

# Rock Monitoring

Bachelorthesis

Studiengang:	Elektro- und Kommunikationstechnik
Autor:	Swen Bischof, Daniel Frei
Betreuer:	Andrea Ridolfi
Auftraggeber:	Andrea Ridolfi
Experten:	Guillaume Maret
Datum:	30.06.2016

## Management Summary

Immer wieder hört man in den Nachrichten, dass Strassen oder Zugstrecken aufgrund von Felsstürzen gesperrt werden mussten. Damit solche Vorfälle möglichst verhindert werden können, braucht es Messvorrichtungen. Diese messen die kleinen Felsbewegungen, welche auf eine Instabilität des Felsen hinweisen. Dadurch können noch vor dem Felssturz Massnahmen getroffen werden um Unfälle vorzubeugen.

Die bestehenden Messvorrichtungen weisen zwei grosse Schwachstellen auf, welche im Rahmen dieses Projektes behoben werden sollen. Die eine Schwachstelle betrifft die Messung. Aktuelle Lösungen messen lediglich die Ausdehnung von Felsspalte, damit aber eine genauere Prognose über die vom Felsen ausgehende Gefahr erstellt werden kann, werden mehr Messdaten benötigt. Eine weitere Schwachstelle betrifft die Übermittlung der Messdaten von der Messvorrichtung zu einer Auswertungseinheit. Diese Messdaten werden momentan mit einem Kabel, welches zuerst mit viel Aufwand verlegt werden muss, übermittelt.

Ziel dieses Projektes ist es, die momentan kabelgebundene Kommunikation zwischen Messvorrichtung und Auswertungseinheit durch eine drahtlose zu ersetzen. Es sollen zwei Messsystemkonfigurationen entwickelt werden. Die erste Messsystemkonfiguration besteht aus einer einzelnen Messvorrichtung welche die Messdaten direkt auf einen Webserver hochlädt. Die zweite Konfiguration besteht aus einem Netzwerk von Messvorrichtungen, welche die Messdaten an einen zentralen Datenlogger senden. Dieser sammelt die Messdaten aller Messvorrichtungen und leitet diese dann an einen Webserver weiter. Ein weiteres Ziel betrifft die Messung selbst. Es soll nebst der eindimensionalen Ausdehnungsmessung durch weitere Sensoren möglich sein die Lagenänderung des Felsens genauer festzustellen. Diese zusätzlichen Sensoren sollen möglichst kompakt und kostengünstig sein. Das Hauptziel des Projektes ist es, einen fertigen Prototyp zu entwickeln, der obengenannte Ziele erfüllt und mit dem erste Messungen möglich sind.

Alle oben genannten Ziele konnten erreicht werden. Es besteht ein fertiger Prototyp welcher in Labormessungen alle gewünschten Funktionen erfüllt. Es wird nebst der Längsausdehnung noch zusätzlich der Pitch- und der Rollwinkel beider Felsen und des Ausdehnungssensors gemessen. Die Einzelanwendung der Messeinrichtung funktioniert und die Messdaten werden erfolgreich auf den Webserver geladen. Mehrere Messvorrichtungen können in einem Netzwerk zusammengefasst werden und ihre Messdaten an den Datenlogger senden. Auch bei dieser Systemkonfiguration funktioniert der Upload der Daten auf den Webserver problemlos.

# Inhaltsverzeichnis

1	Einleitung	5
1.1	Einleitung ins Thema	5
1.2	Bestehende Lösungen	5
2	Projektziele	7
3	Entwicklungsumgebung	8
4	Messung	9
4.1	Messung Felsbewegungen mit Sensoren	9
4.2	Ausdehnungsmessung	9
4.3	Winkelmessung mit Beschleunigungssensoren	10
4.3.1	Winkeldefinitionen	10
4.3.2	Einfache Winkelberechnung mit einer Achse	10
4.3.3	Winkelberechnung mit 3 Achsen	11
4.4	Beschleunigungssensor	12
4.4.1	Theoretische Genauigkeit	13
4.4.2	Rauschen	13
4.4.3	Temperaturverhalten	15
5	Datenübertragung	16
5.1	Differentielles Bussystem	16
5.2	I <sup>2</sup> C Bus	17
6	Systemübersicht	18
6.1	Systemkonfiguration 1	18
6.2	Systemkonfiguration 2	19
7	Kommunikation	20
7.1	Sensornetzwerk	20
7.1.1	Netzwerktopologie	20
7.1.1.1	Ausgewählte Netzwerktopologie	21
7.1.2	Kommunikationsstandard	22
7.1.2.1	Ausgewählter Kommunikationsstandard Gateway <-> Sensorknoten	23
7.1.3	Zugriffsverfahren	24
7.1.3.1	TDMA	24
7.1.3.2	FDMA	24
7.1.3.3	CDMA	24
7.1.3.4	Auswahl des Accessverfahrens	25
7.1.4	Spezifikationen der Netzwerk-Kommunikation	26
7.1.5	Bandspreiztechnik	27
7.2	Kommunikation einzelner Sensorknoten	28
7.2.1.1	Ausgewählter Kommunikationsstandard Gateway <-> Webserver	28
7.2.1.2	Ausgewählter Kommunikationsstandard Sensorknoten <-> Webserver	28
7.2.2	Kommunikation über 3G	29
8	Hardware	30
8.1	Hardwareübersicht	30
8.1.1	Sensorprinte	32
8.1.2	Adapterprint	32
8.1.3	Beschaltung PCA9615	33
8.1.4	I <sup>2</sup> C Bus	33
8.1.4.1	I <sup>2</sup> C Pegel	34
8.2	Laderegler	34
8.2.1	Anschlüsse	35
8.3	Solarpanel	36
8.4	Gehäuse, Mechanik, Wetterbeständigkeit	37
8.5	Bedienelemente	37
8.5.1	Schalter und Led in Schalter	37

8.5.2 Grüne Led	38
8.6 Serielle Schnittstelle	38
8.7 Inbetriebnahme	38
8.7.1 Sensorprint	38
8.7.2 Ladereglerprint	38
8.7.3 Modifikation Or-Gatter und Dioden	39
9 Software	40
9.1 File Struktur	40
9.2 Übersicht eigene Softwarelibrary	40
9.3 Programmieren	42
9.4 Kommunikationsablauf	42
9.4.1 Sensorknoten	42
9.4.2 Sensornetzwerk	43
9.4.2.1 Data-Request empfangen	45
9.5 Upload der Daten auf Webserver über 3G	45
9.6 Server	46
9.7 Software auf Server	46
10 Optimierungen	47
11 Weiterführung der Arbeit	48
12 Zusammenfassung	49
13 Abbildungsverzeichnis	50
14 Tabellenverzeichnis	51
15 Glossar	52
16 Quellenverzeichnis	53
17 Selbständigkeitserklärung	55
18 Anhang	56

# 1 Einleitung

## 1.1 Einleitung ins Thema

Wie vieles unterläuft auch unsere Umwelt einem stetigen Wechsel, oftmals sollten diese Veränderungen jedoch nicht unbemerkt von statten gehen. An Stellen oberhalb einer Bahnstrecke oder einer gefährdeten Felswand oberhalb eines Wohngebiets möchte man genau wissen, wie sich die Lage der Felsen verändert, um allenfalls die Möglichkeit zu haben Gegenmaßnahmen früh genug einzuleiten noch bevor es zu einem Vorfall kommt.

Dieser Bericht hat zum Ziel einen Überblick über den aktuellen Projektstand zu geben und die Vorgehensweise bei der Entwicklung des Prototyps festzuhalten. Die Struktur des Berichts ist so aufgebaut, dass zuerst das Messverfahren, als zweites die Übermittlung dieser Messdaten lokal und zum Schluss die Übertragung der Messdaten zu einer Auswertungseinheit erklärt werden. Die Dafür nötige Hardware und Software wird zum Schluss des Berichts erklärt.

In diesem Bericht wird für den Begriff Datalogger häufig auch der Begriff Gateway verwendet. Ebenso ist der Begriff Messeinrichtung und Sensorknoten gleichzusetzen. Die Begriffe Sensorstab und Ausdehnungssensor beschreiben ebenfalls dasselbe.

## 1.2 Bestehende Lösungen

Die Messvorrichtungen welche momentan zur Felsüberwachung eingesetzt werden bestehen aus sogenannten Rissmesser. Diese Rissmesser ermöglichen eine Messung der Längenveränderung und durch Kombination von drei solchen Rissmessern kann auch die Veränderung in allen drei Achsen festgestellt werden. Diese Rissmesser entsprechen einem linearen Potentiometer.



Abbildung 1 Rissmesser (links) [1] und Rissmesser 3 Dimensional (rechts) [2]

Diese Sensoren werden per Kabel mit einem Datalogger verbunden welcher die gesammelten Messdaten mit GSM oder einer anderen Mobilfunktechnik an eine Auswertungseinheit weiterleitet.

Die Installation dieser Messanlagen erfordert eine genaue Planung, besonders der Verkabelung des Datenloggers und der Sensoren muss Beachtung geschenkt werden. Denn beim Verlegen dieser Kabel muss berücksichtigt werden, dass die Kabel nicht durch herabstürzende Felsen beschädigt werden können, da ansonsten die Anlage unbrauchbar wird. Das Verlegen der Kabel nimmt daher sehr viel Zeit in anspruch und verursacht dementsprechend hohe Installationskosten. Damit bei der Felsüberwachung eine möglichst gute Prognose des Gefahrenpotentials gemacht werden kann, werden so viel Messdaten wie möglich benötigt. Doch durch die hohen Installationskosten wird das Anbringen mehrerer solcher Messvorrichtung zu kostspielig.



Abbildung 2 Messanlage mit Rissmesser [3]

Damit diese Systeme autonom von jeglichem Stromnetz eingesetzt werden können, verfügen viele ein Solarpanel welches das System über einen längeren Zeitraum mit der nötigen Energie versorgen kann. Das gesamte Messsystem ist starken Temperaturschwankungen und der Witterung ausgesetzt. Dies verlangt ein durchdachtes und robustes Design des gesamten Messsystems

## 2 Projektziele

Grundsätzlich ist das Ziel dieses Projektes die Entwicklung eines Prototyps, der die Bewegung von Felsen messen kann. Die bestehenden Lösungen haben die Hauptprobleme, dass sie nicht genügend Messdaten liefern und die Datenübertragung kabelgebunden ist. Diese zwei Probleme sollen im Rahmen dieses Projekts gelöst werden.

Als Grundlage für die Messung der Felsbewegung wird ein bestehender Sensorstab (ein einfaches Potentiometer) verwendet, das die Ausdehnung zweier Felsen misst. Dieser soll nun so erweitert werden, dass die Veränderung der Lage des Stabes aber auch der zwei Befestigungen am Felsen als je zwei Winkel gemessen werden kann. Dies führt zu mehr Messdaten, mit welchen die Situation besser eingeschätzt und somit eine qualitative bessere Prognose erstellt werden kann. Die gemessenen Winkel sollen mindestens eine Genauigkeit von  $\pm 1$  Grad aufweisen.

Es sollen zwei Systemvarianten entwickelt werden. Mit der ersten Systemvariante können Einzelmessungen durchgeführt werden und mit der zweiten Variante kann ein Netzwerk aus mehreren Messvorrichtungen erstellt werden. Beide Systemvarianten sollen die Messdaten drahtlos, entweder direkt auf einen Webserver oder an einen Datenlogger weiterleiten können. Bei der Entwicklung der einzelnen Systeme soll darauf geachtet werden, dass diese modular aufgebaut sind, so dass einfach von der einen zur anderen Konfiguration gewechselt werden kann.

Die Messeinrichtung wird in Gebirgen montiert, wo sie rauen Umgebungsbedingungen ausgesetzt ist. Daher ist ein weiteres Ziel das wetterfeste Design des Prototyps. Wetterfest bedeutet, dass er wasserdicht ist und den starken Temperaturschwankungen standhält. Nebst den Wettereinflüssen sollte der Prototyp auch resistent gegen elektromagnetische Störungen sein. Um die Messeinrichtung auch in abgelegenen Gebieten mit Energie zu versorgen, soll ein Solarpanel verwendet werden. Damit die Grösse des Solarpanels und des Akkus möglichst gering gehalten werden kann, soll darauf geachtet werden, dass die Elektronik möglichst wenig Energie benötigt.

### Übersicht grundlegende Ziele:

- Entwicklung eines Prototyps
- Messung Ausdehnung
- Messung von Felsbewegungen als Winkeln  $\pm 1$  Grad
- Drahtlose Datenübermittlung
- 2 Systemkonfigurationen (Einzelmessung, Sensornetzwerk)
- Energieversorgung mit Solarpanel
- Wetterfest

### Erweiterte Ziele:

- Sensornetzwerk funktioniert mit mehreren Sensorknoten
- Alarm ausgeben bei grossen Änderungen
- Eine Woche Energie ohne Solarpanel
- Daten werden zwischengespeichert
- Messeinrichtung kann über GUI konfiguriert werden
- Die Temperatureffekte können kompensiert werden (genauere Messung)

### 3 Entwicklungsumgebung

Das Ziel des Projektes ist es nicht, ein serienreifes Gerät zu entwickeln, sondern einen funktionierenden Prototyp zu erhalten. Da das Hauptziel die Fertigstellung des Prototypen ist, wird ein Entwicklungsboard verwendet und nicht eine Eigenentwicklung. Dadurch kann wertvolle Zeit eingespart und in andere Problemstellungen investiert werden.

Die Anforderungen an das Entwicklungsboard sind, dass die Schnittstelle I<sup>2</sup>C und die Funkverbindungen LoRa und 3G Mobilfunk unterstützt werden und diese sowie das Board flexibel eingesetzt werden können. Die Mobilfunkstandards 3G und LoRa werden im Kapitel 7.1.2 genauer beschrieben. Um den Zeitaufwand zur Softwareentwicklung zu minimieren, ist es ebenfalls von Vorteil, wenn bereits bestehende Libraries zur Verfügung stehen.

Kriterium	Wert
Schnittstelle	I <sup>2</sup> C
Funkverbindung	LoRa, 3G
Software	Bestehende Libraries
Preis	möglichst tief
Standby- Strom	möglichst tief
Allgemein	Möglichst flexibel

Tabelle 1 Anforderungen Entwicklungsboard

Als Lösung für das Entwicklungsboard fiel die Wahl auf das Libelium Waspote. Ein Hauptgrund dabei war, dass es sehr flexibel einsetzbar ist und alle benötigten Kommunikationsstandards in Form von aufsteckbaren Modulen bereits zur Verfügung stellt. Das Waspote wird an der Berner Fachhochschule im Rahmen von diversen Kursen als Entwicklungsplattform eingesetzt, daher besteht die Möglichkeit auf Erfahrungswerte anderer Entwickler zurückgreifen zu können. Die vielen Beispiele und umfangreichen Libraries, die für dieses Entwicklungsboard verfügbar sind, sind ebenfalls ein grosser Pluspunkt. Das Libelium Waspote benötigt im DeepSleep- Modus einen Strom von nur 66uA, was sehr klein ist und für unsere Low Power Anwendung sehr gut passt.



Abbildung 3 Libelium Waspote [4]



## 4 Messung

### 4.1 Messung Felsbewegungen mit Sensoren

Um die Verschiebung der Felsen messen zu können werden Sensoren benötigt. Der Ausdehnungssensor (blau im Bild), der als Grundlage genommen wird, misst die Ausdehnung zwischen den beiden Felswänden (orange). Um genauere Aussagen über die Bewegung machen zu können wird eine Winkelmessung benötigt. Es werden die Pitch- (rot) und Roll- Winkel (grün) gemessen. Dazu werden Beschleunigungssensoren eingesetzt (Acc1, Acc2, Acc3). Zwei der drei Beschleunigungssensoren (Acc1, Acc3) werden eingesetzt, damit die Winkeländerungen der beiden Felswände gemessen werden können. Mit dem Ausdehnungssensor, sowie dem dritten Beschleunigungssensor (Acc2) wird festgestellt, wie sich die beiden Felsen im Vergleich zueinander bewegen. Wie die Winkel anhand der Beschleunigungssensoren gemessen werden können wird im Kapitel 4.3 beschrieben.

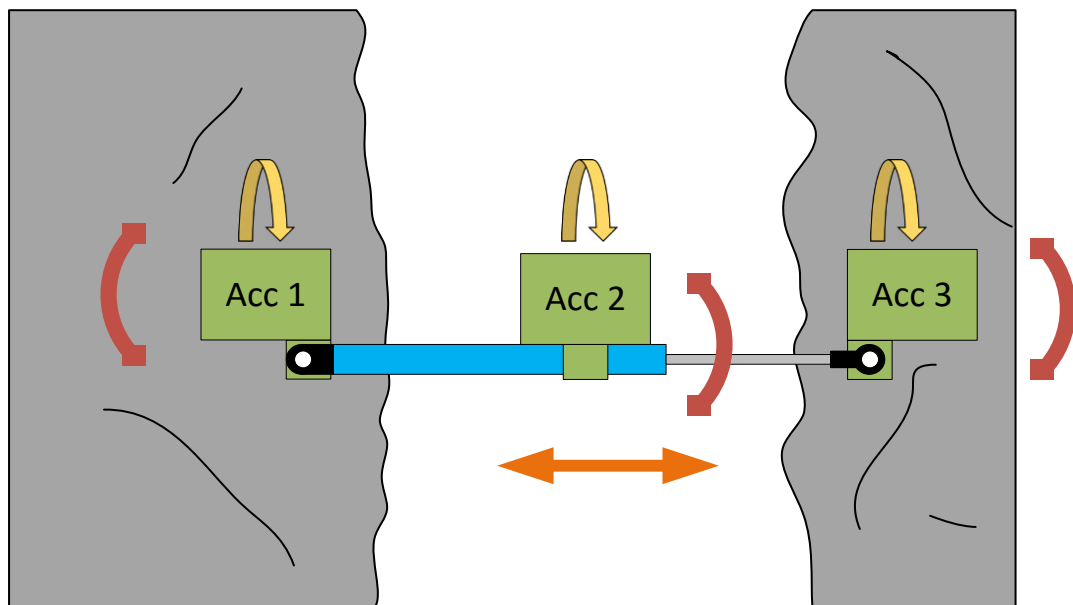
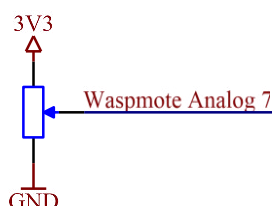


Abbildung 4 Messgrößen Ausdehnung (orange) und Winkel

### 4.2 Ausdehnungsmessung

Mithilfe eines Ausdehnungssensors wird die Verschiebung von zwei Felsen gemessen. Dieser Sensor funktioniert wie ein einfaches Potentiometer. Er gibt die Ausdehnung in Form eines sich linear ändernden Widerstandes an. Die Spannung die über diesem Widerstand abfällt, wird mit einem AD Wandler gemessen. Der AD- Wandler auf dem Libelium Waspnote hat eine Auflösung von 10 Bit, was einer Aufteilung von 1024 Stufen entspricht. Der Ausdehnungssensor kann über eine Strecke von 30.5 cm auseinander- und zusammengeschoben werden. Somit beträgt die Genauigkeit der Distanzbestimmung 0.3mm.



$$\text{Genauigkeit} = \frac{\text{Distanz total}}{\text{Anzahl Stufen}} = \frac{30.5\text{cm}}{1024} = 0.299\text{mm} \approx 0.3\text{mm}$$

Abbildung 5 Schema und Berechnung Ausdehnungssensor

### 4.3 Winkelmessung mit Beschleunigungssensoren

Die Beschleunigungssensoren messen die Beschleunigung der X-, Y- und Z-Achse. Durch die Aufteilung der Erdgravitation von 1G auf die drei Achsen können durch die Verhältnisse die Winkel berechnet werden. Einzelheiten dazu werden in den folgenden Unterkapiteln beschrieben.

#### 4.3.1 Winkeldefinitionen

Der Beschleunigungssensor wird zur Bestimmung der Winkel des Sensorstabes benötigt. Mit dem Beschleunigungssensor mit 3 Achsen, können der Roll- und Pitch Winkel berechnet werden. Dabei entspricht der Roll Winkel der Drehung um die X- Achse und der Pitch Winkel einer Drehung um die Y- Achse. Eine Drehung um die Z- Achse kann mit einem Beschleunigungssensor nicht festgestellt werden, da sich dabei die Beschleunigungswerte auf allen drei Achsen nicht ändern. Um eine solche Winkeländerung feststellen zu können, müsste ein Magnetometer eingesetzt werden, der die Lage des Sensors im Bezug zum Erdmagnetfeld misst. Ein solches Magnetometer wird allerdings nicht eingesetzt, da er relativ anfällig auf magnetische Störungen ist und besonders bei der Installation in der Nähe von Felsen von diesen gestört würde. Des Weiteren, wird die Ausdehnung des Stabes mit dem Ausdehnungssensor gemessen. Somit können wir anhand diese Daten eine Drehung um die Z- Achse annehmen, wenn die beiden anderen Winkel nicht geändert haben, aber die Ausdehnung des Stabes sich geändert hat.

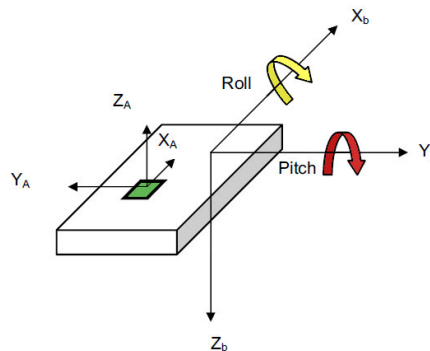


Abbildung 6 Roll- und Pitch Winkel ( [5] S.6)

#### 4.3.2 Einfache Winkelberechnung mit einer Achse

Der Beschleunigungssensor misst die Beschleunigung auf drei Achsen. Um nun damit einen Winkel zu berechnen, wird einfache Trigonometrie verwendet. Untenstehend wird das Prinzip an einem Beispiel einer Winkelberechnung mit einer Achse und einem Winkel gezeigt. Der Beschleunigungssensor (grün) liegt mit einem Winkel  $\alpha$  zur Horizontalen im Raum. Um diesen Winkel zu berechnen, wird der Winkel Alpha über den Arkussinus der gemessenen Beschleunigung X einer Achse geteilt durch die Erdgravitation G. (siehe untenstehendes Bild)

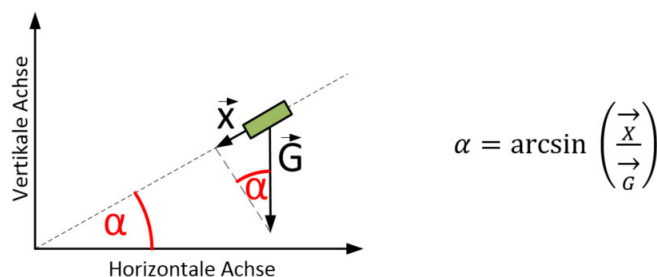


Abbildung 7 Winkelmessung mit Beschleunigungssensor

Diese Berechnung ist relativ einfach, hat aber den Nachteil, dass die Sensitivität des Sensors nicht über den ganzen Winkelbereich konstant ist. Das bedeutet, dass die Genauigkeit der Winkelberechnung je nach Winkel unterschiedlich ist. Um das zu verhindern werden die Winkel mit drei Achsen berechnet.

#### 4.3.3 Winkelberechnung mit 3 Achsen

Damit eine konstante Sensitivität und eine höhere Genauigkeit der Winkelberechnung erreicht werden kann, werden alle drei Achsen für die Berechnung der Winkel mit einbezogen. Wiederum können die Winkel über Trigonometrie anhand untenstehender Formeln berechnet werden. Die Beschleunigungssensoren messen die Beschleunigung in Form einer Spannung und wandeln diese mit dem internen AD- Wandler in einen Beschleunigungswert in Bits. Dieser Beschleunigungswert muss jedoch für die Winkelberechnung in eine Beschleunigung in G umgewandelt werden. Dazu wird der Beschleunigungswert der X-, Y- und Z Achsen mit der Sensitivität des Beschleunigungssensors multipliziert. Dies ergibt die Werte  $A_{X1}$ ,  $A_{Y1}$  und  $A_{Z1}$ , diese entsprechen dabei der Beschleunigung in G der jeweiligen Achse. Mit diesen drei Beschleunigungen können nach untenstehender Formel die Pitch- und Roll- Winkel berechnet werden.

$$Pitch = \alpha = \arctan\left(\frac{A_{X1}}{\sqrt{(A_{Y1})^2 + (A_{Z1})^2}}\right)$$

$$Roll = \beta = \arctan\left(\frac{A_{Y1}}{\sqrt{(A_{X1})^2 + (A_{Z1})^2}}\right)$$

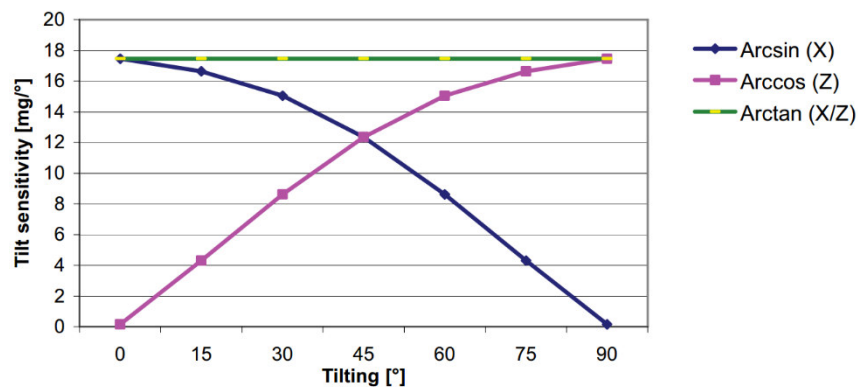


Abbildung 8 Sensitivität bei Berechnung mit 3 Achsen ( [5] S.13)

#### 4.4 Beschleunigungssensor

Bei der Auswahl der Beschleunigungssensoren gab es mehrere Kriterien. Ein Kriterium ist die Betriebsspannung, welche gleich sein soll wie die Betriebsspannung des Waspmotes (3.3V). Als nächstes Kriterium wird für eine möglichst genau Berechnung des Winkels und einer konstanten Sensitivität im gesamten Winkelbereich, wird mindestens ein Beschleunigungssensor mit 2 Achsen benötigt. Um noch genauere Resultate zu erhalten, sollte ein Beschleunigungssensor mit 3 Achsen eingesetzt werden. Um die Verkabelung zwischen den Sensoren zu minimieren, wird ein Bus benötigt. Grundsätzlich unterstützen die meisten Beschleunigungssensoren entweder den I<sup>2</sup>C- Bus oder SPI. Jedoch wird der I<sup>2</sup>C Bus oftmals bevorzugt, da dieser die Auswahl des Bausteines innerhalb des Protokolls festlegt und nicht weitere Slave-Select Leitung benötigt wie dies bei SPI der Fall ist. Der I<sup>2</sup>C-Bus ermöglicht es die drei Sensoren alle an dasselbe Kabel anzuschliessen, somit wird von Sensor zu Sensor jeweils nur eine Verbindung benötigt.

Da für die Winkelbestimmung nur die Erdanziehung von einem G benötigt wird und keine grösseren Beschleunigungen gemessen werden müssen, soll der Bereich auf  $\pm 2G$  einstellbar sein. Dadurch erhält man eine höhere Genauigkeit, da die Auflösung sich auf nur  $\pm 2G$  aufteilt.

Da es sich um eine Low-Power Anwendung handelt, soll der Standby Strom möglichst tief sein. Der Faktor Wirtschaftlichkeit darf natürlich nicht ausser Acht gelassen werden, darum wird auf einen geringen Stückpreis geachtet. Alle Anforderungen an den Beschleunigungssensor sind in Tabelle 2 übersichtlich dargestellt.

Kriterium	Wert
Speisespannung	3.3V
Verbindungen	I <sup>2</sup> C
Achsen	3 (X,Y und Z)
ADC- Auflösung	mind. 12 Bit
Bereich	einstellbar auf $\pm 2G$
Preis	möglichst tief
Standby- Strom	möglichst tief
Grösse	möglichst klein

Tabelle 2 Anforderungen Beschleunigungssensor

Der Entscheid fiel auf den Beschleunigungssensor FXLS8471Q von NXP. Dieser Beschleunigungssensor wurde ausgewählt, weil er alle oben genannten Anforderungen zu einem günstigen Preis erfüllt und ausserdem bereits interne Signalverarbeitung besitzt, was zum Beispiel chipinternes Oversampling oder Ausgabe von Alarmen bei gewissen Änderungen ermöglicht.

#### 4.4.1 Theoretische Genauigkeit

Wie in der Abbildung 8 auf Seite 11 ersichtlich ist, kann mit der Winkelberechnung mit 3 Achsen eine konstante Sensitivität von  $17.45\text{mg}/^\circ$  erreicht werden. Um nun zu berechnen, wie genau der Winkel mit dem Sensor FXLS8471Q bestimmt werden kann, muss dessen Sensitivität berücksichtigt werden.

Die Sensitivität von Beschleunigungssensoren errechnet sich durch den Bereich des internen AD-Wandlers und dem eingestellten Messbereich. Für den Sensor FXLS8471Q ergibt das einen Wert von  $0.244\text{mg}/\text{LSB}$ .

$$\text{AD – Auflösung: } 14\text{Bit, Bereich: } \pm 2G \quad \text{Sensitivität FXLS8471Q} = \frac{\text{eingestellter Bereich in G}}{\text{Bereich AD-Wandler}} = \frac{4G}{2^{14}} = 0.244\text{mg}/\text{LSB}$$

Mit einer Sensitivität der Winkel von  $17.45\text{mg}/^\circ$  lässt sich der Winkel theoretisch auf  $13.9\text{mGrad}$  genau berechnen.

$$\text{Genauigkeit Winkel} = \frac{\text{Sensitivität Sensor}}{\text{Sensitivität Winkel}} = \frac{0.244\text{mg}/\text{LSB}}{17.45\text{mg}/^\circ} = 13.9\text{m}^\circ$$

Diese Genauigkeit ist allerdings ein theoretischer Wert und in der Praxis werden das Rauschen und die Temperatureinflüsse die Genauigkeit stark beeinflussen.

#### 4.4.2 Rauschen

In diesem Kapitel wird untersucht, was das Rauschen für einen Einfluss auf die Genauigkeit des Beschleunigungssensors FXLS8471Q hat. Um dies abzuschätzen, wurde zuerst eine Messung durchgeführt, in dem der Beschleunigungssensor im normalen Modus konfiguriert ist. Dabei wurde untenstehendes Rauschen gemessen. Der Sensor wurde während der Messung nicht bewegt, somit ist jede Änderung der Beschleunigungen als Rauschen zu beurteilen.

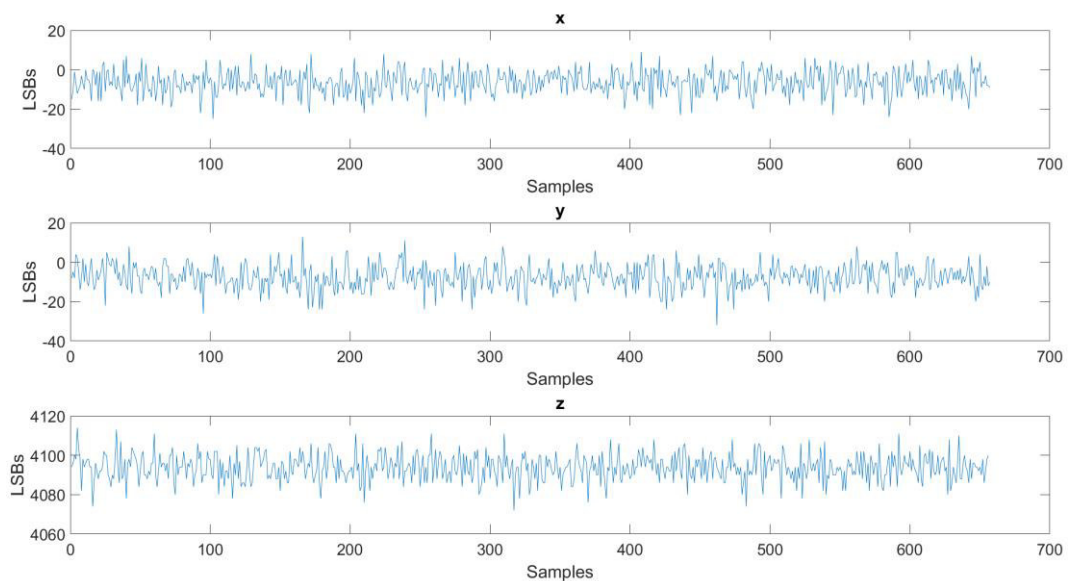


Abbildung 9 Rauschen mit Beschleunigungssensor im normalen Modus

Es ist zu sehen, dass das Rauschen auf den drei Achsen bis zu 40 LSBs beträgt. Dadurch sinkt die Genauigkeit, die Auflösung des Winkels beträgt nun  $0.56^\circ$ .

$$\text{Auflösung Winkel} = \frac{\text{Sensitivität Sensor mit Rauschen}}{\text{Sensitivität Winkel}} = \frac{\frac{0.244\text{mg}}{\text{LSB}} * 40}{17.45\text{mg}/^\circ} = 0.56^\circ$$

Der Beschleunigungssensor FXLS8471Q hat eine interne digitale Signalverarbeitung, die diverse Funktionen zur Verfügung stellt. Eine dieser Funktionen ist das interne Oversampling. Damit kann eingestellt werden, dass die Beschleunigung mehrmals gemessen wird, aber schlussendlich nur ein Durchschnittswert aller Messungen ausgegeben wird. Dadurch kann der Effekt des Rauschens minimiert werden. Es wurde eine Messung durchgeführt mit einem maximalen Oversampling von 1024. Das heisst, dass jeder ausgegebene Wert aus 1024 gemessenen Werten berechnet wurde und somit viel genauer ist. In Abbildung 10 ist nun das Rauschen zu sehen, mit maximalem internen Oversampling.

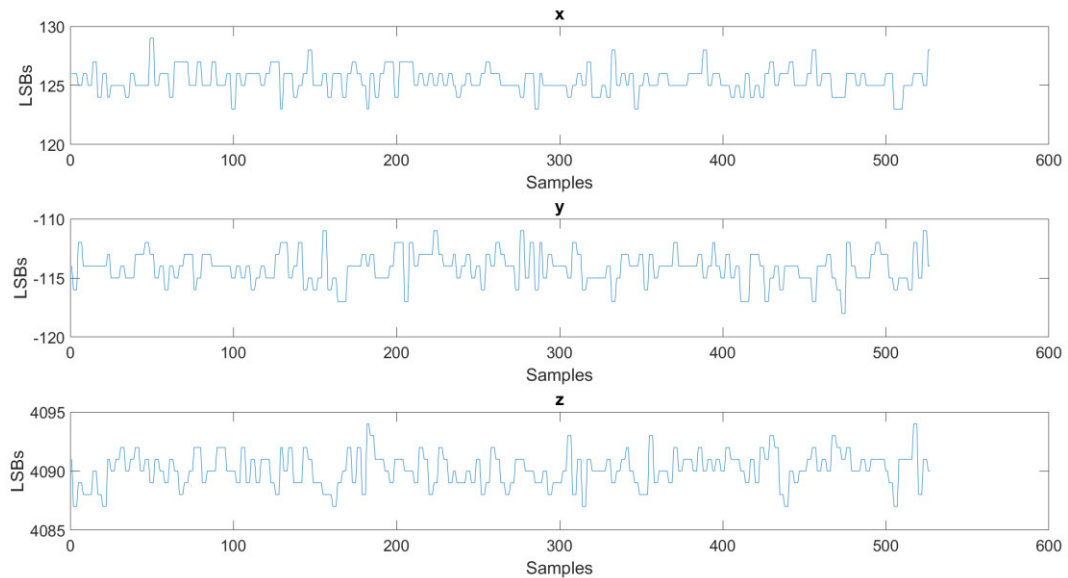


Abbildung 10 Rauschen mit Beschleunigungssensor im 1024-Oversampling Modus

Durch das interne Oversampling, wurde der Einfluss des Rauschens auf die Messung stark minimiert. Das Rauschen beträgt nun noch maximal 7 LSBs, dadurch ist die Genauigkeit der Winkelmessung nun wieder sehr stark angestiegen. Die Auflösung des Winkels beträgt nun  $97.94m^{\circ}$ . In dieser Auflösung ist allerdings der Einfluss der Temperatur noch nicht berücksichtigt.

$$Auflösung\ Winkel = \frac{Sensitivität\ Sensor\ mit\ Rauschen}{Sensitivität\ Winkel} = \frac{\frac{0.244mg}{LSB} * 7}{17.45mg/^{\circ}} = 97.94m^{\circ}$$

#### 4.4.3 Temperaturverhalten

Laut Datenblatt hat der Beschleunigungssensor FXLS8471Q eine temperaturabhängige Abweichung von  $\pm 0.2\text{mg}/^\circ\text{C}$ . Ausserdem kann die Sensitivität des Sensors um  $\pm 0.01\%$  pro Grad Celsius abweichen. Der Einfluss dieser beiden Parameter auf die berechneten Winkel wurde in einer Klimamessung untersucht. Dazu wurde der Sensor mit dem Libelium Wasp mote in einen Klimaschrank gelegt. Es wurde das Verhalten im Temperaturbereich von  $-10^\circ\text{C}$  bis  $70^\circ\text{C}$  analysiert. Während des Tests wurden die Winkel ständig neu berechnet und in einer Tabelle abgespeichert. Der Sensor wurde während dieser Zeit nicht bewegt. Alle Abweichungen der Winkel sind somit als Fehler aufgrund der Temperaturveränderung zu beurteilen. Das Ergebnis des Temperaturtests ist in Abbildung 11 ersichtlich.

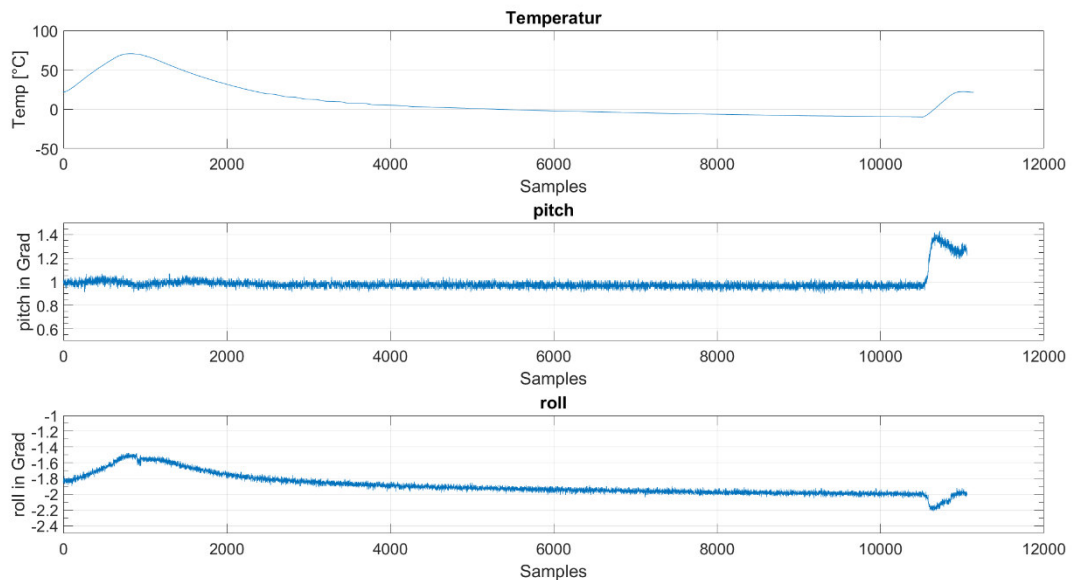


Abbildung 11 Pitch- und Rollwinkel unter Temperatureinfluss

Als Ergebnis dieser Messung kann gesagt werden, dass für den Bereich von  $-10^\circ\text{C}$  –  $70^\circ\text{C}$  der Winkel nicht genauer als  $\pm 0.4$  Grad angegeben werden kann. Dies ist lediglich eine erste Messung und mit nur einer Messung kann nicht das komplette Temperaturverhalten des Beschleunigungssensors beschrieben werden. Um das Temperaturverhalten genauer herauszufinden, wären eine Vielzahl an Messungen mit unterschiedlichen Sensoren und Temperaturkurven notwendig.

Wäre das Temperaturverhalten des Beschleunigungssensors bekannt, so könnte man diese Effekte kompensieren und somit eine höhere Genauigkeit erhalten. Damit die Möglichkeit offen bleibt diese Effekte zu einem späteren Zeitpunkt zu kompensieren, wurde neben jedem Beschleunigungssensor noch ein Temperatursensor platziert.

## 5 Datenübertragung

Um die Daten von den Sensoren zu der restlichen Elektronik zu übertragen, wird ein serieller Bus benötigt. Dieser Bus muss sehr robust gegen Störungen sein. Obwohl die Übertragungsdistanz nur 1.5m beträgt, besteht bereits die Möglichkeit für elektromagnetische Störungen, welche die Zuverlässigkeit der Datenübertragung beeinträchtigen könnten. Damit Störungsimpulse keinen Einfluss auf die übertragenen Daten haben, wird ein differentieller Bus verwendet.

### 5.1 Differentielles Bussystem

Bei einem differentiellen Bus werden pro Signal zwei Leitungen geführt. Diese verlaufen jeweils gegenphasig und die Information liegt in der Spannungsdifferenz dieser zwei Signale. Da Störungen sich auf beide Leitungen gleich auswirken, bleibt die Differenz gleich und es tritt kein Übertragungsfehler auf (siehe untenstehende Abbildung). Wichtig dabei ist, dass die Datenleitungen symmetrisch verlaufen (in Kabel verdreht) und gleich lang sind. Denn bei Asymmetrien kommt es zu Gleichtaktströmen auf dem Bus und so zu Störabstrahlungen. [6]

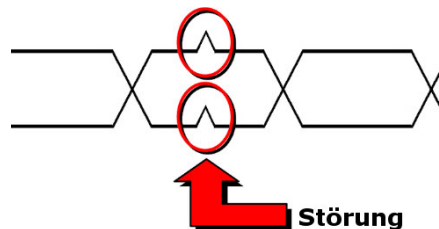


Abbildung 12 Störung auf differentielles Signal [7]

Als Übertragungsbus kommen die differentiellen Industriebusse CAN, RS232 und differentielles I<sup>2</sup>C in Frage. Um den Implementierungsaufwand möglichst klein zu halten, wurde differentielles I<sup>2</sup>C als Übertragungsbus ausgewählt. Dies erlaubt die Ansteuerung der Sensoren über den Standard I<sup>2</sup>C-Bus. Dieser wird dann lediglich für die Übertragung in ein differentielles Signal umgewandelt. Diese Umwandlung wird von dem Chip PCA9615 von NXP vorgenommen und benötigt keine Software. Das Libelium Waspote kann die Sensoren weiterhin über den Standard I<sup>2</sup>C-Bus auslesen werden. Der differentielle I<sup>2</sup>C Bus ist für die Bausteine die mit dem Standard I<sup>2</sup>C angesteuert werden nicht sichtbar.

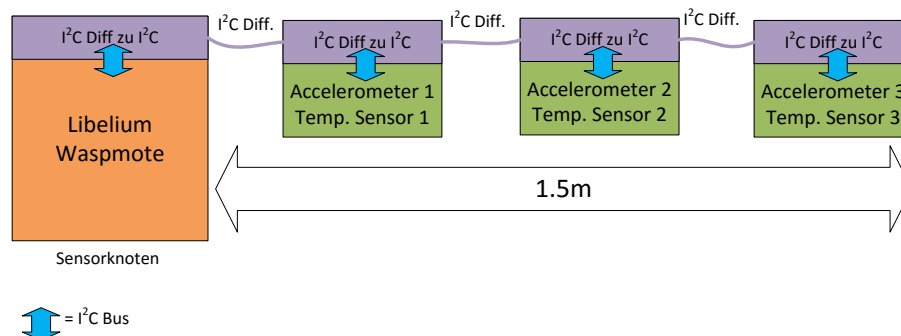


Abbildung 13 Übersicht I<sup>2</sup>C und Diff. I<sup>2</sup>C Bus



## 5.2 I<sup>2</sup>C Bus

Damit die Daten der Sensoren von einem Mikrocontroller ausgelesen werden können, müssen diese Daten über einen Bus übertragen werden. Die meisten Sensoren unterstützen dabei die Übertragung per SPI- oder I<sup>2</sup>C Bus. Da aufgrund der Verwendung von differentielltem I<sup>2</sup>C auch das Standard- I<sup>2</sup>C zur Anwendung kommt, wird dieses Protokoll hier kurz beschrieben.

Bei dem I<sup>2</sup>C Bus gibt es einen Master, der den Bus steuert und mehrere Slaves, die auf den Bus reagieren. Die Slaves werden über eine Adresse angesprochen, deshalb ist es wichtig, dass alle Slaves unterschiedliche I<sup>2</sup>C Slave-Adressen haben.

Der I<sup>2</sup>C Bus ist ein Bus, der lediglich aus den zwei Busleitungen SCL und SDA besteht. Die SCL Leitung überträgt das Clocksignal, das vom Master zur Verfügung gestellt wird. Die Datenleitung funktioniert als bidirektionale Leitung. Eine Datenübertragung wird immer vom Master initiiert und durch eine Startbedingung gestartet. Danach folgt die Adresse des Slaves, den er ansprechen will und ein Bit, das angibt ob Daten geschrieben oder gelesen werden. Danach werden Daten ausgetauscht und jeweils vom Empfänger mit einem Low Pegel bestätigt. Zum Schluss folgt eine Stopp- Bedingung.

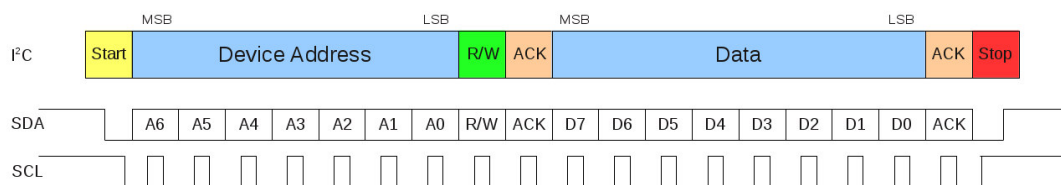


Abbildung 14 I<sup>2</sup>C Bus Protokoll [8]

## 6 Systemübersicht

Um ein möglichst vielseitig einsetzbares Messsystem zu erhalten wurde bei der Entwicklung auf Modularität gesetzt. Es wurden wie in der Zielsetzung bereits erwähnt zwei Systemkonfigurationen entwickelt. Durch geringe Hardwareänderungen lassen sich diese beiden Varianten in einander umwandeln.

In den folgenden Kapiteln wird für das Messsystem der Begriff Messvorrichtung oder Sensorknoten verwendet. Ein solcher ist in untenstehender Abbildung zu sehen. Er besteht grundsätzlich aus Sensoren, einem Solarpanel und einer Elektronik, die die Messdaten auswertet und weiter sendet. Die einzelnen Bestandteile des Sensorknoten werden im Kapitel Hardwareübersicht noch genauer beschrieben.

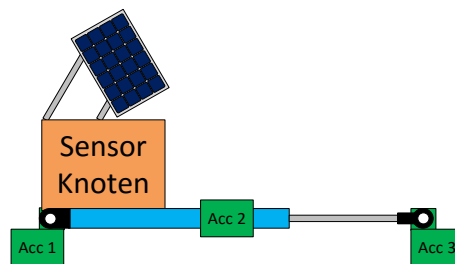


Abbildung 15 Sensorknoten

### 6.1 Systemkonfiguration 1

Die erste der beiden Systemkonfigurationen besteht aus einem einzelnen Sensorknoten. Der einzelne Sensorknoten sendet seine Messdaten direkt per 3G an den Webserver. Solch ein System könnte beispielsweise für Kurzzeitmessungen verwendet werden, da die Installation sehr simpel ist und keine zusätzliche Hardware nötig ist (siehe Abbildung 16). Für weitere Informationen über den 3G Mobilfunk siehe Kapitel 29.

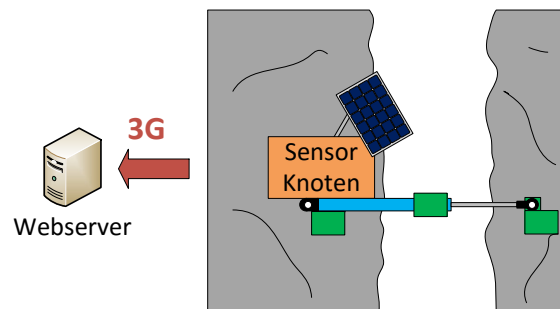


Abbildung 16 Systemkonfiguration einzelner Sensorknoten

#### Vorteile

- + Flexibel in Positionierung
- + Einfache Installation
- + Keine zusätzliche Hardware nötig

#### Nachteile:

- Teuer (speziell für mehrere)
- Energieverbrauch
- Benötigt Mobilfunkdatenempfang (3G)

## 6.2 Systemkonfiguration 2

Die zweite Systemkonfiguration besteht aus mehreren Sensorknoten welche in einem Netzwerk zusammengefasst werden. Diese Systemvariante kann zum Beispiel für Langzeit Messkampagnen mit mehreren Sensorknoten in einem grösseren Gebiet benutzt werden. Auf die Funkstandards 3G und Lora wird im folgenden Kapitel Kommunikation näher eingegangen.

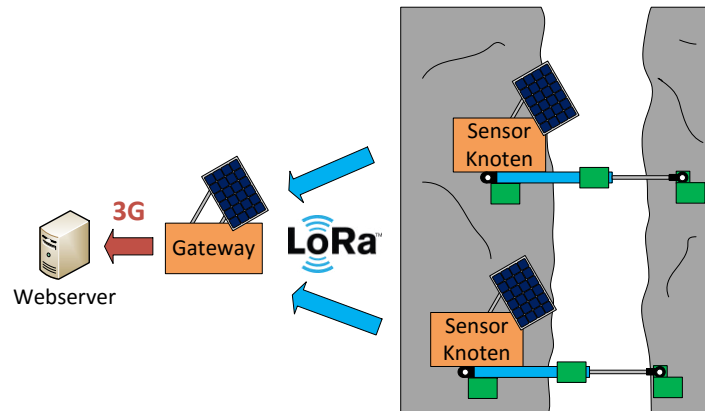


Abbildung 17 Sensornetzwerk

### Vorteile:

- + Optimal für mehrere Messungen
- + Billiger (bei vielen Messeirrichtungen)
- + Niedriger Energieverbrauch
- + Braucht kein Mobilfunkdatenempfang (ausser Gateway)

### Nachteile:

- Zusätzliche Hardware nötig (Gateway)
- Aufwändigere Installation im Vergleich zu Variante 1

## 7 Kommunikation

Eines der Ziele des Projektes besteht darin, den Datenaustausch zwischen den Sensorknoten und einer Auswertungseinheit drahtlos zu realisieren. Es besteht die Möglichkeit einen einzelnen Sensorknoten zu verwenden oder auch mehrere Sensorknoten, welche dann ein Sensornetzwerk bilden.

### 7.1 Sensornetzwerk

Um Daten drahtlos in einem Sensor-Netzwerk austauschen zu können, muss ein Kommunikationsprinzip entworfen werden. Dieser Entwurf beinhaltet die Auswahl einer Netzwerktopologie, eines Kommunikationsstandards und eines Zugriffsverfahrens. Diese drei Elemente werden in den folgenden Kapiteln im Detail erklärt.

#### 7.1.1 Netzwerktopologie

Eine der ersten Schritte im Entwurf einer Kommunikation in einem Netzwerk, ist es sich zu überlegen welche Netzwerktopologie eingesetzt werden soll. In Abbildung 18 Netzwerk Topologien werden die Netzwerk-Topologien schematisch dargestellt, wobei diejenigen welche für unsere Anwendung in Frage kommen rot markiert sind.

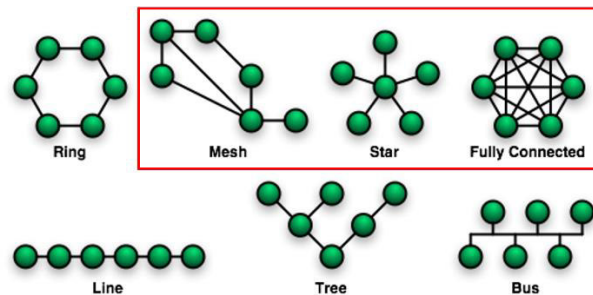


Abbildung 18 Netzwerk Topologien [9]

Da wir in unserem Netzwerk ein Gateway haben, welches die Daten aller im Netzwerk beteiligten Sensorknoten erhält und diese anschliessend weiterleitet, macht es Sinn eine Mesh- oder Star-Topologie in Betracht zu ziehen.

Fully Connected stellt einen Spezialfall eines Mesh dar, in welcher alle Teilnehmer des Netzes eine Verbindung zu einander haben. Mesh und Fully Connected werden deshalb in der folgenden Analyse zusammengefasst.

#### Mesh

##### Vorteile:

- Sicherste Variante eines Netzwerkes
- Bei Ausfall eines Gerätes oder einer Verbindung ist durch Umleitung die Datenkommunikation weiterhin möglich
- Sehr leistungsfähig
- Gute Lastverteilung
- Keine zentrale Verwaltung

##### Nachteile:

- Hoher Energieverbrauch
- Vergleichsweise komplexes Routing nötig
- Jedes Endgerät arbeitet als Router und ist demnach oft aktiv
- Die Endgeräte sollten möglichst eingeschaltet bleiben, um die Beständigkeit des Netzwerkes zu garantieren

[10]

## Star

### Vorteile:

- + Ausfall eines Endgerätes hat keine direkten Auswirkung das Netzwerk.
- + Leicht erweiterbar
- + Leicht verständlich
- + Sehr gute Eignung für Multicast-/Broadcastanwendungen
- + Kein Routing nötig

### Nachteile:

- Ausfall des Verteilers macht Netzwerk unbrauchbar
- Koordinierter Datenaustausch

[11]

### 7.1.1.1 Ausgewählte Netzwerktopologie

Für unsere Kommunikation wurde die Startopologie ausgewählt, weitere Begründungen für die Wahl werden im Kapitel 7.1.2.1 gegeben.

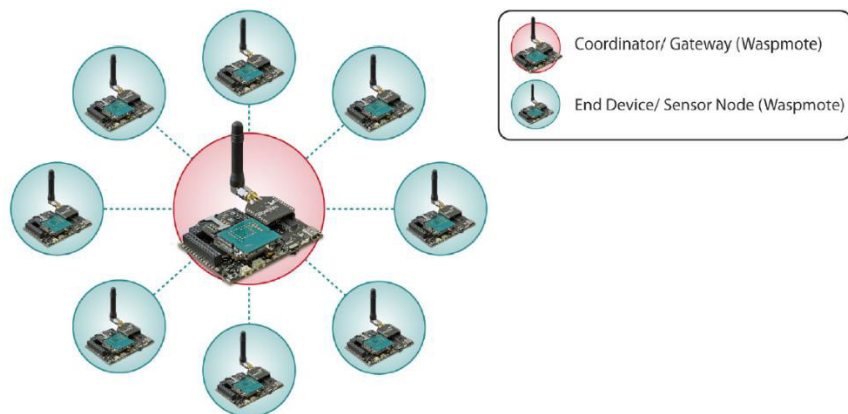


Abbildung 19 Star-Topologie [12]

### 7.1.2 Kommunikationsstandard

Das Spektrum an möglichen Kommunikationsstandards ist riesig. Es gilt zu evaluieren welcher dieser Standards für unsere Anwendung am besten geeignet ist.

Der ausgewählte Kommunikationsstandard sollte im Bereich der Reichweite, der Sendeleistung, beziehungsweise Energieverbrauchs, und der Verbindungseinstellungen eine möglichst hohe Flexibilität aufweisen. Dadurch kann die Kommunikation der Situation vor Ort angepasst und energieeffizient gestaltet werden.

Die riesen Auswahl an Kommunikationsstandards wurde durch die Wahl des Libelium Waspote als Entwicklungsplattform bereits ein bisschen eingeschränkt. Dies ist der Fall, da das Waspote nur gewisse Kommunikationsstandards als aufsteckbare Module zur Verfügung stellt. Folgende Standards sind für das Waspote verfügbar:

- ZigBee
- Bluetooth low energy
- LoRa
- LoRaWAN
- Sigfox
- WiFi
- GSM/GPRS
- 3G

Die Kommunikation stellt einer der Hauptenergieverbraucher des Sensorknotens dar und sollte deshalb einen möglichst tiefen Energieverbrauch haben. In Anbetracht dieses Kriteriums fallen WiFi, GSM/GPRS und 3G bereits weg, da sie einen zu hohen Energiebedarf haben. Übrig bleiben somit noch:

- ZigBee
- Bluetooth low energy
- LoRa
- LoRaWAN
- Sigfox

Sigfox benutzt für die Kommunikation, ähnlich wie bei der Mobilkommunikation, ein Netz welches durch einen Netzbetreiber (Provider) zur Verfügung gestellt wird. Die Verwaltung des Netzes wird durch den Netzbetreiber gewährleistet. Aus diesem Grund müssen alle Geräte welche das Netzwerk benutzen möchten für eine geringe Gebühr angemeldet werden. [13]

Die Kommunikation per Sigfox erfolgt dann nach dem in Abbildung 20 dargestellten Schema.

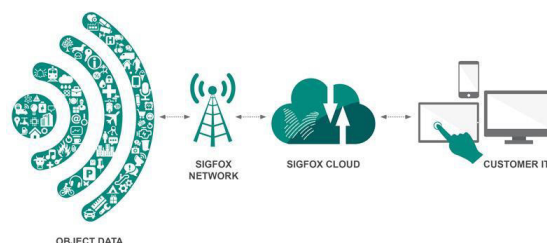


Abbildung 20 Übersicht Kommunikation Sigfox [14]

Um zu gewährleisten, dass das Netz nicht überbelastet wird, gibt der Netzbetreiber Vorgaben zum maximalen Datenvolumen pro Tag und der grössten zugelassenen Message- Grösse. Dadurch wird die gewünschte Flexibilität bereits stark eingeschränkt.

Nebst dem Flexibilitätsverlust fällt Sigfox als möglicher Kommunikationsstandard weg, da in der Schweiz noch kein Sigfox-Netzwerk aufgebaut ist, dies ist in der Abbildung 21 ersichtlich (weisse Gebiete nicht erschlossen).

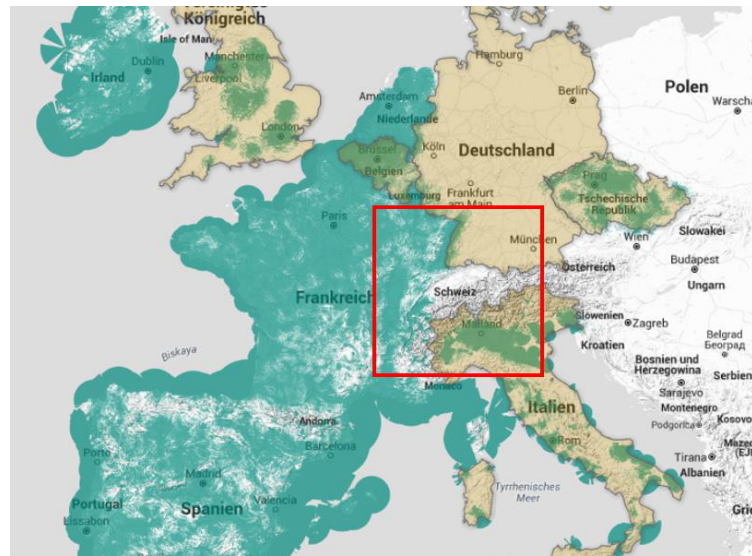


Abbildung 21 Sigfox- Abdeckung [15]

LoRaWan ist vom Konzept her ähnlich wie Sigfox. LoRaWan benötigt ebenfalls ein bereits bestehendes Netz. Ganz aktuell hat der Mobilfunknetzbetreiber Swisscom sein Interesse geäußert ein LoRaWan Netzwerk aufzubauen im Hinblick auf das Internet of Things (IoT) und dessen grossen Wachstumspotenzial. Jedoch ist dieses Netzwerk erst im Aufbau und somit entfällt LoRaWan ebenfalls als möglicher Kommunikationsstandard. [16]

Übrig bleiben somit noch:

- ZigBee
- Bluetooth low energy
- LoRa

Die verbleibenden Kommunikationsstandards werden nun genauer verglichen um den am besten geeigneten Standard zu finden.

	Reichweite	Datenrate	Energieverbrauch	Verbindung
<b>XBee-802.15.4-Pro [17]</b>	1.6 km norm. Antenne 7km zus. Antenne	250 kbit/s	Senden: 10mW Idle: 10uA	Line of Sight (LoS) Mesh-Topologie
<b>XBee-PRO 868 [18]</b>	40 km zus. Antenne	24 kbit/s	Senden: 315 mW Idle: 3uA	LoS Mesh-Topologie
<b>XBee-PRO 900HP [19]</b>	14 km (10Kbps) 6.5km (200Kbps)	10 kbit/s 200 kbit/s	Senden: 225mW Idle: 55uA	LoS Mesh Topologie
<b>LoRa [20] S.16 [21]</b>	21km (LoS) 2km	377 bit/s	Senden: 25mW Idle: 1uA	LoS nicht nötig Punkt zu Punkt
<b>Bluetooth [22]</b>	100m (LoS)	236.7 kbit/s	Senden: 10-500mW Idle: 2mA	Frequenz Hopping LoS

Tabelle 3 Kommunikationsstandards

#### 7.1.2.1 Ausgewählter Kommunikationsstandard Gateway <-> Sensorknoten

Um eine möglichst hohe Flexibilität beim Installieren der Sensorknoten gewährleisten zu können wurde **LoRa** als Kommunikationsstandard ausgewählt. Die hohe Flexibilität wird im Wesentlichen durch die hohe Reichweite erreicht. Da die Line of Sight nicht vorhanden sein muss, können die Sensorknoten nochmals flexibler platziert werden.

Die geringe Datenrate von LoRa stellt für unsere Anwendung keinen Nachteil dar, da nur wenige Daten übertragen werden müssen. Durch die Modulationsart welche in LoRa-Modul verwendet wird, ist die Kommunikation weniger anfällig auf Störungen durch andere Systeme im gleichen Frequenzbereich. Genauere Informationen betreffend der Modulationsart können im Kapitel Kommunikation über 3G entnommen werden.

### 7.1.3 Zugriffsverfahren

Mit dem Begriff Zugriffsverfahren, wird die Art beschrieben wie eine Funkübertragung organisiert ist. Es bestehen die drei Access-Verfahren: Frequency Division Multiple Access (**FDMA**), Time Division Multiple Access (**TDMA**) und Code Division Multiple Access (**CDMA**). Diese Access-Verfahren werden in unterstehendem Bild schematisch dargestellt. Für bestmögliche Resultate wird manchmal auch eine Kombination dieser Access-Techniken eingesetzt. Die Begriffe werden in den folgenden Unterkapiteln noch genauer beschrieben.

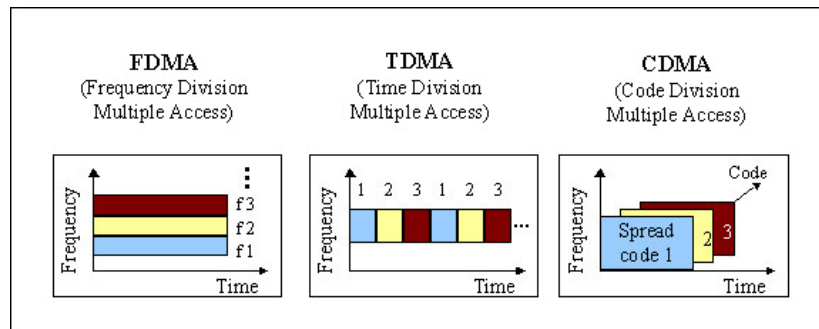


Abbildung 22 Accessverfahren [23]

Die Kommunikation ist der Teil des Sensorknotens, der am meisten Energie verbraucht. Aus diesem Grund muss die Kommunikation möglichst effizient und energiesparend ausgelegt werden. Diese Kriterien werden erreicht, indem die Sendezeit und die Sendeleistung möglichst minimal gehalten werden. In den folgenden Kapiteln werden die einzelnen Access-Techniken erklärt.

In den folgenden Kapiteln wird der Begriff Channel oft verwendet, deshalb wird dieser kurz erklärt. Unter Channel wird ein Frequenzband verstanden, welches für die Kommunikation verwendet werden kann.

#### 7.1.3.1 TDMA

Bei TDMA werden, wie der Abbildung 22 entnommen werden kann, jedem Sender einen bestimmten Zeitschlitz mit dem gesamten Frequenz-Spektrum zum Senden zugewiesen.

Durch die zeitliche Verschiebung der einzelnen Sender, kann das gleiche Frequenzband von mehreren Sendern benützt werden ohne Interferenzen.

Eine regelmässige Synchronisierung ist bei diesem Access-Verfahren nötig, damit das Kommunikationssystem ohne Zwischenfälle funktioniert. Diese Synchronisation ist nötig um zu gewährleisten, dass jeder Sender in seinem Zeitschlitz sendet und nicht durch Zeitverschiebungen, einen anderen Sender während seinem Sendeintervall stört.

#### 7.1.3.2 FDMA

Bei FDMA wird jedem Sender ein Channel zugeteilt, somit kann jeder Sender ständig senden, ohne einen anderen Teilnehmer zu stören (siehe Abbildung 22).

Bei diesem Verfahren ist ein präziser Transmitter nötig, welcher nicht Störungen auf den anderen Frequenzbändern verursacht und somit einen benachbarten Channel stört.

#### 7.1.3.3 CDMA

CDMA steht für Code Division Multiple Access, die einzelnen Sender werden durch deren individuelle Codes unterschieden. CDMA kommt beispielsweise in der Bandspreiztechnik zum Einsatz, für mehr Informationen zur Bandspreiztechnik siehe Kapitel Bandspreiztechnik. Durch die Unterscheidung anhand der unterschiedlichen Codes können alle Sender gleichzeitig und auf dem gleichen Channel senden (siehe Abbildung 22).



#### 7.1.3.4 Auswahl des Accessverfahrens

Die einzelnen Access-Verfahren werden analysiert und Vorteile sowie Nachteil aufgelistet. Anhand der Vor- und Nachteile wird entschieden welches Verfahren für unsere Kommunikation eingesetzt wird.

##### FDMA

Betrachten wir als erstes das FDMA Verfahren. Mit FDMA müsste jeder Sensorknoten einen eigenen Channel bzw. Frequenzband besitzen. Doch gerade in dem Frequenzbereich, in welchem LoRa angesiedelt ist, gibt das Bundesamt für Kommunikation (BAKOM) sehr strikte Vorgaben betreffend der Art der Benutzung, Sendeleistung und die maximale Sendezeit pro Stunde (Dutycycle). [24]

Vorteile:

- + Einfacher Transmitter
- + Ständiges Senden möglich

Nachteile

- Unterschiedliche Spezifikationen für Channels
- Receiver aufwendig

Wird FDMA benutzt, so müsste für jeden Channel unterschiedliche Einstellungen gemacht werden und der Receiver müsste alle benutzen Frequenzbänder unterstützen. Aus diesen beiden Gründen entfällt dieses Access-Verfahren als mögliche Lösung, somit bleiben noch:

- CDMA
- TDMA

##### CDMA

LoRa basiert auf der Bandspreiztechnik welche bereits eine Anwendung von **CDMA** darstellt. Durch die unterschiedlichen Spreizcodes ist es möglich, dass mehrer Sender gleichzeitig auf dem gleichen Frequenzband senden können ohne bzw. mit geringer gegenseitigen Interferenzen. Für weitere informationen betreffend Bandspreiztechnik siehe Kapitel Bandspreiztechnik.

Vorteile:

- + Einfache implementation → bereits in LoRa enthalten
- + Gute Störfestigkeit

Nachteile:

- Alle Sensor-Knoten müssen Senden bis sie durch Gateway „gehört“ wurden → Energie
- Jeder Sensorknoten muss von Beginn an einen anderen Spreizcode haben → Unterschiedliche Software für jeden Sensor-Knoten → Softwareaufwand.

Aufgrund des zusätzlichen Softwareaufwandes und des Energieverbrauchs wurde dieses Access-Verfahren nicht ausgewählt. Das CDMA Verfahren wird vom LoRa Funkstandard weiterhin für die Übertragung verwendet. Für unser Kommunikationsprinzip, das dem Funkstandard übergeordnet ist, wird es jedoch nicht mehr in Betracht gezogen.

## TDMA

Im Gegensatz zu **FDMA** hat **TDMA** keine Schwierigkeiten mit den unterschiedlichen Channel-Spezifikationen, da die gesamte Kommunikation nur über einen Channel stattfindet und somit der Channel nicht ständig gewechselt werden muss.

**TDMA** hat gegenüber **CDMA** den grossen Vorteil, dass die Sensorknoten in einen Ruhemodus gehen können, nachdem sie die Daten gesendet haben und nicht ständig senden müssen bis sie vom Gateway gehört werden.

Vorteile:

- + Einfache Implementation
- + Einfach erweiterbar
- + Energiesparend

Nachteile:

- Synchronisation nötig

Aufgrund der einfachen Implementation und Erweiterbarkeit wurde **TDMA** als Access-Verfahren ausgewählt.

### 7.1.4 Spezifikationen der Netzwerk-Kommunikation

Durch die Wahl von LoRa als Kommunikationsstandard wurde die Netzwerktopologie bereits vorgegeben, da LoRa nur Punkt zu Punkt Verbindungen ermöglicht. Somit wird die Stern-Topologie für das Sensor-Netzwerk ausgewählt.

Kommunikationsstandard	LoRA
Frequenz	868 MHz-868.3MHz
Sendeleistung	25mW (max)
Netzwerktopologie	Stern
Access-Verfahren	TDMA

Tabelle 4 Kommunikation Spezifikationen

### 7.1.5 Bandspreiztechnik

Das eingesetzte Modul von Libelium baut auf dem Transceiver Chip SX1272 von Semtech auf. Dieser Chip unterstützt unterschiedliche Funkstandards, unter anderem auch den Long Range, welcher im Allgemeinen unter dem Namen LoRa bekannt ist.

Dieser Funkstandard verwendet die Bandspreiztechnik Direct Sequenc Spread-Spectrum (DSSS). Durch DSSS wird ein schmalbandiges Signal durch eine Spreizung mit einem spezifischen Code in ein Signal mit einer grösseren Bandbreite umgewandelt, siehe Abbildung 23.

Durch diese Umwandlung muss die Sendeleistung nicht mehr höher als der allgemeine Rauschteppich sein. Somit wird es möglich die Sendeleistung zu verkleinern → low power, welches ganz im Sinne unseres akkubetriebenen Systems ist.

Nebst dem Low Power Aspekt ist LoRa durch die Bandspreiztechnik auch weniger anfällig auf Störungen. Wie in Abbildung 23 zu sehen ist, werden schmalbandige Störungen durch das Zurückspreizen „verschmiert“ und beeinflussen das Nutzsignal fast nicht mehr. [25]

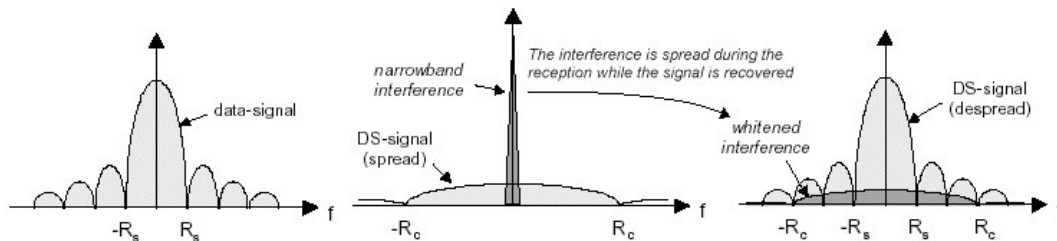


Abbildung 23 Direct Sequence Spread Spectrum (DSSS) [26]

Durch unterschiedliche Spreizcodes ist es möglich, dass mehrere User den gleichen Frequenzbereich benutzen können, da die anderen User nur als Rauschen im Nutz-Signal erscheinen.

## 7.2 Kommunikation einzelner Sensorknoten

Bei der Anwendung mit nur einem Sensorknoten, wird ein effizienter Kommunikationsstandard für die direkte Datenübertrag benötigt. Die Kommunikationsstandards welche wieder zur Auswahl stehen sind:

- ZigBee
- Bluetooth
- LoRa
- LoRaWAN
- Sigfox
- WiFi
- GSM/GPRS
- 3G

Der einzelne Sensorknoten sendet seine Messdaten direkt an den Webserver, dadurch wird ein Kommunikationsstandard mit einer hohen Reichweite benötigt. ZigBee, Bluetooth, LoRa und WiFi können aufgrund der zu geringen Reichweite bereits ausgeschlossen werden. Somit bleiben noch

- LoRaWAN
- Sigfox
- GSM/GPRS
- 3G

Wie bereits im Kapitel 7.1.2 beschrieben wurde, bestehen die Netze welche für Sigfox und LoRaWAN benötigt werden momentan noch nicht. Somit bleiben noch 3G und GSM/GPRS als Optionen übrig. Beide dieser Standards sind für unsere Anwendung bestens geeignet. Beide haben eine hohe Reichweite und unterstützen Mobilfunkdatenverbindung (Mobiles Internet). Der grösste Unterschied dieser zwei Protokolle ist deren Geschwindigkeit. Mit 3G werden die Daten schneller übertragen, was jedoch mehr Energie benötigt. Wegen der kürzeren Sendezeit mit 3G ist der Energieverbrauch etwa gleichzusetzen mit dem von GSM/GPRS.

### 7.2.1.1 Ausgewählter Kommunikationsstandard Gateway <-> Webserver

Für die Weiterleitung der im Gateway gesammelten Messdaten wurde **3G** als Kommunikationsstandard ausgewählt. GSM/GPRS hätte ebenfalls eingesetzt werden können, doch da wir bereits ein 3G-Modul zur Verfügung hatten fiel der Entscheid auf 3G. Dies ermöglicht es uns per Datenverbindung die Messdaten auf einen Webserver zu laden. Der höhere Energieverbrauch ist beim Gateway mit einem grösseren Solarpanel berücksichtigt. Detailinformationen betreffend dem 3G-Modul werden im Kapitel beschrieben.

### 7.2.1.2 Ausgewählter Kommunikationsstandard Sensorknoten <-> Webserver

Damit der einzelne Sensorknoten direkt auf den Webserver laden kann, kommt auch hier das 3G-Modul zum Einsatz. Der Upload der Messdaten wird gleich wie beim Gateway per 3G Datenverbindung gemacht. Detailinformationen betreffend 3G-Modul werden im Kapitel Kommunikation über 3G beschrieben.

### 7.2.2 Kommunikation über 3G

Mit 3G wird die dritte Generation der Mobilfunktechnologie beschrieben, welche auch unter dem Namen UMTS bekannt ist. UMTS hat zum Ziel die mögliche Datenrate im Vergleich zu GPRS drastisch zu erhöhen. Durch diese Verbesserung steht uns eine schnellere Datenverbindung zur Verfügung.

Als 3G-Modul kommt ein bereits bestehendes Modul von Libelium Waspote zum Einsatz. Dieses verwendet das 3G Modul SIM5218E von Simcom. Das 3G-Modul kann auf den Radio Socket 1 von dem Libelium Waspote gesteckt werden und wird dann über eine UART Schnittstelle gesteuert. Für die Kommunikation über UART kann die von Libelium zur Verfügung gestellte API verwendet werden.

Um auf die Sim-Karte zuzugreifen, muss der Pin Code richtig eingegeben werden. Die verwendete Sim Karte hat folgende Eigenschaften:

Eigenschaft	Wert
Tel. Nr.	079 487 78 99
Pin 1	8067
PUK 1	47829932
Pin 2 (wird aktuell verwendet)	5858
PUK 2	69524201
Abo	Swisscom unlimited

Tabelle 5 Informationen Sim- Karte

Wenn Daten über das Datennetz ausgetauscht werden, muss der APN (Access Point Name) entsprechend dem Betreiber richtig gesetzt werden. Damit wird dem Modul bekannt gemacht, zu welchem Netz es die Verbindung aufbauen soll und wie es sich dort einloggt. Bei der Swisscom können untenstehende Informationen verwendet werden, wobei Login und Passwort leergelassen werden können.

Eigenschaft	Wert
APN	gprs.swisscom.ch
Login	
Passwort	

Tabelle 6 APN Informationen Swisscom

## 8 Hardware

### 8.1 Hardwareübersicht

In untenstehendem Blockschaltbild ist die gesamte Hardware grafisch dargestellt. Es ist zu beachten, dass je nach Systemkonfiguration nicht alle Elemente vorhanden sind. Da der Prototyp aber modular aufgebaut ist, ist es mit kleinem Aufwand möglich die Konfiguration zu wechseln. In Tabelle 7 ist ersichtlich, bei welchen Konfigurationen welche Bestandteile verwendet werden.

Konfiguration	vorhanden
Einzelner Sensorstab	<ul style="list-style-type: none"> <li>- Libelium Waspote mit 3G Modul und Antenne</li> <li>- Sensorenprinte mit Adapterprint</li> <li>- Ausdehnungssensor</li> <li>- Solarpanel mit Laderegler und Akku</li> </ul>
Sensornetzwerk, konfiguriert als Sensorknoten	<ul style="list-style-type: none"> <li>- Libelium Waspote mit LoRa Modul und Antenne</li> <li>- Sensorenprinte mit Adapterprint</li> <li>- Ausdehnungssensor</li> <li>- Solarpanel mit Laderegler und Akku</li> </ul>
Sensornetzwerk, konfiguriert als Gateway	<ul style="list-style-type: none"> <li>- Libelium Waspote mit 3G- und LoRa Modul (je mit Antenne)</li> <li>- Solarpanel mit Laderegler und Akku</li> <li>- (Adapterprint optional)</li> </ul>

Tabelle 7 Konfigurationen der Hardware

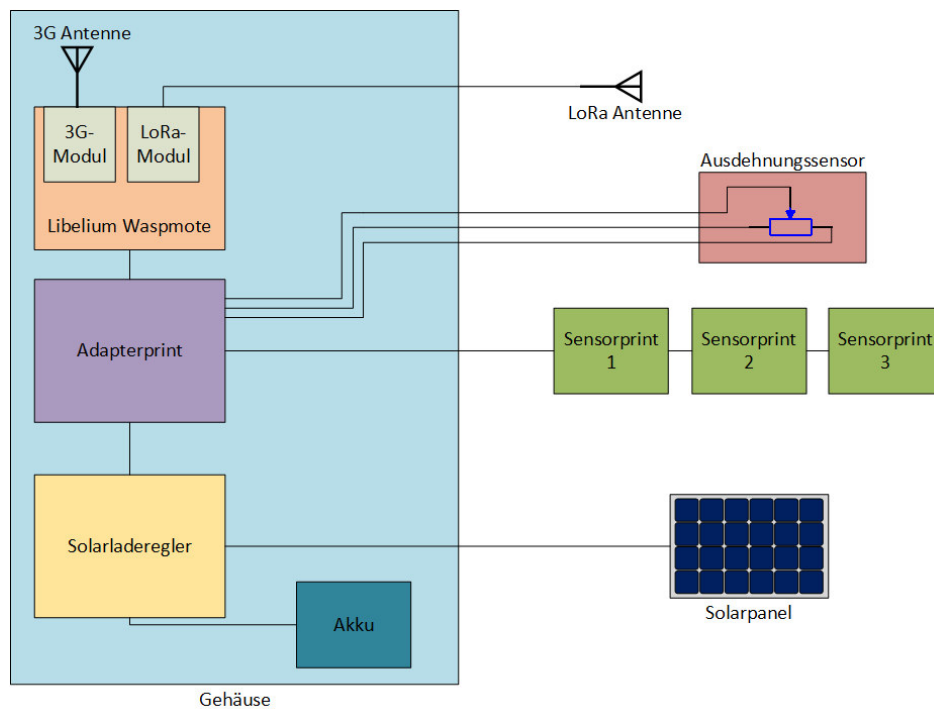


Abbildung 24 Blockschaltbild Hardware

Die Felsbewegungen werden mithilfe von drei Beschleunigungssensoren auf den Sensorprinten und einem Ausdehnungssensor gemessen. Alle gemessenen Daten werden dann über differentielles I<sup>2</sup>C per Kabel zum Adapterprint und von dort per Standard I<sup>2</sup>C zum Prozessor auf dem Entwicklungsboard Libelium Waspote geführt. Die Daten können dann über eine LoRa- Funkverbindung oder über die Mobilfunkverbindung 3G weitergesendet werden (je nach Konfiguration). Für die Energieversorgung ist der interne Akku zuständig. Der Akku wird über den Solarladeregler, an welchen ein Solarpanel angeschlossen ist aufgeladen. Der gesamte Prototyp ist in untenstehendem Foto ersichtlich.



Abbildung 25 Erstellter Prototyp eines Sensorknotens

### 8.1.1 Sensorprinte

Mithilfe der Sensorprinte werden die Winkel und die Temperaturen gemessen. Auf dem Sensorprint sind ein Beschleunigungssensor FXLS8471Q, ein Temperatursensor PCT2075 und der Wandler PCA9615, der Standard I<sup>2</sup>C zu differentielltem I<sup>2</sup>C wandelt, implementiert. Für einen Sensorknoten werden drei dieser Sensorprinte benötigt. Die Sensorprinte werden in einem 3D- gedrucktes Gehäuse eingesetzt und mit Silikon vergossen. Die Verbindung der drei Sensorprinte wird über ein geschirmtes CAT5 Kabel gemacht.

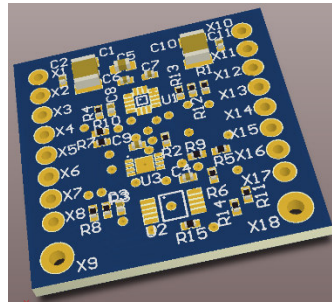


Abbildung 26 Sensorprint

### 8.1.2 Adapterprint

Damit das Libelium Waspote die Sensordaten auslesen kann, muss der differentielle I<sup>2</sup>C Bus wieder in ein Standard I<sup>2</sup>C Bus umgewandelt werden. Dies wird mit diesem Adapterprint bewerkstelligt. Ausserdem werden diverse Anschlüsse und GPIOs des Libelium Waspotes auf Stecker geführt.

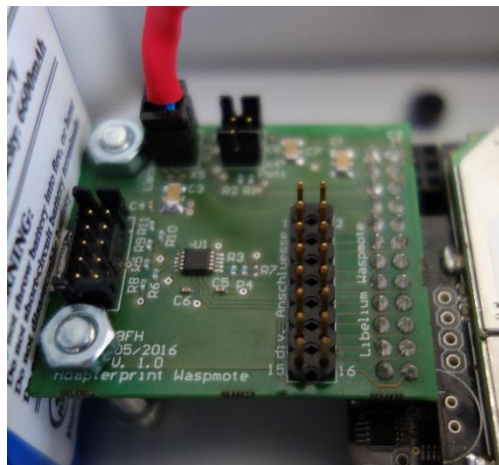


Abbildung 27 Adapterprint



### 8.1.3 Beschaltung PCA9615

Um Reflexionen zu verhindern, muss der differentielle I<sup>2</sup>C Bus an beiden Enden mit dem Wellenwiderstand des Kabels abgeschlossen werden (beim Übergangssprint zum Waspote und auf Sensorboard 3). In dem Datenblatt des ICs PCA9615 [27] wird dabei empfohlen ein CAT5 Kabel zu verwenden, da dies verdrehte Kabelpaare führt, was für eine differentielle Übertragung optimal ist. Für CAT5 Kabel wird empfohlen ein 120Ω Abschlusswiderstand einzusetzen. Ausserdem muss beachtet werden, dass die negative Leitung des differentiellen Busses im passiven Zustand ein tieferes Potential besitzt als die positive. Dies wird durch den Spannungsteiler mit zwei 390Ω- Widerständen und dem Abschlusswiderstand erreicht (Berechnung siehe Anhang).

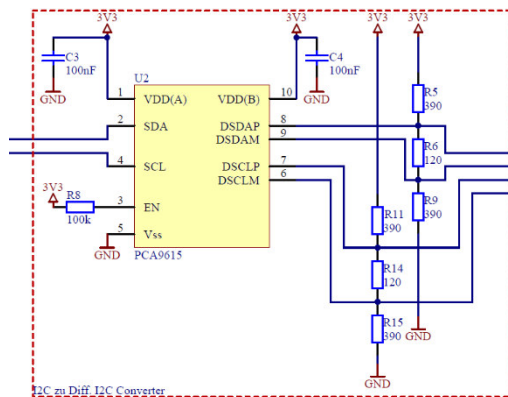


Abbildung 28 Beschaltung PCA9615

### 8.1.4 I<sup>2</sup>C Bus

Die Rolle des Masters übernimmt das Libelium Waspote und die Temperatur- sowie Beschleunigungssensoren agieren als I<sup>2</sup>C- Slaves. Die Slave- Adressen wurden bei dem System folgendermassen zugewiesen:

Sensor	I <sup>2</sup> C Slave Adresse in Hex
Beschleunigungssensor 1	1E
Beschleunigungssensor 2	1D
Beschleunigungssensor 3	1C
Temperatursensor 1	48
Temperatursensor 2	49
Temperatursensor 3	4A

Tabelle 8 Übersicht I2C Slave Adressen

### 8.1.4.1 I<sup>2</sup>C Pegel

Das I<sup>2</sup>C Signal ist ein digitales Signal und sollte deshalb für ein High 3.3V und für ein Low 0V betragen. Der Wandler PCA9615, der den Bus in einen differentiellen I<sup>2</sup>C Bus umwandelt hat allerdings laut Datenblatt bei einem Low-Pegel eine Spannung von 0.52V, dies ist jedoch kein Fehler. Dies führt zu ungewohnten Messresultaten (siehe untenstehendes Bild), da die Sensoren und das Wasmote aber einen Pegel bis 0.99V als Low erkennen ist dies kein Problem. [28] [29]

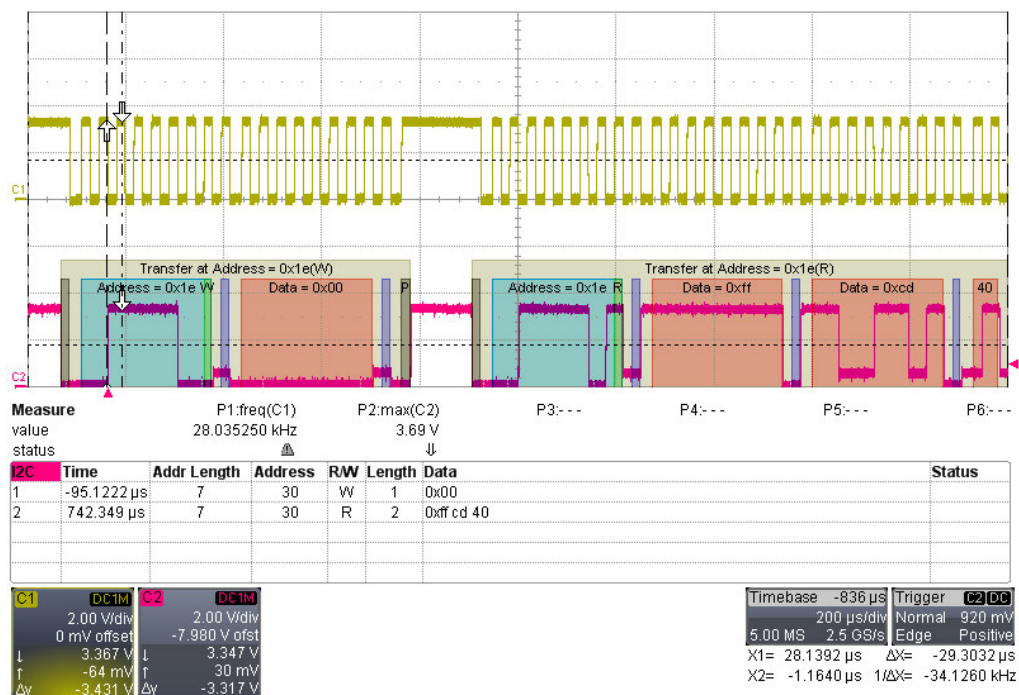


Abbildung 29 Pegel auf dem I<sup>2</sup>C Bus

## 8.2 Laderegler

Da der Sensor-Knoten sowie das Gateway keinen direkten Stromanschluss zur Verfügung haben, ist die Energieversorgung per Solarpanel vorgesehen. Das Wasmote bietet bereits die Möglichkeit ein Solarpanel anzuschliessen mit welchem ein LiPo Akku geladen werden kann.

Nach Recherchen im Libelium-Forum wurde klar, dass bei der Kombination des GSM-Moduls und des LoRa-Moduls das Solarpanel von Libelium nicht genügend Energie zur Verfügung stellen kann um den Akku auf längere Zeit zu stützen bzw. zu laden.

Der Solarpanel-Anschluss des Wasmotes hat sehr strikte Begrenzungen, welche zum einen die maximale Panelspannung betreffen sowie den maximalen Eingangsstrom. Die Begrenzungen sind folgende:

	Parameter	Bedingung
Solaranschluss	$V_{in}$	12V (max), 6V(optimal)
	$I_{in}$	280mA
Akkuanschluss	$V_{in}$	4.2V

Tabelle 9 Begrenzungen der Anschlüsse Libelium Wasmote

Aufgrund dieser Restriktionen, ist es nicht möglich ein leistungsfähigeres Panel anzuschliessen welches den Akku des Sensorknotens längerfristig stützen/laden könnte. Um diese Restriktionen zu umgehen wurde anstelle eines Akkus, ein selbst entwickelter Lade-Regler an den Akkuanschluss des Waspnotes angeschlossen.

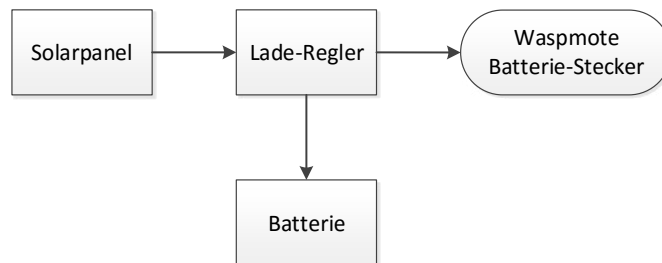


Abbildung 30 Übersicht Stromversorgung

Als Laderegler-Chip, wurde der LT3652 von Linear Technologies eingesetzt. Dieser Chip bietet viele Einstellungsmöglichkeiten, wie zum Beispiel die Ladeschlussspannung des Akku-Typs, den Ladestrom oder eine Akku-Fehler Erkennung. Durch den grossen Eingangsspannungsbereich von 5V-32V wird es möglich die Grösse des Solarpanels bzw. die Leistung des Panels an die Anwendung anzupassen. Beim Design des Ladereglers wurde darauf geachtet, dass alle Komponenten so dimensioniert werden, dass die maximale Eingangs- und Ausgangs-Spannung und der maximale Ausgangsstrom des Laderegler-Chips unterstützt werden. Die Spezifikationen des Ladereglers sind:

Parameter	Bedingung
$V_{in}$	4,95V - 32V
$I_{in (max)}$	3A
$V_{out}$	14,4V
$I_{out}$	2A

Tabelle 10 Spezifikationen Ladereglerprint

Die in Tabelle 9 beschriebenen Parameter können auch dem Datenblatt des LT3652 entnommen werden. [30]

### 8.2.1 Anschlüsse

Für die Verkabelung des Ladereglerprints mit der übrigen Hardware wurden verpolungssichere Stecker ausgewählt.

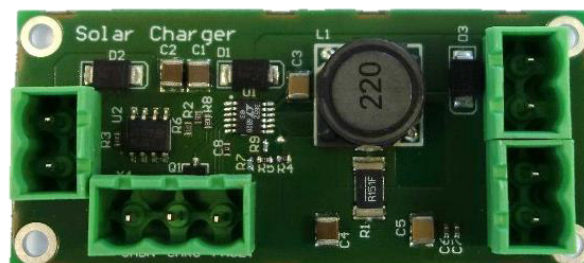


Abbildung 31 Solar-Laderegler

Der Laderegler-Chip kann anzeigen ob er momentan den Akku lädt oder ob ein Fehler beim Ladevorgang vorliegt. Zusätzlich kann der Laderegler extern durch einen Pin ausgeschaltet werden. Diese Informationen sind ebenfalls auf einen Stecker geführt und stehen für die Weiterverarbeitung zur Verfügung.

Das Wasmote kann mittels der Spannung des Akkus den Akkustand messen. Da wir jedoch einen Laderegler mit konstanter Ausgangsspannung an den Akkuanschluss des Wasmotes angeschlossen haben ist dies nicht mehr direkt möglich. Damit dies weiterhin möglich ist, muss die Ladeschaltung vom Akku getrennt werden, damit das Wasmote nur noch die effektive Akkuspannung misst. Der Laderegler kann mittels des SHDN-Pins ausgeschaltet werden und macht diese Akkustandmessung somit wieder möglich.

Anschluss	Funktion
CHRG (low aktiv)	Low: Akku wird geladen High: Akku ist Voll
FAULT (low aktiv)	(kann nur verwendet werden, falls eine zeitgesteuerte Ladung ausgewählt wird) Low: Akku Fehler High: Akku okay
SHDN	wird benutzt um den Ladevorgang zu unterbrechen High: Laderegler wird abgeschaltet Low: Laderegler eingeschalten

Tabelle 11 Anschlüsse Ladereglerprint

### 8.3 Solarpanel

Die Grösse der Solarpanels für den Sensorknoten und das Gateway wurden mit einem Excel-Sheet ausgerechnet. Hierfür wurden die Sendeleistungen und die Sendezeiten der Kommunikationsmodule verwendet. Anhand dieser beiden Angaben konnte ein Energiebedarf pro Tag ausgerechnet werden. Aus diesem Energiebedarf wurde anschliessend die Panelgrösse berechnet. Diese ist etwas überdimensioniert, damit die Versorgung besser gewährleistet werden kann.

Die Sensorknoten erhalten somit ein 10 Watt und das Gateway ein 20 Watt Solarpanel. Bei der Auswahl des Solarpanels wurde darauf geachtet, dass die Panels Wetterbeständig und solide sind.

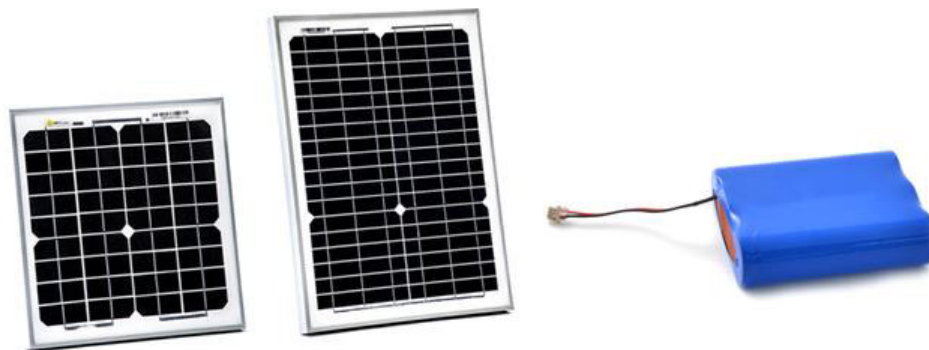


Abbildung 32 Solarpanel und Akku für Sensorknoten und Gateway [31] [32]

Um die erzeugte Solarenergie zu speichern und gleichzeitig das Wasmote zu Speisen wird ein Lithium Polymer Akku eingesetzt. Der Akku reicht für 1 Woche Betrieb ohne Aufladung durch das Solarpanel bei einem Sendeintervall von einer halben Stunde.

## 8.4 Gehäuse, Mechanik, Wetterbeständigkeit

Das gewählte Gehäuse sowie alle ausgewählten Stecker und Kabel entsprechen mind. der IP65 Norm. Der Sensorknoten ist somit ausreichend vor Wasser geschützt. Bei den Gehäusen der Sensorprinte werden die Kabel mithilfe von Kabeldurchführungen befestigt, somit kann gewährleistet werden, dass kein Wasser zur Elektronik durchdringt. Außerdem wurde die Sensorelektronik mit dem Silikon Elastosil RT 625 A von Wacker Silicones vergossen, um die Elektronik vor Wasser und Feuchtigkeit zu schützen. Es wird für die Verbindungen zwischen den Sensoren ein abgeschirmtes Kabel verwendet, die Gehäuse der Sensoren selbst sind jedoch nicht abgeschirmt. Der Schirm des Kabels wird immer beidseitig mit dem GND verbunden.

## 8.5 Bedienelemente

Damit gewisse Informationen von aussen sichtbar sind, wurden ein Taster mit integrierter LED und eine separate LED verwendet. Die genauere Funktion wird in den folgenden Kapiteln erklärt.



Abbildung 33 Taster mit Led (unten) und grüne Led (oben)

### 8.5.1 Schalter und Led in Schalter

Mithilfe des Schalters, kann der Sensorknoten oder das Gateway ausgeschaltet werden. Falls der Schalter gedrückt ist, ist das Gerät eingeschaltet, ansonsten ist die Stromzufuhr zu der Elektronik unterbrochen.

**Achtung:** Der Schalter muss gedrückt sein, damit das Gerät aktiv ist!

(Ausnahme: Wenn über USB ein PC angeschlossen ist, ist das Gerät ebenfalls aktiv)

Beim Einschalten wird während fünf Sekunden der Zustand des Akkus mithilfe der Led im Schalter signalisiert. Untenstehende Tabelle gibt einen Überblick über die Anzeige der Led.

Eigenschaft	Akkustand
Led leuchtet	Über 80%
Led blinkt (0.5 Sekunden an, 0.5 Sekunden aus)	Zwischen 40% und 80%
Led blinkt (0.1 Sekunden an, 0.9 Sekunden aus)	Unter 40%
Led leuchtet nicht	Akku leer

Tabelle 12 Anzeige Led in Schalter

### 8.5.2 Grüne Led

Die grüne Led hat je nachdem ob es sich um das Gateway oder den Sensorknoten handelt verschiedene Funktionen:

Gateway: leuchtet, wenn Daten über LoRa empfangen  
Blinkt, wenn Daten auf Server geladen werden konnten  
Leuchtet nicht, wenn in SleepMode

Sensorknoten: leuchtet, wenn LoRa Verbindung zu Gateway  
Blinkt, wenn Daten gesendet werden  
Leuchtet nicht, wenn in SleepMode

## 8.6 Serielle Schnittstelle

Es ist möglich über die USB- Schnittstelle eine serielle Verbindung mit dem Gerät herzustellen und zu sehen in welchem Status sich das Gerät befindet. Beachtet werden muss, das bei jedem Einstecken des USB- Stecker ein Reset ausgeführt wird und das Programm neu startet. Damit der Computer eine serielle Verbindung herstellen kann, muss ein Treiber für den FTDI – Chip vorhanden sein. Dieser kann falls nötig auf der Seite von FTDI heruntergeladen werden. [33]

Die Einstellungen für die serielle Verbindung sind hier zu sehen:

Einstellung	Wert
Baudrate	115'200
Daten	8 Bit
Parität	Keine
Stopp-Bit	1 Bit
Flusskontrolle	keine

Tabelle 13 Einstellungen serielle Schnittstelle

Mithilfe eines Terminalprogramms kann die serielle Schnittstelle ausgelesen werden und so kann der Ablauf der Kommunikation und die Resultate der Messungen beobachtet werden.

## 8.7 Inbetriebnahme

### 8.7.1 Sensorprint

Bei der Inbetriebnahme der Sensorprinte und dem Adapterprint für das Libelium Waspmote ist aufgefallen, dass manchmal der Prozessor beim Einschalten mehrmals neu startet. Der Grund dafür war, dass auf all diesen Printen mehrere 100µF Kondensatoren bestückt waren. Diese brauchten zum Laden am Anfang zu viel Strom, was einen Neustart des Waspmotes zur Folge hatte. Die 100µF Kondensatoren wurden alle durch 1µF Kondensatoren ersetzt und das Problem tritt nicht mehr auf.

### 8.7.2 Ladereglerprint

Der Laderegler Print, welcher für die Ladung des Akkus per Solarpanel zuständig ist, wurde zuerst ohne Akku und Solarpanel in Betrieb genommen.

Als erstes wurde ein Netzgerät an den Solaranschluss des Ladereglers angeschlossen und die Leerlaufspannung am Akkuanschluss gemessen. Bei dieser Messung schaltete der Laderegler ohne irgendeinen Grund aus und nach unbestimmter Zeit wieder ein. Nach genauerer Analyse der Schaltung konnte der Grund für dieses unerklärliche Verhalten gefunden werden. Die Ursache war das floating Gate des MOSFETS welcher für das Ein- und Ausschalten des Ladereglers zuständig ist. Nachdem ein Pulldown-Widerstand nach GND auf der Rückseite des Ladereglers bestückt wurde, funktionierte die Schaltung wie gewünscht. Dieser Widerstand wurde im Schema eingezeichnet. Anschliessend wurde der Laderegler mit Panel und Akku getestet und funktionierte wie vorgesehen.

### 8.7.3 Modifikation Or-Gatter und Dioden

Das Libelium Wasmote wird durch den Li-Po Akku gespeisen. Der Akku wird normalerweise durch den Ladereglerprint per Solarpanel geladen. Wird ein Libelium Wasmote per USB mit einem PC verbunden, so wird der Akku standardmässig durch eine Beschaltung auf dem Wasmote geladen.

Um mögliche Probleme durch gleichzeitiges Laden per USB und Ladereglerprint zu vermeiden, muss der Ladereglerprint den Ladevorgang unterbrechen, sobald ein USB Kabel eingesteckt wird. Das Ausschalten des Ladereglerprints kann durch einen High-Pegel am SHDN Pin vorgenommen werden. Dafür muss eine Spannung grösser als 1.1V angelegt werden.

Da der Laderegler jedoch auch durch einen Output Pin des Wasmotes ausgeschaltet werden kann, wird hier eine Oder-Verknüpfung benötigt. Die USB Speisung und der Output Pin werden mit der in Abbildung 34 dargestellten Schaltung logisch Or verknüpft.

Aufgrund der unterschiedlichen Spannungspegel (USB 5V, Wasmote 3.3V) wurde einfachheitshalber ein passives Or-Gatter mit zwei Dioden und einem Widerstand realisiert. Dieses hat am Ausgang entweder den High Pegel vom USB (5V) oder den High Pegel des Wasmotes (3.3V), jeweils minus die Diodenspannung. Falls beide Eingänge Low sind, wird der Ausgang mit einem Pulldown Widerstand ebenfalls auf Low (0V) gezogen.

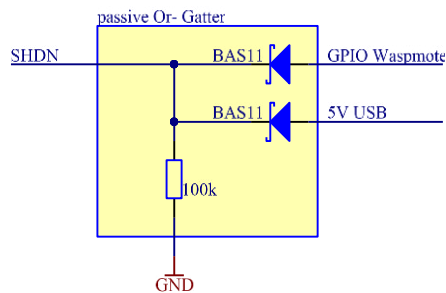


Abbildung 34 passives Or- Gatter

Da jedoch die USB-Schnittstelle auch für die Ausgabe von Debug-Messages verwendet wird, hat sich die Speisespannung des USB-Ports, selbst wenn kein Kabel angeschlossen ist, nicht unter 1.9V gesenkt. Durch diese Rest-Spannung wurde das Laden des Wasmotes durch den Ladereglerprint unterbrochen. Eine mögliche Lösung für dieses Problem bestand darin alle Debug-Nachrichten aus dem Source-Code zu entfernen. Als Alternativlösung wurden auf der Hardware mit Hilfe von drei Dioden in Serie ein Spannungsabfall von 2.1V erzeugt. Durch diese Beschaltung konnte sichergestellt werden, dass trotz der Rest-Spannung von 1.9V der Ladereglerprint eingeschaltet bleibt.

Somit ist der SHDN Pin bei einem High Pegel vom Wasmote (3.3V) nur noch 1.2V und bei einem High Pegel vom USB (5V) nur noch 2.9V. Dies reicht jedoch um den Mosfet auf dem Ladereglerprint durchzuschalten und den Laderegler somit auszuschalten. Damit auch der Low Zustand sauber durchgeschaltet wird, trotz Dioden, wurde ein zusätzlicher Pulldown- Widerstand direkt auf dem Ladereglerprint eingefügt.

Diese Beschaltung sollte in Zukunft durch einen Schmitt Trigger ersetzt werden, der bei den gewünschten Pegeln durchschaltet und somit wieder ein digitales Signal generiert.

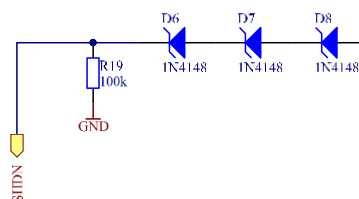


Abbildung 35 drei Dioden in Serie für 2.1V Spannungsabfall

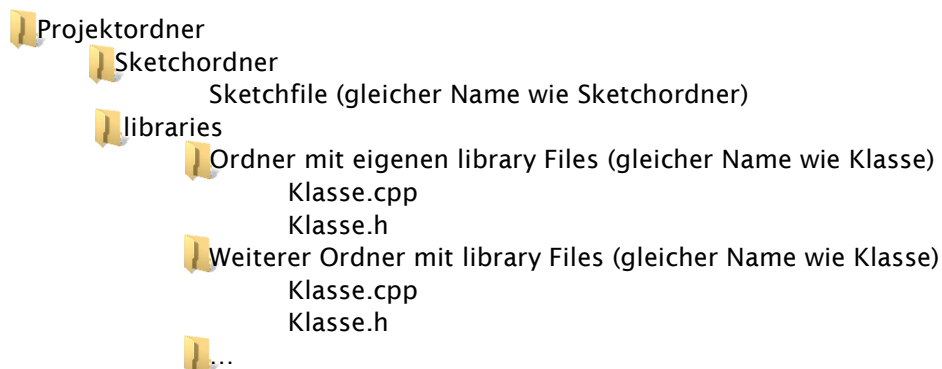
## 9 Software

Die Software wurde geschrieben um auf das Libelium Waspote Pro V1.2 zu laden. Dazu wird die Pro IDE V4.0 von Libelium zusammen mit der API V.023 von Libelium verwendet. Die API von Libelium stellt bereits sehr viele Funktionen zur Verfügung, was das Erstellen von Software beschleunigt. Die eigene Softwarebibliothek muss in der Programmiersprache C++ verfasst werden. Dabei wird der objektorientierte Ansatz verwendet. Es wurde eine eigene Softwarebibliothek in C++ geschrieben. Diese liest die Daten der Sensoren aus, und übermittelt diese per LoRa / 3G. Die ganze Software wurde mithilfe von Doxygen dokumentiert. Damit wurde eine Dokumentation zur Software in HTML- Form erzeugt. Diese Software Dokumentation kann direkt unter diesem Link aufgerufen werden: <http://rockmonitoring.netpresence.ch/Software/>

### 9.1 File Struktur

Da die Software mit der Libelium IDE kompiliert und auf das Waspote geladen wird, muss eine genaue Ordner- und File Struktur gegeben sein, damit die Software richtig funktioniert.

Als oberste Schicht ist der Software- Projektordner, darin enthalten sind die Ordner libraries und Sketchordner. Der Sketchordner muss dabei gleich heissen, wie das darin enthaltene Sketchfile (.pde). Im Sketchfile können die Waspote- API aus dem Installationspfad der IDE und die eigenen Softwarebibliotheken im Ordner Libraries verwendet werden. Eine Vorgabe dieser Libraries ist es, dass pro C++ Klasse ein Ordner gibt, der gleich heisst wie die Klasse und in welchem sich das entsprechende Cpp- und Header- File befindet. Mit der Libelium IDE wird dann das Sketch File angepasst und auf das Waspote geladen (inklusive aller verwendeten Libraries). Die Dateien der eigenen Softwarebibliothek sollten idealerweise in einem externen Editor geschrieben werden. Ausserdem muss bei den Einstellungen der Libelium IDE der Sketchbook Speicherort den Pfad des Projektordners enthalten, damit die Libraries gefunden werden.



### 9.2 Übersicht eigene Softwarelibrary

Die Software wurde nach dem objektorientierten Ansatz geschrieben. Dabei wurde für die Temperatursensoren und die Beschleunigungssensoren je eine eigene Klasse erstellt, welche die Sensoren konfiguriert und ausliest. Alle Sensoren wurden dann in der Klasse Sensors zusammengefasst, damit alle Sensoren mit einem Aufruf angesprochen werden können.

Nebst der Sensors Klasse wurde eine weitere Klasse für die Steuerung der Leds erstellt. Die Methoden dieser Klasse enthalten noch viele Delays, welche das Programm blockieren. Damit dies nicht mehr passiert, sollten dafür zukünftig Interrupts eingesetzt werden.

Auch für die beiden Kommunikationsstandards 3G und LoRa wurde je eine Klasse erstellt. Diese Klassen steuern den Ablauf der Kommunikation und die Einstellungen der jeweiligen Kommunikations-Module.



Da unsere Hardware modular aufgebaut ist, wurde dieses Prinzip auch in der Software so umgesetzt. Die einzelnen Systemkonfigurationen bestehen aus unterschiedlichen Kombination der oben genannten Klassen. Um eine Übersicht zu erhalten, welche Systemkonfiguration welche Klassen benützt, können die untenstehenden Klassendiagramme betrachtet werden. Ein genauerer Beschrieb der Klassen mit ihren Attributen und Methoden ist in der Doxygen Dokumentation zu finden (siehe Kapitel Software).

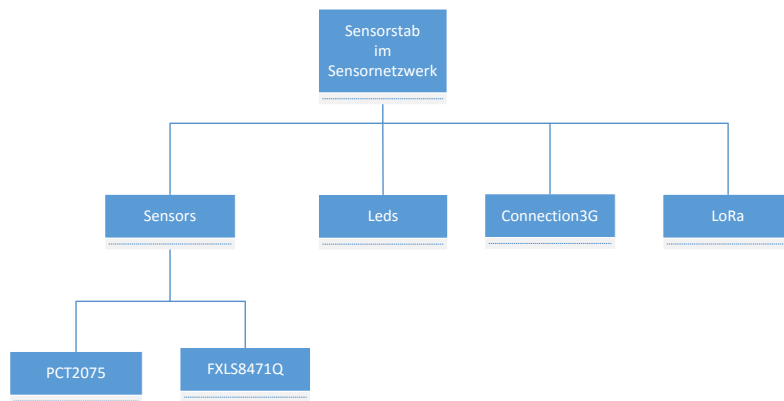


Abbildung 36 Klassendiagramm Sensorstab im Netzwerk

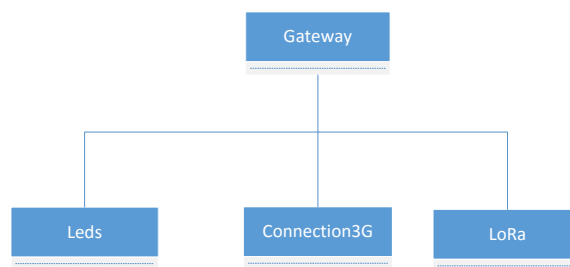


Abbildung 37 Klassendiagramm Gateway

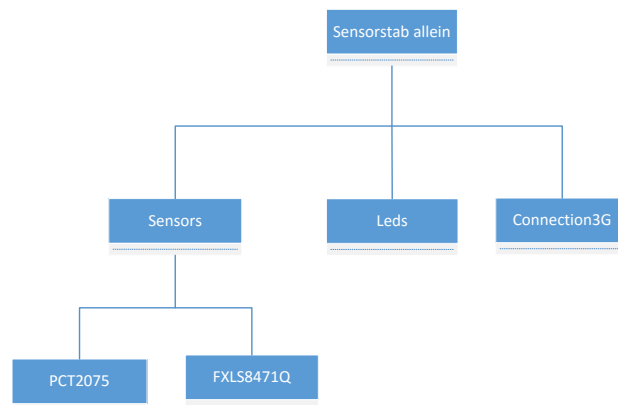


Abbildung 38 Klassendiagramm Sensorstab alleine

### 9.3 Programmieren

Für das Programmieren des Gerätes wird die Entwicklungsumgebung mit der API von Libelium benötigt. Diese kann auf der Webseite von Libelium heruntergeladen werden. [34]

Danach muss sichergestellt werden, dass die Libraries kopiert wurden und die Ordnerstruktur aussieht wie dies im Kapitel 9.1 beschrieben wird. Je nach gewünschter Systemkonfiguration kann dann folgendes Sketchfile auf das Gerät geladen werden:

Konfiguration	Programm
einzelner Sensorstab	Sensorstab_GSM
Sensornetzwerk, konfiguriert als Sensorknoten	Gateway_LoRa
Sensornetzwerk, konfiguriert als Gateway	Gateway_LoRa

Tabelle 14 Überblick über Software der verschiedenen Systemkonfigurationen

### 9.4 Kommunikationsablauf

Wie bereits in den Projektzielen erwähnt besteht die Möglichkeit einen einzelnen Sensorknoten zu installieren oder mehrere Sensorknoten zu einem Sensor-Netzwerk zusammen zu fassen. Für beide Systemvarianten wurde eine geeignete Software für die Kommunikation entworfen.

#### 9.4.1 Sensorknoten

In der Systemvariante mit nur einem Sensorknoten, besteht das System nur aus dem Sensorknoten und dem Webserver. Der Datenaustausch geschieht direkt per 3G.

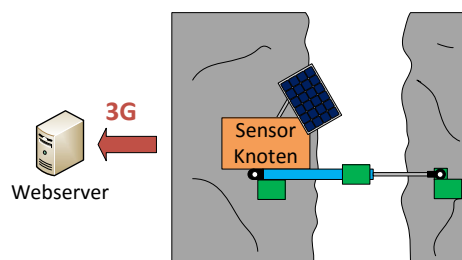


Abbildung 39 Sensorknoten direkte Datenübertragung

#### Aufgaben Sensorknoten:

- Messung machen
- Daten auf Webserver laden
- Deep Sleep

#### Aufgaben Webserver:

- Daten abspeichern
- Daten grafisch darstellen
- ACK an Sensorknoten schicken

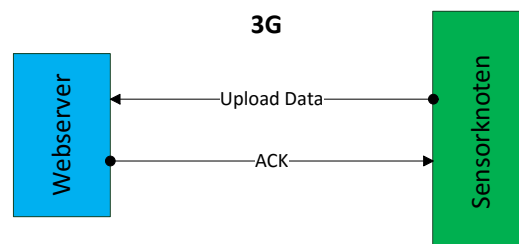


Abbildung 40 Kommunikationsablauf eines einzelnen Sensorknotens

Diese Systemvariante ist vom Softwareaufwand sehr überschaubar und konnte zu grossen Teilen aus der Software für das Sensornetzwerk zusammengestellt werden.

### 9.4.2 Sensornetzwerk

Beim Sensornetzwerk sind Sensorknoten, das Gateway und ein Webserver an der Kommunikation beteiligt. Um sich einen Überblick zu verschaffen werden als erstes die Aufgaben der einzelnen Systeme angeschaut.

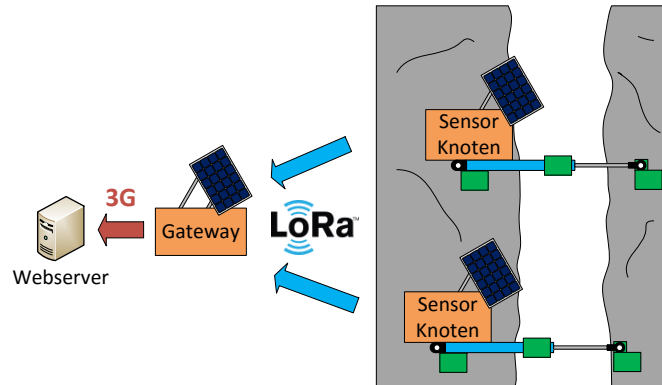


Abbildung 41 Sensor-Netzwerk

#### Aufgaben Sensorknoten:

- Messung machen
- Data Request empfang und mit ACK bestätigen
- Daten per LoRa senden
- Warten auf Sleep Command
- Deep Sleep

#### Aufgaben Gateway:

- Data Request an alle beteiligten Sensorknoten schicken
- Messdaten von jedem Sensorknoten empfangen
- Daten per 3G auf Webserver laden
- Deep Sleep Command per LoRa senden

#### Aufgaben Webserver:

- Daten abspeichern
- Daten grafisch darstellen
- ACK an Gateway schicken

Der Datenaustausch wird durch den Data Request des Gateway initiiert und läuft anschliessend nach dem Schema in Abbildung 42 ab.

Damit die Datenübermittlung möglichst sicher ist, wird jeder Befehl/Daten mit einem Acknowledgment bestätigt.

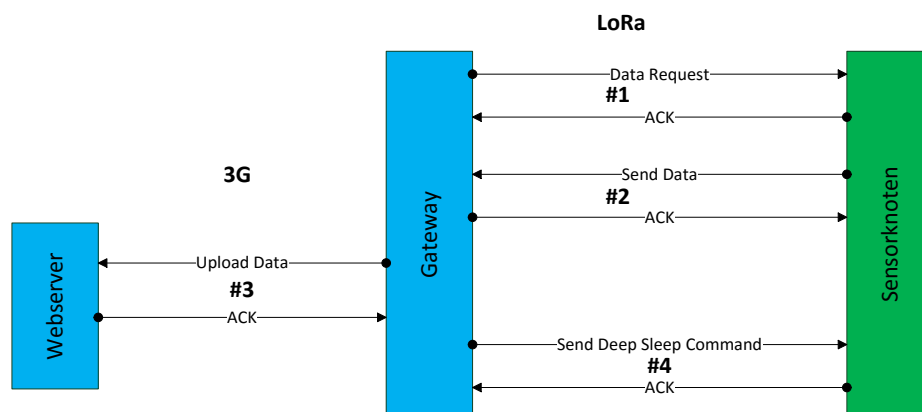


Abbildung 42 Kommunikationsablauf eines Sensorknotens

Für mehrere Sensorknoten sieht der Ablauf relativ ähnlich aus, zuerst werden wieder alle Informationen gesammelt und anschliessend auf den Webserver geladen.

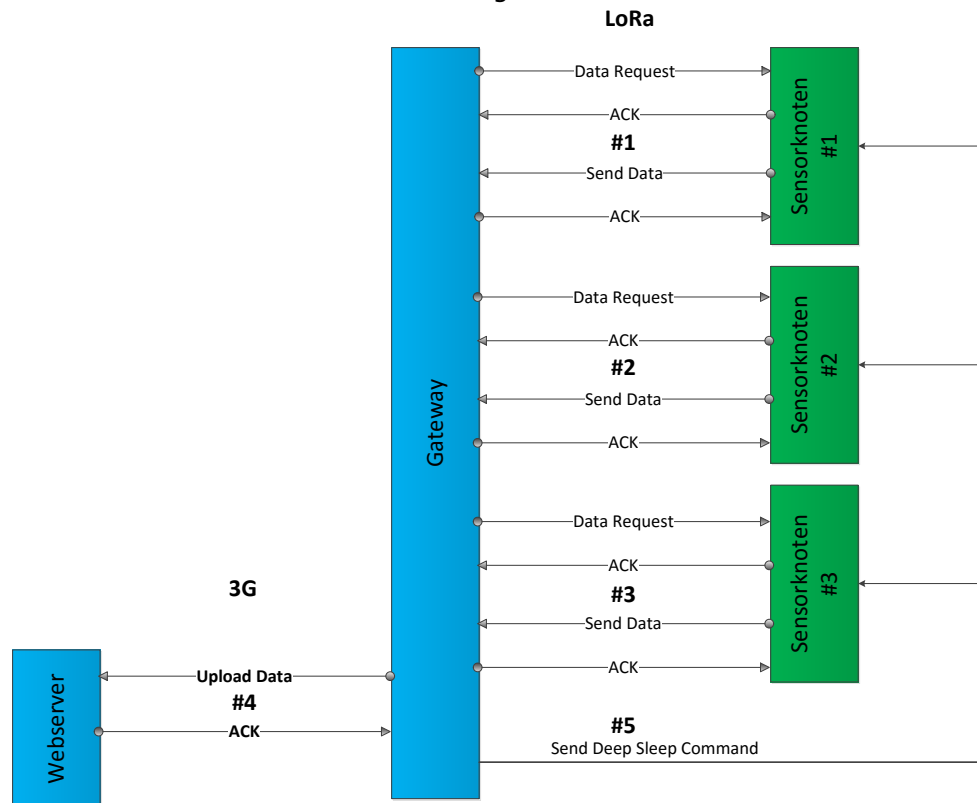


Abbildung 43 Kommunikationsablauf mehrerer Sensorknoten

Durch das ausgewählte Access-Verfahren TDMA müsste die Sensorknoten und das Gateway regelmässig synchronisiert werden, um sicher zu stellen dass die Sensorknoten zur richtigen Zeit aus dem Ruhemodus erwachen um den Data Request des Gateways zu erhalten.

Damit wir diese Synchronisation umgehen können, sendet das Gateway nachdem es alle Aufgaben erledigt hat einen Deep Sleep Command an alle Sensorknoten. Nach empfangen des Deep Sleep Commands, gehen die Sensorknoten in den Deep Sleep mode und wachen nach einer vordefinierten Zeit wieder auf (siehe Abbildung 44).

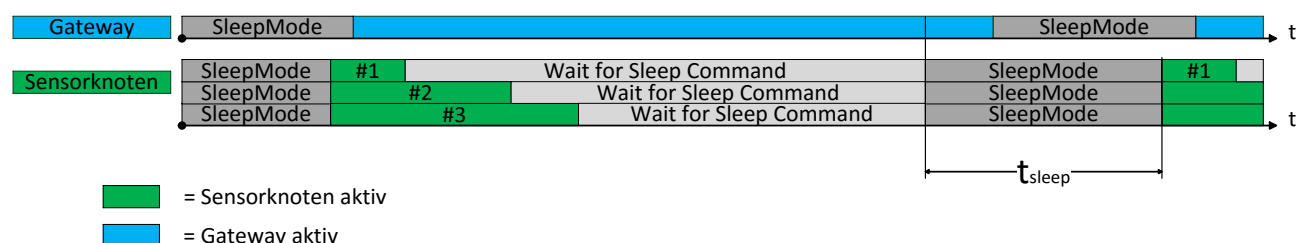


Abbildung 44 Zeitlicher Kommunikationsablauf

Diese Request/Response Vorgehen wurde ausgewählt um zukünftige Funktions- bzw. Befehls-Erweiterungen möglichst einfach implementieren zu können.

#### 9.4.2.1 Data-Request empfangen

Um der Anforderung nach Low Power Kommunikation gerecht zu werden, wurde darauf geachtet so wenig wie möglich zu senden. Das LoRa-Modul bietet die Möglichkeit Aktivitäten auf dem Übertragungskanal anhand des Signalpegels zu erkennen. Diese Channel Activity Detection (CAD) benötigt weniger Energie als vergleichsweise das Empfangen.

Wie bereits beschrieben schickt der Sensorknoten seine Daten nur wenn er einen Data Request des Gateways erhalten hat. Dieser Prozess erfolgt entsprechend der Grafik in Abbildung 45.

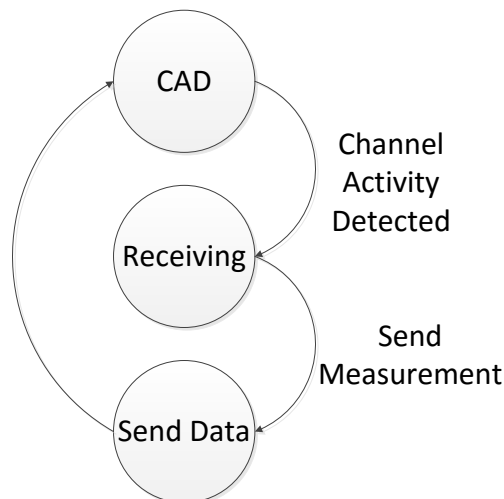


Abbildung 45 Data Request Ablauf

#### 9.5 Upload der Daten auf Webserver über 3G

Um die Sensordaten auf einen Webserver zu laden, wird das HTTP Protokoll verwendet. Zum Übertragen der Daten wird der Befehl GET verwendet. Dieser ist ursprünglich dazu gedacht Daten abzurufen. Es ist aber sehr einfach mit Hilfe von Argumenten bei dem Get- Aufruf Daten zu übertragen. Der Get- Aufruf hat untenstehenden Aufbau:

GET/**Skript.php**?**name1=value1&name2=value2&name3=value3**

(Orange: Auszuführendes Skript, Grün: Variablenname, Blau: Variablenwert)

Übertragen werden folgende Daten als Argumente:

Variable	
name	Name des Sensors
pitch1	Pitch- Winkel des Sensors 1
roll1	Roll- Winkel des Sensors 1
pitch2	Pitch- Winkel des Sensors 2
roll2	Roll- Winkel des Sensors 2
pitch3	Pitch- Winkel des Sensors 3
roll3	Roll- Winkel des Sensors 3
pot	Ausdehnung des Ausdehnungssensors
temp1	Temperatur des Sensors1
temp2	Temperatur des Sensors2
temp3	Temperatur des Sensors3

Tabelle 15 Eigenschaften der Get- Anfrage

Alle Winkel und die Ausdehnung werden als 16 Bit Integer mit 2 Kommastellen (z.B. 1234 für 12.34°) gesendet. Die Temperaturen werden direkt als vorzeichenbehaftete Bytes gesendet. So wird die Menge an übertragenen Daten minimiert. Der Server beantwortet die erfolgreiche Anfrage mit einer Bestätigung.

## 9.6 Server

Die Daten der Sensoren gelangen schlussendlich auf einen Webserver. Auf diesem werden alle erhaltenen Daten in einer Datei gespeichert. Diese Daten sind jederzeit abrufbar und auf der Webseite in einer übersichtlichen Grafik dargestellt. Um Zugriff auf die Daten und die Skripts zu erhalten, werden untenstehende Zugangsdaten benötigt. Für den Benutzer reicht es jedoch aus, wenn er mit der untenstehenden URL die Webseite aufruft. Dies ist ein privater Server von Benjamin Habegger und steht in Basel. In Zukunft könnte man aber auch einen anderen Webserver einrichten.

Zugangsdaten Webserver:

URL: <http://rockmonitoring.netpresence.ch/>  
Server: <ftp.netpresence.ch>  
User: [rockmonitoring@webworkz.ch](mailto:rockmonitoring@webworkz.ch)  
Password: Network4Sensors!

## 9.7 Software auf Server

Auf dem Server laufen nebst einem kurzen Teil in HTML mehrere PHP Skripts. Diese werten die Empfangenen Daten aus und speichern die Daten in einer .csv Datei. Des Weiteren gibt es eine Klasse, die die Daten formatiert und ein Skript, das die Daten ausliest und in einer Grafik mit Hilfe von Highcharts darstellt.

File	Beschreibung
index.php	Wird beim Öffnen der obenstehenden URL gestartet, beinhaltet HTML- und Css- Code für die Darstellung der Webseite Ausserdem werden im PHP- Teil ein Objekt der Klasse DataFormatter erstellt und die von den Sensorknoten zuletzt erhaltenen Daten vom Objekt DataFormatter geholt und angezeigt.
LogData.php	Nimmt die GET- Anfragen entgegen, liest die damit übertragenen Daten aus und antwortet auf die Anfrage. Die Daten werden in einem csv- File abgespeichert. Darin enthalten sind die Daten und der Zeitpunkt bei dem die Daten empfangen wurden im Unix Timeformat. Somit wird die Zeit mit GMT-0 abgespeichert. Der Schweiz hat im Sommer GMT+2 und im Winter GMT+1. Dies muss bei der Auswertung der Daten berücksichtigt werden.
DataFormatter.php	Die Klasse DataFormatter dient dazu die abgespeicherten Daten aus dem csv- File auszulesen. Dazu stellt sie mehrere Methoden zur Verfügung.
DisplaySingleSensor.php	Diese Seite wird aufgerufen, wenn man die Daten eines Sensorknotens betrachten möchte. Es wird wiederum ein Objekt der Klasse DataFormatter erstellt. Dann werden die Daten in Grafiken dargestellt. Dies geschieht mithilfe der HighCharts Bibliothek und den Daten vom DataFormatter. Die Zeitachse zeigt Zeit für GMT-0 an. (siehe LogData.php)

Tabelle 16 Server Skripts

Ordner	Beschreibung
HighCharts	Enthält die Softwarebibliothek HighCharts, mit dieser werden die Daten graphisch dargestellt. [35]
Data	Enthält pro Sensorknoten eine csv-Datei, in der alle Daten die empfangen wurden gespeichert sind

Tabelle 17 Server Ordner

## 10 Optimierungen

Wie im Kapitel Inbetriebnahme bereits erwähnt, sollen bei den Sensorboards und dem Adapterprint bei Verwendung mit dem Libelium Waspote 1  $\mu$ F Kondensatoren eingesetzt werden anstatt der geplanten 100  $\mu$ F Kondensatoren.

Der DeepSleep Modus des Libelium Waspotes bietet verschiedene Modi an. Um am meisten Energie sparen zu können, wird die Funktion All\_OFF gesetzt. Diese konnte jedoch nicht verwendet werden, da die Spannungsversorgung der Sensoren dann ausgeschaltet wird und dies zu Problemen auf dem I<sup>2</sup>C Bus des Waspotes führt. Darum sollten die Sensoren in Zukunft extern mit Spannung versorgt werden (direkt von Batterie über Spannungsregler, nicht über Waspote) oder es wird ein I<sup>2</sup>C Isolator eingesetzt. Momentan wurde der DeepSleep verwendet mit der Option Socket0\_Off. [36]

Die Temperatursensoren und die Beschleunigungssensoren lösen eine Pegeländerung einer Interrupt Leitung aus, wenn die Temperatur gewisse Grenzen über-/ unterschreitet oder gewisse Beschleunigungsevents geschehen sind. Dieser Pin wird momentan auf einen GPIO Pin des Entwicklungsboards geführt und nicht ausgewertet. Um die Interrupts als Alarm zu verwenden, müssen diese auf die Interrupt Pins des Waspotes umverdrahtet werden. Ausserdem müssen bei den Sensoren die Ausgabe der Interrupts konfiguriert werden und bei dem Libelium Waspote die Interrupts ausgelesen werden. Damit könnte das Waspote jederzeit (auch im DeepSleep Modus) von einem externen Interrupt aufgeweckt werden und einen Alarm ausgeben (z.B. per SMS).

Solange die Sensorboards in Silikon vergossen werden, ist das momentane Design ok. Falls aber der Print in Zukunft einmal anders fixiert wird, muss an Löcher zur Halterung auf dem PCB gedacht werden. Zudem sollte bei den Sensorprinten Stecker für die Verbindung mit dem Kabel vorgesehen werden. Dies vereinfacht die Montage der Sensoren im Gehäuse.

Um die Elektronik optimal gegen elektromagnetische Störungen zu schützen, müsste die ganze Elektronik in abgeschirmten Gehäusen montiert werden und auch die Stecker sollten eine Abschirmung haben.

Die Or- Schaltung, die im Kapitel Modifikation Or-Gatter und Dioden erklärt wird, wird optimalerweise auf einem PCB platziert (z.B. auf dem Ladereglerprint) und die drei Dioden durch einen Schmitt-Trigger ersetzt. Der Pulldown-Widerstand nach den drei Dioden kann dann ebenfalls auf dem Ladereglerprint platziert werden.

## 11 Weiterführung der Arbeit

Die Systemkonfiguration 1, mit einem einzelnen Sensorknoten, funktionierte in den Labormessungen einwandfrei. Deshalb können in der Weiterführung des Projektes erste Messungen mit dieser Systemkonfiguration an einer Felswand durchgeführt werden.

Die Systemkonfiguration Sensornetzwerk funktionierte bei Labortests ebenfalls stabil. Es blieb allerdings nicht genügend Zeit, um umfassende Tests durchführen zu können. Deshalb sollte das LoRa Netzwerk mit mehreren Sensorknoten noch einmal gründlich getestet werden, bevor auch damit Messungen an einer Felswand durchgeführt werden.

Um ein komplettes Sensornetzwerk, mit mehreren Sensorknoten aufzubauen, müsste noch zusätzliche Hardware erstellt werden. Momentan steht eine Hardware als Sensorknoten und eine als Gateway (ist umkonfigurierbar) zur Verfügung.

Das User Interface kann noch erweitert beziehungsweise optimiert werden. Zum Beispiel könnte, ein GUI auf dem Webserver entworfen werden, mit welchem verschiedene Konfigurationen an den Sensorknoten vorgenommen oder Zustände abgefragt werden könnten.

Um die Konfiguration der Sensorknoten über die USB-Schnittstelle auch durch einem Laien zu ermöglichen (anstatt Änderung in Code), könnte ein weiteres GUI entwickelt werden.

Als Software Erweiterung, könnten die Messdaten auf einer SD-Karte zwischen gespeichert werden, damit die Daten auch verfügbar sind wenn sie nicht gesendet werden können.

Falls das Projekt in einem grösseren Umfang weitergeführt wird und ein Produkt daraus hervorgehen soll, muss die komplette Hardware überarbeitet werden. Dazu sollte ein eigenes Microcontroller-Board designt werden, um auf die teure Hardware von Libelium verzichten zu können. Die Schaltungen, die momentan sehr modular auf mehreren PCBs verteilt sind, können dann auf einem einzelnen Print zusammengefasst werden. Dadurch kann dann auch die Grösse der Elektronik drastisch verkleinert werden.



## 12 Zusammenfassung

Es wurde in dieser Arbeit ein Prototyp zur Messung von Felsbewegungen entwickelt. Dieser misst mithilfe eines Ausdehnungssensors die Ausdehnung von zwei Felsen. Weiter messen drei Beschleunigungssensoren die Winkel der zwei Felswände sowie den Winkel des Sensorstabs. Der Sensorstab entspricht dabei dem Unterschied der Bewegung der beiden Felsen. Die Ausdehnung kann mit einer Genauigkeit von 0.3mm genau bestimmt werden und bei der Winkelmessung kann bei konstanter Raumtemperatur eine Genauigkeit von  $\pm 0.1$  Grad und in einem Bereich von  $-10^{\circ}\text{C}$  bis  $+70^{\circ}\text{C}$  von  $\pm 0.6$  Grad garantiert werden.

Die Messdaten können erfolgreich direkt via 3G Mobilfunknetz auf einen Webserver geladen werden. Alternativ kann ein Sensornetzwerk eingerichtet werden, das die Daten mehrerer Messeinrichtungen sammelt und diese dann an den Webserver weiterleitet.

Für Gehäuse, Stecker und Kabel wurden nur Produkte ausgewählt, die mindestens der Norm IP65 entsprechen und somit gegen Eindringen von Staub und Strahlwasser aus allen Richtungen geschützt sind. [37]

Die Elektronik wird über einen Akku mit Energie versorgt. Dieser Akku kann zuverlässig über das Solarpanel oder den USB- Anschluss aufgeladen werden. Mit einem Messintervall von 30 Min kann der Sensorknoten insgesamt sieben Tage lang ohne Aufladung durch das Solarpanel betrieben werden.

Es wurden alle grundlegenden Ziele der Arbeit erreicht. Zusätzlich konnten noch die erweiterten Ziele erreicht werden, bei denen das Sensornetzwerk mit mehreren Sensorknoten funktioniert und dass die Messeinrichtung eine Woche ohne Solarpanel auskommt.

## 13 Abbildungsverzeichnis

Abbildung 1 Rissmesser (links) [1] und Rissmesser 3 Dimensional (rechts) [2]	5
Abbildung 2 Messanlage mit Rissmesser [3]	6
Abbildung 3 Libelium Waspote [4]	8
Abbildung 4 Messgrößen Ausdehnung (orange) und Winkel	9
Abbildung 5 Schema und Berechnung Ausdehnungssensor	9
Abbildung 6 Roll- und Pitch Winkel ( [5] S.6)	10
Abbildung 7 Winkelmessung mit Beschleunigungssensor	10
Abbildung 8 Sensitivität bei Berechnung mit 3 Achsen ( [5] S.13)	11
Abbildung 9 Rauschen mit Beschleunigungssensor im normalen Modus	13
Abbildung 10 Rauschen mit Beschleunigungssensor im 1024-Oversampling Modus	14
Abbildung 11 Pitch- und Rollwinkel unter Temperatureinfluss	15
Abbildung 12 Störung auf differentiell Signal [7]	16
Abbildung 13 Übersicht I <sup>2</sup> C und Diff. I <sup>2</sup> C Bus	16
Abbildung 14 I <sup>2</sup> C Bus Protokoll [8]	17
Abbildung 15 Sensorknoten	18
Abbildung 16 Systemkonfiguration einzelner Sensorknoten	18
Abbildung 17 Sensornetzwerk	19
Abbildung 18 Netzwerk Topologien [9]	20
Abbildung 19 Star-Topologie [12]	21
Abbildung 20 Übersicht Kommunikation Sigfox [14]	22
Abbildung 21 Sigfox- Abdeckung [15]	23
Abbildung 22 Accessverfahren [23]	24
Abbildung 23 Direct Sequence Spread Spectrum (DSSS) [26]	27
Abbildung 24 Blockschaltbild Hardware	30
Abbildung 25 Erstellter Prototyp eines Sensorknotens	31
Abbildung 26 Sensorprint	32
Abbildung 27 Adapterprint	32
Abbildung 28 Beschaltung PCA9615	33
Abbildung 29 Pegel auf dem I <sup>2</sup> C Bus	34
Abbildung 30 Übersicht Stromversorgung	35
Abbildung 31 Solar-Laderegler	35
Abbildung 32 Solarpanel und Akku für Sensorknoten und Gateway [31] [32]	36
Abbildung 33 Taster mit Led (unten) und grüne Led (oben)	37
Abbildung 34 passives Or- Gatter	39
Abbildung 35 drei Dioden in Serie für 2.1V Spannungsabfall	39
Abbildung 36 Klassendiagramm Sensorstab im Netzwerk	41
Abbildung 37 Klassendiagramm Gateway	41
Abbildung 38 Klassendiagramm Sensorstab alleine	41
Abbildung 39 Sensorknoten direkte Datenübertragung	42
Abbildung 40 Kommunikationsablauf eines einzelnen Sensorknotens	42
Abbildung 41 Sensor-Netzwerk	43
Abbildung 42 Kommunikationsablauf eines Sensorknoten	43
Abbildung 43 Kommunikationsablauf mehrerer Sensorknoten	44
Abbildung 44 Zeitlicher Kommunikationsablauf	44

## 14 Tabellenverzeichnis

Tabelle 1 Anforderungen Entwicklungsboard	8
Tabelle 2 Anforderungen Beschleunigungssensor	12
Tabelle 3 Kommunikationsstandards	23
Tabelle 4 Kommunikation Spezifikationen	26
Tabelle 5 Informationen Sim- Karte	29
Tabelle 6 APN Informationen Swisscom	29
Tabelle 7 Konfigurationen der Hardware	30
Tabelle 8 Übersicht I2C Slave Adressen	33
Tabelle 9 Begrenzungen der Anschlüsse Libelium Waspote	34
Tabelle 10 Spezifikationen Ladereglerprint	35
Tabelle 11 Anschlüsse Ladereglerprint	36
Tabelle 12 Anzeige Led in Schalter	37
Tabelle 13 Einstellungen serielle Schnittstelle	38
Tabelle 14 Überblick über Software der verschiedenen Systemkonfigurationen	42
Tabelle 15 Eigenschaften der Get- Anfrage	45
Tabelle 16 Server Skripts	46
Tabelle 17 Server Ordner	46

## 15 Glossar

**API**

application programming interface, Schnittstelle zur Anwendungsprogrammierung

---

**CAN**

Controller Area Network, industrielles serielles Bussystem

---

**CDMA**

Code Division Multiple Access, Trennung von mehreren Sendern anhand Spreizcode

---

**DSSS**

Direct Sequenc Spread-Spectrum, Bandspreiztechnik

---

**FDMA**

Frequency Division Multiple Access, Verwendung verschiedener Frequenzbänder

---

**GPIO**

General Purpose Input/Output, Anschlusspin des Prozessors, konfigurierbar als Ein- oder Ausgang

---

**GSM**

Global System for Mobile Communications, weltweiter Funkstandard

---

**I<sup>2</sup>C**

Inter-Integrated Circuit, ein Kommunikationsbus für kurze Distanzen

---

**IDE**

Integrated Development Environment, integrierte Software Entwicklungsumgebung

---

**IOT**

Internet Of Things, Internet der Dinge, verbindet Alltagsgegenstände zunehmend mit dem Internet

---

**LoRa**

Long Range Radio, Spezifikation für drahtlose Verbindungen

---

**LoRaWAN**

Long Range Wide Area Network, Spezifikation für drahtlose Netzwerke

---

**LoS**

Line of Sight, direkte Sichtverbindung

---

**Pitch- Winkel**

Winkel der Drehung um Y- Achse

---

**Roll- Winkel**

Winkel der Drehung um X- Achse

---

**TDMA**

Time Division Multiple Access, zeitliche Trennung von mehreren Sendern

---

**UART**

Universal Asynchronous Receiver Transmitter, serielle Schnittstelle

---

**UMTS**

Universal Mobile Telecommunications System, auch bekannt als 3G

---

**Zigbee**

Spezifikation für drahtlose Netzwerke

---

## 16 Quellenverzeichnis

- [1] „Interfels,“ [Online]. Available: <http://www.interfels.de/pages/lineare+potentiometer+rissmesser+crackmeter>. [Zugriff am 29 Juli 2016].
- [2] „Geo-Instrumentation,“ [Online]. Available: <http://www.geo-instrumentation.com/fissurometer-fi3d>. [Zugriff am 29 Juni 2016].
- [3] „Sommer Messtechnik,“ [Online]. Available: <http://www.sommer.at/de/anwendungen/geologie/warnsystem-steinbruch-spitz>. [Zugriff am 29 Juni 2016].
- [4] Libelium, „Libelium,“ [Online]. Available: <http://www.libelium.com/>. [Zugriff am 16 Mai 2016].
- [5] ST Microelectronics, „AN 3182 Tilt measurement using a low-g 3 axis accelerometer,“ 2010.
- [6] M.-. u. F. (. VDE/VDI-Gesellschaft Mikroelektronik, Elektromagnetische Verträglichkeit in der KFZ-Technik, Berlin: VDE, 2005.
- [7] „Prad,“ 27 Juni 2016. [Online]. Available: [http://www.prad.de/images/monitore/specials/inputlag/dvi\\_tmids-differential\\_noise.png](http://www.prad.de/images/monitore/specials/inputlag/dvi_tmids-differential_noise.png).
- [8] „Opencores,“ 27 Juni 2016. [Online]. Available: <http://opencores.org/usercontent,img,1352582784>.
- [9] „kl2217,“ [Online]. Available: <https://kl2217.wordpress.com/2009/06/30/network-topologies/>. [Zugriff am 29 Juni 2016].
- [10] „wikipedia,“ 25 Juni 2016. [Online]. Available: [https://de.wikipedia.org/wiki/Vermaschtes\\_Netz](https://de.wikipedia.org/wiki/Vermaschtes_Netz). [Zugriff am 25 Juni 2016].
- [11] „Wikipedia,“ 25 Juni 2016. [Online]. Available: [https://de.wikipedia.org/wiki/Topologie\\_\(Rechnernetz\)](https://de.wikipedia.org/wiki/Topologie_(Rechnernetz)). [Zugriff am 25 Juni 2016].
- [12] L. Waspnote, „Lora-Networking Guide,“ 2016.
- [13] LinkLabs, „Link Labs,“ 13 Januar 2016. [Online]. Available: <http://www.link-labs.com/sigfox-vs-lora/>. [Accessed 21 April 2016].
- [14] „Newelectronics,“ [Online]. Available: <http://www.newelectronics.co.uk/electronics-technology/developers-vie-to-create-the-communications-standard-of-choice-for-the-iot/110411>. [Zugriff am 29 Juni 2016].
- [15] „Sigfox,“ [Online]. Available: <http://www.sigfox.com/coverage>. [Zugriff am 29 Juni 2016].
- [16] „NZZ,“ 25 Juni 2016. [Online]. Available: <http://www.nzz.ch/nzzas/nzz-am-sonntag/swisscom-lanciert-ein-mobilnetz-fuer-das-internet-der-dinge-1.18502444>.
- [17] „www.digi.com,“ [Online]. Available: <http://www.digi.com/products/xbee-rf-solutions/modules/xbee-802-15-4#specifications>. [Zugriff am 19 April 2016].
- [18] „www.digi.com,“ [Online]. Available: <http://www.digi.com/products/xbee-rf-solutions/modules/xbee-pro-868#specifications>. [Zugriff am 19 April 2016].
- [19] „www.digi.com,“ [Online]. Available: <http://www.digi.com/products/xbee-rf-solutions/modules/xbee-pro-900hp#specifications>. [Zugriff am 19 April 2016].
- [20] *Swiss National Frequency Allocation Plan 2016*, Schweiz, 2016.
- [21] „www.semtech.com,“ [Online]. Available: <http://www.semtech.com/images/datasheet/sx1272.pdf>. [Zugriff am 19 April 2016].
- [22] „/www.cooking-hacks.com,“ [Online]. Available: <https://www.cooking-hacks.com/bluetooth-module-pro-for-arduino>. [Zugriff am 19 April 2016].
- [23] „International Telecommunication Union,“ [Online]. Available: <http://www.itu.int/osg/spu/ni/3G/technology/>. [Zugriff am 29 Juni 2016].
- [24] B. f. Kommunikation, „Swiss\_National\_Frequency\_Allocation\_Plan\_2016,“ 2016.
- [25] „Maximintegrated,“ 25 Juni 2016. [Online]. Available: <https://www.maximintegrated.com/en/app-notes/index.mvp/id/1890>.

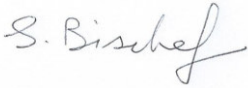
- [26] „okob,“ [Online]. Available: <https://www.okob.net/texts/mydocuments/80211physlayer/>. [Zugriff am 29 Juni 2016].
- [27] NXP Semiconductors, „PCA9615, Product Datasheet, 2-channel multipoint Fast-mode Plus differential I2C-bus buffer with hot-swap logic,“ 2014.
- [28] NXP, „FXLS8471Q Datasheet,“ 2015.
- [29] NXP, „PCT2075 Datasheet,“ 2014.
- [30] Linear Technology, „LT3652 Datasheet,“ 2010.
- [31] „solarkontor,“ [Online]. Available: <https://solarkontor.de/20w-solarmodul-monokristallin-12V>. [Zugriff am 29 Juni 2016].
- [32] „Cooking-Hacks,“ [Online]. Available: <https://www.cooking-hacks.com/6600ma-h-rechargeable-battery>. [Zugriff am 29 Juni 2016].
- [33] FTDI, „FTDI Chip,“ [Online]. Available: <http://www.ftdichip.com/FTDrivers.htm>. [Zugriff am 29 Juni 2016].
- [34] Libelium, „Libelium SDK,“ [Online]. Available: [http://www.libelium.com/development/waspmote/sdk\\_applications](http://www.libelium.com/development/waspmote/sdk_applications). [Zugriff am 29 Juni 2016].
- [35] Highcharts, „Highcharts,“ [Online]. Available: <http://www.highcharts.com/>. [Zugriff am 28 Juni 2016].
- [36] Libelium, „Libelium Forum,“ [Online]. Available: <https://www.libelium.com/forum/viewtopic.php?f=28&t=21707>. [Zugriff am 10 Juni 2016].
- [37] Reinmedical, „Reinmedical,“ [Online]. Available: <http://www.reinmedical.com/de/technik/ip-schutzklassen.html>. [Zugriff am 29 Juni 2016].

## 17 Selbständigkeitserklärung

Wir bestätigen, dass wir die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der im Literaturverzeichnis angegebenen Quellen und Hilfsmittel angefertigt habe. Sämtliche Textstellen, die nicht von uns stammen, sind als Zitate gekennzeichnet und mit dem genauen Hinweis auf ihre Herkunft versehen.

Ort, Datum: 30.06.2016

Ort, Datum: 30.06.2016

Unterschrift:   
Swen Bischof

Unterschrift:   
Daniel Frei

## 18 Anhang

### - Berechnungen

- Abschlusswiderstand differentielles I<sup>2</sup>C
- Leds

### Hardware

- Schemen
- Anschlüsse



## Berechnung Abschlusswiderstand differentieller I2C Bus

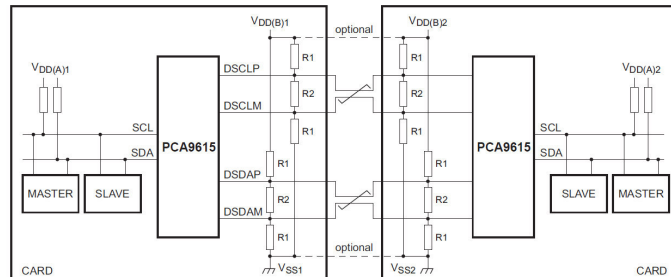


Abbildung Abschluss differentielles I2C aus Datenblatt PCA9615

Empfohlen vom Datenblatt bei 5V Speisespannung:

Abschlusswiderstand R2: 120 Ohm

Spannungsteiler R1: 600 Ohm

Dies ergibt bei passiven Leitungszuständen folgende Leitungspegel:

Leitung Positiv: 
$$U_p = \frac{V_{cc}}{R_1 + R_1 + R_2} \cdot (R_1 + R_2) = \frac{5V}{600\Omega + 600\Omega + 120\Omega} \cdot (600\Omega + 120\Omega) = 2.727V$$

Leitung Negativ: 
$$U_n = \frac{V_{cc}}{R_1 + R_1 + R_2} \cdot (R_1) = \frac{5V}{600\Omega + 600\Omega + 120\Omega} \cdot (600) = 2.273V$$

Spannungsdifferenz: 
$$U_{pn} = U_p - U_n = 2.727V - 2.273V = 0.454V$$

Nun wurden die Widerstände angepasst für eine 3.3V Speisespannung, so dass die Spannungsdifferenz (0.454V) und der Abschlusswiderstand (120 Ohm) gleich gross bleiben.

Spannungsdifferenz: 
$$U_{pn} = 0.454V$$

Leitung Positiv: 
$$U_p = \frac{V_{cc}}{2} + \frac{U_{pn}}{2} = \frac{3.3V}{2} + \frac{0.454V}{2} = 1.877V$$

Leitung Negativ: 
$$U_n = \frac{V_{cc}}{2} - \frac{U_{pn}}{2} = \frac{3.3V}{2} - \frac{0.454V}{2} = 1.423V$$

Strom durch Spannungsteiler 
$$I = \frac{0.454V}{120\Omega} = 3.78mA$$

Spannungsteiler R1: 
$$R_1 = \frac{1.423V}{3.78mA} = 376.123\Omega \Rightarrow \underline{\underline{390\Omega}}$$

Rückrechnung mit R1 = 390 Ohm und R2 = 120 Ohm:

Leitung Positiv: 
$$U_p = \frac{V_{cc}}{R_1 + R_1 + R_2} \cdot (R_1 + R_2) = \frac{3.3V}{390\Omega + 390\Omega + 120\Omega} \cdot (390\Omega + 120\Omega) = 1.87V$$

Leitung Negativ: 
$$U_n = \frac{V_{cc}}{R_1 + R_1 + R_2} \cdot (R_1) = \frac{3.3V}{390\Omega + 390\Omega + 120\Omega} \cdot (390\Omega) = 1.43V$$

Spannungsdifferenz: 
$$U_{pn} = U_p - U_n = 0.44V$$

ursprünglicher Wert der Spannungsdifferenz war 0.454V, somit 14mV Differenz, was einer Abweichung von unter 4 Prozent entspricht und somit sehr genau ist.

## Berechnung Led Vorwiderstände

Led in Schalter:

$$U_{Led} = 2.1V \text{ typ. (aus Datenblatt)}$$

$$U_R = V_{cc} - U_{Led} = 3.3V - 2.1V = 1.2V$$

$$I = 20mA \text{ (aus Datenblatt)}$$

$$R = \frac{U}{I} = \frac{1.2V}{20mA} = 60 \Omega \Rightarrow 62\Omega$$

Led grün:

$$U_{Led} = 2V \text{ typ. (aus Datenblatt)}$$

$$U_R = V_{cc} - U_{Led} = 3.3V - 2V = 1.3V$$

$$I = 15mA \text{ (aus Datenblatt)}$$

$$R = \frac{U}{I} = \frac{1.3V}{15mA} = 86.7 \Omega \Rightarrow 91\Omega$$

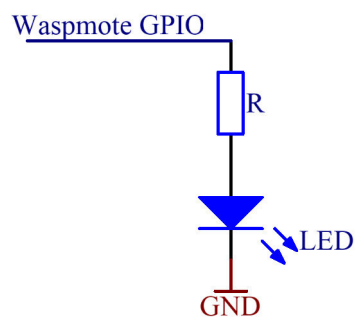
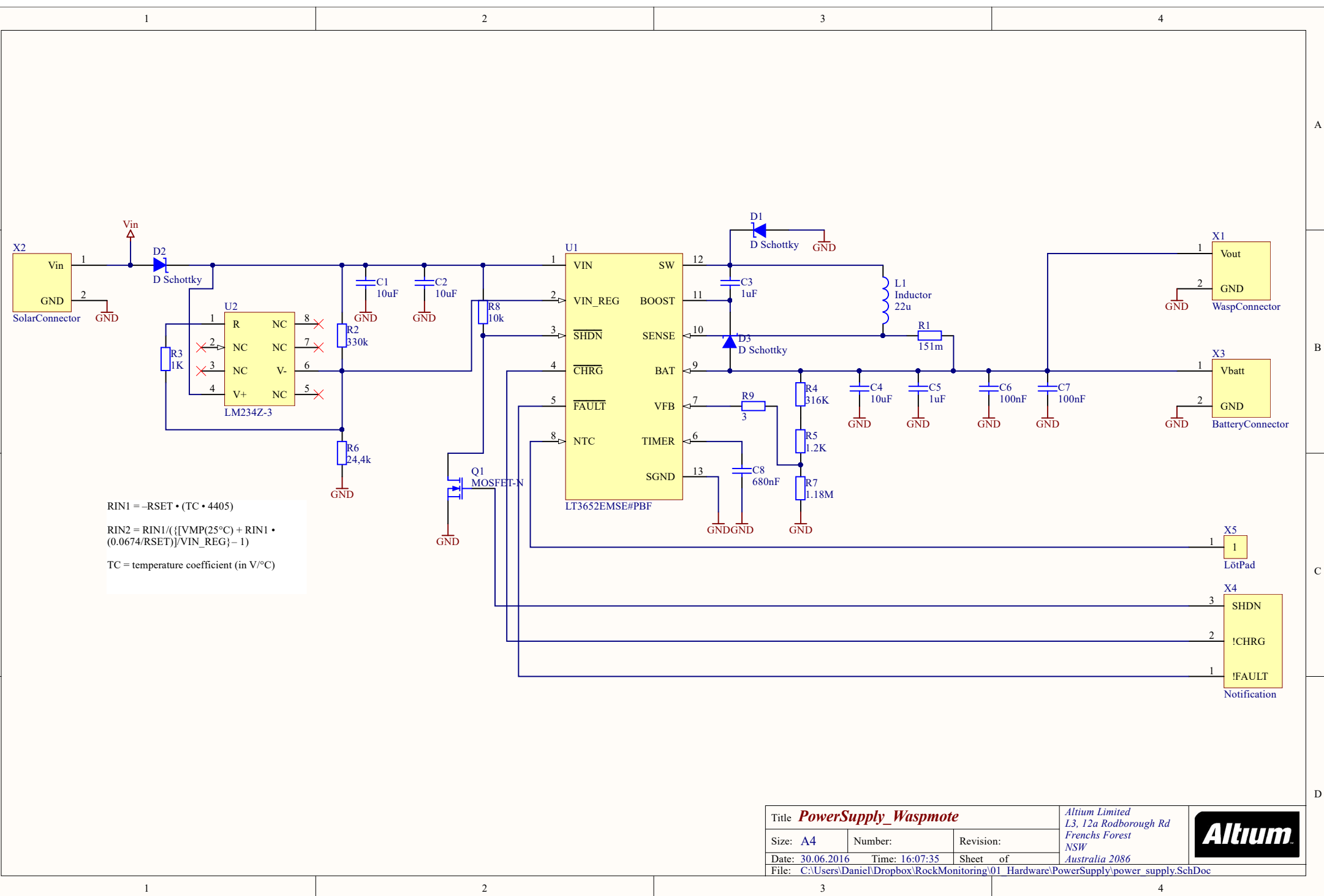
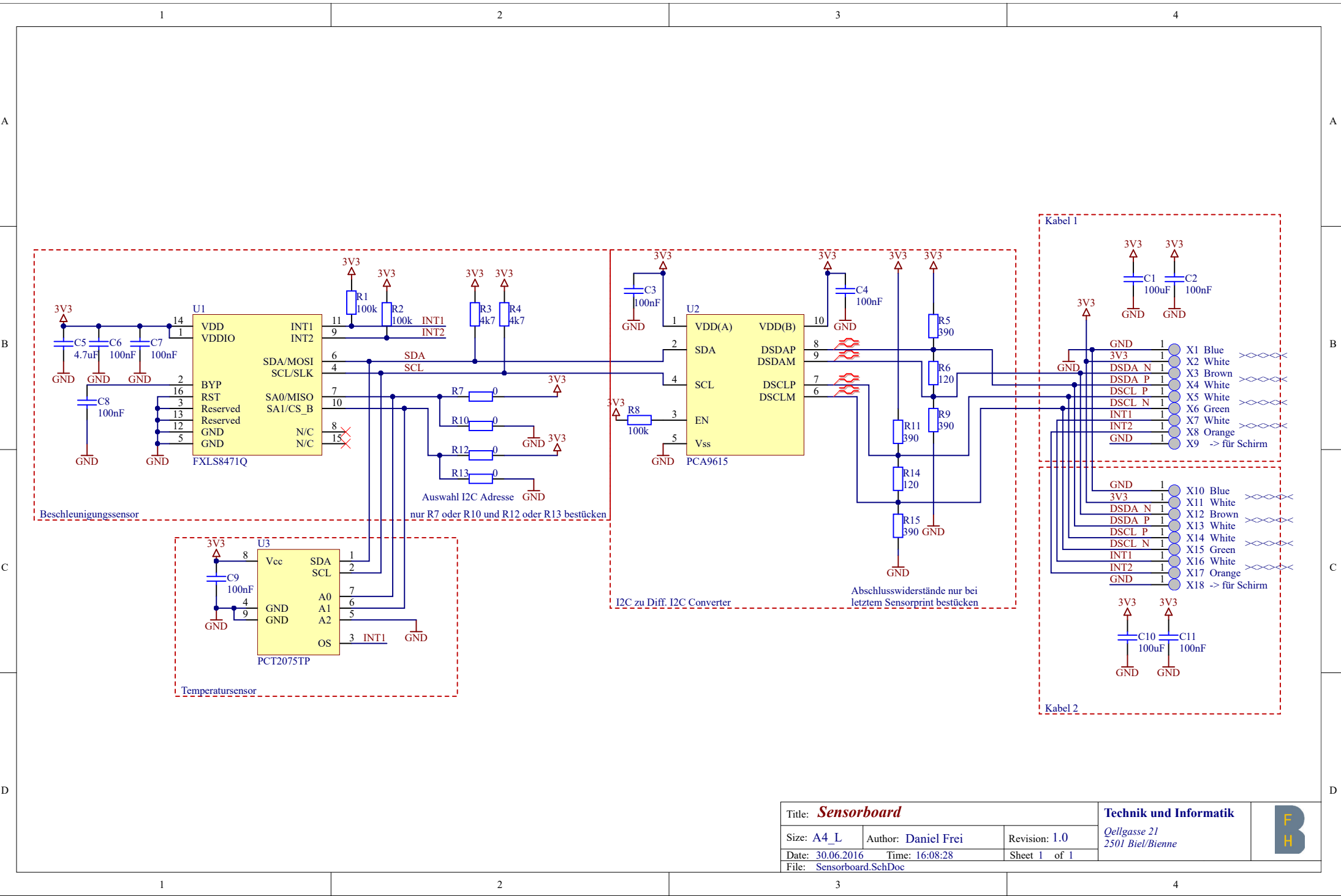
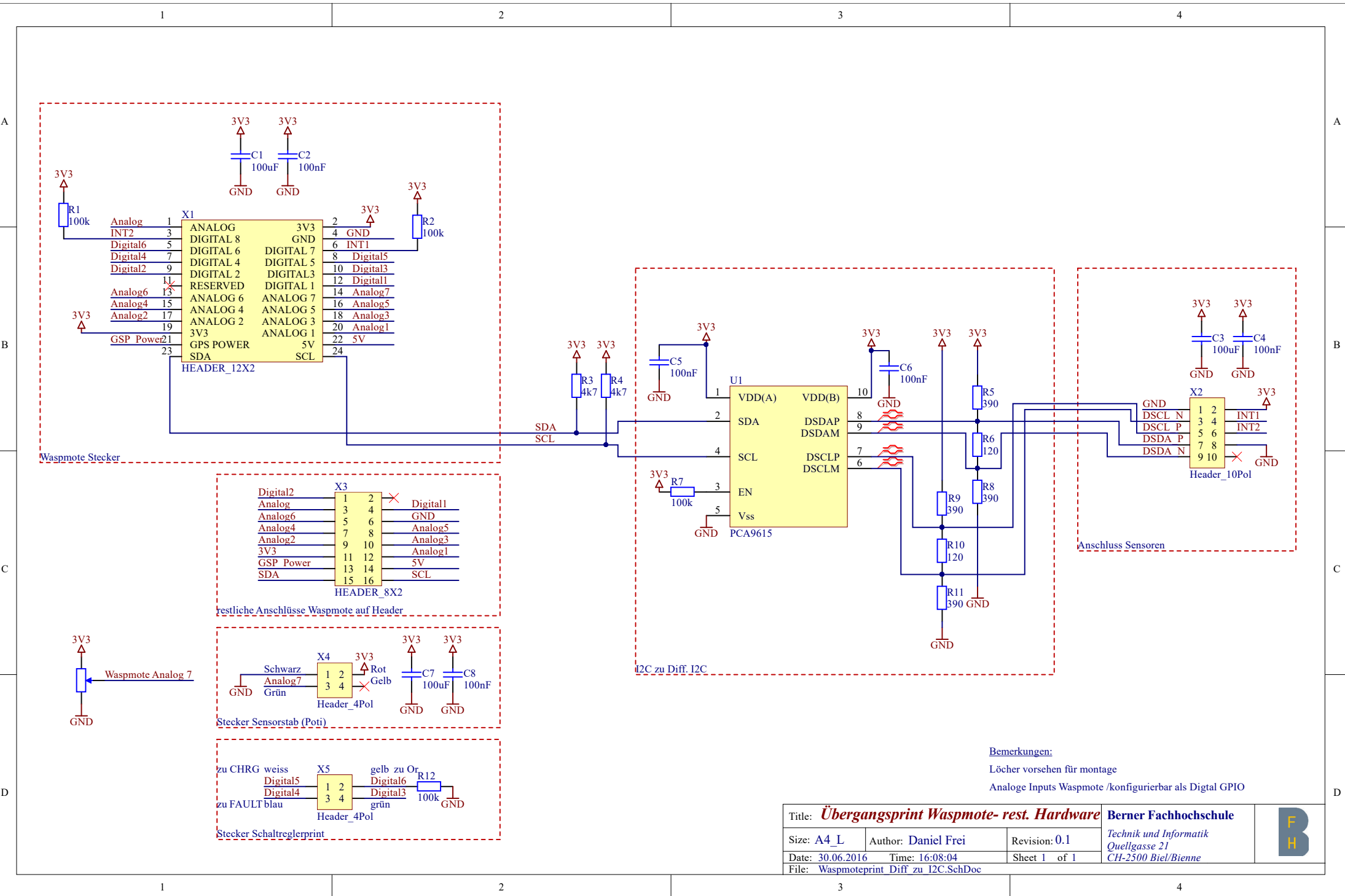



Abbildung Schema Led







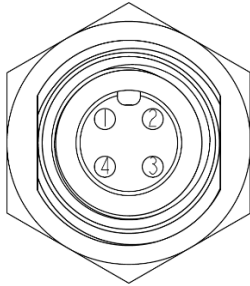
Bemerkungen:  
Löcher vorsehen für montage  
Analoge Inputs Waspote /konfigurierbar als Digital GPIO

Title: <b>Übergangsprint Waspote- rest. Hardware</b>			<b>Berner Fachhochschule</b>  <i>Technik und Informatik</i> <i>Quellgasse 21</i> <i>CH-2500 Biel/Bienne</i>	
Size: <b>A4_L</b>	Author: <b>Daniel Frei</b>	Revision: <b>0.1</b>		
Date: <b>30.06.2016</b>	Time: <b>16:08:04</b>	Sheet <b>1</b> of <b>1</b>		
File: <b>Waspoteprint Diff zu I2C.SchDoc</b>				



# Anschlüsse

## Stecker Ausdehnungssensor

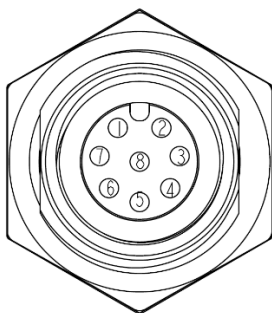


Stecker Pin	Signal	Farbe Kabel	Beschreibung
1	GND	Schwarz	Spannungsversorgung
2	3.3V	Rot	Spannungsversorgung
3	Analog7	Grün	Spannung über Ausdehnungssensor
4	-	-	

Abbildung Frontansicht Stecker Ausdehnungssensor 4 Pol

Tabelle Steckerbelegung Ausdehnungssensor 4 Pol

## Stecker Sensoren



Stecker Pin	Signal	Farbe Kabel	Beschreibung
1	GND	Blau	Spannungsversorgung
2	3.3V	Weiss-Blau	Spannungsversorgung
3	DSDA_N	Braun	Diff. Daten Minus
4	DSDA_P	Weiss-Braun	Diff. Daten Plus
5	DSCL_P	Weiss-Grün	Diff. Clock Plus
6	DSCL_N	Grün	Diff. Clock Minus
7	Schirm	Schirm	Abschirmung
8	INT1	Weiss-Orange	Interrupt Pin

Abbildung Frontansicht Stecker Sensoren 8 Pol

Tabelle Steckerbelegung Sensoren 8 Pol

## Solarstecker

Über die Solarstecker kann ein Solarpanel angeschlossen werden, das dann den Akku lädt. Das Solarpanel muss den Anforderungen in untenstehender Tabelle entsprechen, ansonsten kann das Gerät beschädigt werden.

Anforderung Solarpanel	Minimal	Maximal
Ausgangsspannung	5V	32V
Ausgangsstrom	-	3A

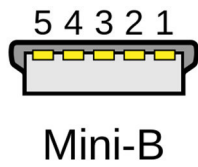
Tabelle Anforderungen Solarpanel



Abbildung Solarstecker Plus (rechts) und Minus (links)

### Mini USB-B Stecker

Mit dem USB- Stecker können das Gateway und der Sensorknoten neu programmiert werden. Der Akku wird über die USB Verbindung ebenfalls geladen, wenn das Gerät eingeschalten ist.



Stecker Pin	Signal	Farbe Kabel	Beschreibung
1	5V	Rot	Spannungsversorgung
2	D-	Weiss	Diff. Daten negativ
3	D+	Grün	Diff. Daten positiv
4	-	-	
5	GND	Schwarz	Spannungsversorgung