

Pascal Häfliger



Die vorliegende Semesterarbeit von Lino Schmid und Pascal Häfliger ist im Frühlingssemester 2009 am Institut für Technische Informatik (TIK) des Departementes Informationstechnologie und Elektrotechnik (ITET) der ETH Zürich entstanden. Die Arbeit wurde von beiden Bearbeitern gemeinsam zu gleichen Teilen durchgeführt.

---

Lino Schmid

---

Pascal Häfliger



# Inhaltsverzeichnis

<b>1 Zusammenfassung und Ausblick</b>	<b>1</b>
1.1 Problemstellung . . . . .	1
1.2 Lösungsmethoden . . . . .	1
1.3 Ergebnisse . . . . .	2
1.4 Ungelöste Probleme . . . . .	2
<b>2 Einleitung</b>	<b>3</b>
2.1 Aufgabenstellung . . . . .	3
2.2 Systemübersicht . . . . .	6
<b>3 Sensor</b>	<b>7</b>
3.1 Bereits vorhandene Programme . . . . .	7
3.1.1 Windows . . . . .	7
3.1.2 Linux . . . . .	8
3.2 Anschluss und Heizung . . . . .	9
3.2.1 Kabel und Sonde . . . . .	9
3.2.2 8-pin M12 Connector . . . . .	9
3.2.3 USB-Kabel . . . . .	10
3.2.4 Heizung . . . . .	10
3.3 Protokolle . . . . .	11
3.3.1 Verschiedene Protokolle . . . . .	11
3.3.2 SDI-12 . . . . .	11
<b>4 Messungen und Auswertung</b>	<b>13</b>
4.1 Erklärung der Messungen und des Messaufbaus . . . . .	13
4.2 Standby . . . . .	14
4.2.1 Standby SDI-N . . . . .	14
4.2.2 Standby SDI-C . . . . .	15
4.2.3 SDI-N Standby bei -25 °C . . . . .	16
4.3 Regenmessung . . . . .	17
4.4 Ein-/Ausschalten . . . . .	18
4.5 Messung Heizung . . . . .	19
4.5.1 Versuchsaufbau Kältekammer . . . . .	19
4.5.2 Heizung . . . . .	19
4.6 Messkommandos . . . . .	23
4.6.1 Windmessung . . . . .	23

4.6.2	Vergleich Wind-, Regen- Temperatur- und Supervisormessung . . . . .	25
4.7	Auswertung . . . . .	27
4.7.1	Einschalten . . . . .	27
4.7.2	Berechnung Verbrauch . . . . .	29
<b>5</b>	<b>Auswahl</b>	<b>35</b>
5.1	Solarzelle . . . . .	36
5.2	Laderegler . . . . .	37
5.3	Akku . . . . .	38
<b>6</b>	<b>Entlade-/Aufladeprozess</b>	<b>39</b>
6.1	Simulation Energieverbrauch ohne zusätzliche Funktionen verglichen mit Regen	40
6.2	Ladung im Akku bei unterschiedlichen Ladeströmen . . . . .	41
6.3	Vergleich Regen/Heizen . . . . .	42
<b>7</b>	<b>Software</b>	<b>43</b>
7.1	Fest einkompilierte Parameter . . . . .	43
7.2	Prototyp (auf Gumstix) . . . . .	44
7.2.1	Konfigurierbare Parameter . . . . .	44
7.2.2	Ablauf . . . . .	46
7.2.3	Softwarestruktur . . . . .	47
7.2.4	GSN . . . . .	49
7.2.5	Kommandozeilen Parameter . . . . .	51
7.3	Endsystem (auf TinyNode) . . . . .	52
7.3.1	Konfigurierbare Parameter . . . . .	52
7.3.2	Ablauf . . . . .	52
7.3.3	GSN . . . . .	55
7.4	Befehle für die Wetterstation . . . . .	56
<b>8</b>	<b>Zusammenbau/Inbetriebnahme</b>	<b>57</b>
<b>A</b>	<b>Pflichtenheft</b>	<b>59</b>
A.1	Zielbestimmung . . . . .	59
A.1.1	Musskriterien . . . . .	59
A.1.2	Wunschkriterien . . . . .	60
A.1.3	Abgrenzungskriterien . . . . .	60
A.2	Produkteinsatz . . . . .	60
A.2.1	Anwendungsbereiche . . . . .	60
A.2.2	Betriebsbedingungen . . . . .	60
A.3	Produktumgebung . . . . .	61
A.3.1	Allgemein . . . . .	61
A.3.2	Prototyp . . . . .	61
A.3.3	Endsystem . . . . .	61
A.4	Produktfunktionen . . . . .	61
A.4.1	Konfigurationsdatei [P] . . . . .	61
A.4.2	Funktionen . . . . .	61
A.5	Produktdaten . . . . .	63



---

A.6	Produktleistungen . . . . .	63
A.7	Benutzeroberfläche . . . . .	63
A.8	Energieversorgung . . . . .	64
A.8.1	Solarzelle . . . . .	64
A.8.2	Akku . . . . .	64
A.9	Ergänzungen . . . . .	64
<b>B</b>	<b>CD Inhalt</b>	<b>65</b>
	<b>Bibliography</b>	<b>67</b>



# Kapitel 1

## Zusammenfassung und Ausblick

Die Integration der Wetterstation geschieht schrittweise. Zuerst schreiben wir für die Basisstation ein Java-Programm, womit die Wetterstation angesteuert wird. Nachher wird in Zusammenarbeit mit Roman Lim ein Programm für das TinyNode geschrieben, so dass die Wetterstation autonom betrieben werden kann. Als nächster Schritt, wird die Wetterstation auf dem Dach des ETH-Gebäudes installiert und getestet. Später wird sie in angepasster Form auf dem Matterhorn installiert.

### 1.1 Problemstellung

- Der Schwerpunkt der Arbeit ist die Systemintegration. Die Wetterstation soll auf allen Ebenen in das bestehende System integriert werden, angefangen bei der Mechanik bis zum Speichern der Daten im GSN<sup>1</sup>.
- Gewisse Teile, die für das Vorhaben benötigt werden, sind noch nicht vorhanden (Software und Hardware).
- Eine weitere Herausforderung ist die hochalpine Umgebung, in welcher die Wetterstation eingesetzt wird.

### 1.2 Lösungsmethoden

- Die Systemintegration erfolgt in Teilschritten. Zuerst wird eine Java-Software für den Gumstix geschrieben, womit der Wettersensor angesteuert werden kann. Gleichzeitig wird der Energieverbrauch des Sensors ausgemessen. Für das Endsystem muss schliesslich ein Programm für das TinyNode geschrieben und ein mechanischer Aufbau für die Solaranlage angefertigt werden.
- Um die Daten ins GSN transportieren zu können, werden ein Wrapper und vier virtuelle Sensoren implementiert.
- Anhand der Energiemessungen werden eine Solarzelle, ein Laderegler und eine Batterie dimensioniert und ausgesucht.

---

<sup>1</sup>Global Sensor Networks (GSN) ist eine Software Middleware, die entworfen wurde um die Entwicklung von Sensor Netzwerken zu vereinfachen

### 1.3 Ergebnisse

Die erreichten Ziele sind:

- Die Wetterstation kann mit der Basisstation betrieben werden.
- Die Wetterstation kann mit dem TinyNode betrieben werden.
- Mit der gewählten Solaranlage ist die Wetterstation autonom.

Im folgenden Foto wird die Wetterstation mit einem Gumstix betrieben. Die Daten werden im GSN gespeichert und können grafisch dargestellt werden.



Abbildung 1.1: Solarzelle in Betrieb

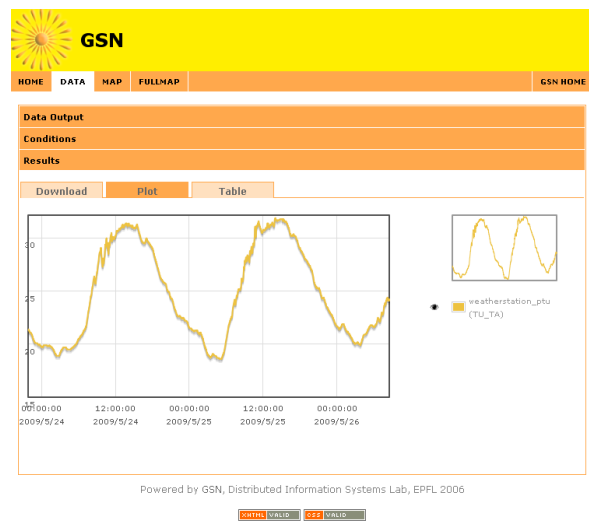


Abbildung 1.2: GSN Grafik Temperatur

### 1.4 Ungelöste Probleme

- Gehen bei der Datenübertragung zum GSN Pakete verloren, wird dies nicht bemerkt.
- Wenn die Wetterstation eingeschneit ist, ist ihre Funktionalität eingeschränkt.

# Kapitel 2

## Einleitung

### 2.1 Aufgabenstellung

#### Einleitung

Im Rahmen des PermaSense-Projektes wurden an zwei Standorten in der Schweiz funkbetriebene Sensornetzwerke installiert (Matterhorn seit 2008, Jungfrauoch seit 2009). Beide Installationen befinden sich auf einer Höhe von über 3.000 m und sind dabei extremen, hochalpinen Wetterbedingungen ausgesetzt.

Im Hinblick auf den Systembetrieb ist die Integration einer Wetterstation in die bestehende Infrastruktur geplant. Während die Wetterstation für reine Überwachungszwecke in der Nähe der vorhandenen Basisstation (Embedded Linux-System, Stromversorgung durch Solaranlage) betrieben werden kann, sind im späteren Verlauf abgelegene Orte ebenso vorstellbar.

Die hieraus entstehende Idee ist die Implementierung und Systemintegration einer autonom arbeitenden Wetterstation in das bestehende funkbetriebene Sensornetzwerk.

#### Aufgabenstellung

1. Erstellen Sie einen Projektplan und legen Sie Meilensteine sowohl zeitlich wie auch thematisch fest [1]. Erarbeiten Sie in Absprache mit dem Betreuer ein Pflichtenheft.
2. Machen Sie sich mit den relevanten Arbeiten im Bereich Sensornetze, Wetterbeobachtung mit funkbetriebenen Sensornetzwerken und Arbeiten die auf einem ähnlichen Sensor (Vaisala WXT520) basieren vertraut. Hierbei ist interdisziplinäre Arbeit (Geoscience) erwünscht. Führen Sie eine Literaturrecherche durch. Suchen Sie auch nach relevanten neueren Publikationen. Prüfen Sie welche Ideen/Konzepte Sie aus diesen Lösungen verwenden können.
3. Arbeiten Sie sich in die relevanten Technologien ein: Wettersensor, Protokolle zur Kommunikation über eine serielle Schnittstelle, Gumstix Verdex Embedded Linux System, architekturübergreifendes Erstellen von Software (Cross-Compiling), relevante Komponenten der PermaSense-Architektur (Nachrichtentypen, Global Sensor Network (GSN)).

4. Implementieren Sie einen Prototypen zur Verwendung des Wettersensors mit einer Basisstation des PermaSense Projektes (Gumstix Verdex Embedded Linux System). Folgende Komponenten werden dabei gefordert: Serielle Kommunikation mit Wettersensor, Steuerung der Funktionen über Konfigurationsdatei oder Frontend, Übermittlung und Speicherung der Daten im Global Sensor Network (GSN).
5. Eine autonome Wetterstation soll über die Anbindung der Wetterstation an einen Sensorknoten vom Typ TinyNode mit einem Sensor Interface Board (SIB) realisiert werden. Für die autonome Stromversorgung muss eine korrekt dimensionierte Solaranlage ausgewählt und eingerichtet werden.
6. Dimensionieren und entwerfen Sie eine elektrische Schaltung zur Stromversorgung des Wettersensors über eine Solaranlage. Herstellerangaben über den Energieverbrauch des Sensors können mit vorhandener Messtechnik verifiziert werden.
7. Entwickeln und erstellen Sie einen ausführlichen Plan zur Integration des Wettersensors in die vorhandene Infrastruktur. Im Vorfeld muss dazu vor allem die Menge der zu implementierenden Funktionen erörtert werden. Setzen Sie sich mit allen zur Integration notwendigen Aspekten auseinander, dies umfasst zum Beispiel auch die Integration in die vorhandenen Kommunikationsprotokolle im Sensornetzwerk. Die Integration soll beide Kommunikationswege (Senden der Messdaten an die Basis, Empfangen von Steuerungsbefehlen von der Basis) abdecken. Die gesammelten Daten sollen zuletzt ebenso im GSN gespeichert werden.
8. Implementieren Sie dieses Konzept in einem lauffähigen Prototypen. Denken Sie dabei auch an den mechanischen Aufbau der Anlage. Die Implementierung der notwendigen Software auf dem TinyNode unter TinyOS/Permadozer wird dabei von uns übernommen.
9. Dokumentieren Sie Ihre Arbeit sorgfältig mit einem Vortrag, einer kleinen Demonstration, sowie mit einem Schlussbericht.

## Durchführung der Semesterarbeit

### Allgemeines

- Der Verlauf des Projektes soll laufend anhand des Projektplanes und der Meilensteine evaluiert werden. Unvorhergesehene Probleme beim eingeschlagenen Lösungsweg können Änderungen am Projektplan erforderlich machen. Diese sollen dokumentiert werden.
- Sie verfügen über PC's mit Linux/Windows für Softwareentwicklung und Test. Für die Einhaltung der geltenden Sicherheitsrichtlinien der ETH Zürich sind Sie selbst verantwortlich. Falls damit Probleme auftauchen wenden Sie sich an Ihren Betreuer.
- Stellen Sie Ihr Projekt zu Beginn der Semesterarbeit in einem Kurzvortrag vor und präsentieren Sie die erarbeiteten Resultate am Schluss im Rahmen des Institutskolloquiums Ende Semester.
- Besprechen Sie Ihr Vorgehen regelmässig mit Ihren Betreuern. Verfassen Sie dazu auch einen kurzen wöchentlichen Statusbericht (email).

**Abgabe**

- Geben Sie zwei unterschriebene Exemplare des Berichtes spätestens am *08. Juni 2009* dem betreuenden Assistenten oder seinen Stellvertreter ab. Diese Aufgabenstellung soll vorne im Bericht eingefügt werden.
- Räumen Sie Ihre Rechnerkonten soweit auf, dass nur noch die relevanten Source- und Objectfiles, Konfigurationsfiles, benötigten Directorystrukturen usw. bestehen bleiben. Der Programmcode sowie die Filestruktur soll ausreichend dokumentiert sein. Eine spätere Anschlussarbeit soll auf dem hinterlassenen Stand aufbauen können.

**Wichtige Termine**

- 18.03.2009, 9-10 Uhr: Einführungspräsentation im TEC Group Meeting
- 02.06.2009: Abschlusspräsentation
- 08.06.2009: Abgabe des Abschlussberichtes

**Ressourcen**

- SVN Repository: <svn://svn.ee.ethz.ch/permasense-weatherstation>
- Embedded Linux System zur Entwicklung: [forecast.ethz.ch](http://forecast.ethz.ch), SSH-Zugang freigeschaltet
- Allgemeine Tipps zur Erstellung einer SA/DA/MA:  
<http://www.tik.ee.ethz.ch/~kellmatt/ThesisSurvivalGuide>

## 2.2 Systemübersicht

Abbildung 2.1 zeigt die Systemübersicht. Links, im blauen Kasten sind die im Netzwerk verteilten Sensorknoten dargestellt. Im roten Kasten in der Mitte ist die Basisstation abgebildet und im rechten, grünen Kasten ist der Server, welcher die Daten speichert.

Der Prototyp soll von der Basisstation aus gesteuert werden. Als Stromversorgung kann die bereits installierte Solaranlage mitbenutzt werden. Das Endprodukt soll durch einen TinyNode gesteuert werden, damit die Wetterstation, wie die anderen Sensoren, dezentral im Netzwerk platziert werden kann. Dazu ist noch eine Spannungsversorgung bestehend aus einer Solarzelle, einer Batterie und einem Laderegler notwendig.

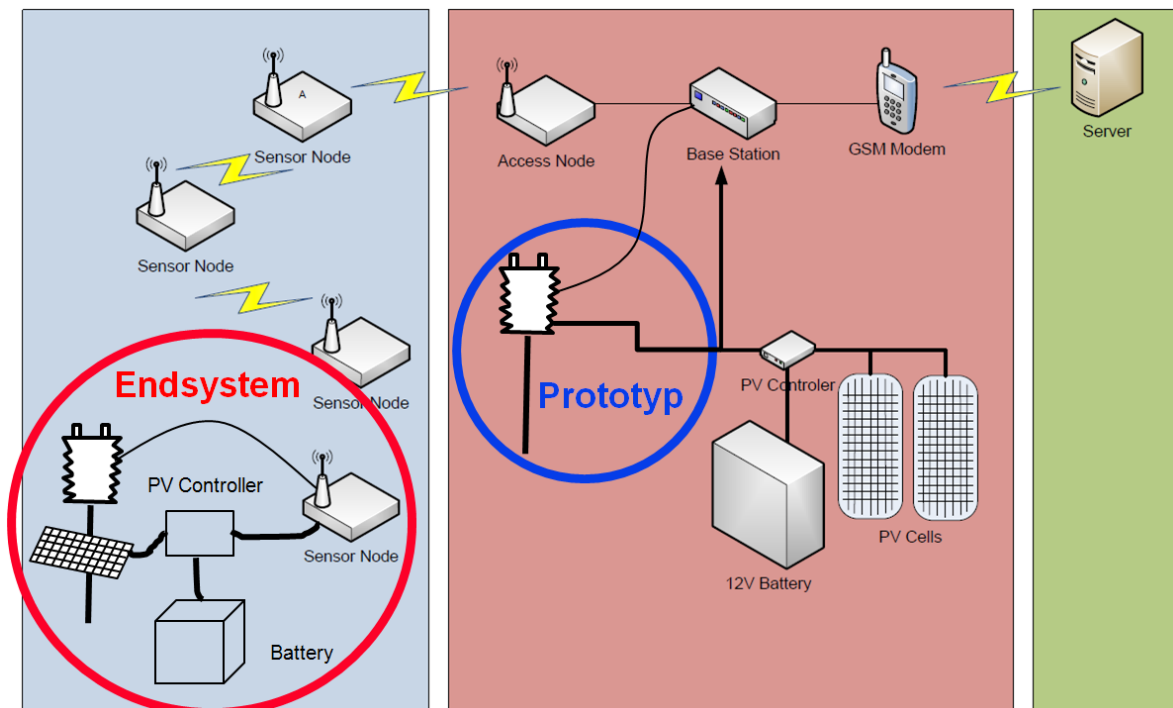


Abbildung 2.1: Systemübersicht



# Kapitel 3

## Sensor

Dieses Kapitel ist eine Zusammenfassung des Handbuches [2], welches für die Anwendung wichtige Hinweise enthält.

### 3.1 Bereits vorhandene Programme

#### 3.1.1 Windows

Von der Firma Vaisala wird bereits eine Windowssoftware mitgeliefert, das Vaisala Configuration Tool. Es besteht aus den Programmen *Tool* und *Monitor*. Das Programm *Tool* zeigt die aktuellen Werte der Sonde und die einstellbaren Parameter können verändert werden. Im Programm *Monitor* ist ersichtlich, ob eine Verbindung zur Sonde vorhanden ist. Die Meldungen dieser Sonde werden in einem Fenster aufgelistet. Werden die Standardwerte (wie z.B. die Baudrate) abgeändert und vergessen, kann mittels Service-Port (siehe Abschnitt 3.2.3 USB-Kabel) immer noch auf die Sonde zugegriffen werden. Die beiden Programme sind in der nächsten Abbildung aufgezeigt. Das *Tool* in Abbildung 3.1 und der *Monitor* in Abbildung 3.2.

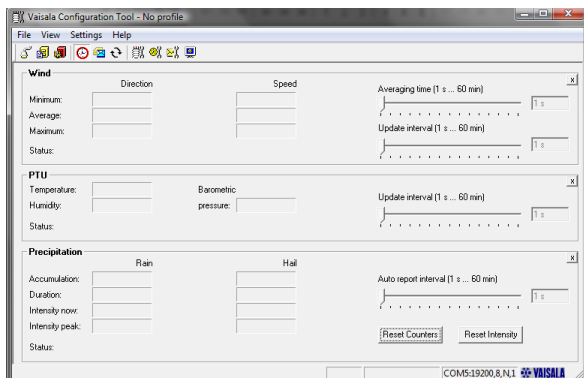


Abbildung 3.1: Vaisala Tool

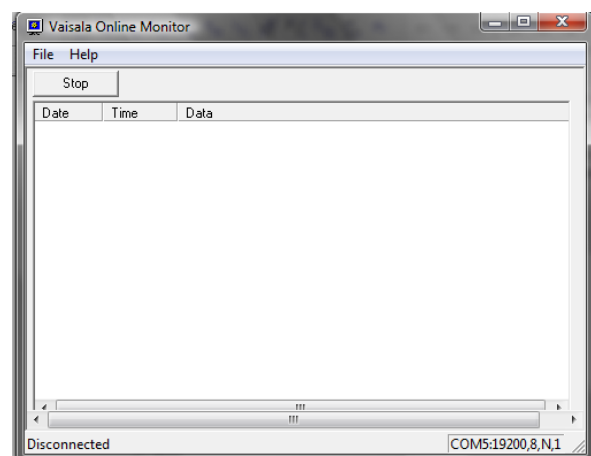


Abbildung 3.2: Vaisala Monitor

### 3.1.2 Linux

Auch für Linux ist bereits ein Programm vorhanden. Es heisst wview und wird von der Herstellerfirma nicht unterstützt. Wview ist eine Sammlung von linux/unix Dämons. Das Programm erfasst aktuelle Wetterdaten und archiviert sie in einer Datenbank. In einem konfigurierbaren Intervall ruft wview die archivierten Daten auf, generiert damit Grafiken und stellt diese auf einer Webseite. Abbildung 3.3 zeigt ein Beispiel einer solchen Webseite.

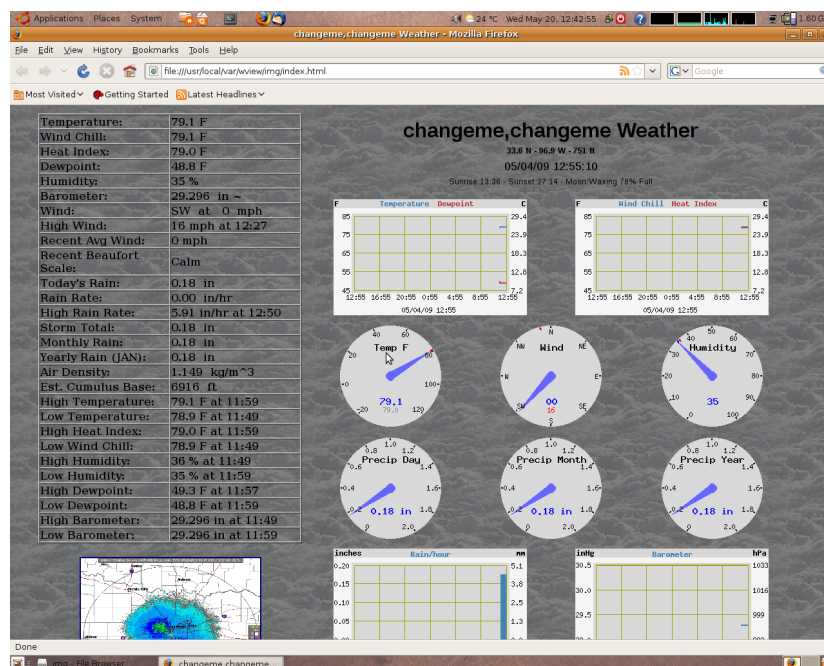


Abbildung 3.3: Linux wview

## 3.2 Anschluss und Heizung

### 3.2.1 Kabel und Sonde

Der Sensor kann verschieden angeschlossen werden. In der Abbildung 3.4 werden die wichtigsten Anschlüsse gezeigt. Zur Verfügung stehen zwei Kabel, das USB-Kabel mit dem 4-pin M8 snap-on connector und das 8-pin M12 Kabel mit unabgeschlossenem Ende. Am Ende des M12 Kabels ist ein DE-9 Stecker montiert.

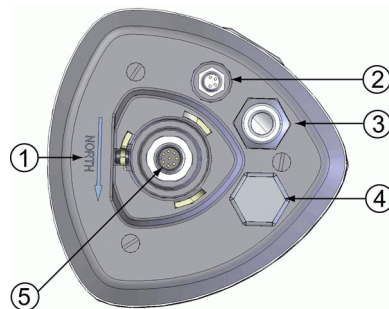


Abbildung 3.4: Anschluss Sensor

Wichtigste Anschlüsse:

2 = 4-pin M8 connector für den Service Port

5 = 8-pin M12 connector für das Datenkabel

### 3.2.2 8-pin M12 Connector

Der DE-9 Stecker des M12 Kabels ist wie folgt angeschlossen:

Tabelle 3.1: Anschluss M12 Connector

Wire Color	M 12 Pin #	RS-232	SDI-12
Blue	7	Data out (TxD)	Data in/out (Tx)
Gray	5	-	-
White	1	Data in (RxD)	Data in/out(Rx)
Green	3	GND for data	GND for Data
Pink	6	GND for Vh+	GND for Vh +
Yellow	4	Vh+ (heating)	Vh+ (heating)
Red	8	GND for Vin+	GND for Vin+
Brown	2	Vin+ (operating)	Vin+ (operating)

Es ist nur die RS-232 und die SDI-12 Schnittstelle aufgeführt (weitere Anschlussmöglichkeiten siehe [2] Seite 64). Wie aus der Tabelle 3.1 ersichtlich ist, müssen Speisung für Heizung und Sensor separat angeschlossen werden.

### 3.2.3 USB-Kabel

Das USB-Kabel wird auch als Service Kabel bezeichnet. Mit diesem Kabel kann über einen USB-Port mit der Sonde kommuniziert werden. Dabei stellt der USB Port die Energie für den Sensor zur Verfügung. Die Heizung ist beim Betrieb des Sensors über den USB-Anschluss nicht verwendbar. Die serielle Schnittstelle ist auf 19200, 8, N, 1 einzustellen. Dabei ist zu beachten, dass die Kommunikation über das M12 Kabel nicht funktioniert, wenn der Service-Port angeschlossen ist. Auf Windows kann der USB-Treiber von Vaisala mit der mitgelieferten CD installiert werden.

### 3.2.4 Heizung

Die Heizung muss separat angeschlossen werden. Die Heizung ist unter dem Niederschlagssensor und in den Windsensoren angebracht. Sie hält den Sensor in der Nähe des Gefrierpunktes ( $-4\text{ °C}$  bis  $+4\text{ °C}$ ) frei von Eis und Schnee. Ist die Heizung eingeschaltet, wird intern alle 15 Sekunden eine Messung ausgeführt. Die Heizung kann abgefragt und an- oder abgeschaltet werden (siehe [2] Seite 122). Wenn die Heizung aktiviert ist, hat sie folgende Heizkurve (Abb. 3.5). In Abbildung 3.6 ist die benötigte Heizleistung ersichtlich.

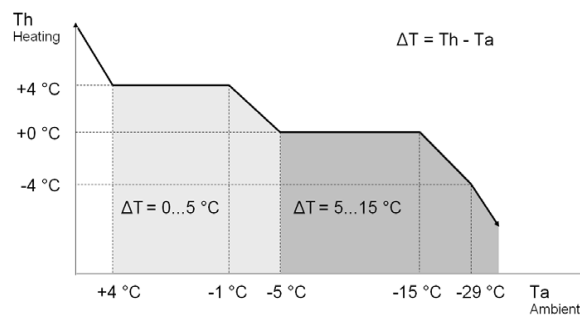


Abbildung 3.5: Heizkurve

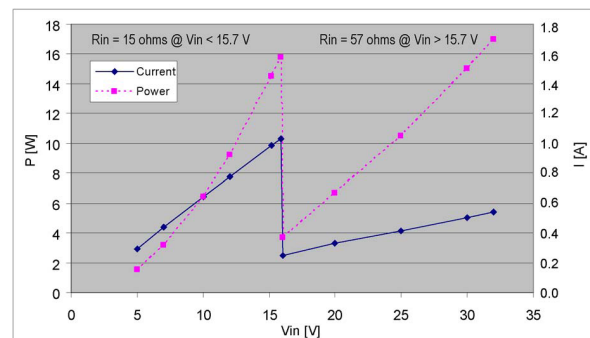


Abbildung 3.6: Heizleistung

## 3.3 Protokolle

### 3.3.1 Verschiedene Protokolle

Der Vaisala Sensor stellt für die Kommunikation verschiedene Protokolle zur Verfügung, welche unterschiedlich im Energieverbrauch und der Anwendung sind. Von Bedeutung sind das SDI-12 und RS-232 Protokoll. Eine Übersicht über die vom Hersteller zur Verfügung gestellten Protokolle sind in der Tabelle 3.2 ersichtlich.

Tabelle 3.2: Übersicht Protokolle

Serial Interface	Communication Protocols Available
RS-232	ASCII automatic and polled NMEA 0183 v3.0 automatic and query SDI-12 v1.3 and SDI-12 v1.3 continuous measurement
RS-485	ASCII automatic and polled NMEA 0183 v3.0 automatic and query SDI-12 v1.3 and SDI-12 v1.3 continuous measurement
RS-422	ASCII automatic and polled NMEA 0183 v3.0 automatic and query SDI-12 v1.3 and SDI-12 v1.3 continuous measurement
SDI-12	SDI-12 v1.3 and SDI-12 v1.3 continuous measurement

Es ist darauf zu achten, dass die Schnittstellen jeweils richtig angeschlossen sind. Auf RS-232 und SDI-12, kann direkt von der seriellen Schnittstelle eines PC's zugegriffen werden, für RS-485 und RS-422 braucht es einen Wandler. Die Standardeinstellungen des Sensors bei der seriellen Schnittstelle sind in Tabelle 3.3 zu sehen.

Tabelle 3.3: Protokoll Baudrate

Serial Interface	Serial Settings
SDI-12	1200 Baud, 7, E, 1
RS-232	19200 Baud, 8, N, 1

Das TinyNode stellt eine SDI-12 Schnittstelle zur Verfügung. Eine Anbindung per SDI-12 ist laut Vaisala die energiesparsamste Möglichkeit um den Wettersensor zu betreiben.

### 3.3.2 SDI-12

Der Vaisala Sensor stellt bei der SDI-12 Schnittstelle zwei Protokolle zur Auswahl, den SDI-12 Native mode und den SDI-12 Continuous mode. Wird der Sensor im SDI-12 Native mode betrieben, hat er den kleinsten Standbyverbrauch, laut Hersteller 1 mW. Betreibt man den Sensor im Continuous mode, verbraucht er nach Angaben des Herstellers 3 mW. Jede Messung verbraucht zusätzlich Energie. Im Continuous mode werden die Messungen in vom Benutzer einstellbaren Abständen gemacht. Die Messresultate müssen mit einem Befehl angefragt werden. Die dafür notwendigen Befehle können in [2] im Kapitel 7 nachgeschlagen werden. Der Modus SDI-12 Native und SDI-12 Continuous werden im Folgenden auch als SDI-N und SDI-C bezeichnet.



# Kapitel 4

## Messungen und Auswertung

### 4.1 Erklärung der Messungen und des Messaufbaus

Für die Dimensionierung der Komponenten muss der genaue Energieverbrauch des Vaisala WXT520 Sensors in den verschiedenen Modi bekannt sein. Die Messungen wurden mit dem Agilent Technologies N6705A DC Power Analyzer gemacht. Mit dem Power Analyzer werden Strom und Spannung aufgezeichnet. Die Grössen werden in einer Textdatei abgespeichert und anschliessend mit Matlab und Excel weiterverarbeitet. Die Resultate werden untereinander, so wie mit den Angaben des Herstellers verglichen. Auf der beigelegten CD ist die vollständige Auswertung der Messresultate aufgelistet.

#### **Messungen:**

- Kommandos
  - Wind 1 Hz/2 Hz/4 Hz Samplingrate
  - Wind mit 10/30 Sekunden Average Time
  - Temperatur
  - Regen
  - Supervisor
- Protokolle
  - SDI native
  - SDI continuous
- Einschaltvorgang
- Messung des Standbystromes
- Verschiedene Spannungen 6/12/24V
- Simulation von Regen
- Simulation von Schnee

Zur Auswahl der Messungen:

Wie vom Hersteller beschrieben, ist das SDI-12 Native Protokoll das Energiesparsamste. Der SDI-12 Continuous mode verbraucht gleich viel wie die anderen Modi. Die meisten Messungen werden im SDI-12 Native Modus durchgeführt. Zum Vergleich wurde noch ein zusätzlicher Modus ausgemessen. Die ausgemessenen Spannungen sind typisch für Akkumulatoren und liegen im Spannungsbereich der Sonde.

## 4.2 Standby

Die folgenden Auswertungen (Abb. 4.1) vergleichen den benötigten Standbystrom in Abhängigkeit vom ausgewählten Kommunikationsprotokoll, der Spannung und der Temperatur.

### 4.2.1 Standby SDI-N

Das SDI-N Protokoll zeichnet sich durch einen geringen Energieverbrauch aus. Die Sonde wird deshalb im SDI-N Modus mit verschiedenen Spannungen betrieben. Der dabei verbrauchte Strom und die Leistung werden gemessen und anschliessend verglichen.

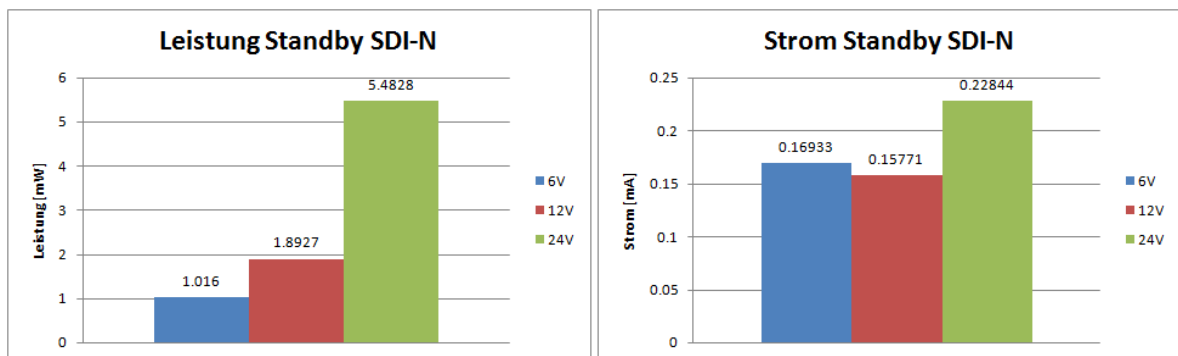


Abbildung 4.1: Standby SDI-N

Es ist ersichtlich, dass bei einer Betriebsspannung von 12 V ein Strom von  $150 \mu\text{A}$  gemessen werden kann. Dieser Standbystrom erzeugt eine Verlustleistung von ca. 2 mW. Laut Hersteller soll die Verlustleistung im Bereich von 1 mW sein. Diesen Wert erreicht man mit der Messung bei 6 V. Der genaue Verlauf des Standbystromes über 30 s kann der Abbildung 4.2 entnommen werden.

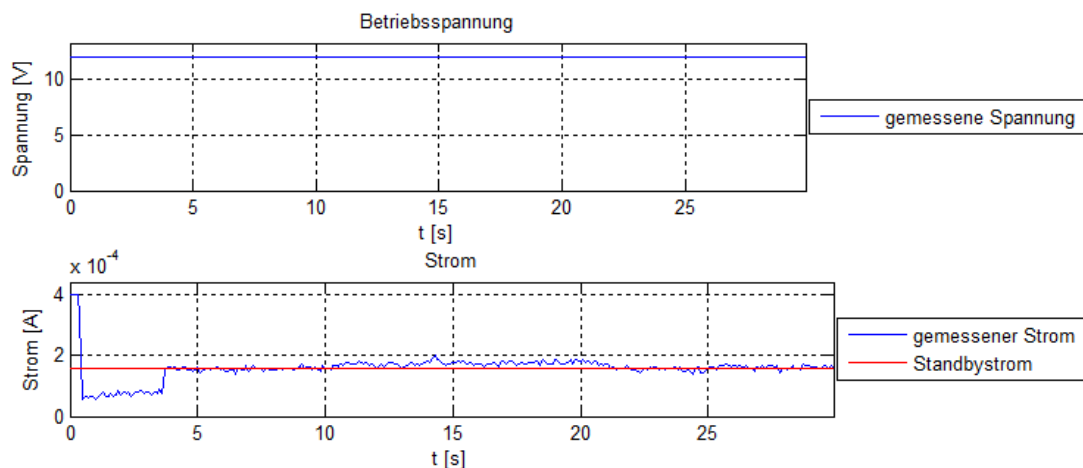


Abbildung 4.2: Messung Standbystrom bei 12 V im SDI-N Modus

In Abbildung 4.2 ist der Stromverlauf bei 12 V aufgezeigt. Die Messung wird für jeden Standbystrom gemacht. Die anderen Grafiken sind im fig-Format auf der CD zu finden. Vom



Poweranalyser werden nur Strom und Spannung gemessen, alle anderen Größen werden daraus abgeleitet. Dies gilt auch für alle nachfolgenden Messungen. Der Wert *Standbystrom* (rote Kurve) ist aus dem Mittel des gemessenen Stromes (blaue Kurve) berechnet. Wie man sehen kann, passt dieser mehrheitlich mit den Messwerten überein. Dieser Mittelwert wird auch im vorhergehenden Bild Standby SDI-N (Abb. 4.1) erörtert.

#### 4.2.2 Standby SDI-C

Um einen konkreten Vergleich zu haben, wurde der SDI-C Modus untersucht. Der SDI-C Modus sollte benutzerfreundlicher sein als der SDI-N Modus. In Abbildung 4.3 ist der Mittelwert des Standbystromes aufgezeigt:

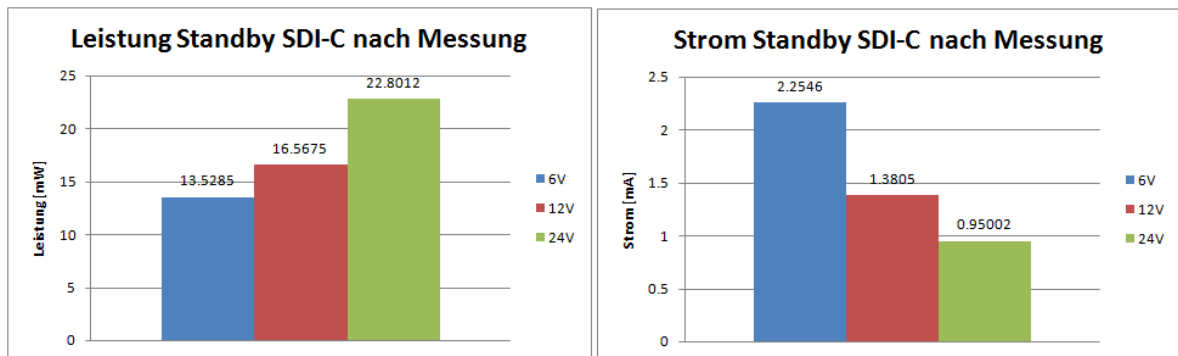


Abbildung 4.3: Standby SDI-C nach Messung

Um ein besseres Bild vom Verlauf des Stromes zu bekommen, sind die Messwerte in einer Grafik (Abb. 4.4) wiedergegeben:

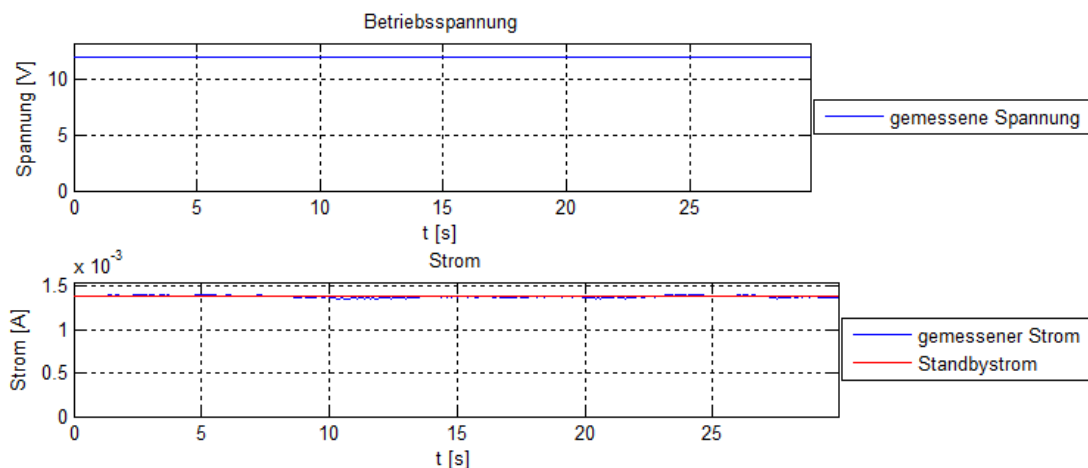


Abbildung 4.4: Standby SDI-C bei 12 V nach Messung

Bei der Messung fällt auf, dass sich der Standbystrom bei unterschiedlichen Messungen ändert. Nach dem Supervisor Befehl (0M5!) ist der Standbystrom am tiefsten. Die Messung in Abbildung 4.4 wird mit dem Windkommando (0M1!) abgeschlossen, daher der höhere

Standbystrom. Um die beiden Ströme vergleichen zu können, wird in Abbildung 4.5 der Verlauf nach dem Supervisor Befehl gezeigt, was den tieferen Stromverbrauch zur Folge hat.

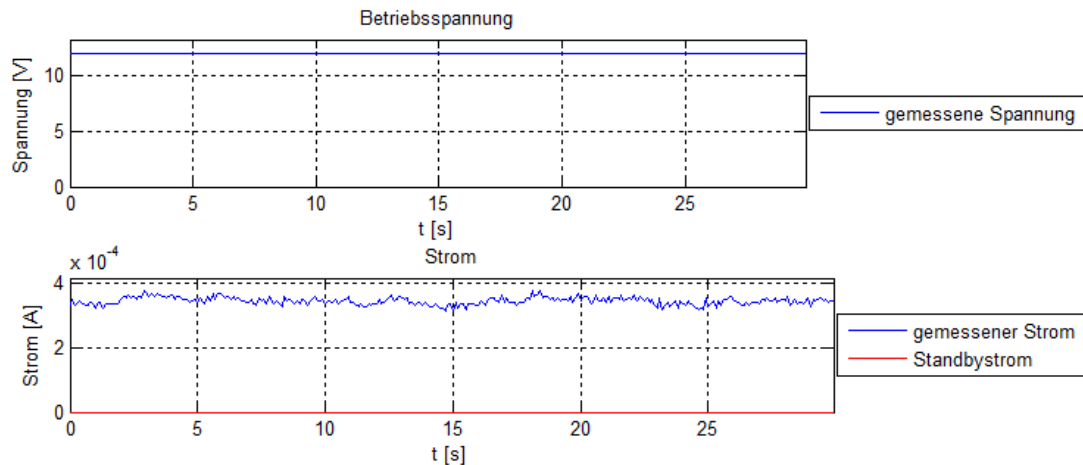


Abbildung 4.5: Standby SDI-C bei 12 V nach SUDO Messung

Wie den beiden Grafiken entnommen werden kann, ist der Standbystrom im Normalfall 0,35 mA. Bei gewissen Messungen geht der Wert aber auf 1,38 mA und würde in dieser Größenordnung bleiben, wenn keine andere Messung gemacht werden würde. Eine Abhängigkeit des Standbyverbrauchs von der Messung wurde nur im SDI-C Protokoll beobachtet, im SDI-N Protokoll ist ein solches Verhalten nicht aufgefallen.

### 4.2.3 SDI-N Standby bei -25 °C

Da sich der Sensor in hochalpinem Gebiet befindet, wurde der Standby auch bei -25 °C gemessen.

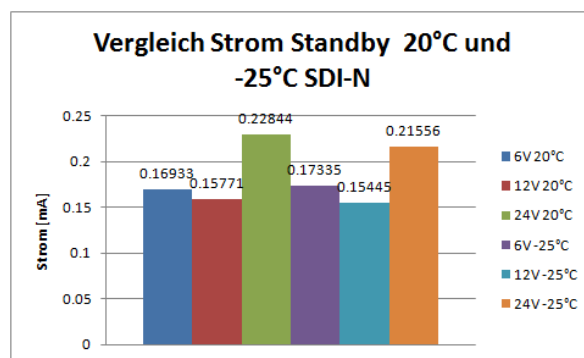


Abbildung 4.6: Messung Standby -25 °C

In Abbildung 4.6 ist der Standbystrom in Abhängigkeit der Spannung und der Temperatur dargestellt. Es zeigt sich, dass der Verbrauch bei -25 °C und bei 25 °C ungefähr gleich ist. Konkret verbraucht der Sensor 0,158 mA bei 25 °C und 12 V während er 0,154 mA bei -25 °C und 12 V verbraucht.

### 4.3 Regenmessung

Die Regenmessung verbraucht, wie der Standbymodus, dauerhaft Strom. Sie unterscheiden sich aber dadurch, dass die Regenmessung nur Strom verbraucht, wenn es regnet. Nach Herstellerangaben wird ein Strom von 0.07 mA zum Standbystrom hinzugefügt. Ein einzelner Regentropfen soll dabei einen extra Strom von 0.04 mA zum bereits fließenden Strom hinzufügen. Um eine Vorstellung zu bekommen, wie das bei mittlerem bis starkem Regenfall aussieht, wurde mit den Fingern auf den Sensor geklopft, welcher dies als Regen interpretiert. Wenn der Regen aufhört, schaltet der Sensor die automatische Regenmessung innerhalb von 10 bis 20 Sekunden ab. In Abbildung 4.7 sieht man wie der Stromverbrauch bei simuliertem Regen verläuft. Um den zusätzlichen Energieverbrauch einer Messung zu kennen, wird vom Gesamtverbrauch der Standbyverbrauch abgezogen.

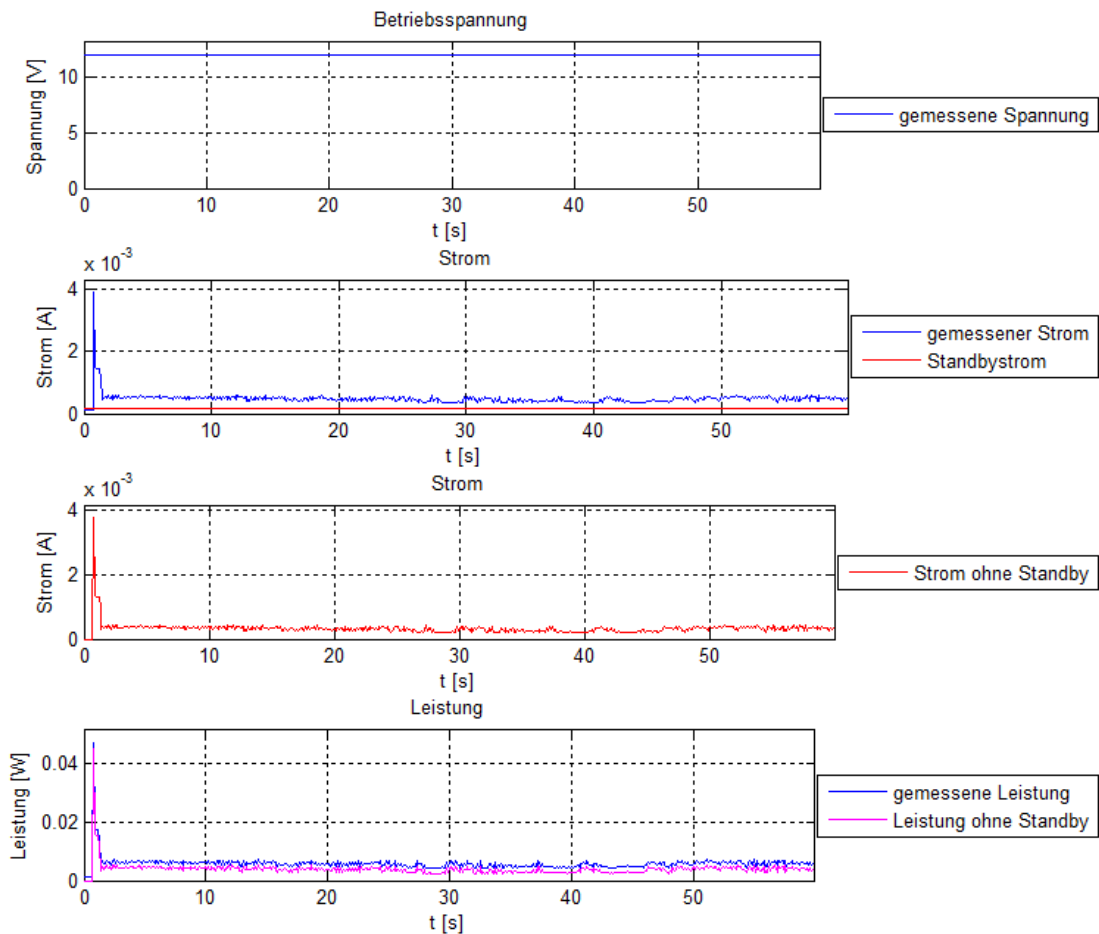


Abbildung 4.7: Spannungs- Strom- und Leistungsverlauf bei Regen

Zur genaueren Beschreibung der Grafik 4.7: Der Sensor wird über 60 Sekunden mit 12 V betrieben. Dabei wurde das Verhalten bei Regen simuliert. Das erste Fenster zeigt den zeitlichen Verlauf der Spannung. Im zweiten Fenster ist der gemessene Strom und der Standbystrom

bei 12 V sichtbar. Am Anfang gibt es eine Spitze wo der Strom kurzzeitig auf 4 mA ansteigt. Der im Schnitt zusätzlich zum Standbystrom benötigte Strom beträgt 0.321 mA, eindeutig mehr, als wenn man nur die 0.07 mA des Herstellers mit einbeziehen würde.

Bei anderen Betriebsspannungen sieht der Verbrauch bei Regen folgendermassen aus:

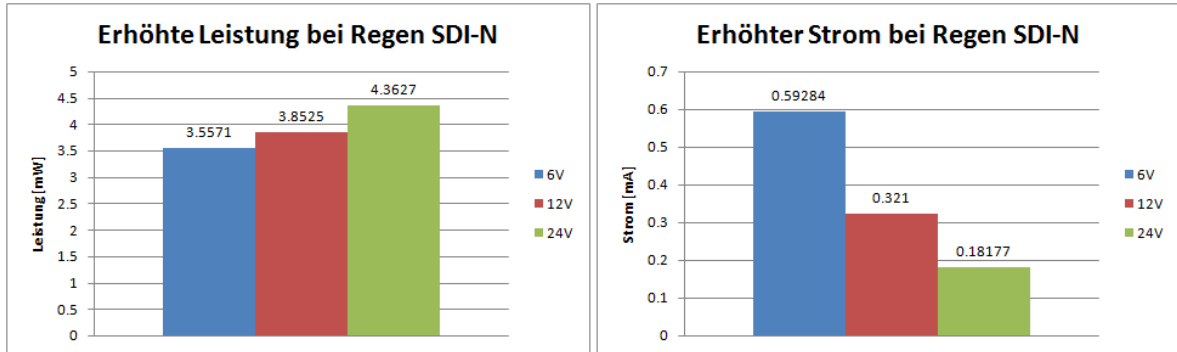


Abbildung 4.8: Zusätzliche benötigte Leistung bei Regen

Dieser Versuch wird im SDI-N Modus gemacht. Durch seinen niedrigen Energieverbrauch fallen auch weitere geringe Energieverbräuche, wie die des zusätzlichen Verbrauchs bei Regen, ins Gewicht. Anders ist dies beim SDI-C Modus. Der zusätzliche Verbrauch von 0.3 mA fällt bei einem durchschnittlichen Verbrauch von 1.3 mA nicht mehr so stark ins Gewicht.

#### 4.4 Ein-/Ausschalten

In diesem Abschnitt wird der Einschaltstrom gemessen. Daraus kann ermittelt werden, ab welchem Zeitpunkt es sich lohnen würde, den Sensor auszuschalten, um Energie gegenüber dem Standbybetrieb zu sparen. Da der Einschaltstrom höher ist als der Standbystrom muss dies messtechnisch untersucht werden. In Abbildung 4.9 ist das Einschaltverhalten dargestellt.

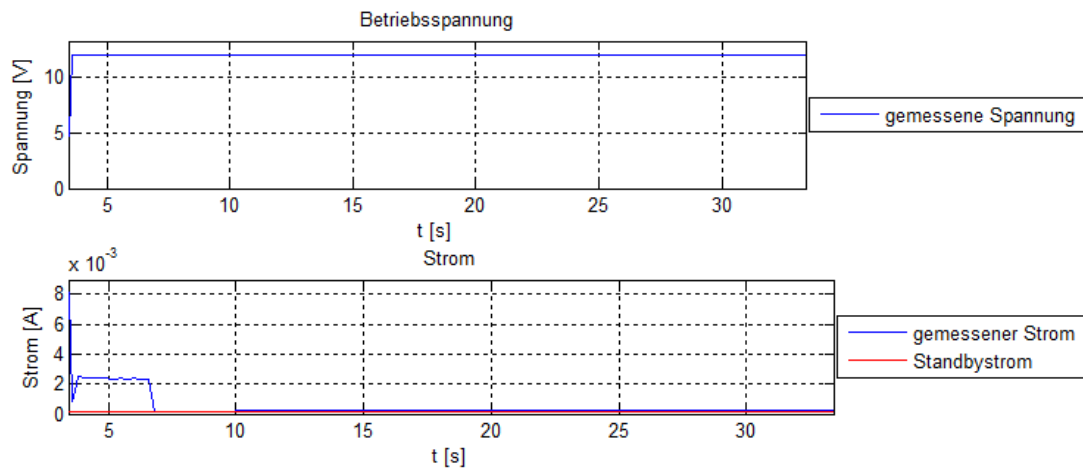


Abbildung 4.9: Einschalten bei 12 V SDI-N

## 4.5 Messung Heizung

### 4.5.1 Versuchsaufbau Kältekammer

Es wurden detaillierte Analysen der Wetterstation in einer Kältekammer durchgeführt. Der Versuchsaufbau sieht folgendermassen aus (Abb. 4.10):



Abbildung 4.10: Versuchsaufbau Kältekammer

Abbildung 4.10 linkes Bild: Auf dem Tisch steht der Laptop, mit welchem die Sonde im Kühlraum über eine RS-232 Schnittstelle verbunden ist. Die Speisung der Heizung und die Speisung des Sensors sind getrennt. Dies wird durch die Verwendung von zwei unabhängigen Ausgängen des Messgerätes realisiert. Die Sonde (rechtes Bild) befindet sich bei  $-25\text{ }^{\circ}\text{C}$  in einer Kühlkammer. Die Kabel werden durch dafür vorgesehene Öffnungen geführt. Alle Messungen sind ausführlich in einem Messprotokoll dokumentiert, welches auf der CD beigelegt ist.

### 4.5.2 Heizung

Die Heizung wird bei  $6\text{ V}$ ,  $12\text{ V}$  und  $24\text{ V}$  ausgemessen, obwohl die Leistung nach Herstellerangaben bei  $15.5\text{ V}$  und bei  $32\text{ V}$  maximal ist. Für die Spannungen berücksichtigen wir die üblichen Batterie- und Solarspannungen. Bei einer Spannung von  $15.7\text{ V}$  wird der Heizwiderstand im Sensor automatisch umgeschaltet. Der Hersteller hat bei  $12\text{ V}$  einen Strom von  $1.1\text{ A}$  und  $0.6\text{ A}$  bei  $24\text{ V}$  angegeben. Bei den Messungen ist zu beachten, dass es auf dem Kabel einen Spannungsabfall gibt. Die Spannung am Sensor entspricht somit nicht der Spannung des Poweranalyzers. Bei  $6\text{ V}$  ist der Spannungsabfall über den Kabel am kritischsten, weil anstatt der  $6\text{ V}$  nur noch  $4.7\text{ V}$  am Sensor gemessen werden. Dies liegt gemäss Handbuch sogar unter der Betriebsspannung.

### Messung 6 V

Die Heizung wird bei  $-25\text{ °C}$  20 Minuten betrieben. In dieser Zeit erhöht sich die Temperatur des Messkopfes auf  $-22.5\text{ °C}$ , was keinen wesentlichen Unterschied zur Umgebungstemperatur von  $-25\text{ °C}$  darstellt. Die Temperaturwerte können in der Tabelle 4.1 betrachtet werden. Die Messresultate für Strom und Spannung sehen bei 6V folgendermassen aus (Abb. 4.11):

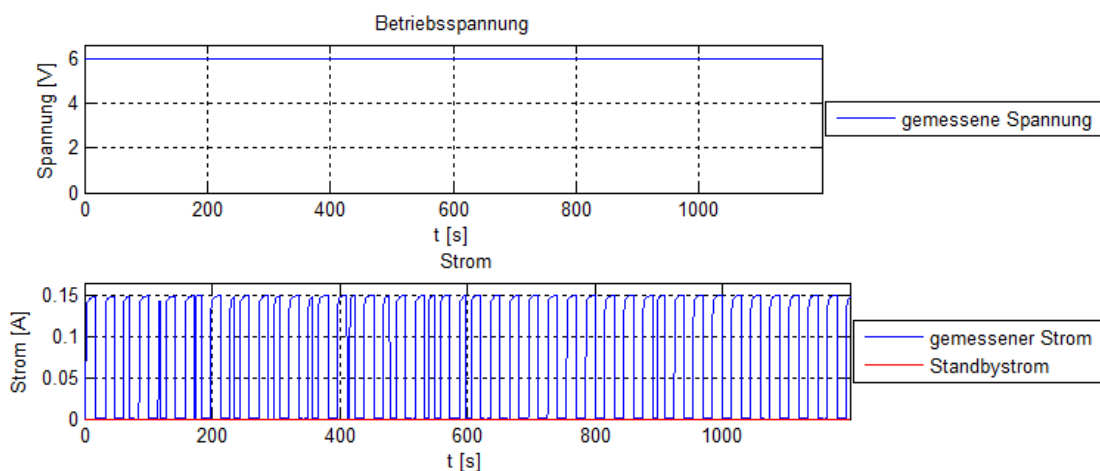


Abbildung 4.11: Heizung 6 V

Die Heizleistung wird geregelt und sollte sich bei einer gewissen Temperatur einpendeln. Wie der genaue Verlauf der Heizkurve aussieht, kann in Abbildung 3.5 nachverfolgt werden. Der Mittelwert der gemessenen Leistung beträgt  $0.445\text{ W}$ . Trotz der kleinen Leistung fliesst ein Strom von  $0.1\text{ A}$ , was gegenüber dem Verbrauch des Sensors gross ist.

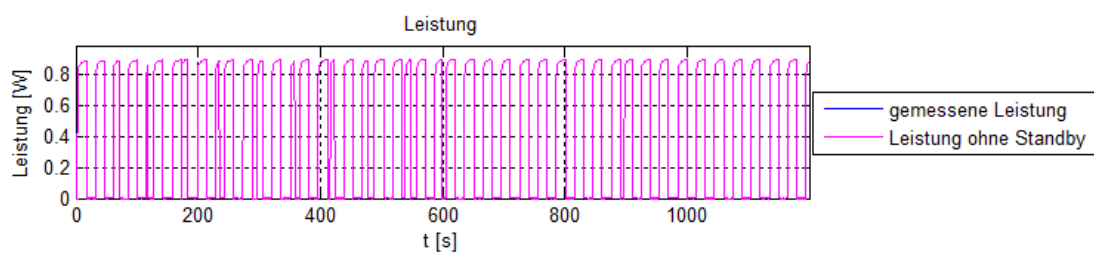


Abbildung 4.12: Leistung Heizung bei 6V

Tabelle 4.1: Temperaturmessung Heizung bei 6 V

Zeit[ <b>min</b> ]	Lufttemp. [ <b>°C</b> ]	Interne Temp. [ <b>°C</b> ]	Temp. Heizung [ <b>°C</b> ]
Vor Messung	-22.6	-22.0	-23.2
1	-22.8	-22.1	-23.0
2	-22.9	-22.2	-23.0
3	-22.9	-22.3	-23.0
4	-23.0	-22.3	-23.0
5	-23.1	-22.4	-23.0
6	-23.1	-22.4	-22.6
7	-23.2	-22.4	-22.8
8	-23.2	-22.4	-22.6
9	-23.2	-22.4	-22.1
10	-23.3	-22.5	-22.5
15	-23.4	-22.5	-22.6
20	-23.6	-22.6	-22.5

### Messung 12 V

Bei 12 V wird die Sonde auch bei  $-25\text{ °C}$  betrieben. Die Messzeit beträgt 20 Minuten. Bei 12 V kann die Temperatur auf  $-15\text{ °C}$  angehoben werden. Dies ist eine Steigerung der Temperatur um ca.  $10\text{ °C}$ , was jedoch immer noch nicht ausreicht um Eis zu schmelzen. Die Kurven des Stroms und der Leistung von 12 V sehen denen von 6 V ähnlich. Die mittlere Leistung beträgt 4 W.

Tabelle 4.2: Temperaturmessung Heizung bei 12V

Zeit[ <b>min</b> ]	Lufttemp. [ <b>°C</b> ]	Interne Temp. [ <b>°C</b> ]	Temp. Heizung [ <b>°C</b> ]
Vor Messung	-23.8	-23.0	-22.8
1	-23.8	-23.0	-22.6
2	-23.8	-22.9	-21.6
3	-23.9	-22.8	-20.9
4	-23.9	-22.6	-20.0
5	-24.0	-22.5	-19.5
6	-24.0	-22.3	-19.1
7	-24.0	-22.1	-19.0
8	-24.0	-22.0	-18.3
9	-24.0	-21.8	-17.7
10	-24.0	-21.6	-17.6
15	-23.9	-20.8	-16.5
20	-23.8	-20.3	-15.6

### Messung 24V

Für 24 V wurden dieselben Messungen durchgeführt. Auch in diesem Fall gibt es keine drastischen Änderungen zu den vorangegangenen Messungen. Der mittlere Strom beträgt 0.21 A

und die Leistung 5 W. Die Leistungszunahme ist nicht linear da der Widerstand der Heizung geändert wird.

### Weitere Messungen

In der Kühlkammer wurden weitere Messungen gemacht, bei welchen unter anderem erfolglos versucht wurde Eis zu schmelzen. Detaillierte Angaben dazu sind auf dem Messprotokoll auf der beigelegten CD zu finden.



Abbildung 4.13: Eisklotz schmelzen bei -1 °C

Nach diesen Messungen stellt sich heraus, dass die Heizung äusserst ineffektiv ist, vor allem für Solarbetrieb. Die beim Heizen verbrauchte Energie würde ohne Effekt den ganzen Akku entladen. Zudem ist es kritisch die Heizung mit 6 V zu betreiben, da durch den Spannungsabfall am Anschlusskabel, die Spannung am Sensor unter die Betriebsspannung fällt. Abbildung 4.14 zeigt Leistung und Strom in Abhängigkeit von der Spannung.

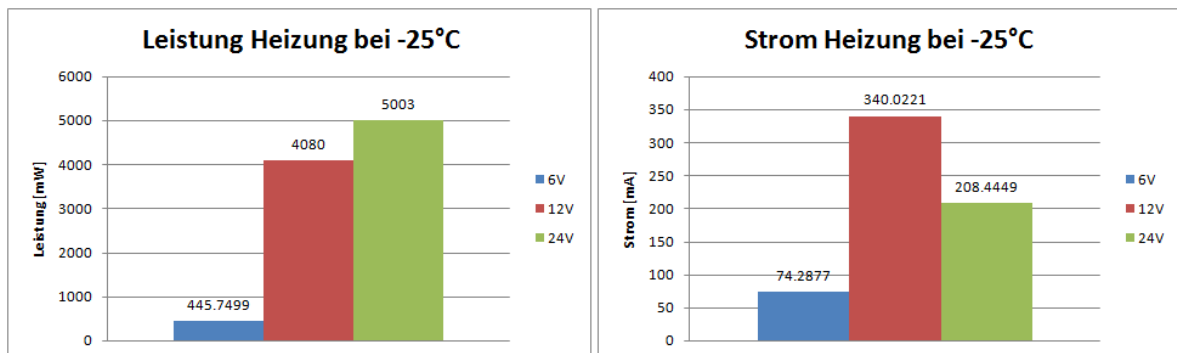


Abbildung 4.14: Übersicht Leistungen und Strom

Die Resultate des Stroms, der Spannung und der Leistung sind mit Vorsicht zu betrachten, weil nur die am Poweranalyzer gemessenen Werte ausgewertet sind. Die Spannungen am Sensor sind wegen des Spannungsabfalls über dem Kabel kleiner.

Die Heizung schaltet sich nahe dem Gefrierpunkt (4 °C) ein. Sie soll bei Schneeregen die Fläche für die Regenmessung schneefrei halten.



## 4.6 Messkommandos

Für die Verbrauchsberechnung sind die Messungen der einzelnen Messkommandos von Bedeutung, denn bei der Durchführung einer Messung wird zusätzlich Energie verbraucht. Sowohl die Leistungsaufnahme wie auch die Dauer der Messungen unterscheiden sich.

### 4.6.1 Windmessung

Bei der Windmessung können verschiedene Parameter verändert werden, wie die Messdauer und die Abtastrate. Diese Einstellungen haben einen Einfluss auf den Verbrauch. In der ersten Messung wird bei einer Abtastrate von 1 Hz für 10 Sekunden gemessen. Weitere Messungen wurden mit Messdauern von 30 Sekunden gemacht, um zu zeigen, dass der Verbrauch proportional zur Messzeit ist. Der Energieverbrauch wird im SDI-N und SDI-C Modus analysiert. Abbildung 4.15 zeigt die Stromkurve der Windmessung im SDI-N-Modus.

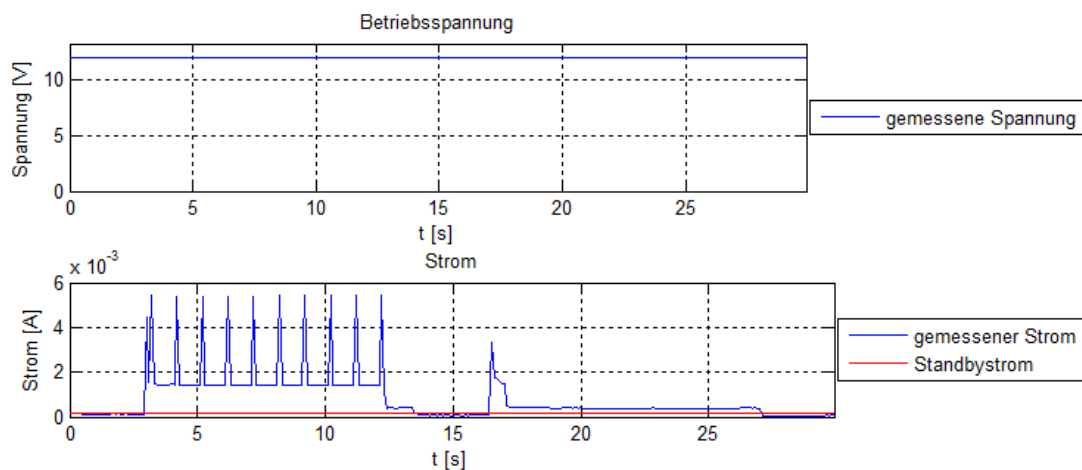


Abbildung 4.15: Stromverlauf Windmessung bei 12 V SDI-N Modus 1 Hz

In Abbildung 4.15 ist der Stromverlauf der Windmessung bei 12 V dargestellt. Auffällig sind die Spitzen im Stromverlauf, welche sich mit der Messfrequenz erhöhen. In diesem Fall wird mit 1 Hz gemessen. Die Windmessung dauert 10 Sekunden, danach werden die Daten abgerufen. Die Messung beginnt nach 3.9 Sekunden und endet bei 13.4 Sekunden.

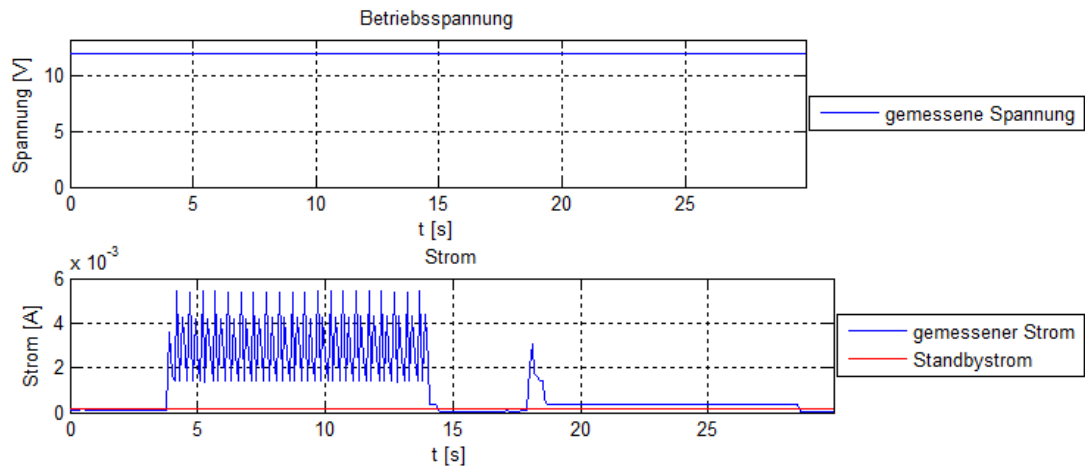


Abbildung 4.16: Stromverlauf Windmessung bei 12 V SDI-N Modus 4 Hz

Als Vergleich wird in Abbildung 4.16 die Messung bei 4 Hz gezeigt. Die Häufigkeit der Stromspitzen nimmt wie angenommen zu, was eine erhöhte Leistungsaufnahme zur Folge hat.

Um die Stromverbräuche miteinander vergleichen zu können, wird nachfolgend von einem durchschnittlichen Strom gesprochen. Er berechnet sich, indem man die Ladung, die für eine Messung verbraucht wird, durch die Messdauer teilt. Dieser durchschnittliche Stromverbrauch stellt zwar keinen absoluten Wert dar, kann jedoch für den relativen Vergleich bei gleicher Messdauer verwendet werden. Folgend sind die Windmessungen miteinander verglichen, dabei sind die Werte des durchschnittlichen Stromverbrauchs auf 30 bzw. 60 Sekunden bezogen.

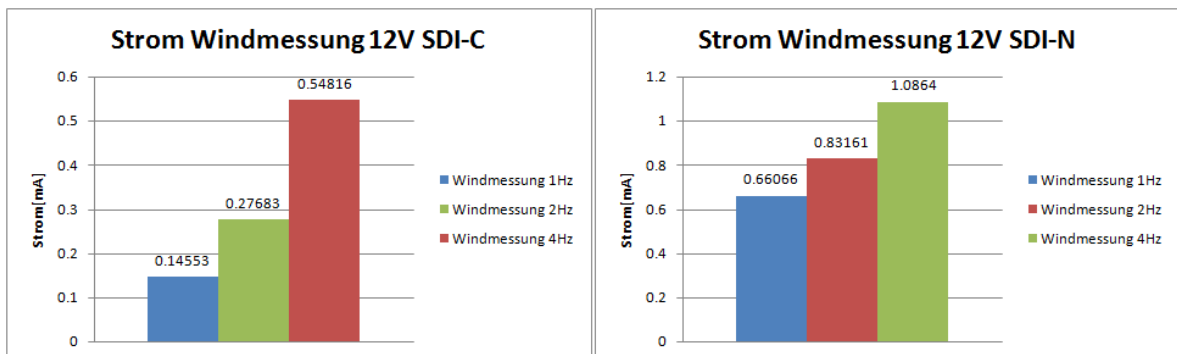


Abbildung 4.17: Vergleich des Stromes der Windmessung bei verschiedene Messfrequenzen

Abbildung 4.17 zeigt den Vergleich zwischen dem SDI-C und dem SDI-N Modus. Wie aus dem Verlauf der Ströme sichtbar ist, braucht der Sensor bei einer Abtastrate von 4 Hz am meisten Strom. Im Vergleich mit einer Messung im SDI-C Modus wird im SDI-N Modus mehr Strom für die Windmessung benötigt.

Der wirkliche Stromverbrauch kann über die Energie berechnet werden. Die Energie ist unabhängig von der Messzeit, sofern der Standbyverbrauch abgezogen wird. Dieser Wert charakterisiert also die Messung. Abbildung 4.18 zeigt den Energieverbrauch der Windmessung und den durchschnittlichen Stromverbrauch während der Windmessung.

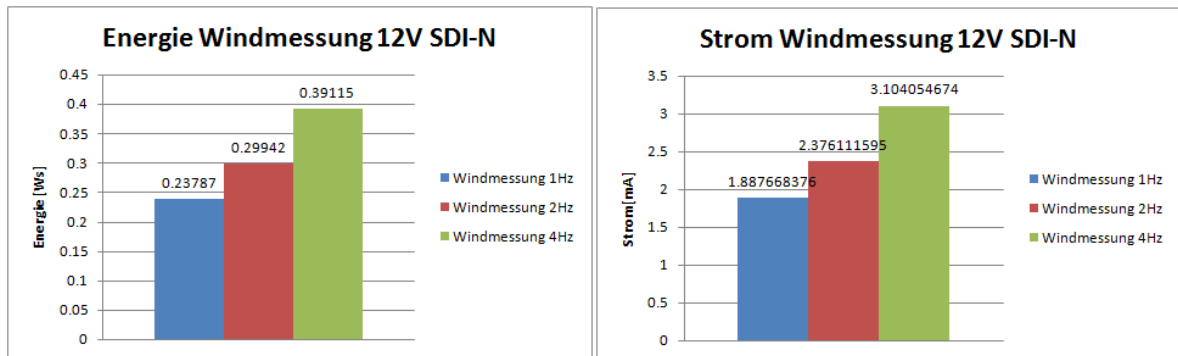


Abbildung 4.18: Energie und durchschnittlicher Stromverbrauch der Windmessung

Vergleicht man die Werte des durchschnittlichen Stromverbrauchs mit den Angaben des Herstellers, so liegen sie in den gleichen Größenordnungen. Nach Hersteller benötigt die Windmessung 2-5 mA.

#### 4.6.2 Vergleich Wind-, Regen- Temperatur- und Supervisormessung

Wie in Abschnitt 4.6.1 auf Seite 23 erklärt, sind die durchschnittlichen Werte auf die Messdauer des Poweranalyzer bezogen (30 s) und nicht auf die Dauer der Messungen der Vaisala Wettersonde. Zuerst werden die Leistungen der Messkommandos bei unterschiedlichen Spannungen im SDI-C Modus verglichen (Abb. 4.19 und Abb. 4.20).

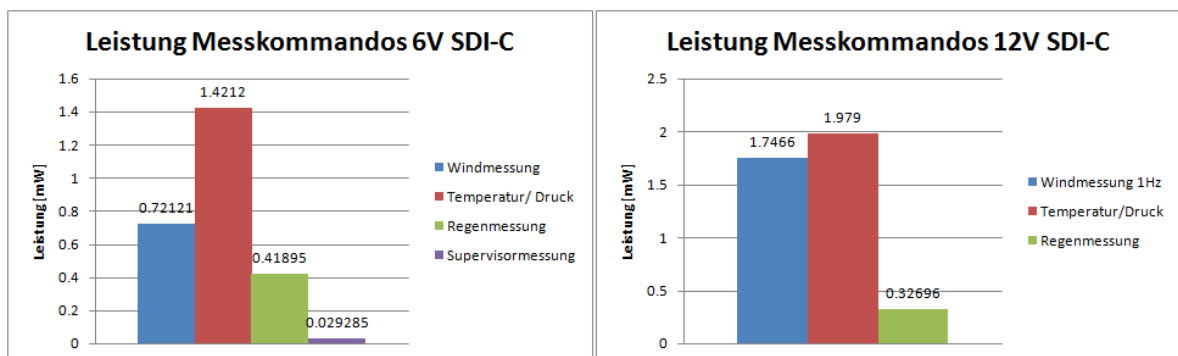


Abbildung 4.19: Leistungen der Messkommandos bei 6 V und 12 V im SDI-C Modus

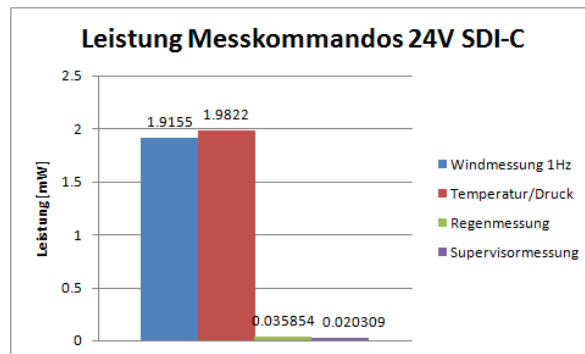


Abbildung 4.20: Leistungen der Messkommandos bei 24 V im SDI-C Modus

Die Messungen im SDI-N Modus sehen folgendermassen aus (Abb. 4.21). Da der SDI-N Modus der Energiesparsamste ist, wird auch eine detailliertere Analyse gemacht.

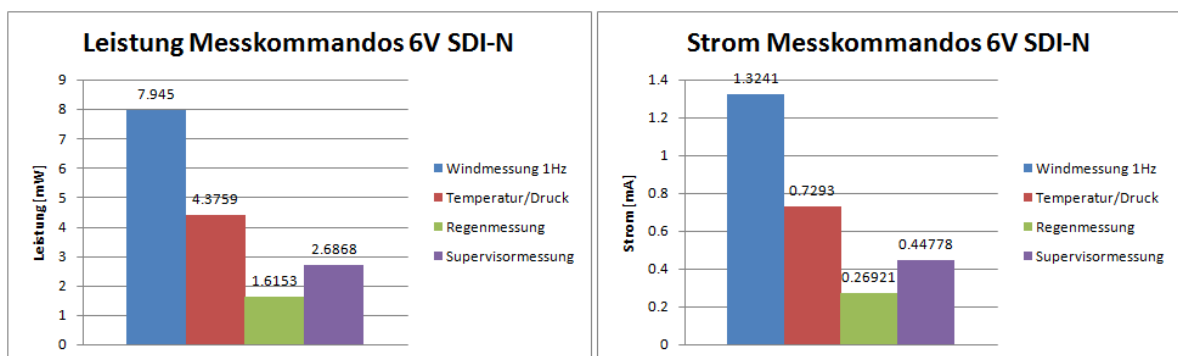


Abbildung 4.21: Leistungen und Stromverbräuche der Messkommandos bei 6 V im SDI-N Modus

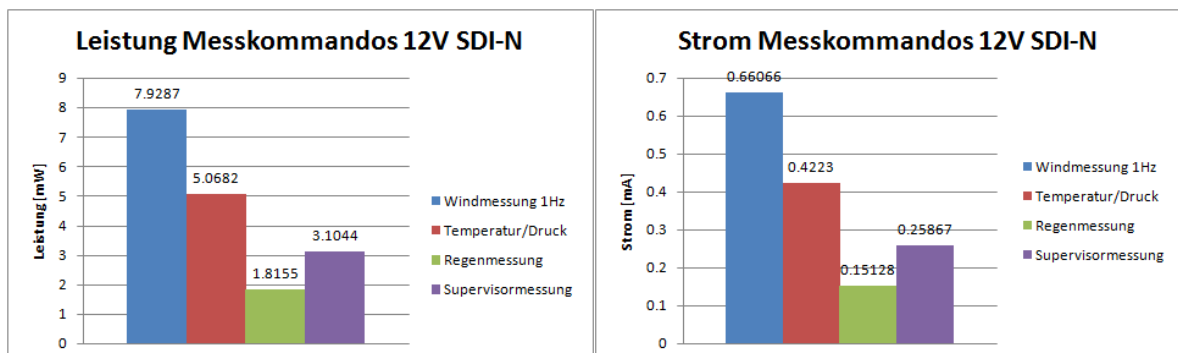


Abbildung 4.22: Leistungen und Stromverbräuche der Messkommandos bei 12 V im SDI-N Modus

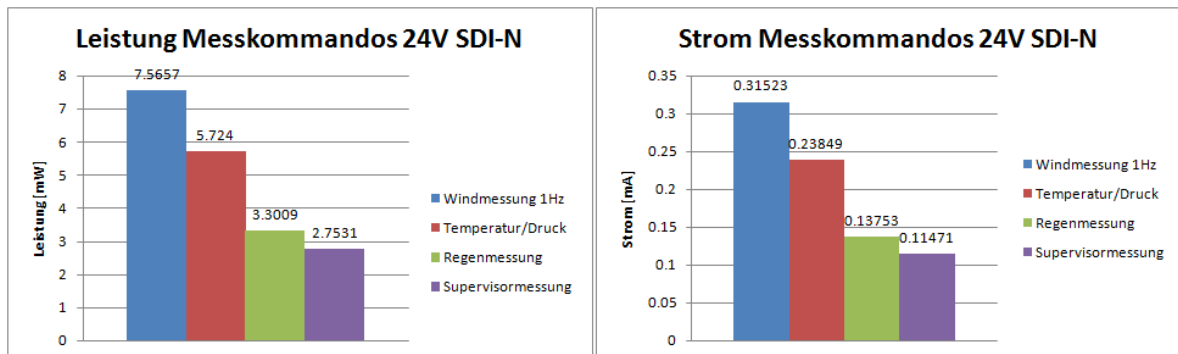


Abbildung 4.23: Leistungen und Stromverbräuche der Messkommandos bei 24 V im SDI-N Modus

Den Messwerten zufolge, verbraucht die Windmessung am meisten Energie, dies entspricht auch den Angaben des Herstellers. Wird bei der Windmessung die Verbindung der Messstäbe zum Beispiel durch Schnee unterbrochen, so ergibt sich bei allen Messwerten der Windmessung eine Null.

## 4.7 Auswertung

In diesem Abschnitt werden die Messungen der Abschnitte 4.2 bis 4.6 ausgewertet.

### 4.7.1 Einschalten

Wie schon bei den Messungen von Wind usw. benötigt auch die Einschaltdauer eine gewisse Zeit. Werden also wieder Durchschnittswerte gebildet, sind diese abhängig von der Messdauer des Poweranalyzers. Darum wird wieder über die Energie gerechnet, denn die verbrauchte Energie bleibt konstant.

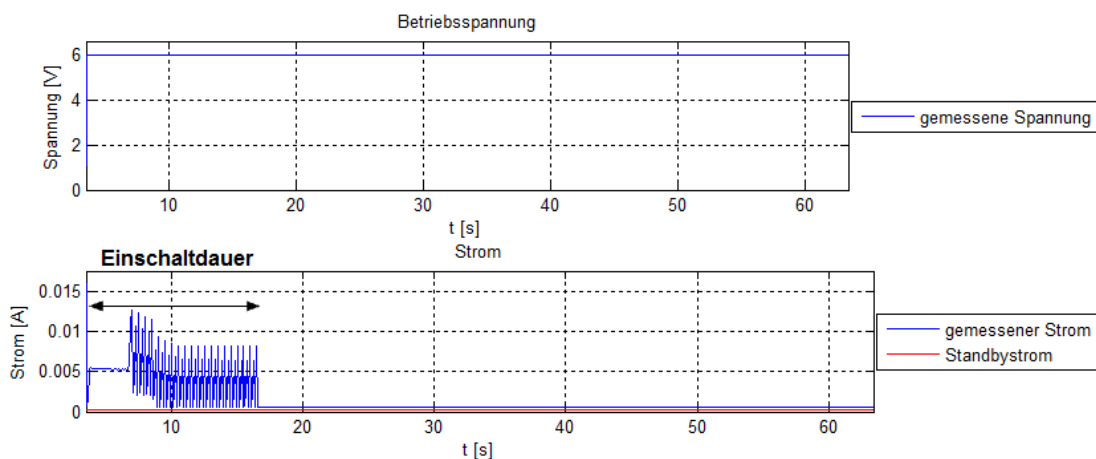


Abbildung 4.24: Einschaltdauer

Zur Berechnung der Leistung wird die Gesamtenergie durch 17 Sekunden, die Dauer des Einschaltens, geteilt. In diesem Fall wurde angenommen, dass der Standbyverbrauch nach

dem Einschalten vernachlässigt werden kann.

Tabelle 4.3: Überblick Einschaltvorgang ganze Messperiode

Protokoll	Spannung [V]	Energie [Ws]	Leistung [mW]	Strom [mA]
SDI-C	6	0.513	30.199	5.039
SDI-C	12	0.354	20.801	1.740
SDI-C	24	0.527	31.001	1.296
SDI-N	6	0.112	13.980	2.338
SDI-N	12	0.165	20.605	1.720
SDI-N	24	0.295	36.840	1.734

Der Standbyverbrauch nach dem Einschalten kann jedoch nicht vernachlässigt werden. Deshalb wird nach dem Einschaltvorgang das Messresultat abgeschnitten. Die Resultate sind bezogen auf die Einschaltdauer von 17 Sekunden im SDI-C-Modus und 8 Sekunden im SDI-N-Modus. In Tabelle 4.4 werden die Messresultate gezeigt, welche nach dem Einschaltvorgang abgeschnitten wurden.

Tabelle 4.4: Überblick Einschaltvorgang Messperiode abgeschnitten

Protokoll	Spannung [V]	Energie [Ws]	Leistung [mW]	Strom [mA]
SDI-C	6	0.356	20.922	3.491
SDI-C	12	0.263	15.449	1.292
SDI-C	24	0.359	21.118	0.883
SDI-N	6	0.078	9.800	1.639
SDI-N	12	0.092	11.467	0.957
SDI-N	24	0.126	15.741	0.741

Die verbrauchte Energie beim Einschaltvorgang, wurde mit der Energie im Standby verglichen. Daraus ergibt sich dann die Zeit, in der die Sonde abgeschaltet werden muss, damit es sich energetisch überhaupt lohnt, die Sonde auszuschalten. Nun zur Aufreihung der Resultate: Zuerst wird der SDI-C Modus, danach der SDI-N Modus, und schlussendlich der SDI-C Modus mit tiefem Standby in Tabelle 4.5 festgehalten (SDI-C SU).

Tabelle 4.5: Abwägung der Ausschaltzeit gegen den Standbyverbrauch

U/Protokoll	Verhältnis P	$T_{\text{ein}}$ [s]	$T_{\text{aus}}$ [s]	$W_{\text{ein}}$ [mWs]	$W_{\text{stby}}$ [mWs]
6 V SDI-C	1.546	17.000	<b>26.290</b>	355.670	355.670
12 V SDI-C	0.932	17.000	<b>15.852</b>	262.630	262.630
24 V SDI-C	0.926	17.000	<b>15.745</b>	359.010	359.010
6 V SDI-N	9.646	8.000	<b>77.164</b>	78.399	78.399
12 V SDI-N	6.058	8.000	<b>48.467</b>	91.733	91.733
24 V SDI-N	2.871	8.000	<b>22.968</b>	125.930	125.930
6 V SDI-C SU	6.207	17.000	<b>105.512</b>	355.670	355.670
12 V SDI-C SU	3.718	17.000	<b>63.211</b>	262.630	262.630
24 V SDI-C SU	2.633	17.000	<b>44.760</b>	359.010	359.010

$U$  ist die angelegte Spannung am Sensor. Das Protokoll entspricht dem verwendeten Kommunikationsprotokoll, im untersuchten Fall das SDI-12 Protokoll mit den beiden verschiedenen Moden. Verhältnis  $P$  ist das Verhältnis der beiden Leistungen, der Leistung beim Einschalten und der Standbyleistung.  $T_{\text{ein}}$  ist die Zeit beim Einschalten, in der der Sensor einen erhöhten Stromverbrauch aufweist.  $T_{\text{aus}}$  ist die Zeit, in der der Sensor ausgeschaltet bleiben muss, damit es sich lohnt den Sensor auszuschalten. Ansonsten würde mit dem Ein- und Ausschaltvorgang mehr Energie aufgewendet, als wenn der Sensor die ganze Zeit im Standby laufen würde.  $W_{\text{ein}}$  ist die Energie, die beim Einschalten verbraucht wird.  $W_{\text{stby}}$  ist die Energie, die vom Standby in der Zeit in welcher der Sensor ausgeschaltet ist, verbraucht wird.  $W_{\text{ein}}$  und  $W_{\text{stby}}$  werden zur Kontrolle aufgeführt. Sie müssen im Grenzpunkt gleich sein.

#### 4.7.2 Berechnung Verbrauch

Um den Energieverbrauch der Sonde zu messen, wurde nun angenommen die Vaisala Messsonde messe im zwei-Minuten-Takt. Dabei werden der SDI-N und der SDI-C Modus bei 12 V untersucht und verglichen. Zuerst wird der Stromverbrauch der einzelnen Messungen aufgelistet, plus der Standby, danach wird alles auf den Tag umgerechnet. Es erscheint sinnvoll 12 V als Spannung zu nehmen, da die bereits bestehende Basisstation auch mit 12 V läuft. Damit erreicht man eine einheitliche Spannung. Ein weiterer Grund ist, dass man bei 6 V am unteren Ende der Betriebsspannung ist. Bei der Heizung hat man schlussendlich 4.7 V am Sensor was unter der Betriebsspannung liegt. Der Vorteil von 6 V gegenüber 12 V und 24 V ist, dass die Solarkomponenten kleiner sind, damit aber auch leistungsschwacher. Mit 12 V wird somit die Grösse und auch die Anwendung optimiert. Mit 12 V kann man nicht dauerhaft Heizen, jedoch wird die Heizfunktion ermöglicht, was bei 6 V zu Schwierigkeiten führt. Bei 24 V werden die Komponenten zu gross.

In der folgenden Tabelle wird der Sensor mit 12 V im SDI-C-Modus betrieben. In der Auflistung nicht berücksichtigt sind, der zusätzliche Verbrauch bei Regen und wenn die Heizung eingeschaltet ist. Der Sensor wird nicht ausgeschaltet, sondern bleibt im Standby wenn nichts gemessen wird.

Tabelle 4.6: Stromverbrauch SDI-C 12 V ohne Regen und ohne Heizen

	Strom [mA]	Dauer [s]	Ladung [As]
1 Messzyklus 12 V SDI-C		120	
Wind 1 Hz 10 sek	0.548	30	0.01644
Temperatur/Druck/Feuchtigkeit	0.165	30	0.00495
Regen	0.027	30	0.00082
Standby	2.255	120	0.27055
<b>Total</b>			<b>0.29276</b>

Die durchschnittlichen Werte der Messungen werden auf die 30 Sekunden Messdauer des Poweranalyzers gerechnet. Wird jetzt mit diesen 30 Sekunden multipliziert, so erhält man die Ladung abzüglich Standby, welche für eine Messung verbraucht wird.

In Tabelle 4.7 wird der ganze Verbrauch auf den Tag umgerechnet und somit die Anzahl Tage berechnet, welche die Vaisala Messsonde messen könnte, ohne dass der anfänglich volle Akku nachgeladen wird.

Tabelle 4.7: Tagesverbrauch SDI-C 12 V ohne Heizen und ohne Regen

Anzahl Minuten pro Tag	1440
Anzahl Messungen pro Tag	720
Ladung pro Tag für alle Messungen (in As)	135.27
As $\rightarrow$ Ah	0.038
Bsp Batterie.	2.5 Ah
<b>Anzahl Tage</b>	<b>66.54</b>



Die gleiche Auswertung wurde auch für den SDI-N Modus bei 12 V durchgeführt.

Tabelle 4.8: Stromverbrauch SDI-N 12 V ohne Regen und ohne Heizen

	<b>Strom [mA]</b>	<b>Dauer [s]</b>	<b>Ladung [As]</b>
1 Messzyklus 12 V SDI-N		120	
Wind 1 Hz 10 sek	0.79728	30	0.0239184
Temperatur/Druck/Feuchtigkeit	0.55524	30	0.0166572
Regen	0.29335	30	0.0088005
Standby	0.158	120	0.01893
<b>Total</b>			<b>0.06830</b>

Tabelle 4.9: Tagesverbrauch SDI-N 12V ohne Heizen / ohne Regen

Anzahl Minuten pro Tag	1440
Anzahl Messungen pro Tag	720
Ladung pro Tag für alle Messungen (in As)	49.18
As → Ah	0.014
Bsp Batterie.	2.5 Ah
<b>Anzahl Tage</b>	<b>183.01</b>

Wie man sieht, wird im SDI-N Modus weniger Energie verbraucht. Zusätzlich könnte man den Sensor nach der Messung abschalten, dies spart zusätzlich Energie. Was hingegen vernachlässigt wird, ist der Laderegler, den man einsetzt, damit die Batterie nicht über- oder allzu fest entladen wird. Dieser Regler hat einen permanenten Standbystrom. Da der SDI-C-Modus keine wirklichen Vorteile mit sich bringt, haben wir uns für den verbrauchsärmeren SDI-N-Mode entschieden.

Der Verbrauch mit Laderegler und Ein-/Ausschalten wird in der Tabelle 4.10 mit berücksichtigt. In den nachfolgenden Tabellen werden nur noch die Resultate des SDI-N-Modes gezeigt. Alle weiteren Tabellen können auf der beiliegenden CD angeschaut werden, die Tabellen sind im File AuswertungMessungen.xlsx gespeichert.

Tabelle 4.10: Stromverbrauch SDI-N 12 V mit Reglerstandby und Ausschalten, ohne Regen und ohne Heizen

	<b>Strom [mA]</b>	<b>Dauer [s]</b>	<b>Ladung [As]</b>
1 Messzyklus 12 V SDI-N		120	
Wind 1 Hz 10 sek	0.79728	30	0.0239184
Temperatur/Druck/Feuchtigkeit	0.55524	30	0.0166572
Regen	0.29335	30	0.0088005
Einschalten	0.957315139	8	0.007658521
Standby	0.158	120	0.01893
Standby Laderegler Solarzelle	3	120	0.36
<b>Total</b>			<b>0.421135081</b>

Tabelle 4.11: Tagesverbrauch SDI-N 12 V ohne Heizen und ohne Regen

Anzahl Minuten pro Tag	1440
Anzahl Messungen pro Tag	720
Ladung pro Tag für alle Messungen (in As)	303.22
As → Ah	0.08423
Bsp Batterie.	2.5 Ah
<b>Anzahl Tage</b>	<b>29.6816878</b>

In der Tabelle 4.12 wird vom ungünstigsten Fall für die Regenmessung ausgegangen. Es wird angenommen, dass es immer regnet und die Sonde dauernd eingeschaltet ist. Dabei sieht man, dass die Sonde im ganzen Verbrauch relativ wenig ausmacht, weil der Laderegler einen Standby von 3 mA aufweist.

Tabelle 4.12: Stromverbrauch SDI-N 12 V mit Dauerregen und ohne Heizen

	<b>Strom [mA]</b>	<b>Dauer [s]</b>	<b>Ladung [As]</b>
1 Messzyklus 12 V SDI-N		120	
Wind 1 Hz 10 sek	0.79728	30	0.0239184
Temperatur/Druck/Feuchtigkeit	0.55524	30	0.0166572
Regen	0.29335	30	0.0088005
Standby	0.158	120	0.01893
Standby Laderegler Solarzelle	3	120	0.36
Zusätzlicher Standby für Regen	0.321	120	0.03852
<b>Total</b>			<b>0.4492</b>

Tabelle 4.13: Tagesverbrauch SDI-N 12 V mit Dauerregen und ohne Heizen

Anzahl Minuten pro Tag	1440
Anzahl Messungen pro Tag	720
Ladung pro Tag für alle Messungen (in As)	323.39
As → Ah	0.08983
Bsp Batterie.	2.5 Ah
<b>Anzahl Tage</b>	<b>27.83</b>

In der folgenden Tabelle werden die Berechnungen mit eingeschalteter Heizung durchgeführt. Die Heizung ist der grösste Verbraucher der Wetterstation.

Tabelle 4.14: Stromverbrauch SDI-N 12 V mit Heizen

	Strom [mA]	Dauer [s]	Ladung [As]
1 Messzyklus 12 V SDI-N		120	
Wind 1 Hz 10 sek	0.79728	30	0.0239184
Temperatur/Druck/Feuchtigkeit	0.55524	30	0.0166572
Regen	0.29335	30	0.0088005
Standby	0.158	120	0.01893
Standby Laderegler Solarzelle	3	120	0.36
Heizung zusätzlicher Standby	340.02	120	40.80
<b>Total</b>			<b>41.23</b>

Tabelle 4.15: Tagesverbrauch SDI-N 12V mit Heizen

Anzahl Minuten pro Tag	1440
Anzahl Messungen pro Tag	720
Ladung pro Tag für alle Messungen (in As)	29686.29
As → Ah	8.25
Bsp Batterie.	2.5 Ah
<b>Anzahl Tage</b>	<b>0.30</b>
<b>Anzahl Stunden</b>	<b>7.28</b>

# Kapitel 5

## Auswahl

Die Kriterien beim Auswahlverfahren der Komponenten sind:

- Das Bauteil sollte von einem Grossverteiler wie Farnell stammen, da dort die Lieferfrist in der Regel nur ein Tag beträgt.
- Die Wetterstation muss den hochalpinen Kriterien gerecht werden. Das heisst, die Komponenten müssen auch bei  $-25\text{ °C}$  noch funktionieren und Schnee, Wind und Kälte über längere Dauer überstehen können.
- Das Bauteil muss den gestellten technischen Anforderungen genügen und somit ausreichend dimensioniert sein.

Aus den verschiedenen Produkten haben wir die passenden bestellt. Die ausgewählten Komponenten sind in den Absätzen 5.1, 5.2 und 5.3 näher beschrieben. Aufgezeigt werden die wichtigsten Eckdaten. Weitere Informationen findet man im Datenblatt.

## 5.1 Solarzelle

Wir haben eine Solarzelle von der Firma Megasol<sup>1</sup> ausgesucht. Die Firma Megasol nennt ihre Solarzelle *Mobile*. Ihren Angaben zufolge, kann die Solarzelle von -25 °C bis 50 °C eingesetzt werden.

- Begehrbar und bruchfest (ohne Glas)
- Geringes Gewicht
- Lange Lebensdauer
- Hoher Zell-Wirkungsgrad dank optimaler Hinterlüftung und Wärmeableitung
- Geeignet für Extrem-Einsätze



Tabelle 5.1: Kenngrößen Solarzelle

Leistung	7 Wp
Spannung	17 V
Strom	411 mA
Abmessung	205*360*15 mm
Gewicht	0.49 kg

Abbildung 5.1: Megasol Solarzelle

Mit der Einheit Wp ist die Spitzenleistung gemeint. Das bedeutet, die Leistung bei maximaler Bestrahlung des Solarpanels. Die höhere Spannung eignet sich um den 12 V Akku zu laden.

<sup>1</sup>[http://www.megasol.ch/shop/Solar%20unterwegs/megapowerset/de/pdf/MegaPower%20Set.pdf/file\\_view](http://www.megasol.ch/shop/Solar%20unterwegs/megapowerset/de/pdf/MegaPower%20Set.pdf/file_view)

## 5.2 Laderegler

Der Laderegler ist von der Firma Steca<sup>2</sup>. Der volle Produktname lautet Steca Solsum 8.8c. Der Laderegler wird benötigt um sicherzustellen, dass der Akku weder über- noch unterladen wird. Ein Nachteil des Ladereglers ist, dass er einen Standbystrom von bis zu 4 mA verbraucht. Die Eckdaten des Ladereglers:



Tabelle 5.2: Kenngrößen Regler

Systemspannung	12 V
Max. Eingangsspannung	47 V
max. Laststrom	8 A
Abmessung	130*88*39 mm
Gewicht	0.165 kg

Abbildung 5.2: Steca Solarregler

Der Regler ist von -25 °C bis +50 °C einsetzbar. Er ist spannungsgesteuert und passt sich automatisch der Eingangsspannung 12 V/24 V an. Weitere Funktionen sind:

- Verpolschutz Last, Moduleingang
- Kurzschlussabsicherung Last, Modul
- Überspannungsschutz am Moduleingang
- Leerlaufschutz, falls keine Batterie angeschlossen ist
- Verhinderung Rückstrom in das Modul

<sup>2</sup><http://www.stecagmbh.de/index.php>

### 5.3 Akku

Als Akku haben wir einen Bleiakku gewählt, mit dem Produktnamen Hawker Cyclon Battery<sup>3</sup>. Der Bleiakкумуляtor zeichnet sich durch geringe Selbstentladung und Langlebigkeit aus. Zudem kann er über einen weiten Temperaturbereich verwendet werden (-40 °C bis 80 °C). Es folgen die wichtigsten Kenngrößen:



Tabelle 5.3: Kenngrößen Akku

Spannung	12 V
Energiespeicherung	2.5 Ah
Bauform	D
Abmessung	113*90*70 mm
Gewicht	1.2 kg

Abbildung 5.3: Akku Hawker Energy

Wichtig in unserer Anwendung ist die Selbstentladung der Batterie. Diese ist in einem Diagramm des Herstellers ersichtlich. In der Annahme, dass die Temperatur meist unter 20 °C liegt, wäre die Kapazität nach 100 Wochen immer noch bei 70 %. Das heisst, die Entladung spielt eine kleine Rolle in unserer Anwendung, da wir davon ausgehen, nicht länger als einige Tage oder ein bis zwei Wochen ohne Sonne zu sein. Die Selbstentladung wird in Abbildung 5.4 dargestellt. Abbildung 5.5 zeigt die Lebensdauer des Akkus, wobei angenommen wird, dass der Akku eine Kapazität von 80 % der Nennkapazität aufweist. Weiter gibt der Hersteller an, dass der Akku 300 Vollladezyklen und bis zu 10000 Teilladezyklen aushält. Die Betriebsdauer liegt zwischen 10 und 15 Jahren.

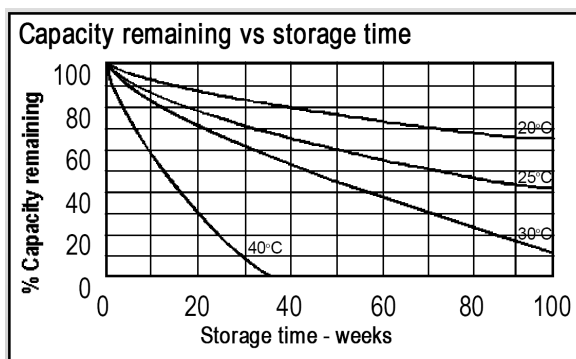


Abbildung 5.4: Restkapazität in Abhängigkeit der Lagerzeit

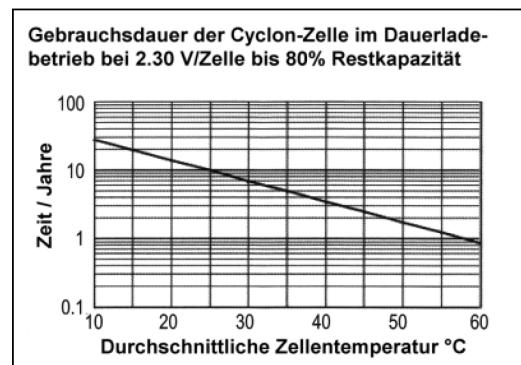


Abbildung 5.5: Gebrauchsdauer des Akku abhängig von der Temperatur

<sup>3</sup><http://ch.farnell.com/hawker-energy/9819-0020/battery-12v-2-5ah-2x3-cyclon/dp/1218977>



## Kapitel 6

# Entlade-/Aufladeprozess

Um zu prüfen ob der ausgewählte Akku ausreichend dimensioniert ist, wurde anhand von Wetterdaten vom Klein Matterhorn die Ladung im Akku berechnet. Die Wetterdaten vom 1.1.2008 bis 31.12.2008 sind von Kachelmann<sup>1</sup> zur Verfügung gestellt worden. In den Wetterdaten ist die Sonnenscheindauer in Minuten enthalten. Als Sonnenschein zählt, wenn die Bestrahlungsstärke den Grenzwert  $120 \text{ W/m}^2$  überschreitet.<sup>2</sup>

Da bei  $120 \text{ W/m}^2$  die Solarzelle keine volle Leistung abgibt, wird angenommen, dass von den möglichen 411 mA bei voller Sonneneinstrahlung, lediglich 80 %, also 328 mA fließen. Für die Berechnung wird angenommen, dass der Akkumulator immer mit diesen 328 mA geladen wird, sobald der Akku nicht mehr voll geladen ist. Dies entspricht nicht der Wirklichkeit. Zusätzlich wird angenommen, dass die Spannung immer konstant ist, unabhängig vom Ladezustand des Akkus. Der Tagesverbrauch wird immer von der Energie, die täglich produziert wird, abgezogen. Dieser negative oder positive Wert wird zur Ladung des Vortages addiert. Ist dieser Wert negativ, so wird der Akku entladen, ist er positiv, wird der Akku geladen. In den folgenden Absätzen werden verschiedene Ausgangslagen behandelt. Der ganze Berechnungsvorgang ist auf der CD zu finden.

---

<sup>1</sup>Die Wetterdaten wurden von jschug@meteomedia.ch zugeschickt

<sup>2</sup><http://de.wikipedia.org/wiki/Sonnenstunde>

## 6.1 Simulation Energieverbrauch ohne zusätzliche Funktionen verglichen mit Regen

Bei der Simulation ohne zusätzliche Funktionen wird angenommen, dass der Sensor alle zwei Minuten eine Messung macht, ansonsten ist er im Standby. Zum Standbyverbrauch der Wetterstation addiert sich der Standbyverbrauch des Ladereglers. Der daraus resultierende totale Verbrauch wurde mit dem Verbrauch verglichen, wenn es zusätzlich die ganze Zeit regnen würde (siehe Absatz 4.3).

Die Anfangskapazität des Akkumulators, wird mit 0 Ah angenommen. Der verwendete Akku hat eine Kapazität von 2.5 Ah. Diese Annahmen gelten für alle kommenden Beispiele, ausser es wird auf eine Abweichung hingewiesen.

Abbildung 6.1 zeigt die Akkuladung in Amperestunden über das gesamte Jahr. Dabei zeigt die blaue Kurve die Ladung bei der normalen Messung, während die grüne Kurve die Ladung bei Dauerregen darstellt.

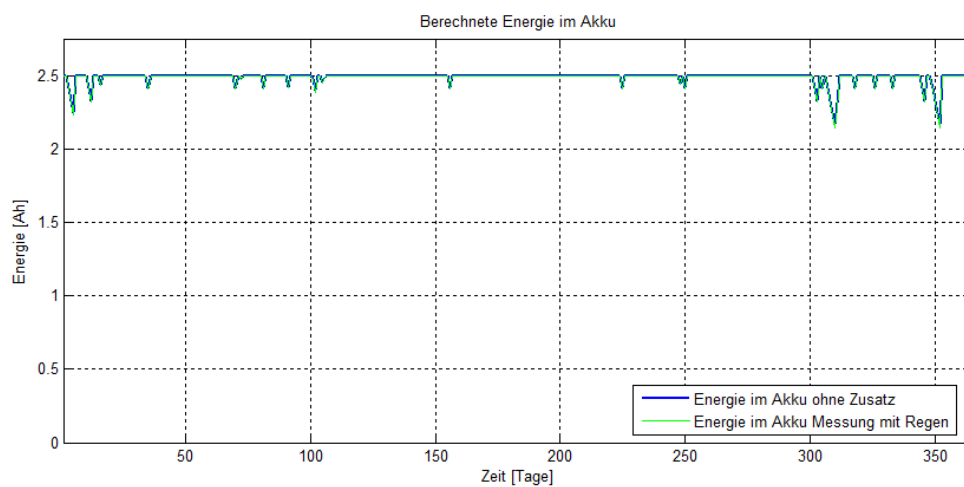


Abbildung 6.1: Ladung des Akkus bei realistischem Wetter

Vergleicht man die beiden Kurven, so stellt man fest, dass der Regensensor nicht der dominierende Verbraucher ist. Da 80 % Ladestrom bei schlechtem Wetter zu optimistisch ist, folgt ein Vergleich bei unterschiedlichen Ladeströmen.

## 6.2 Ladung im Akku bei unterschiedlichen Ladeströmen

In diesem Abschnitt wird ein Ladestrom von 80 % (bei schönem Wetter) mit einem Ladestrom von 10 % (bei schlechtem Wetter) verglichen. Dies entspricht einem Ladestrom von 328 mA bzw. 41 mA. Der angenommene Verbrauch entspricht dem Verbrauch bei Regen.

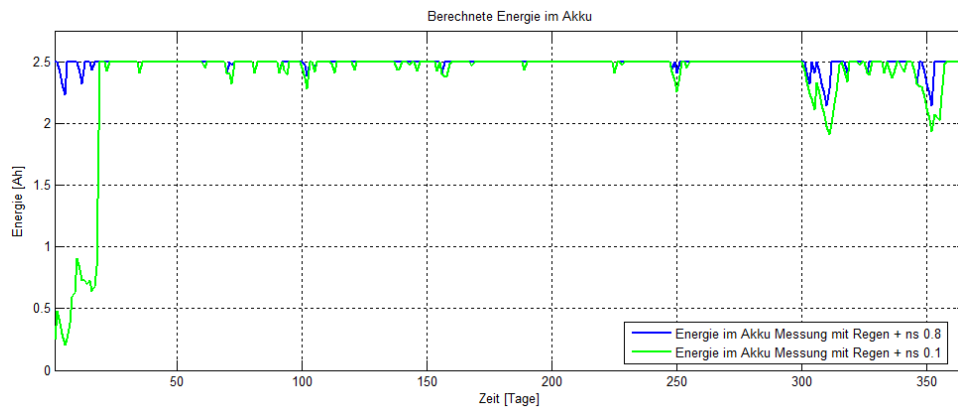


Abbildung 6.2: Aufladung mit unterschiedlichem Ladestrom

In Abbildung 6.2 fällt auf, dass vor allem am Anfang ein Unterschied entsteht, deshalb wird dieser genauer betrachtet.

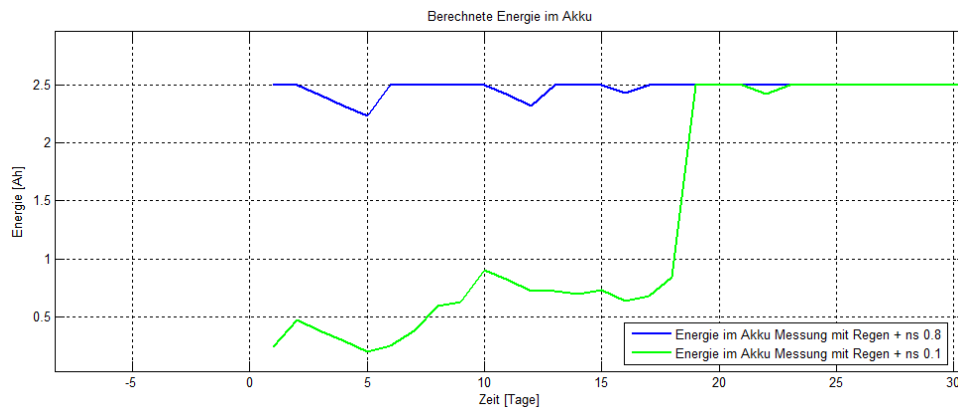


Abbildung 6.3: Vergrößerung Anfang mit unterschiedlichen Ladeströmen

Bei einem Ladestrom von 328 mA vermag die Solarzelle den Akku noch am ersten Tag zu füllen. Bei einem Ladestrom von 41 mA kann der Akku in den ersten 15 Tagen nicht gefüllt werden. Dies auch weil an diesen Tagen Winter ist und die Sonnenscheindauer sehr kurz ist. Dennoch kommt es zu keinem Unterbruch in den Messungen. Nicht berücksichtigt ist das Einschneien des Solarpanels.

### 6.3 Vergleich Regen/Heizen

In diesem Abschnitt wird die Ladung im Akku betrachtet, wenn die Heizung dauernd eingeschaltet ist und mit einem Strom von 328 mA geladen wird. Es wurde dabei nicht berücksichtigt, dass die Temperatur auch höher wird als die Ausschaltswelle der Heizung. Verglichen wird dabei der normale Verbrauch bei Dauerregen mit dem Verbrauch bei dauernd eingeschalteter Heizung.

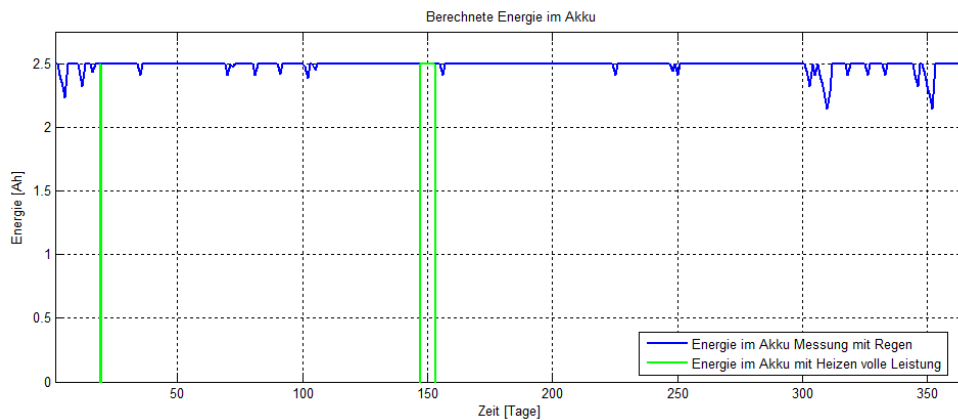


Abbildung 6.4: Vergleich Energie Heizen / Regen

In der Grafik ist sichtbar, dass bei eingeschalteter Heizung die meiste Zeit keine Energie im Akku ist. Die ganze Energie, die die Solarzelle abgibt, wird verheizt. Um die Kapazität des Akkus besser ausnutzen zu können, wurden Berechnungen mit 1 h bzw. 2 h heizen pro Tag durchgeführt.

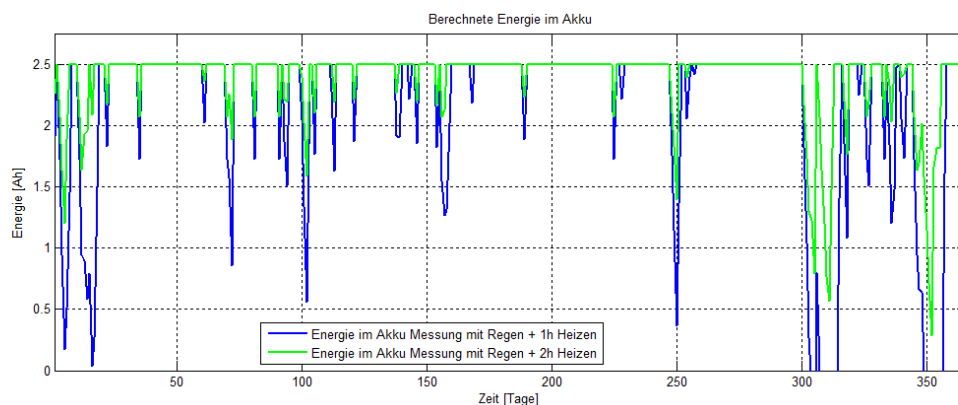


Abbildung 6.5: Heizung 1 h oder 2 h pro Tag aktiv

Mit einer Heizzeit von 1 h pro Tag würde mit den gemachten Annahmen der Akku nie leer. Anders sieht das bei einer Heizzeit von 2 h pro Tag aus. Vor allem in den Wintertagen gibt es Tage, an denen der Akku leer sein wird. Eine Möglichkeit wäre die Heizung Spannungsgesteuert zu aktivieren, das heisst, wenn die Akkuspannung einen gewissen Wert unterschreitet, schaltet die Heizung aus.

# Kapitel 7

## Software

### 7.1 Fest einkompilierte Parameter

Tabelle 7.1 listet die Parameter auf, die fest im Programmcode gespeichert sind.

Tabelle 7.1: Fest einkompilierte Parameter.

	Parameter	Wert	Erklärung
Einstellungen	Xu_M	S	SDI-12 v 1.3 native
	Xu_C	1	Serial Interface: SDI-12
	Xu_I	0	Keine automatische Messung
	Xu_B	1200	Baudrate 1200
	Xu_D	7	7 Data Bits
	Xu_P	E	Parity Even
	Xu_S	1	Ein Stop Bit
Messungen	Wu_U	M	Die Windgeschwindigkeit wird in m/s gemessen
	Tu_P	H	Der Druck wird in hPa gemessen
	Tu_T	C	Die Temperatur wird in °C gemessen
	Ru_U	M	Metrische Einheiten für die Regenmenge
	Ru_S	M	Metrische Einheiten für die Hagelmenge
	Ru_M	–	Autosend mode wird nicht verwendet
	Ru_Z	M	Der Zähler wird automatisch zurückgesetzt
	Ru_X	–	Limits Regenmenge werden nicht verwendet
	Ru_Y	–	Limits Hagelmenge werden nicht verwendet
	Su_S	Y	Fehlerbericht eingeschaltet

## 7.2 Prototyp (auf Gumstix)

### 7.2.1 Konfigurierbare Parameter

Alle Parameter, die nicht fest einkompiliert sind, können konfiguriert werden. Dazu ist im Arbeitsverzeichnis des Programms die Datei `ConfigurationFile.xml` zu erstellen. In dieser Konfigurationsdatei kann jeder Parameter von Tabelle 7.2 auf der nächsten Seite konfiguriert werden. Wird in der Datei ein Parameter nicht aufgeführt, wird der Standardwert eingesetzt. Existiert die Datei nicht, werden für alle Parameter Standardwerte eingesetzt. Listing 7.1 zeigt beispielhaft die Konfiguration einiger Parameter in der Konfigurationsdatei. Die Windmessung wird ein-, die Druckmessung ausgeschaltet, der Richtungs-Offset wird auf 27° gesetzt und die Heizungsregelung wird ausgeschaltet. Alle anderen Parameter nehmen ihre Standardwerte an.

---

```
<?xml version="1.0" encoding="UTF-8"?>
<java version="1.6.0_12" class="java.beans.XMLDecoder">
  <object class="weatherstation.ConfigurationFile">

    <void property="wu">
      <boolean>true</boolean>
    </void>

    <void property="tu_Pa">
      <boolean>>false</boolean>
    </void>

    <void property="wu_D">
      <int>27</int>
    </void>

    <void property="su_H">
      <string>N</string>
    </void>

  </java>
```

---

Listing 7.1: Deklaration einiger Parameter in der Konfigurationsdatei

Tabelle 7.2: Parameter, die via Konfigurationsdatei eingestellt werden können.

	Abk.	Mögliche Werte	Standard	Erklärung
	Xu_A	0..9,A..Z	0	Gerätenummer der Wetterstation
Wind	Wu	true/false	true	Die Windmessung ein- bzw. ausschalten
	Wu_Dn	true/false	true	Windrichtung min ein- bzw. ausschalten
	Wu_Dm	true/false	true	Windrichtung Durchschnitt ein- bzw. ausschalten
	Wu_Dx	true/false	true	Windrichtung max. ein- bzw. ausschalten
	Wu_Sn	true/false	true	Windgeschwindigkeit min. ein- bzw. ausschalten
	Wu_Sm	true/false	true	Windgeschwindigkeit Durchschnitt ein- bzw. ausschalten
	Wu_Sx	true/false	true	Windgeschwindigkeit max. ein- bzw. ausschalten
	Wu_I	1 ... 3600	120	Mess-Intervall in s
	Wu_A	1 ... 3600	3	Zeit für Durchschnittsberechnung (siehe Handbuch [2], Appendix D)
	Wu_G	1 oder 3	3	Algorithmus zur Berechnung der min/max Windgeschwindigkeit (siehe Handbuch [2], Absatz „Wind Measurement Principle“ auf Seite 25)
	Wu_D	-180 ... 180	0	Richtungs-Offset
	Wu_F	1, 2, 4	2	Abtastfrequenz
PTU	Tu	true/false	true	Die Druck-, Temperatur- und Feuchtigkeitsmessung ein- bzw. ausschalten
	Tu_Pa	true/false	true	Druckmessung ein- bzw. ausschalten
	Tu-Ta	true/false	true	Temperaturmessung ein- bzw. ausschalten
	Tu_Tp	true/false	true	Interne Temperatur ein- bzw. ausschalten
	Tu_Ua	true/false	true	Feuchtigkeitsmessung ein- bzw. ausschalten
	Tu_I	1 ... 3600	120	Mess-Intervall in s
Niederschlag	Ru	true/false	true	Niederschlagsmessung ein- bzw. ausschalten
	Ru_Rc	true/false	true	Regenmenge Messung ein- bzw. ausschalten
	Ru_Rd	true/false	true	Regendauer Messung ein- bzw. ausschalten
	Ru_Ri	true/false	true	Regenintensität Messung ein- bzw. ausschalten
	Ru_Hc	true/false	true	Hagelmenge Messung ein- bzw. ausschalten
	Ru_Hd	true/false	true	Hageldauer Messung ein- bzw. ausschalten
	Ru_Hi	true/false	true	Hagelintensität Messung ein- bzw. ausschalten
	Ru_Rp	true/false	true	Maximalwert Regen Messung ein- bzw. ausschalten
	Ru_Hp	true/false	true	Maximalwert Hagel Messung ein- bzw. ausschalten
	Ru_I	1 ... 3600	120	Mess-Intervall in s
Supervisor	Su	true/false	true	Die Heizungs- und Supervisormessung ein- bzw. ausschalten
	Su_Th	true/false	true	Heiztemperaturmessung ein- bzw. ausschalten
	Su_Vh	true/false	true	Heizspannungsmessung ein- bzw. ausschalten
	Su_Vs	true/false	true	Betriebsspannungsmessung ein- bzw. ausschalten
	Su_Vr	true/false	true	3.5V Referenzspannungsmessung ein- bzw. ausschalten
	Su_Id	true/false	true	Information ein- bzw. ausschalten
	Su_I	1 ... 3600	120	Mess-Intervall in s
	Su_H	Y/N	Y	Heizungsregelung ein- bzw. ausschalten

### 7.2.2 Ablauf

- System starten
  1. Konfigurationsdatei lesen
  2. Serielle Schnittstelle einrichten
  3. Gerätenummer des Wettersensors lesen und gemäss Konfiguration neu setzen
  4. Kommunikationseinstellungen vornehmen (auf SDI-12 native setzen)
  5. Alle Messungen stoppen, Wettersensor Reset
  6. Seriennummer lesen
  7. Wind, PTU<sup>1</sup>, Niederschlag und Supervisor Einstellungen vornehmen
  8. Timer setzen für Wind-, PTU-, Niederschlag- und Supervisormessung
  9. Timer starten
  
- Messung tätigen
  1. Timer-Interrupt wird ausgelöst
  2. Entsprechende Messung starten, alle anderen Messungen sperren. Der Sensor gibt die Zeit bekannt, die er braucht um die Messung auszuführen
  3. Gemäss Antwort des Sensors einen Timer setzen. Wird die Zeit (plus Marge) überschritten, wird ein Fehler in der Kommunikation vermutet. Die Messung wird abgebrochen und die anderen Messungen werden wieder frei gegeben
  4. Treten in der Zwischenzeit Interrupts für andere Messungen auf, die in einer Warteschlange vermerken
  5. Wenn der Sensor meldet, dass er die Messung gemacht hat, Messwerte holen
  6. Daten zum GSN senden
  7. Alle Messungen wieder freigeben
  8. Nächste Messungen in der Warteschlange starten
  9. Wenn keine: Power down
  
- Konfiguration ändern
  1. Neue Konfigurationsdatei wird auf dem File-System gespeichert
  2. Software muss neu gestartet werden

---

<sup>1</sup>PTU (Pressure, Temperature and Humidity) steht für die Messung für Druck, Temperatur und Feuchtigkeit



### 7.2.3 Softwarestruktur

Das Programm besteht aus den drei Komponenten serial, weatherstation und webserver.

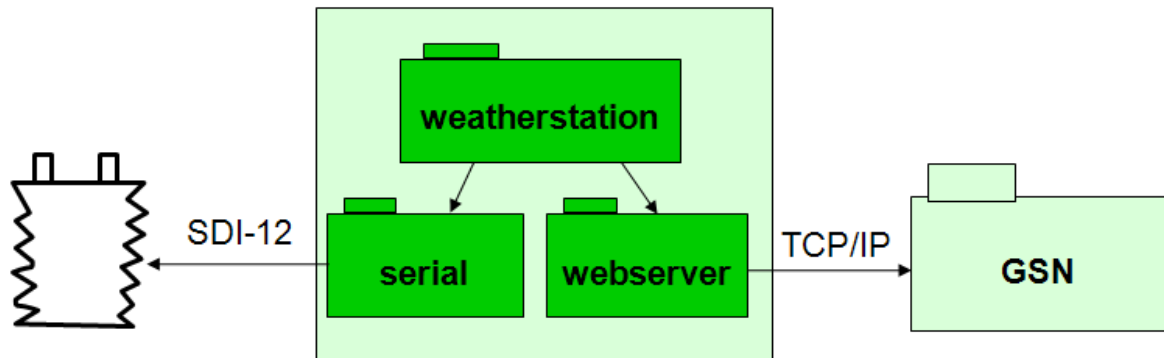


Abbildung 7.1: Softwarestruktur Prototyp

#### Komponente serial

Die Komponente serial ist ein Treiber für die serielle Schnittstelle. Sie besteht unter anderem aus folgenden Klassen:

- SerialIO** Klasse, die anzuwenden ist um Daten über die serielle Schnittstelle zu senden.
- connect** Baut die Verbindung auf.
  - sendData** Sendet Daten über die serielle Schnittstelle. Es wird keine Antwort erwartet.
  - sendAndReceiveData** Wird innerhalb von 500 ms mit einer Antwort gerechnet, kann diese Methode verwendet werden.
  - addDataAvailableListener** Listener zum empfangen von Daten über der seriellen Schnittstelle.

**DataAvailableEvent** Event um zu signalisieren, dass Daten empfangen worden sind.

**DataAvailableEventListener** Diese Klasse muss implementiert werden um Interrupts für den Empfang von Daten über der seriellen Schnittstelle empfangen zu können.

**ComException** Wird bei Fehler in der Kommunikation über die serielle Schnittstelle geworfen.

#### Komponente webserver

In der Komponente webserver ist ein Webserver implementiert, der die Daten für das GSN zu Verfügung stellt. Die Komponente besteht aus folgenden Klassen:

**Server** Kreiert je nach Konfiguration einen Webserver für das GSN oder einen Webserver, der die aktuellen Daten in einer für den Menschen gut lesbaren Form darstellt.

**Database** Die Datenbank, die pro Sensor bis zu 1'000 Messwerte für das GSN speichert.

**WebserverException** Ausnahme, die geworfen wird, wenn ein Fehler im Webserver entsteht.

### Komponente weatherstation

**Main** Liest die Konfigurationsdatei, konfiguriert die Wetterstation, startet zwei Webserver und startet die Timer, die die Messungen starten.

**MeasurementControl** Enthält die run-Methode der Timer. Beim Auslösen eines Timers wird die entsprechende Messung gestartet.

**WeatherStation** Dieses Objekt stellt die Wetterstation dar.

**initialize** Initialisiert die Wetterstation.

**start** Startet den Zustandsautomaten.

**onNewMeasurementToDo** Veranlasst den Zustandsautomaten eine neue Messung zu starten.

**onNewMeasurementAvailable** Veranlasst den Zustandsautomaten neue Daten vom Sensor zu holen.

**startMeasurement** Sendet den Befehl zum Starten der Messung zum Wettersensor.

**getNewDataFromSensor** Holt die Daten vom Wettersensor.

**run** Zustandsautomat (siehe Abbildung 7.2).

**Parser** Wird gebraucht um die Antworten des Wettersensors zu parsen.

**GSN** Sendet die Daten zum GSN.

**Configuration** Enthält die Konfiguration gemäss Konfigurationsfile.

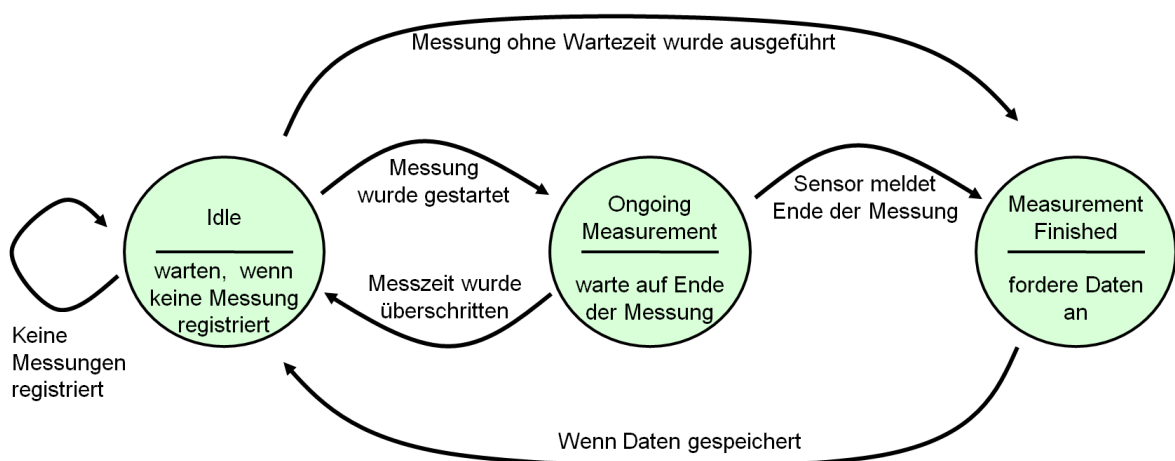


Abbildung 7.2: Zustandsautomat

## 7.2.4 GSN

Tabelle 7.3 zeigt die Parameter, die zum GSN gesendet werden können. *Kursiv* gedruckte Parameter müssen immer mitgesendet werden:

Tabelle 7.3: Parameter, die zum GSN gesendet werden.

	<b>Abk.</b>	<b>Name</b>	<b>Einheit</b>	<b>Datentyp</b>
<b>Wind</b>	<i>Sn</i>	Seriennummer	Text	String
	<i>Wu.T</i>	Zeitstempel der Windmessung <sup>a</sup>		long
	Wu_Dn	Windrichtung Minimum	°	integer
	Wu_Dm	Windrichtung Durchschnitt	°	integer
	Wu_Dx	Windrichtung Maximum	°	integer
	Wu_Sn	Windgeschwindigkeit Minimum	m/s	double
	Wu_Sm	Windgeschwindigkeit Durchschnitt	m/s	double
	Wu_Sx	Windgeschwindigkeit Maximum	m/s	double
<b>PTU</b>	<i>Sn</i>	Seriennummer	Text	String
	<i>Tu.T</i>	Zeitstempel der PTU-Messung <sup>a</sup>		long
	Tu_Pa	Luftdruck	hPa	double
	Tu-Ta	Lufttemperatur	°C	double
	Tu_Tp	Interne Temperatur	°C	double
	Tu_Ua	Relative Feuchtigkeit	%	double
<b>Niederschlag</b>	<i>Sn</i>	Seriennummer	Text	String
	<i>Ru.T</i>	Zeitstempel der Regenmessung <sup>a</sup>		long
	Ru_Rc	Regenmenge	mm	double
	Ru_Rd	Regendauer	sec	integer
	Ru_Ri	Regenintensität	mm/h	double
	Ru_Hc	Hagelmenge	Einschläge/cm <sup>2</sup>	double
	Ru_Hd	Hageldauer	sec	integer
	Ru_Hi	Hagelintensität	Einschläge/cm <sup>2</sup> *h	double
	Ru_Rp	Maximalwert Regenintensität	mm/h	double
	Ru_Hp	Maximalwert Hagelintensität	Einschläge/cm <sup>2</sup> *h	double
<b>Supervisor</b>	<i>Sn</i>	Seriennummer	Text	String
	<i>Su.T</i>	Zeitstempel der Supervisorormessung <sup>a</sup>		long
	Su_Th	Heiztemperatur	°	double
	Su_Vh	Heizspannung	°	double
	Su_Vs	Betriebsspannung	V	double
	Su_Vr	3.5 V Referenzspannung	V	double
	Su_Id	Information	Text	String
	Su_I	Identifikation <sup>b</sup>	Text	String

<sup>a</sup>Anzahl Millisekunden seit 01. Januar 1970, 00:00:00 GMT.

<sup>b</sup>Die Identifikation besteht aus folgenden Elementen: 2 Zeichen SDI-12 Version; 8 Zeichen Lieferantenname; 6 Zeichen Modellnummer; 3 Zeichen Firmware Version; 8 Zeichen Seriennummer.

Die Messintervalle für die vier Sensoren sind konfigurierbar. Auf dem Gumstix laufen zwei Server, die auf zwei verschiedene Ports „hören“. Ein Server stellt die aktuellen Wetterdaten in einer für den Menschen gut lesbaren Form zur Verfügung. Der andere Server liefert die Daten für das GSN gemäss folgendem Algorithmus:

Das GSN generiert am Anfang eine Zufallszahl, die bei jedem HTTP-Request<sup>2</sup> mitgesendet wird. Der Server sendet beim ersten Request die aktuellen Wetterdaten, sammelt ab diesem Zeitpunkt alle Messwerte, die generiert werden (max. 1000 Messwerte) und sendet sie beim nächsten HTTP-Request dem GSN. Ändert sich die Zufallszahl, greift ein anderes GSN auf den Server zu, was nicht unterstützt wird. Der Server löscht alle gesammelten Daten und beginnt wieder von vorne.

Die Nachrichten beginnen mit *Sn* [*Seriennummer*] und *...u-T* [*Zeitstempel*]. Diese zwei Parameter müssen immer gesendet werden. Nun folgen die Messwerte. Jeder Messwert besteht aus der Abk. (siehe Tabelle 7.3), einem Leerzeichen, dem Wert und dem Zeichen „|“. Es müssen nicht immer alle Messwerte gesendet werden. Die Reihenfolge der Messwerte spielt keine Rolle. Jede Messung wird mit einem zusätzlichen „|“ abgeschlossen. Nachfolgend 5 Beispiele:

#### Wind Komplet

```
Sn_E1120013|Wu_T_1242227383300|Wu_Dn_229|Wu_Dm_244|Wu_Dx_279|
Wu_Sn_229|Wu_Sm_244|Wu_Sx_279||
```

#### Druck, Temperatur und Feuchtigkeit

Es müssen nicht alle Parameter angefügt sein

```
Sn_E1120013|Tu_T_1242227417800|Tu-Ta_25.9|Tu_Pa_957.0||
```

#### Niederschlag

Die Reihenfolge spielt keine Rolle

```
Sn_E1120013|Tu_T_1242227568400|Ru_Hd_0|Ru_Rc_0.02|Ru_Rd_50|Ru_Hc_0.0|
Ru_Ri_0.0|Ru_Rp_1.4|Ru_Hp_0.0|Ru_Hi_0.0||
```

#### Supervisor

Diese Parameter sind obligatorisch

```
Sn_E1120013|Su_T_1242227687800||
```

#### Mehrere Messungen

Die Messungen sind durch ein zusätzliches „|“ voneinander getrennt

```
Sn_E1120013|Tu_T_1242227417800|Tu-Ta_25.9|Tu_Pa_957.0||Sn_E1120013|
Tu_T_1242227758300|Tu-Ta_26.1|Tu_Pa_956.5||Sn_E1120013|
Tu_T_1242228053700|Tu-Ta_26.3|Tu_Pa_957.0||
```

Auf der GSN-Seite werden die Daten empfangen und geparsed. Dazu wurde der Wrapper *WeatherStationWrapper.java* geschrieben. Dieser Wrapper kann für alle vier Sensortypen (*WeatherStation\_Wind*, *WeatherStation\_PTU*, *WeatherStation\_Rain* und *WeatherStation\_Supervisor*) verwendet werden. Bei der Initialisierung müssen dem Wrapper, folgende Werte übergeben werden:

**measurement** Name des Sensors (wind, ptu, precipitation oder supervisor).

**url** Adresse des Webservers auf dem Gumstix.

**rate** Periodendauer in ms, mit der der Webserver abgefragt werden soll.

<sup>2</sup>Das Hypertext Transfer Protocol (HTTP, dt. Hypertext-Übertragungsprotokoll) ist ein Protokoll zur Übertragung von Daten über ein Netzwerk. Es wird hauptsächlich eingesetzt, um Webseiten aus dem World Wide Web (www) in einen Webbrowser zu laden.

Nachfolgend ist ein Beispiel aufgeführt, wie die Definition des Wrappers in einem virtuellen Sensor aussehen kann:

---

```
<address wrapper="weatherStation">
  <predicate key="measurement">supervisor</predicate>
  <predicate key="url">http://forecast:52000</predicate>
  <predicate key="rate">120000</predicate>
</address>
```

---

Listing 7.2: Definition des Wrappers weatherStation in einem virtuellen Sensor

### 7.2.5 Kommandozeilen Parameter

Die Software kann über Kommandozeilen-Argumente beeinflusst werden. Listing 7.3 zeigt die Ausgabe von `runnablePrototyp --help`. Zusätzliche Informationen über den Programmablauf können mit der Option `-d` eingeschaltet werden. `-db` ist dazu gedacht, die Software zu Debug-Zwecken zu einem anderen Verhalten zu veranlassen. In der Endversion der Software hat dieser Parameter keinen Einfluss. Über die Option `-gp` wird der Port für den GSN-Webserver eingestellt. Standardmässig wird der Port 52000 genutzt. Der Port für den zweiten Webserver, der die aktuellen Daten in einer für den Menschen gut lesbaren Form darstellt, kann über die Option `-hp` eingestellt werden. Standardmässig wird der Port 80 genutzt, weil das der standardmässige Port für Webserver mit HTTP-Request ist. Die serielle Schnittstelle wird mit der Option `-sp` eingestellt. Standardmässig wird COM5 für Windows und `/dev/ttyS1` für Linux verwendet. Wird die Software auf Windows ausgeführt, muss die Option `-w` eingegeben werden.

---

```
usage: WeatherStation
-?,--help          Print usage
-d,--debug         Debug outputs
-db,--diffBeh     Different behave (for debug only)
-gp,--gsnport <arg> Port to use for GSN (if nothing defined ,
                    port 52000 is used)
-hp,--hmiport <arg> Port to use for HMI (if nothing defined ,
                    port 80 is used)
-sp,--serialport <arg> Port to use for Serial (if nothing
                    defined , COM5 is used for Windows and
                    /dev/ttyS1 for Linux)
-w,--windows      Operating system is Windows. If Option not
                    set , Operating System is Linux
```

---

Listing 7.3: Kommandozeilen Parameter

## 7.3 Endsystem (auf TinyNode)

### 7.3.1 Konfigurierbare Parameter

Folgende Parameter sind konfigurierbar:

Tabelle 7.4: Konfigurierbare Parameter auf Endsystem.

Parameter	Mögliche Werte	Erklärung
Wu	Y/N	Windmessung ein- bzw. ausschalten
Ru	Y/N	Niederschlagsmessung ein- bzw. ausschalten
Su_H	Y/N	Heizungsregelung ein- bzw. ausschalten
Ru_Z	Y/N	Niederschlagszähler zurücksetzen

### 7.3.2 Ablauf

SENDE kennzeichnet die Befehle, die vom TinyNode zum Wettersensor gesendet werden, EMPFANGE kennzeichnet die Antworten vom Wettersensor.

Die Werte in [eckigen Klammern] sind Parameter.

Nach jeder Aktivität der RS232-Schnittstelle muss 20 ms gewartet werden, bevor neue Daten zum Wettersensor gesendet werden können.

**System starten** Bei jedem Reset

- Konfigurierbare Parameter lesen:  
Siehe Tabelle 7.4.
- Serielle Schnittstelle einrichten:  
Für SDI-12 gilt: Bitrate 1200, 7 Datenbits, Parität gerade, 1 Stopbit.
- Gerätenummer des Wettersensors lesen:  
SENDE            ?!  
EMPFANGE   [a]<cr><lf>  
[a]: Gerätenummer des Wettersensors
- Kommunikationseinstellungen vornehmen (auf SDI-12 native setzen):  
SENDE           [a]XXU,M=S,C=1,I=0!  
EMPFANGE   [a]<cr><lf>
- Alle Messungen stoppen, Wettersensor resettet:  
SENDE           [a]XZ!  
EMPFANGE   [a]<cr><lf>  
SENDE           [a]XZ!  
Keine Antwort. Warte ca. 50 ms.
- Seriennummer lesen:  
SENDE           [a]!  
EMPFANGE   [a][I]<cr><lf>  
[I] ist die Identifikation und hat folgende Struktur: [sccccccmmmmmmv-vxxxxxxx].  
Die 8 Zeichen am Schluss (xxxxxxx) sind die Seriennummer [Sn].

## 7. Wind-Einstellungen vornehmen:

Muss nur gemacht werden, wenn  $Wu$  in Tabelle 7.4 auf der vorherigen Seite Y ist.

SENDE [a]XWU,R=11111100&00000000!

EMPFANGE [a]<cr><lf>

SENDE [a]XWU,A=3,G=3,U=M,D=0,F=2!

EMPFANGE [a]<cr><lf>

## 8. Druck-, Temperatur- und Feuchtigkeits-Einstellungen vornehmen:

SENDE [a]XTU,R=11110000&00000000!

EMPFANGE [a]<cr><lf>

SENDE [a]XTU,P=H,T=C!

EMPFANGE [a]<cr><lf>

## 9. Niederschlags-Einstellungen vornehmen:

Muss nur gemacht werden, wenn  $Ru$  in Tabelle 7.4 auf der vorherigen Seite Y ist.

SENDE [a]XRU,R=11111111&00000000!

EMPFANGE [a]<cr><lf>

SENDE [a]XRU,U=M,S=M,Z=M!

EMPFANGE [a]<cr><lf>

## 10. Supervisor-Einstellungen vornehmen:

SENDE [a]XSU,R=11111000&00000000!

EMPFANGE [a]<cr><lf>

SENDE [a]XSU,S=Y,H=[Su\_H]!

EMPFANGE [a]<cr><lf>

[Su\_H]: Wert von  $Su_H$  in Tabelle 7.4 auf der vorherigen Seite

**Messung tätigen** Alle 2 Min. durchführen

## 1. Windmessung:

Muss nur gemacht werden, wenn  $Wu$  in Tabelle 7.4 auf der vorherigen Seite Y ist.

SENDE [a]M1!

EMPFANGE [a][ttt][n]<cr><lf>

[ttt]: Messdauer in Sekunden

[n]: Anzahl verfügbare Parameter

Nach ttt Sekunden:

EMPFANGE [a]<cr><lf>

Die folgenden Schritte sind so lange zu wiederholen, bis alle [n] Parameter empfangen sind. Der Anfangswert von [x] ist 1 und wird bei jedem Durchgang inkrementiert.

SENDE [a]D[x]!

EMPFANGE [a]+[Daten]<cr><lf>

[Daten]: Gemessene Parameter in der Reihenfolge  $Wu\_Dn$   $Wu\_Dm$   $Wu\_Dx$   $Wu\_Sn$   $Wu\_Sm$   $Wu\_Sx$ , durch + oder - getrennt.

## 2. Temperatur-, Feuchtigkeit- und Druckmessung:

SENDE [a]M2!

EMPFANGE [a][ttt][n]&lt;cr&gt;&lt;lf&gt;

[ttt]: Messdauer in Sekunden

[n]: Anzahl verfügbare Parameter

Nach ttt Sekunden:

EMPFANGE [a]&lt;cr&gt;&lt;lf&gt;

Die folgenden Schritte sind so lange zu wiederholen, bis alle [n] Parameter empfangen sind. Der Anfangswert von [x] ist 1 und wird bei jedem Durchgang inkrementiert.

SENDE [a]D[x]!

EMPFANGE [a]+[Daten]&lt;cr&gt;&lt;lf&gt;

[Daten]: Gemessene Parameter in der Reihenfolge Tu-Ta Tu-Tp Tu-Ua Tu-Pa durch + oder - getrennt.

## 3. Niederschlagsmessung:

Muss nur gemacht werden, wenn *Ru* in Tabelle 7.4 auf Seite 52 Y ist.

SENDE [a]M3!

EMPFANGE [a]000[n]&lt;cr&gt;&lt;lf&gt;

[n]: Anzahl verfügbare Parameter

Die folgenden Schritte sind so lange zu wiederholen, bis alle [n] Parameter empfangen sind. Der Anfangswert von [x] ist 1 und wird bei jedem Durchgang inkrementiert.

SENDE [a]D[x]!

EMPFANGE [a]+[Daten]&lt;cr&gt;&lt;lf&gt;

[Daten]: Gemessene Parameter in der Reihenfolge Ru-Rc Ru-Rd Ru-Ri Ru-Hc Ru-Hd Ru-Hi Ru-Rp Ru-Hp durch + oder - getrennt.

## 4. Supervisor:

SENDE [a]M5!

EMPFANGE [a][ttt][n]&lt;cr&gt;&lt;lf&gt;

[ttt]: Messdauer in Sekunden

[n]: Anzahl verfügbare Parameter

Nach ttt Sekunden:

EMPFANGE [a]&lt;cr&gt;&lt;lf&gt;

Die folgenden Schritte sind so lange zu wiederholen, bis alle [n] Parameter empfangen sind. Der Anfangswert von [x] ist 1 und wird bei jedem Durchgang inkrementiert.

SENDE [a]D[x]!

EMPFANGE [a]+[Daten]&lt;cr&gt;&lt;lf&gt;

[Daten]: Gemessene Parameter in der Reihenfolge Su-Th Su-Vh Su-Vs Su-Vr Su-Id durch + oder - getrennt.

## 5. Daten zum GSN senden:

Siehe dazu Absatz 7.3.3 auf der nächsten Seite.



**Konfiguration ändern** Siehe auch Tabelle 7.4

1. Neue Konfigurationsparameter empfangen:
2. Wenn  $Wu = Y$ , führe Schritt 7 von „System starten“ aus.
3. Wenn  $Ru = Y$ , führe Schritt 9 von „System starten“ aus.
4. Heizungsregelung ein- bzw. ausschalten:  
SENDE           [a]XSU,H=[Su\_H]!  
EMPFANGE   [a]<cr><lf>  
[Su\_H]: Wert von  $Su_H$  in Tabelle 7.4
5. Wenn  $Ru_Z = Y$ :  
SENDE           [a]XZRU!  
EMPFANGE   [a]<cr><lf>

**7.3.3 GSN**

Es werden alle Daten im Abstand von zwei Minuten per SerialForwarder zum GSN gesendet. Wenn die Wind- bzw. die Niederschlagsmessung ausgeschaltet ist, wird sie nicht zum GSN gesendet (siehe Tabelle 7.4 auf Seite 52). Die einzelnen Parameter bleiben gleich wie beim Prototyp und sind in Tabelle 7.3 auf Seite 49 aufgeführt.

## 7.4 Befehle für die Wetterstation

Tabelle 7.5 stellt die verwendeten Befehle für den Wettersensor dar. Es werden nur die SDI-12 Befehle aufgeführt. Die Tabelle ist nicht vollständig. Für weitere Befehle und um die Einstellungen zu ändern siehe im Handbuch [2] (Absätze „Communication Setting Commands“ ab Seite 59, „General Commands“ ab Seite 66, „SDI-12 Protocol“ ab Seite 80 sowie Kapitel „Sensor and Data Message Settings“ ab Seite 107).

[a] ist die Gerätenummer des Wetter-Sensors.

Tabelle 7.5: Verwendete Befehle für den Wettersensor WXT520.

ASCII	Bedeutung	Seitenzahl in [2]
[a]XXU!	Kommunikationsparameter einstellen	60 ff
[a]XZ!	Software reset	66
[a]XZRU!	Niederschlagszähler zurücksetzen	67
[a]XZRI!	Niederschlagsintensität zurücksetzen	68
[a]XZM!	Alle Messungen stoppen	69
?!	Gerätenummer abfragen	81
[a]!	Quittierung	81
[a]A[b]!	Gerätenummer auf b ändern	82
[a]I!	Identifikation senden	83
[a]M[x]!	Messung starten	84
[a]D[x]!	Daten holen	87
[a]R[x]!	Kontinuierliche Messung	91
[a]XWU!	Windsensor Einstellungen	107 ff
[a]XTU!	Druck-, Temperatur- und Feuchtigkeitseinstellungen	113 ff
[a]XRU!	Niederschlagsensor Einstellungen	116 ff
[a]XSU!	Supervisor Einstellungen	122 ff

# Kapitel 8

## Zusammenbau/Inbetriebnahme

In diesem Kapitel wird auf den mechanischen Aufbau des Prototyps eingegangen.

Die Solarzelle wird fest an die Stange montiert. Die restlichen Komponenten sind noch nicht weiter verarbeitet worden. Dies wurde so mit dem Betreuer abgesprochen. Folgend ein paar Bilder, wie die provisorische Montage aussieht.

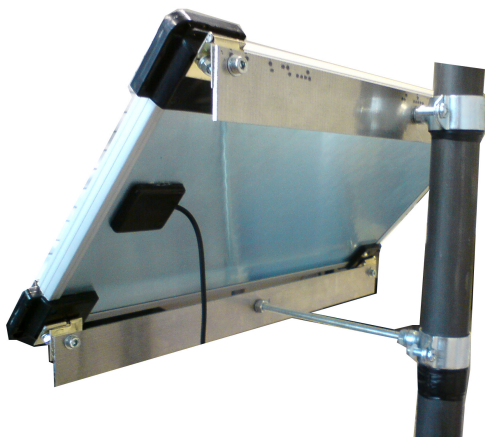


Abbildung 8.1: Befestigung Solarzelle



Abbildung 8.2: Ganzer Testaufbau

Der Laderegler, der Akku und der Gumstix sind in Plastiksäcke eingepackt und in der Kiste unter dem Stuhl verstaut. Das Problem bei dem Aufbau ist, dass der Gumstix ca. 150 mA verbraucht und die Solarzelle die meiste Zeit des Tages im Schatten steht. So kann die Anlage nicht betrieben werden. Deswegen wird der Gumstix extern gespeist. Folgend einige Strommessungen vor Ort, dabei stand die Solarzelle im Sonnenlicht.



Abbildung 8.3: Strommessung Versuch 1

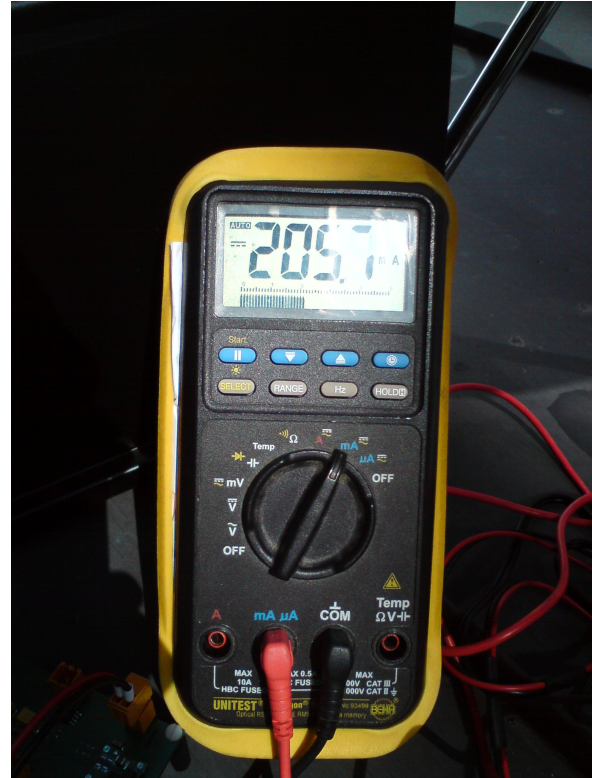


Abbildung 8.4: Strommessung Versuch 2

Gemessen wurde der Strom im Draht der Solarzelle bei verschiedenen Einstrahlungen. Auf der Skala ist die Einheit nicht gut ersichtlich, sie ist mA.

# Anhang A

## Pflichtenheft

Vermerk:

[P] gilt nur für Prototyp

[E] gilt nur für Endsystem

ist nichts geschrieben, gilt es für beide Systeme

### A.1 Zielbestimmung

#### A.1.1 Musskriterien

- Prototyp
  - Der Prototyp besteht aus dem Wettersensor (Vaisala WXT520) und einer Basisstation (Gumstix Verdex Embedded Linux System).
  - Die Basisstation kommuniziert über eine serielle Schnittstelle mit dem Wettersensor.
  - Die Wetterstation ist konfigurierbar.
  - Die Messwerte werden zum GSN<sup>1</sup> übermittelt und dort gespeichert.
- Endsystem
  - Das Endsystem besteht aus dem Wettersensor (Vaisala WXT520), einer autonomen Stromversorgung und einem Sensorknoten vom Typ TinyNode mit einem Sensor Interface Board (SIB).
  - Der TinyNode kommuniziert über eine serielle Schnittstelle mit dem Wettersensor.
  - Die Wetterstation ist konfigurierbar.
  - Die Wetterstation ist im bestehenden funkbetriebenen Sensornetzwerk von PermaSense<sup>2</sup> integriert.
  - Die Messwerte werden zum GSN übermittelt und dort gespeichert.
  - Die Stromversorgung besteht aus einer Solaranlage.

---

<sup>1</sup>Global Sensor Networks (GSN) ist eine Software Middleware, die entworfen wurde um die Entwicklung von Sensor Netzwerken zu vereinfachen

<sup>2</sup>Das Projekt PermaSense bezweckt die Entwicklung und Demonstration eines flexiblen, verteilten Drahtlossensornetzwerkes (WSN) für hoch-alpine Wetterbedingungen, das an hochwertige, betriebssichere geophysikalische Sensoren angepasst ist

### A.1.2 Wunschkriterien

- Prototyp
  - Als Kommunikationsprotokoll wird SDI-12 verwendet.
- Endsystem
  - Keine

### A.1.3 Abgrenzungskriterien

- Die Implementierung der notwendigen Software auf dem TinyNode unter TinyOS/PermaDozer wird nach unseren Vorgaben vom Institut übernommen.
- Bereits bestehende Software/Hardware darf und soll genutzt werden.

## A.2 Produkteinsatz

### A.2.1 Anwendungsbereiche

Die Wetterdaten am Aufstellungsort von PermaSense werden für folgende Fälle gebraucht:

- Für die Funktions-Fernüberwachung, für die Fernwartung und für die Erstdiagnose des Systems.
- Verfügbarkeit eines aktuellen Wetterdatensatz im aktuellen Mikroklima.

### A.2.2 Betriebsbedingungen

- Die Wetterstation wird im hochalpinen Gebiet ( $< 5'000$  m) an exponierter Lage eingesetzt.
- Sonnige Lage.
- Betriebsdauer: täglich 24 Stunden.
- Vor-Ort-Wartung ist nur mit einem immensen Aufwand möglich.

## A.3 Produktumgebung

Folgende Bedingungen müssen erfüllt sein, damit das Produkt eingesetzt werden kann:

### A.3.1 Allgemein

- Schutz vor Umweltgefahren (Blitzschutz, Steinschlag, Lawinenniedergang, Überschwemmung)
- Temperatur: -52 ... +60 °C
- Druck: 600 ... 1100 hPa
- Wind: 0 ... 60 m/s

### A.3.2 Prototyp

- Stromversorgung
- Internet-Anschluss für Gumstix

### A.3.3 Endsystem

- PermaSense Netzwerk
- Sonnige Lage

## A.4 Produktfunktionen

### A.4.1 Konfigurationsdatei [P]

- Die Funktionen des System sind über eine Konfigurationsdatei steuerbar.
- Die Konfigurationsdatei ist eine XML-Datei.
- Die Konfigurationsdatei wird beim Starten des Systems eingelesen.
- Der Benutzer kann im Betrieb aus der Ferne die Konfigurationsdatei austauschen.
- Die neuen Funktionen werden nach einem Reset des Systems ausgeführt.

### A.4.2 Funktionen

- Allgemeines
  - Bei der Software kann ein Reset gemacht werden.
  - Der Niederschlagszähler wird bei einem Reset zurückgesetzt (Regenmenge, Hagelmenge, Regendauer, Hageldauer).
  - Die Niederschlags-Intensitätsspitze wird bei einem Reset zurückgesetzt (Regen- und Hagel Maximalwert).
  - Alle aktuellen Messvorgänge können abgebrochen und neu gestartet werden.

- Wind
  - Die Windmessung ist ein-/ausschaltbar.
  - Jeder Parameter kann einzeln ein/ausgeschaltet werden (Richtung min, Richtung Durchschnitt, Richtung max, Geschwindigkeit min, Geschwindigkeit Durchschnitt, Geschwindigkeit max) [P]
  - Alle Parameter sind eingeschaltet [E]
  - Das Mess-Intervall ist einstellbar (10... 3600 s) [P]
  - Das Mess-Intervall ist 120 s [E]
  - Die Zeit für die Durchschnittsberechnung ist einstellbar (siehe Handbuch [2], Appendix D). [P]
  - Die Zeit für die Durchschnittsberechnung ist 3 s [E]
  - Der Algorithmus zur Berechnung der min/max Windgeschwindigkeit ist einstellbar (siehe Handbuch [2], Absatz „Wind Measurement Principle“ auf Seite 25). [P]
  - Der Algorithmus zur Berechnung der min/max Windgeschwindigkeit ist  $G=3$  [E]
  - Die Windgeschwindigkeit wird in  $\text{m/s}$  angegeben.
  - Der Richtungs-Offset ist einstellbar. [P]
  - Der Richtungs-Offset ist  $0^\circ$ . [E]
  - Die Abtastfrequenz ist einstellbar (1, 2, 4 Hz) [P]
  - Die Abtastfrequenz ist 2 Hz [E]
  
- Druck, Temperatur und Feuchtigkeit
  - Die Messung ist ein-/ausschaltbar. [P]
  - Die Messung ist eingeschaltet. [E]
  - Jeder Parameter kann einzeln ein/ausgeschaltet werden (Druck, Temperatur, Feuchtigkeit). [P]
  - Alle Parameter sind eingeschaltet [E]
  - Das Mess-Intervall ist einstellbar (10... 3600 s) [P]
  - Das Mess-Intervall ist 120 s [E]
  - Der Druck wird in hPa angegeben
  - Die Temperatur wird in  $^\circ\text{C}$  angegeben



- Niederschlag
  - Die Messung ist ein-/ausschaltbar.
  - Jeder Parameter kann einzeln ein/ausgeschaltet werden (Regenmenge, Regendauer, Regenintensität, Hagelmenge, Hageldauer, Hagelintensität, Regen peak, Hagel peak). [P]
  - Alle Parameter sind eingeschaltet [E]
  - Das Mess-Intervall ist einstellbar (10... 3600 s) [P]
  - Das Mess-Intervall ist 120 s [E]
  - Einheiten beim Regen: Menge in mm, Dauer in s, Intensität in mm/h
  - Einheiten bei Hagel: Menge in  $\frac{\text{Einschläge}}{\text{cm}^2}$ , Dauer in s, Intensität in  $\frac{\text{Einschläge}}{\text{cm}^2 \cdot \text{h}}$
  - Die Regenmenge/Hagelmenge kann zurückgesetzt werden [E]
- Heizung, Supervisor
  - Die Messung ist ein-/ausschaltbar. [P]
  - Die Messung ist eingeschaltet. [E]
  - Jeder Parameter kann einzeln ein/ausgeschaltet werden (Heiztemperatur, Heizspannung, Betriebsspannung, 3.5 V Referenzspannung, Information). [P]
  - Alle Parameter sind eingeschaltet [E]
  - Das Mess-Intervall ist einstellbar (10... 3600 s). [P]
  - Das Mess-Intervall ist 120 s. [E]
  - Wenn die Heizung eingeschaltet ist, wird das Mess-Intervall auf 15 s gesetzt.
  - Die Heizung kann ausgeschaltet werden.

## A.5 Produktdaten

Die Daten werden in den vorgesehenen Intervallen abgefragt und zwischengespeichert. So schnell wie möglich werden sie dann über GSM zum GSN gesendet. Es werden folgende Daten gespeichert:

- Konfigurationsdatei auf dem Prototyp.
- Alle Messwerte, wie im Abschnitt A.4.2 definiert

## A.6 Produktleistungen

Abgabetermin ist der 08.06.2009

## A.7 Benutzeroberfläche

Die Benutzeroberfläche wird durch das GSN zur Verfügung gestellt und ist nicht Teil der Semesterarbeit.

## **A.8 Energieversorgung**

Die Energieversorgung besteht aus einer Batterie, einer Solarzelle und einem Laderegler.

### **A.8.1 Solarzelle**

Die Dimensionierung der Solarzelle muss so erfolgen, dass der Akku geladen und das Gerät betrieben werden kann. Ausserdem, ist darauf zu achten, dass die Solarzelle, den leeren Akku, innert nützlicher Frist, wieder laden kann.

### **A.8.2 Akku**

Der Akku:

- Sollte die Wetterstation über mehrere Tage mit Strom versorgen können, auch wenn keine Sonne scheint.
- Muss auch bei  $-20^{\circ}\text{C}$  Leistung abgeben können.
- Muss eine möglichst lange Lebenszeit aufweisen.
- Soll mit einem Laderegler geladen werden.

## **A.9 Ergänzungen**

Keine

# Anhang B

## CD Inhalt

- code/gsn
  - Quellcode des GSN
- code/helloworld
  - Hello World Programm
- code/Prototyp
  - Software des Prototypen
- code/Prototyp/doc
  - Dokumentation der Software**
- code/Prototyp/src
  - Quellcode der Software für den Prototyp
- code/Prototyp/Configurationfile.xml
  - Konfigurationsdatei für die Software**
- code/Prototyp/runnablePrototyp
  - Ausführbares Programm**
- code/Readme
  - Anweisungen um die das Projekt ins Eclipse zu importieren
- code/wview-5.1.3
  - Software wview
- docs/Auswertung
  - Auswertung der Messungen
- docs/Datenblätter
  - Datenblätter der Solaranlage und des Sensors
- docs/Documentation
  - $\LaTeX$  Quellcode für den Schlussbericht
- docs/Documentation/Schlussbericht.pdf
  - Schlussbericht dieser Semesterarbeit**
- docs/GSN
  - Dokumentation und Libraries für das GSN
- docs/Literaturrecherche
  - Recherchierte Websites
- docs/Organisatorisch
  - Dateien zur Arbeitsplanung
- docs/PermaSense

- Dokumentation zu PermaSense
- docs/Photovoltaic
  - Recherchen zu Photovoltaik
- docs/Vorträge
  - Unterlagen zu den Vorträgen
- docs/Vorträge/Kurzvortrag.pptx
  - Einführungspräsentation
- docs/Vorträge/Präsentation.pptx
  - Abschlussvortrag**
- docs/WetterZermatt
  - Wetterdaten von Zermatt
- Messungen20080409
  - Daten, Grafiken, Matlab- und CSV-Files der getätigten Messungen**
- soft Verschiedene benötigte Software und Treiber

# Literaturverzeichnis

- [1] E. Zitzler. Studien- und Diplomarbeiten, Merkblatt für Studenten und Betreuer. computer Engineering and Networks Lab, ETH Zürich, Switzerland, March 1998
- [2] Vaisala Oyj, USER'S GUIDE Vaisala Weather Transmitter WXT520, FIN-00421 Helsinki, 2008