

BLSP1601 Honigwerk

Konzept

Projektbezeichnung	Honigwerke
Auftraggeber	Klaus Rascher (klaus.rascher@web.de)
Verantwortlich	Claudio Hediger (claudio@blselectronics.ch)
Beteiligt	Jérôme McDonald, ev weitere
Bearbeitungszustand	Review
Dateiname	BLSP1601D02_Konzept.docx
Version	1.0
Klassifikation	Intern

Änderungsverzeichnis

Nr.	Datum	Version	Betrifft	Beschreibung	Autor
01	08.10.2016	1.0	Alle	Dokument erstellt	CH, JM

Inhaltsverzeichnis

1	Einleitung	4
1.1	Vorgehen	4
2	Übersicht	5
2.1	Blockschaltbild Prototyp	6
2.2	Architektur Satellit	7
2.3	Architektur Zentrale	7
2.4	Funkverbindung zwischen Zentrale und Satellit	8
2.5	Funkverbindung zwischen Zentrale und Mobilfunknetz	9
2.6	Stromversorgung	9
2.7	Konnektivität zur Konfiguration	11
2.8	Sensoren	11
3	Hardware	12
3.1	Leiterplatten	12
3.2	Gehäuse	12
3.3	Stecker	13
3.3.1	Zentrale & Satelliten	13
3.3.2	Messwandler	13
3.4	Messeingänge	13
3.4.1	Waage	13
3.4.2	Temperatur Sensor	14
3.5	Zeitmodul	14
3.6	Mikrocontroller	15
3.7	Funkmodul	15
3.8	2G / 3G Modul	15
3.9	Stromversorgung	16
3.10	Sonstiges	16
4	Software	17
4.1	Funkprotokoll	17
4.1.1	Pairing	17
4.1.1.1	Methode A:	17
4.1.1.2	Methode B:	17
4.1.1.3	Methode C:	18
4.1.2	OSI-2 (MAC) Framestruktur	18
4.1.2.1	Frameheader	18
4.1.2.2	Nutzdaten	18
4.2	Datenpakete	19
4.2.1	OSI-3 Messprotokoll (MAC Type: 0x01)	19
4.2.2	OSI-3 Statusprotokoll (MAC Type: 0x02)	19

4.2.3	OSI-3 Zeitprotokoll (MAC-Typ: 0x03)	20
4.3	Telegramme	20
4.3.1	OSI-4 Messtelegramm	20
4.3.1.1	Messtelegramm Beispiel	21
4.4	Fehlerhandhabung	21
4.4.1	Übermittlungsfehler von der Zentrale zum Server	21
4.4.2	Übermittlungsfehler / Ausbleibende Übermittlung Satelliten	21
4.4.3	Satelliten Fehler	21
4.4.4	Unterbruch der externen Energiezufuhr	21
4.4.5	Leerer Akku	22
4.4.6	Nicht synchronisierter Start eines Satelliten	22
4.4.7	Nicht synchronisierter Start einer Zentrale	22
4.5	Ablaufdiagramm Zentrale	23
4.6	Ablaufdiagramm Satellit	24
5	Energieverbrauch	25
5.1	Satelliten	25
5.2	Zentrale	25
5.3	Solarzelle	26
6	Kosten	27
6.1	Satelliten	27
6.2	Zentrale	28
6.3	Messsystem	28
7	Anhang	29
7.1	Abbildungsverzeichnis	29

1 Einleitung

Im nachfolgenden Dokument, wird die Realisierung der Hardware sowie der Software für das Projekt „Honigwerke“ beschrieben.

Dabei wird auf die einzelnen Detailspekte soweit eingegangen, dass dem Leser ein klares Bild entsteht, wie das Problem gelöst wird

1.1 Vorgehen

Nach der Gutheissung des Konzeptes durch den Auftraggeber, wird mit der Entwicklung und Umsetzung des darin beschriebenen begonnen.

Das Konzept bildet dabei die Grundlage für alle weiteren Entwicklungsschritte und dient dabei als Leitfaden. Idealerweise bedarf es keinen Änderungen mehr am Konzept während der Entwicklung.

2 Übersicht

Beim Projekt Honigwerke handelt es sich um eine moderne IoT¹ gemässe Plattform, für die Arbeit mit Bienenwaben. Dieses Dokument widmet sich der Hardware, welche im Feld eingesetzt wird. Für die Datenbank ist der Arbeitgeber selbst zuständig.

Das zu entwickelnde System besteht aus zwei Typen Modulen, einer Zentrale und einer finiten Anzahl an Satelliten. Von den Satelliten werden Daten der Bienenwaben (Gewicht, Bruttemperatur) gesammelt und an eine Zentrale gesendet. Die Satelliten kommunizieren über eine Funkstrecke mit der Zentrale. Die Zentrale stellt zugleich eine Verbindung zum Mobilfunknetz her. Dies geschieht über eine 2G GSM Verbindung. Nach der Entwicklung der Prototypen, ist ein Einsatz von 3G bzw. UMTS angestrebt. Dies deshalb, da die 2G Infrastruktur in absehbarer Zeit nicht mehr unterstützt wird.² Der Kunde wird dann lediglich die Zentrale austauschen müssen. Eine Anpassung der Satelliten wird nicht notwendig sein.

Die Module werden über einen Akku versorgt. Dieser wird mit einem Solarpanel geladen. Um Energie zu sparen, gehen die Module in einen Sleep Zustand, wenn sie nicht gebraucht werden.

Alle Module verfügen über genügend Speicher und ein Fehler Handling, um über längere Zeit autonom funktionieren zu können. Updates können per USB und in einer fortgeschrittenen Version auch „Over-the-air“³ aufgespielt werden.

¹ https://en.wikipedia.org/wiki/Internet_of_things

² <http://www.tagesanzeiger.ch/digital/mobil/SwisscomKunden-surfen-ab-2020-mit-5G/story/31780602>

³ STM AN3262, Application note; „Using the over air bootloader with STM32W108 devices“.

2.1 Blockschaltbild Prototyp

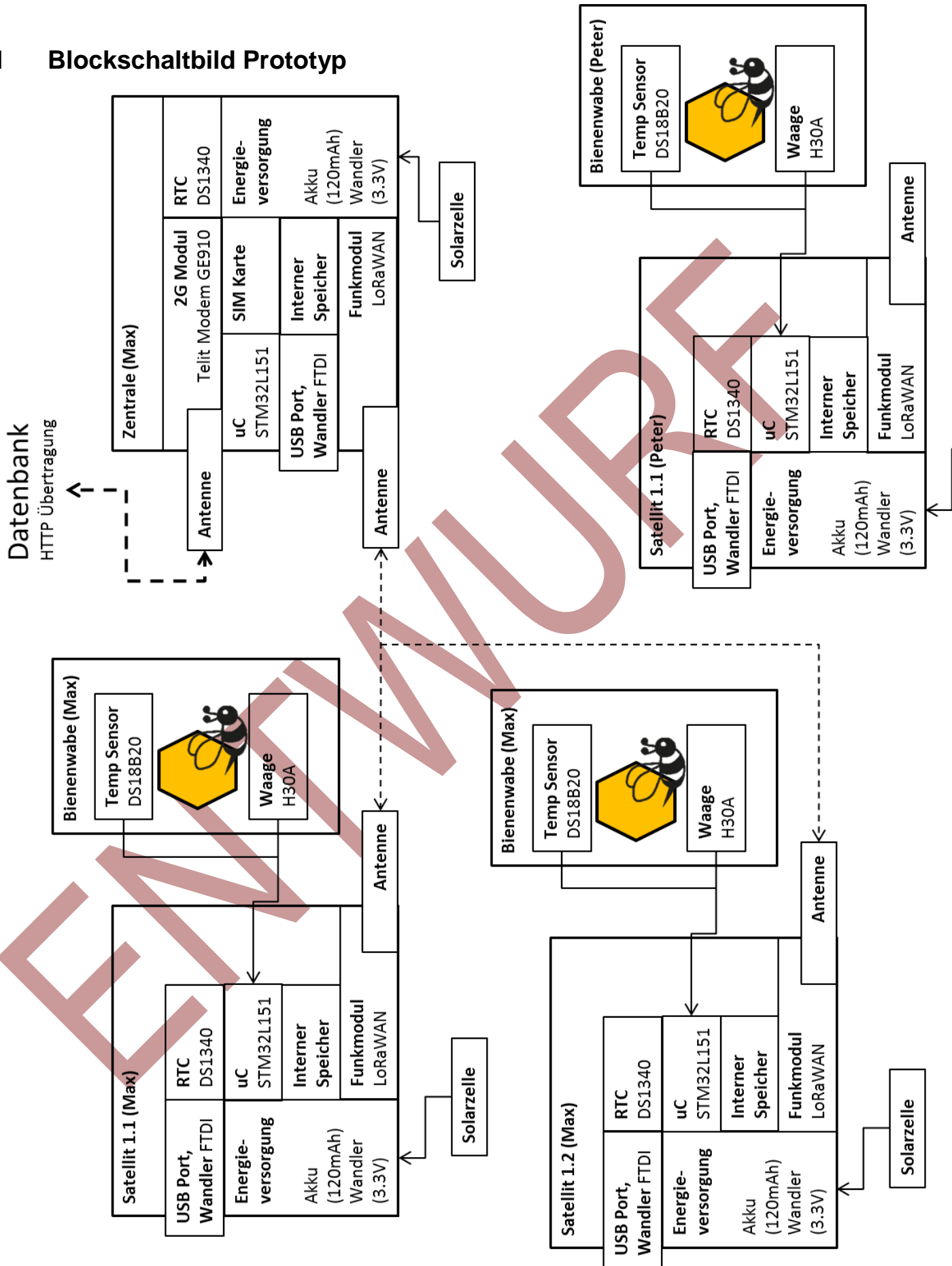


Abbildung 1: Blockschaltbild Prototyp

2.2 Architektur Satellit

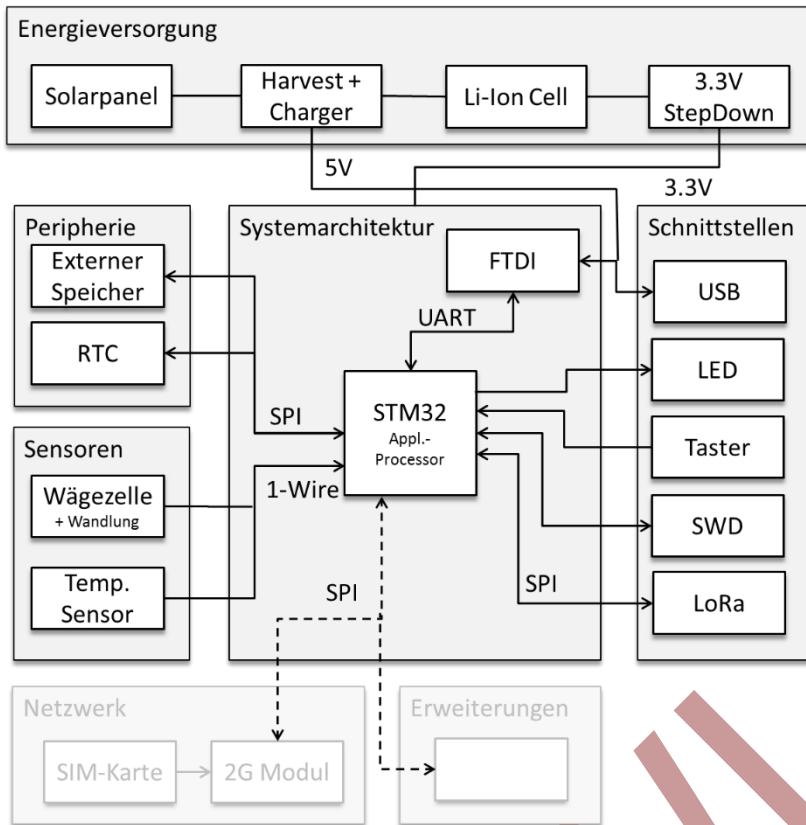


Abbildung 2: Systemarchitektur Satellit

2.3 Architektur Zentrale

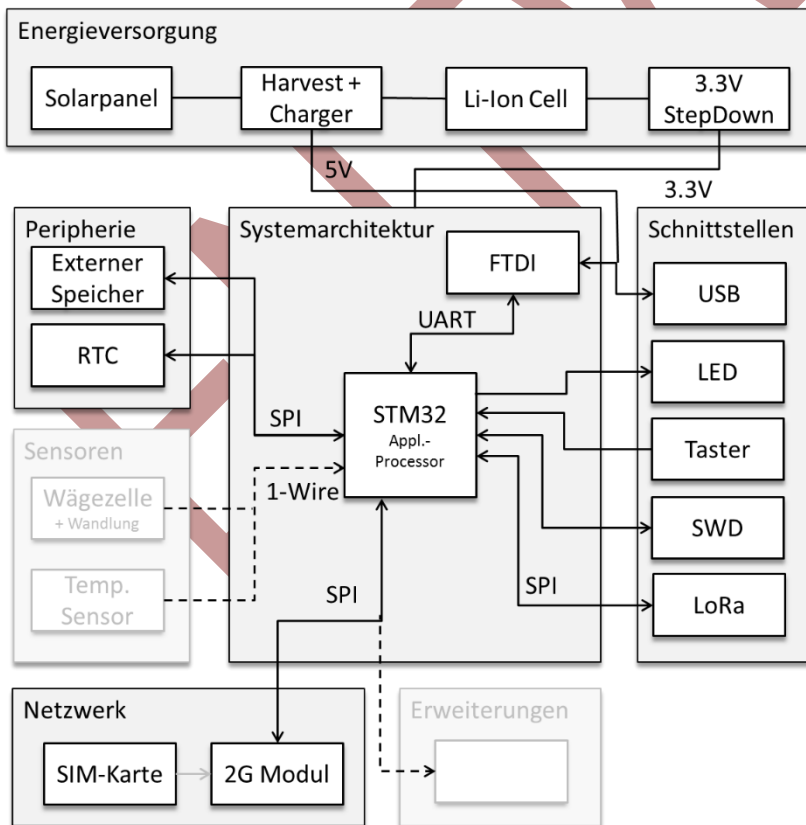


Abbildung 3: Systemarchitektur Zentrale

2.4 Funkverbindung zwischen Zentrale und Satellit

Die Funkverbindung zwischen der Zentrale und den Satelliten wird mittels Funkmodulen oder Funkchips hergestellt, welche mittels LoRa modulierte HF-Signale aussenden. Bei LoRa handelt es sich um ein Frequenzmoduliertes Zirpen (Frequency Modulated Chirp)⁴. LoRa befindet sich auf der ersten Ebene des OSI-Modells und gehört somit zum Physical layer (PHY-OSI Layer 1).

Es beschreibt keinerlei Intelligenz oder Funktionalität welche über dem Physical Layer 1 liegen.

Es handelt sich somit lediglich um ein Verfahren zur Frequenzmodulation und somit um ein Verfahren zur Übertragung von Bits.

Für die eigentliche Übertragung von Daten ist jedoch noch einiges mehr notwendig.

Dafür gibt es zum Beispiel den LoRaWAN⁵ Standard. LoRaWAN ist ein MAC (media access control) Layer Protokoll. Dieses befindet sich im MAC-OSI Layer 2.

Es wurde dazu konzipiert, um grosse offene Netzwerke zu kreieren.

Daneben gibt es noch eine alternative zu LoRaWAN, nämlich den Symphony Link. Dieser setzt ebenfalls auf den LoRa Phy. Dieses Protokoll bietet jedoch noch einige Features.

Was jedoch beide gemeinsam haben ist, dass beide MAC ein Gateway voraussetzt, bei welchem alle Daten zusammenlaufen. Es ist angedacht, dass es Landesweite LoRa bzw. SigFox, LTE-M etc. Netzwerke geben wird. Dadurch würde die Notwendigkeit eines eigenen Gateway entfallen, da die Satelliten dann direkt mit dem öffentlichen Netz kommunizieren könnten.⁶

In der Schweiz, ist die Swisscom bereits daran, ein entsprechendes Netz aufzubauen.

Diese Standards haben viele Vorteile was die Funktionalität betrifft. Jedoch ergeben sich auch gewisse Nachteile im Hinblick auf das aktuelle Projekt. So ist beispielsweise in der LoRaWAN Spezifikation im Abschnitt 7.1.2 spezifiziert, dass das Gateway mindestens auf drei verschiedenen Frequenzen zur selben Zeit auf eingehende Pakete warten muss. Dies erfordert mehrere Transceiver sowie einen Datenkonzentrator welcher die eingehenden Pakete bündelt.

Daraus resultiert ein einiges höherer Hardwareaufwand sowie ein um einiges höherer Softwareaufwand. Dasselbe gilt für den Symphony Link.

Sollte ein konformes Gateway realisiert werden, so wäre dies am schnellsten mit einem RaspberryPi gemacht. Dies hätte den Vorteil, dass damit die Netzwerkanbindung bereits erledigt ist. WLAN wäre auch sehr einfach möglich. Die Hardwarekosten wären jedoch um einiges höher. Schätzungsweise um die 300€ für ein Gateway

Eine alternative wäre ein sogenanntes Single-Channel Gateway. Dieses besteht aus nur einem Transceiver und kann dementsprechend auf nur einem Kanal zur selben Zeit Daten empfangen. Dadurch wird das Gateway allerdings bereits nicht mehr LoRaWAN konform.

Es stellt sich nun die Frage, in wie weit sich ein konformes Gateway lohnt.

⁴ http://perso.ens-lyon.fr/patrick.flandrin/SPIE01_PF.pdf

⁵ <https://www.lora-alliance.org/What-Is-LoRa/Technology>

⁶ https://de.wikipedia.org/wiki/Long_Range_Wide_Area_Network

Ein guter Kompromiss wäre vermutlich der folgende:

Sensoren kommunizieren mit der Zentrale mittels der LoRa Modulation allerdings über einen proprietären MAC-OSI Layer 2. Dies ermöglicht eine sehr schlanke Topologie des Systems. Sinnvolle und nützliche Aspekte von LoRaWAN würden übernommen. Beispielsweise das ALOHA verfahren. Sowie die duty-cycle Regelungen um ETSI konform zu bleiben. Was jedoch entfernt würde wäre die Vorgabe auf mehreren Kanälen zu horchen.

Dadurch geht zuerst einmal auch die ADR (Adaptive Data Rate) verloren. Da diese ebenfalls in LoRaWAN integriert war.

Dies bedeutet, dass die Satelliten wie auch die Zentrale auf fix definierten Frequenzen und Datenraten senden müssen.

Bei LoRaWAN wurde dies spezifisch ausgehandelt. Eventuell könnte man eine eigene Ergänzung in den eigenen MAC-Layer implementieren.

Die Satelliten sollten über das Proprietäre Format wie auch über den LoRaWAN Standard verfügen bzw. diesen sprechen können. Dadurch könnten die Sensoren auch mit bestehenden Netzen und konformen Gateways eingesetzt werden.

2.5 Funkverbindung zwischen Zentrale und Mobilfunknetz

Für die Verbindung ins Mobilfunknetz werden Telit Modems eingesetzt.

Bisher wird ein xE910 angestrebt. Dieses gibt es als 2G, 3G und 4G Variante. Wobei der Preis massiv variiert. Für die 2G Variante liegt dieser bei um die 30€ für 3G um 70€ und 4G um 150€

Die Daten werden nach der Aufbereitung durch die Zentrale direkt an den spezifizierten Server weitergeleitet.

2.6 Stromversorgung

Die Energieversorgung soll durch Anschluss eines Solarpanels gewährleistet werden.

Intern befindet sich bei den Satelliten wie auch bei der Zentrale ein Akku.

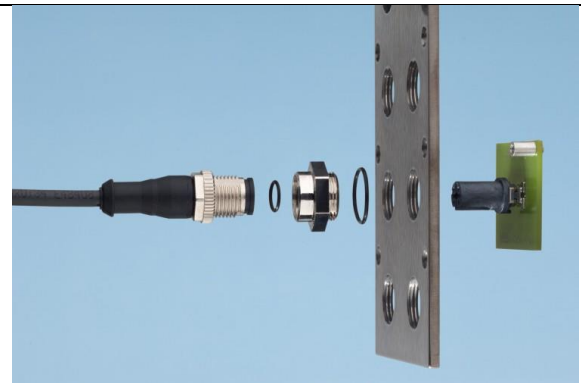
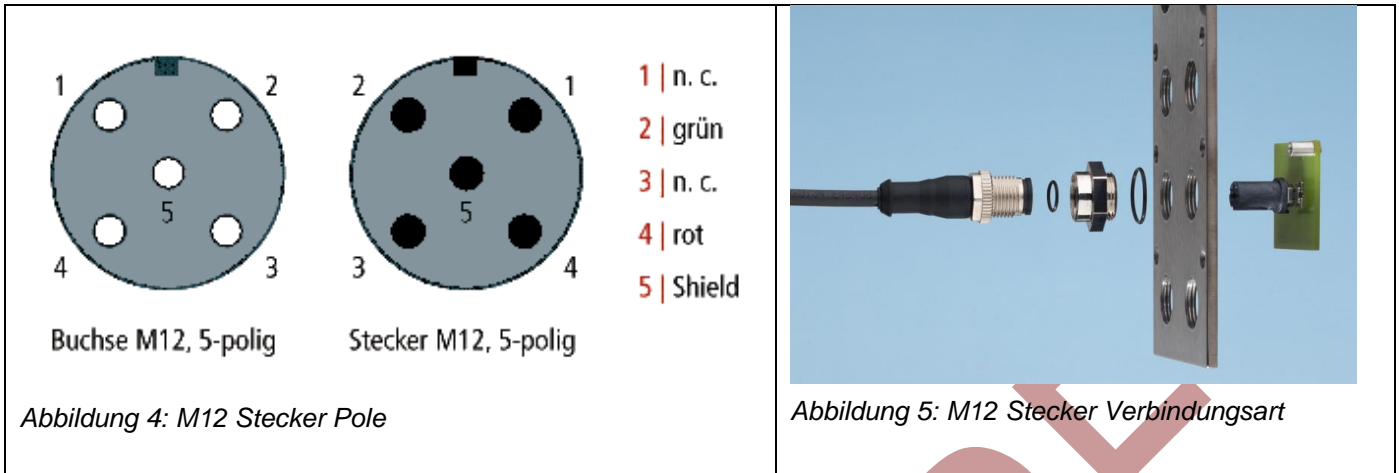
Dieser wird während den Sonnenstunden durch die integrierte Ladeelektronik geladen.

Während der Nacht, wird die Energie des Akkus verwendet um die Elektronik zu versorgen.

Damit möglichst viele verschiedene Solarpanels unterstützt werden können, wird ein möglichst grosser Eingangsspannungsbereich vorgesehen. Dieser kann im Bereich von 5..40V liegen. Damit lassen sich sehr viele Solarpanels anschliessen. Sollte es in einem späteren Verlauf bzw. nach den Prototypen zu einer Veränderung der Anforderungen kommen, so lässt sich die Spannungsversorgung relativ leicht anpassen.

Die Module (Zentrale wie auch Satelliten) verfügen wahlweise über einen oder zwei Anschlüsse zur externen Spannungsversorgung.

Als Anschluss dient ein M12 Stecker



Eine Schleifung bzw. Serienschaltung der Satelliten ist problemlos möglich. Dies kann zum einen durch mehrfachen Anschluss eines Steckers an das gleiche Kabel erfolgen, oder aber durch zwei Buchsen am Gehäuse

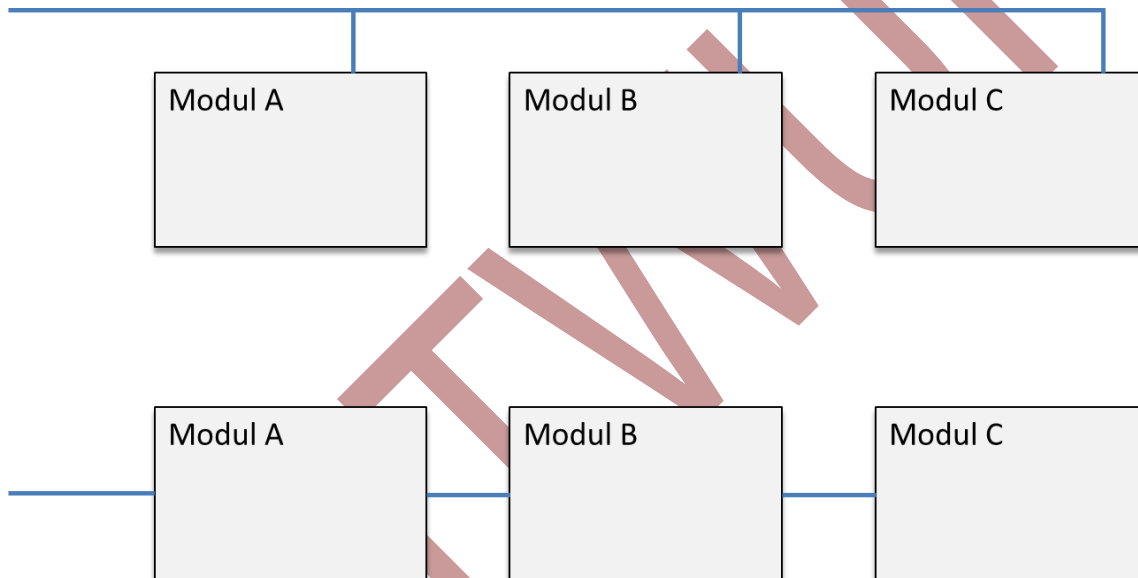


Abbildung 6: Schematische Darstellung für Modul Schleifung

2.7 Konnektivität zur Konfiguration

Um mit den Modulen kommunizieren zu können, und eventuell Einstellungen vorzunehmen, Softwareupdates durchzuführen oder den aktuellen Status abzufragen, befindet sich bei den Modulen eine USB Schnittstelle.

⁷

Diese Schnittstelle ist extern zugänglich über eine entsprechend robuste USB-Schnittstelle.



Abbildung 7: Ruggedized Waterproof Mini-USB connector

Es gäbe auch die Möglichkeit, einen USB-Dongle zu entwickeln. Mit diesem wäre es möglich, über LoRa mit den Modulen zu kommunizieren und diese zu Parametrieren. Dadurch könnte man ohne grossen Aufwand, viele Module innert kurzer Zeit konfigurieren und deren Status prüfen sowie Updates durchführen.

Da der Datendurchsatz von USB massiv höher ist als jener von der LoRa Funkstrecke, würde sich ein solcher Dongle eher für kurze Sessions mit geringem Datenvolumen eignen.

Bei diesem Dongle handelt es sich um einen Stick, welcher ein LoRa Modul beinhaltet. Die eigentliche Kommunikation findet dann wieder mittels Funk statt. Es ist dadurch nicht notwendig, die Module Physisch zu erreichen.

2.8 Sensoren

An das Sensormodul lassen sich nebst Wägezellen auch noch mehrere Temperatursensoren des Typs DS18B20 anschliessen.

Die Wägezellen verfügen über eine eigene winzige Elektronik, von der die Messwerte erfasst und gewandelt werden. Daran wird auch der Temperatursensor angeschlossen. Von der Messelektronik werden die Daten per 1-Wire an die Sensor-Module gesendet.

⁷ <http://www.usbfirewire.com/parts/rr-214300-10.html>

3 Hardware

3.1 Leiterplatten

Spezifikationen:

4 layer PCB

PCB Thickness 1.6mm

Copper Weight 1oz 35um

PCB Size ca. 15cm * 15cm

PCB Color Green

Surface Finish Hasl (lead free)

Beschreibung:

Ziel ist es, Satellit und Zentralen PCBs mit nur einem Layout zu realisieren. Die PCBs würden sich nicht unterscheiden. Die Unterschiede würden mit verschiedenen Bestückungsvarianten realisiert werden. Dadurch ergeben sich viele Vorteile. Kosten und Aufwand können reduziert werden. Ob dies im Layout sinnvoll ist, ergibt sich jedoch erst definitiv durch das Layouten.

Für die Messwandler wird ebenfalls ein kleines PCB erstellt. Dieses wird die Dimensionen von ca. 5cm*5cm haben.

Das Layout wird mit Altium erstellt. Die PCBs lassen wir bei unserem chinesischen Kontakt herstellen.

3.2 Gehäuse

Das Verwendete Gehäuse wird ähnlich wie auf dem unteren Bild aussehen.



Abbildung 8: Mögliches Gehäuse

Spezifikationen

- 140x140x46,5mm
- verdeckte Schrauben und Steckeranschlüsse

BLSP1601D02

- hochwertiges V0-Material und optionale Schutzart IP 65 (Dichtungs-Set als Zubehör) für Endgeräte im Außenbereich
- dichter Elektronikraum mit zwei Platinen-Ebenen
- optionale Abdeckung schließt das Gehäuse nach unten und ermöglicht, eine weitere Platine zu montieren
- einfache, bündige Wandmontage an drei Punkten
- rostfreie Edelstahlschrauben mit sicherem Torx-Antrieb
- NET-BOX auch als Tischgehäuse einsetzbar (Anti-Rutsch-Füßchen als Zubehör)
- dreiteiliges Gehäuse, bestehend aus Unterteil, Oberteil und Blende

Beschreibung

Das elegante, flexible Gehäuse kann an einer Wand oder auch liegend montiert werden. Durch die IP65 konforme Bauart, löst sich damit auch das Problem der Stecker. Da diese unter der Haube montiert werden können, müssen sie nicht speziell robust oder IP65 konform sein.

3.3 Stecker

3.3.1 Zentrale & Satelliten

- 1 x Mini-USB
- 1 x M12-4 für Speisung
- 1 x SMA für Antenne
- 1 x M12-5 für Datenbus

Beschreibung

Die USB Buchse ist dazu da, Daten Manuell abrufen zu können und wenn nötig ein Softwareupdate aufzuspielen.

Die M12 Buchsen sind sehr stabile, Industriestandard Buchsen. In unserem Fall werden sie für die Speisung und den Datenbus verwendet.

Der SMA Anschluss kann wie bei einem Standard W-LAN Router sehr einfach angeschraubt werden.

3.3.2 Messwandler

- 2 x M12-5 für Datenbus
- 2 x 3.5mm Klinkenbuchse für Temperatursensoren

Beschreibung

Die Messwandler haben 2 M12 Anschlüsse für den Datenbus. Zwei weil mehrere Messwandler seriell geschaltet werden können. Die Messwandler werden auch über die M12 Buchse gespeist.

Die Temperatursensoren können an einer Standard Klinkenbuchse wie sie in jedem mp3 Player etc. eingesetzt werden angeschlossen werden. Dies macht es sehr einfach für den Nutzer die Sensoren anzuschliessen. Die Waage kann mit Klemmen angeschlossen werden.

3.4 Messeingänge

3.4.1 Waage

- Hardware:** Bosche Plattform Wägezelle H30A⁸
Schnittstelle: 1 Wire mit externer Übersetzung

⁸ <https://www.bosche.eu/waagenkomponenten/waegezellen/plattform-waegezellen/plattform-waegezellen-h30a>

Beschreibung:

Die Sensoren bzw. die Satelliten verfügen über eine finite Anzahl an Eingängen an welche sich eine Wägezelle anschliessen lässt. Eine Wägezelle besteht aus einem auf einem Metallblock aufgeklebten Dehnmesstreifen. Diese Wägezelle liefert ein Spannungswert analog der Dehnung des Streifens. Diese Spannung ist abhängig vom aufgelegten Gewicht durch welches die Dehnung entsteht.

In diesem Projekt wird der Fokus auf die Wägezelle Bosche H30 gelegt. Diese Wägezelle benötigt eine empfohlene Speisespannung von 5-12V. Da die Elektronik intern mit einer Spannung von 3.3V Arbeitet, bedarf es einer internen Spannungswandlung.

Eine Wägezelle hat einen typischen Widerstand von etwa 350 Ohm +- 100 Ohm. Die Elektronik wird für die Verwendung mit solchen Wägezellen vorgesehen. Es wird zudem eine durch einen Jumper änderbare Konfiguration vorgesehen, welche es ermöglicht, eine andere Spannung für die Zelle zu verwenden. Dadurch ergibt sich die Möglichkeit, auch noch eine alternative Wägezelle zu verwenden, welche stark von den 350 Ohm abweicht.

Die Messelektronik wird an der Waage selbst angebracht und ist nicht im Sensor enthalten. Die Hardware rechnet die Messdaten um und Wandelt sie zu einem Digitalen Signal, welches über eine 1 Wire Verbindung übertragen wird.

3.4.2 Temperatur Sensor

Hardware: Maxim DS18B20 Programmable Resolution 1-Wire Digital Thermometer⁹
Schnittstelle: 1-Wire¹⁰

Beschreibung:

Neben den Eingängen für die Wägezelle, befindet sich noch ein Anschluss für Temperatursensoren des Typs DS18B20. Diese Sensoren werden über 1Wire Interface angesprochen und liefern direkt die gemessenen Temperaturwerte zurück.

1Wire ist eine Busarchitektur und ermöglicht das anschliessen mehrere Teilnehmer am selben Bus. Wird der Bus ohne separate Speisungsleitung benützt, so kann bis maximal etwa 100°C gemessen werden. Denn ab da reicht die parasitäre Speisung nicht mehr aus.

In diesem Projekt wird jedoch die zusätzliche Speisung vorgesehen, da dies einen robusteren Betrieb gewährleistet.

Es können bis zu 20 Sensoren angeschlossen werden.

3.5 Zeitmodul

Hardware: DS1340
Schnittstelle: I²C oder SPI

Beschreibung:

⁹ <https://datasheets.maximintegrated.com/en/ds/DS18B20.pdf>

¹⁰ <https://de.wikipedia.org/wiki/1-Wire>

Damit alle Module (Zentrale wie auch die Satelliten) über eine aktuelle Zeit sowie ein Datum verfügen, befindet sich ein sogenanntes RTC Modul DS1340¹¹ auf jedem Board. Dieses wird jeweils mit der Zentrale synchronisiert und ist auf wenige Sekunden genau mit der Uhr in der Zentrale synchronisiert. Die Zentrale selbst, synchronisiert sich über das Mobilfunknetz mit einem Zeitserver.

3.6 Mikrocontroller

Hardware: STM32F0xx
Technologie: Programmierung in C, FreeRTOS

Beschreibung:

Als Mikrocontroller für die Module wird ein STM32F1xx eingesetzt. Diese Controller sind äusserst leistungsstark und zugleich preisgünstig. Zudem benötigen sie sehr wenig Energie.

Sie bieten zudem schier unendliche Möglichkeiten in Bezug auf interne Peripherie, Konfiguration etc.

3.7 Funkmodul

Hardware: RN2483
Technologie: LoRaWAN

Beschreibung:

Für die Funkverbindung werden die RN2483 Module von Microchip eingesetzt. Diese Module enthalten bereits den LoRaWAN Stack. Zudem ermöglichen diese auch die Kommunikation ohne MAC. Dies ist genau das was benötigt wird. Es ist zudem ein Firmwareupdate der Module möglich. Danke der Kommunikation mittels UART, gestaltet sich die Kommunikation sehr einfach und transparent. Beste Voraussetzungen für ein schnelles TimeToMarket! Das Modul ist zudem R&TTE Approved und hat somit die notwendigen Zertifikate für Europa
Kostenpunkt: um 17 EUR pro Stück.

3.8 2G / 3G Modul

Hardware: UEE910EUD Modem
Technologie: 2G GSM (Erweiterbar auf 3G)

Beschreibung:

Für die Kommunikation der Zentrale mit dem Mobilfunknetz wird ein Telit UE910EUD¹² oder ein GE910EUD eingesetzt. Diese Modems sind für 2G bzw. 3G Netze geeignet.

¹¹ <https://www.maximintegrated.com/en/datasheet/index.mvp/id/3914>

¹² <http://www.telit.com/iot-modules/>

3.9 Stromversorgung

Hardware: Solarpanel, Harvest + Charger, Akku, 3.3V Wandlung

Beschreibung:

Die Module werden über einen Akku gespeist, dieser wird über ein Solarpanel geladen. Dafür wird die Energie gesammelt und über eine Ladeelektronik in den Akku geladen. Da die Akkuspannung ca. 3.8V beträgt, muss sie auf 3.3V gewandelt werden.

Der Akku wird so eingebaut, dass er leicht ausgetauscht werden kann. Dazu wird ein System entwickelt, mit welchem er im Inneren des Deckels, vom Gehäuse, befestigt werden kann.

3.10 Sonstiges

Beim Prototypen werden einige Sicherheitsvorkehrungen getroffen, dh. Es werden teils zusätzliche Verbindungsmöglichkeiten und externe Bauteile vorgesehen, die unter normalen Umständen nicht verwendet werden. Sollte es jedoch zu unerwarteten Resultaten oder Fehlern kommen, kann dadurch besser geprüft werden was falsch ist und das Problem behoben werden.

Dies beinhaltet unter anderem: Mehrere Bestückungsvarianten von pullup/pulldown Widerständen am uC. Schnittstellen alternativen. Überbrückung der Speisung...

4 Software

Die Firmware der Zentrale wie auch der Satelliten wird in C geschrieben.

Dabei kommt ein Betriebssystem zum Einsatz. Wir werden FreeRTOS verwenden, da dies auch für kommerzielle Anwendungen gebührenfrei verwendbar ist.

4.1 Funkprotokoll

Wie bereits weiter oben erwähnt, können wir aufgrund von verschiedenen Limitierungen, kein echtes LoRaWAN in der Zentrale einsetzen.

Stattdessen wird für den OSI Layer 2 (MAC) ein eigenes Protokoll entwickelt. Dieses soll sehr schlank aufgebaut sein und leicht Erweiterungen erlauben.

Ein wichtiger Aspekt hierbei soll die Datenintegrität sowie die Verschlüsselung sein. Wir setzen hierbei auf AES128 Bit Verschlüsselung der Daten.

4.1.1 Pairing

Damit beide Seiten die Daten bearbeiten können, muss beiden ein identischer Schlüssel bekannt sein. Die Bekanntgabe des Schlüssels kann auf verschiedene Arten erfolgen.

4.1.1.1 Methode A:

Durch Drücken der WPP (Wireless Pairing Process) Taste am Satelliten, wechselt dieser für etwa 1-Minute in den dauerhaften Empfangsmodus. Wird nun während dieser Zeit die WPP Taste bei der Zentrale gedrückt, so sendet diese ein einmaliges Paket an alle Satelliten, welches den zufällig generierten Schlüssel enthält. Nach Empfang des Pakets durch die Satelliten, wird der Modus beendet und der Schlüssel wird intern abgespeichert. Zusätzlich versucht das Modul mit dem nun bekannten Schlüssel, ein Testpaket an die Zentrale zu senden. Wenn dies erfolgreich war, wechselt der Satelliten wieder in den regulären Modus. Ansonsten wird die fehlerhafte Konfiguration durch einen Error signalisiert. Zukünftige Übertragungen werden bei erfolgreichem Pairing ab dann mithilfe des neuen Schlüssels durchgeführt. Satelliten

Diese Methode ist die einfachste. Nachteil ist jedoch, dass die Sicherheit bei einer Man in the middle Attacke nicht gewährleistet ist, da der Schlüssel für einen Moment im Klartext übermittelt wird.

Eine Verbesserung dieses Ablaufs wäre, den Schlüssel mit einem hart codierten Schlüssel zuvor doppelt zu verschlüsseln. Solange der intern verwendete Schlüssel nicht bekannt ist, besteht eine höhere Sicherheit. Dies ist jedoch keine Lösung auf Dauer, weshalb davon abgesehen wird.

4.1.1.2 Methode B:

Die Module werden durch USB Anschluss konfiguriert. Der Benutzer vergibt manuell einen Schlüssel und trägt diesen bei jedem Modul manuell ein. Dadurch ist eine MITM Attacke ausgeschlossen.

Diese Methode ist die Zeitintensivste jedoch auch die sicherste.

BLSP1601D02

4.1.1.3 Methode C:

Gleich wie Methode A jedoch wird die Sendeleistung der Zentrale reduziert. Dadurch sinkt die Reichweite für potenzielle Angreifer.

4.1.2 OSI-2 (MAC) Framestruktur

Ein Frame des LSL (LoRa Simple Link) besteht aus den folgenden Daten:

4.1.2.1 Frameheader

Der Frameheader ist bei allen Frames identisch.

Er setzt sich aus insgesamt 4 Bytes zusammen, wobei noch ein Byte für die CRC-Summe hinzukommt.

1	2	3	4	5
AddrTo	AddrFrom	Length	Type	CRC8

1 | uint8_t | 0x00..0xFF | AddrTo

Empfänger des Pakets, 0xFF gilt als Broadcast und erreicht alle empfangsbereiten Teilnehmer

2 | uint8_t | 0x00..0xFF | AddrFrom

Absender des Pakets.

3 | uint8_t | 0x00..0xFF | Length

Die Länge des Pakets, inkl. CRC8 Byte.

4 | uint8_t | 0x00..0xFF | Type

Pakettyp

5 | uint8_t | 0x00..0xFF | CRC8

Checksumme in Form einer CRC8. Wird am Ende jedes Frames angehängt.

Die CRC8 berechnet sich über das gesamte Frame inkl. Payload.

Mit Ausnahme des CRC Bytes selbst natürlich.

4.1.2.2 Nutzdaten

Die eigentlichen Nutzdaten bzw. die sogenannte „Payload“ befindet sich immer zwischen Byte 4 und 5.

4byte	1..250b	1byte
Header	Payload	CRC8

Die maximale Grösse für die Nutzdaten liegt bei 250 Bytes. Dies ergibt sich daraus, dass es nur ein Byte für die Längenangabe gibt. Somit kann ein Paket maximal 255 Byte = 0xFF lang sein. Da 5 Bytes für den Header inkl. CRC benötigt werden, ergibt sich daraus eine maximale Grösse von 250 Bytes. Wie diese Nutzdaten auf dem Device interpretiert werden, hängt vom Paket Typ und der Firmware des Devices ab.

4.2 Datenpakete

Über dem OSI-2 liegen die Datenpakete. Je nach Protokoll kann das Paket verschiedenartige Daten enthalten.

4.2.1 OSI-3 Messprotokoll (MAC Type: 0x01)

Das Paket setzt sich aus mehreren Bytes zusammen

1	2	3	4	5..xx
Length	Count	Offset A	Offset B	Telegram

1 | uint8_t | 0x00..0xFF | Length
Gibt die Länge des Pakets an

2 | uint8_t | 0x00..0xFF | Count
Gibt die Anzahl der Messtelegamme an

3 | uint8_t | 0x00..0xFF | Offset A
Gibt den Startoffset für das erste Telegramm bekannt beginnend bei Byte 1

4 | uint8_t | 0x00..0xFF | Offset B
Analog zu Offset A

Ab Byte 5 folgen die Messtelegamme. Diese sind weiter unten beschrieben.
Das Paket kann beliebig viele Messtelegamme enthalten. Vorausgesetzt, die Payload des MAC Frames wird nicht überschritten.

4.2.2 OSI-3 Statusprotokoll (MAC Type: 0x02)

Das Paket setzt sich aus mehreren Bytes zusammen
Er setzt sich aus insgesamt 4 Bytes zusammen.

1	2	3	4
Length	Err	Res	Res

1 | uint8_t | 0x00..0xFF | Length
Gibt die Länge des Pakets an

2 | uint8_t | 0x00..0xFF | Err
Enthält Angaben zu aufgetretenen Fehlern im Satelliten

3 | uint8_t | 0x00..0xFF | Res
Reserviert für zukünftige Implementierungen

4 | uint8_t | 0x00..0xFF | Res
Reserviert für zukünftige Implementierungen

4.2.3 OSI-3 Zeitprotokoll (MAC-Typ: 0x03)

Dieses Paket dient dazu, Zeiten zu synchronisieren.

1	2	3	4	5	6
Day	Month	Year	Hour	Minute	Delaytime

1 | uint8_t | 0x00..0xFF | Day
Wochentag

2 | uint8_t | 0x00..0xFF | Month
Monat

3 | uint8_t | 0x00..0xFF | Year
Jahr

4 | uint8_t | 0x00..0xFF | Hour
Stunde

5 | uint8_t | 0x00..0xFF | Minute
Minute

6 | uint8_t | 0x00..0xFF | Delaytime
Delaytime in 5 Sekunden schritten. Solange wird das Senden verzögert.

4.3 Telegramme

Die Telegramme setzen auf der OSI-4 Schicht an und beinhalten die eigentlichen Nutzdaten

4.3.1 OSI-4 Messtelegramm

Das Messtelegramm beinhaltet die eigentlichen Messwerte der Sensoren.

Das Telegramm gibt Auskunft über die ID, den Typ und den Status des Satelliten sowie den Wert.

Zu jedem Messtelegramm gehört ein Messpaket (Type 0x01)

1	2	3	4	5
Satelliten ID	Type	H Byte	L Byte	Status

1 | uint8_t | 0x00..0xFF | Satelliten ID
Beinhaltet die Satelliten ID des Moduls. Meist 0 für Kanal A, 1 für Kanal B etc.

2 | uint8_t | 0x00..0xFF | Type
Typ des Satelliten, 0x01 = DS18B20, 0x02 = Wägezelle etc.

3 | uint8_t | 0x00..0xFF | H Byte
Höherwertiges Byte des Rohwertes

4 | uint8_t | 0x00..0xFF | L Byte
Niederwertiges Byte des Rohwertes

5 | uint8_t | 0x00..0xFF | Status
Beinhaltet einen Status und mögliche Fehler.

BLSP1601D02

4.3.1.1 Messtelegramm Beispiel

Die untenstehende Tabelle zeigt eine typische Übertragung zweier Messwerte über das Netzwerk.

OSI-4 Messtelegramm					Telegramm A	Telegramm B	
OSI-3 Datenpaket				Header			
OSI-2 MAC LSL			Header				CRC8
OSI-1 LoRA	Präambel	Start					
			5 Byte	4 Byte	5 Byte	5 Byte	1 Byte

Die gesamte Übertragung besteht aus 20 Bytes für zwei Messtelegramme.
Bei einer Bitrate von 300 Bits/s benötigt der Satelliten dazu 0.54 Sekunden.

4.4 Fehlerhandhabung

Ein Ereignis welches in diese Kategorie fällt ist beispielsweise das abrupte abfallen der Versorgungsspannung oder andere seltene Ereignisse. Die Zentrale erstellt, beim Erkennen eines Fehlers, einen Eintrag in einem Logfile. Das Logfile wird mit den anderen Daten dem Server übermittelt.

Nachfolgend werden für verschiedene Szenarien das vorgesehene Verhalten der Elektronik sowie der Software beschrieben.

4.4.1 Übermittlungsfehler von der Zentrale zum Server

Die Zentrale bekommt den Tag hindurch immer wieder Messwerte der einzelnen Satelliten. Diese werden intern gesammelt, bis diese am Abend oder einem anderen definierten Zeitpunkt mittels Mobilfunk zum Server übermittelt werden. Sollte diese Kommunikation aufgrund unvorhergesehener Ereignisse nicht funktionieren, so werden die Daten weiterhin im internen Speicher der Zentrale behalten. Es wird dann in definierten Abständen versucht, eine neue Verbindung aufzubauen. Fehlversuche werden geloggt. Sobald dies erfolgreich war, werden die Daten, beginnend mit den ältesten, zum Server übermittelt.

4.4.2 Übermittlungsfehler / Ausbleibende Übermittlung Satelliten

Übermittlungsfehler der Satelliten werden von der Zentrale geloggt. Sollten häufige Fehler auftreten oder ein Satellit komplett aufhören zu senden, wird dem Server eine Warnung übermittelt.

4.4.3 Satelliten Fehler

Bemerkt der Satellit, dass etwas nicht stimmt. Z.B. Unrealistische/keine Resultate bei der Messung. Übermittelt er eine Fehlermeldung an die Zentrale.

4.4.4 Unterbruch der externen Energiezufuhr

Durch den integrierten Akku, stellt dieses Ereignis kein Problem dar.

4.4.5 Leerer Akku

Sollte es einmal vorkommen, dass der integrierte Akku nicht mehr genügend Energie hat, um das System zu versorgen, so stellt dies kein Problem dar.

Es sollten keine Daten verloren gehen, da ein empfangenes Paket von einem Satellit zuerst intern abgespeichert wird und erst dann die Bestätigung an den Satellit versendet wird.

Dadurch würde der Satellit im Fehlerfall keine Bestätigung erhalten und das Paket erneut senden bis es erfolgreich zugestellt wurde.

4.4.6 Nicht synchronisierter Start eines Satelliten

Ist ein Satellit längere Zeit nicht mit Energie versorgt gewesen, so wird dieser nach dem Aufstarten solange im Empfangsmodus verbleiben, bis dieser ein Zeitsynchronisierungspaket von der Zentrale erhalten hat.

4.4.7 Nicht synchronisierter Start einer Zentrale

Hat eine Zentrale ihre interne Zeit verloren, so ruft sie diese beim Zeitserver ab.

Wenn als Übertragungsmodus UltraLowPowerWirelessTechnology gewählt wurde, so müssen die Satelliten sich zuerst wieder mit der Zentrale synchronisieren, damit diese weiss wann sie Daten zu erwarten hat. Im regulären (dauernd aktiven Betrieb) ist dies nicht notwendig.

4.5 Ablaufdiagramm Zentrale

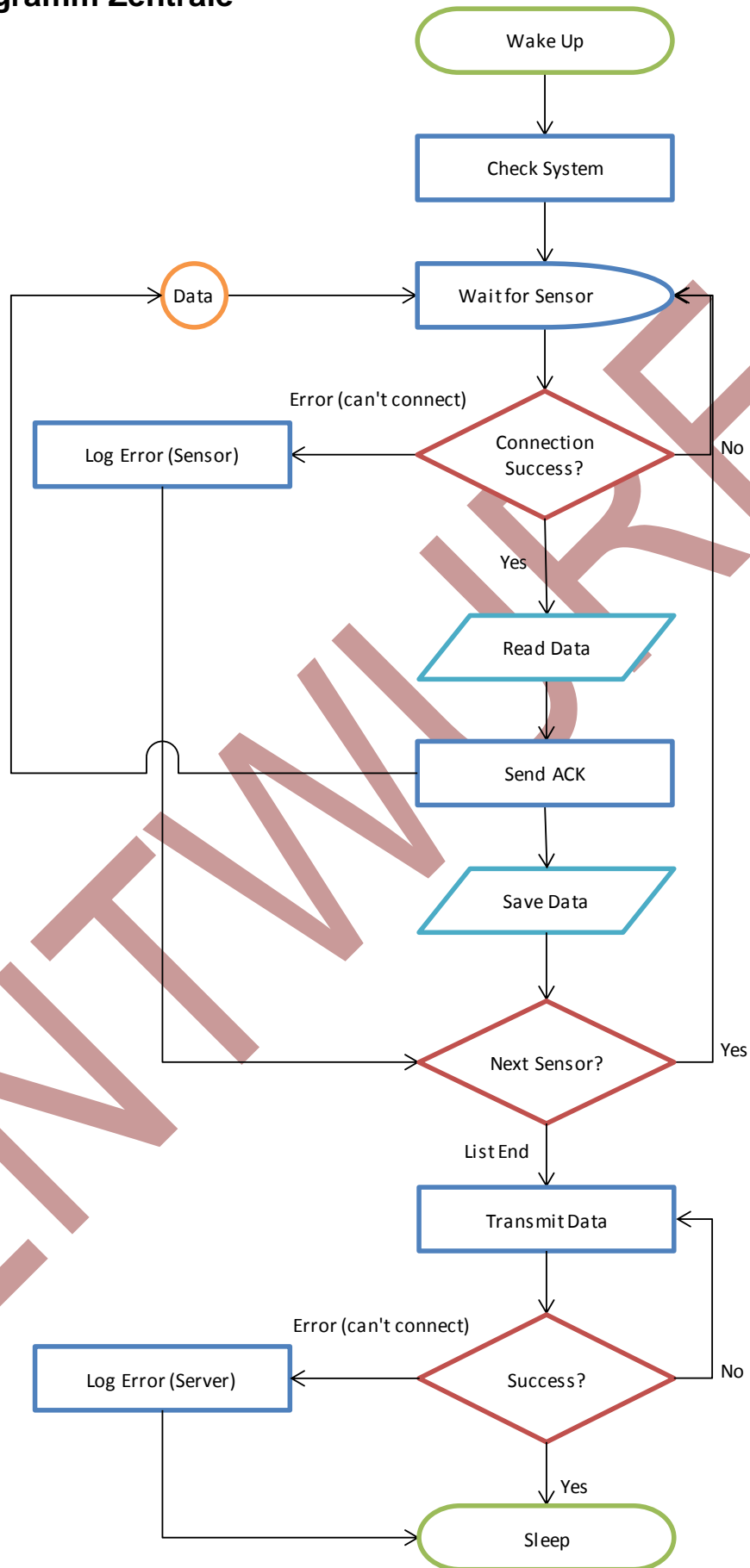


Abbildung 9: FlowChart Zentrale

4.6 Ablaufdiagramm Satellit

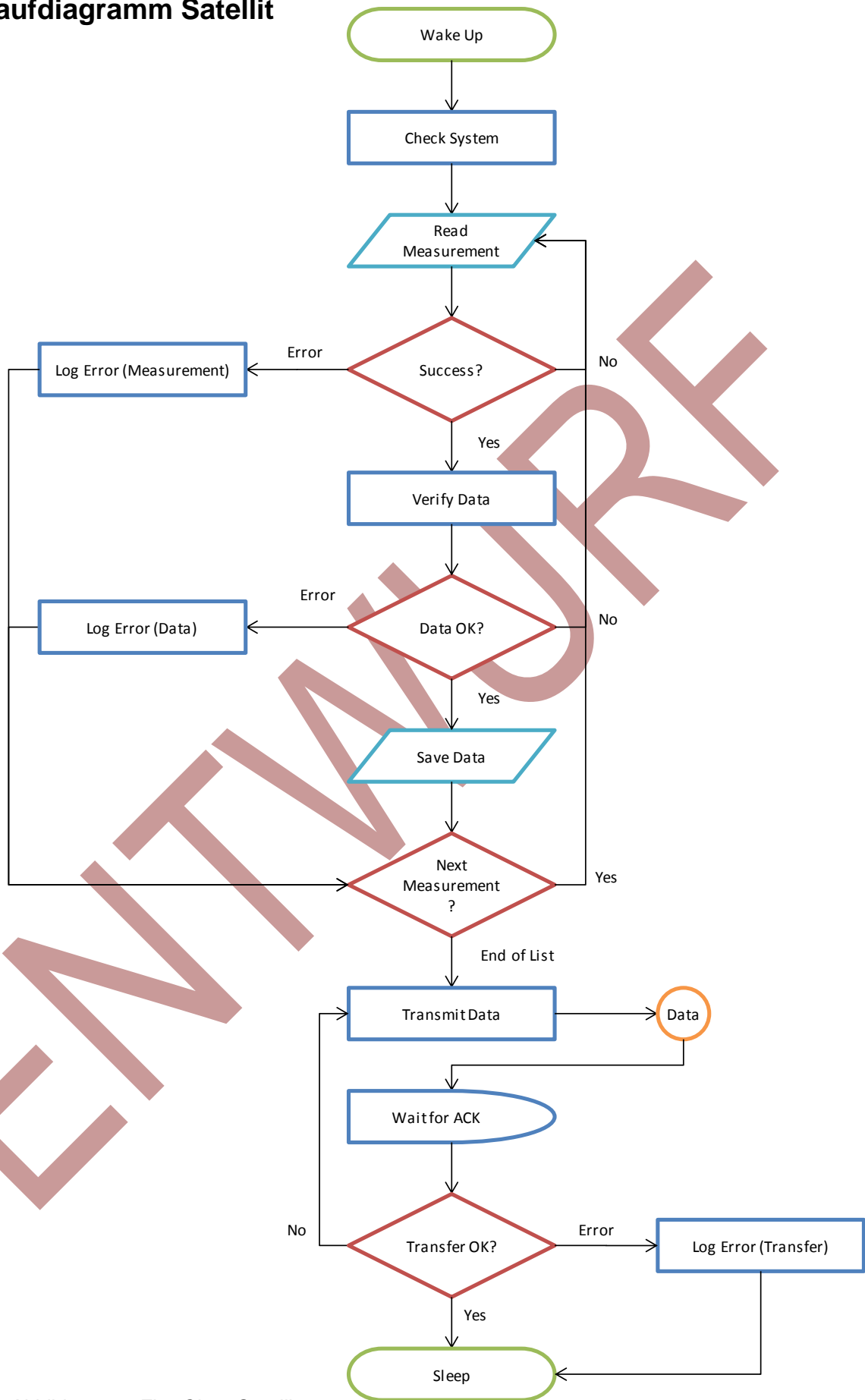


Abbildung 10: FlowChart Satellit

5 Energieverbrauch

5.1 Satelliten

Die Satelliten senden in regelmässigen Abständen Daten zur Zentrale. Dabei ist es wichtig, dass diese einen geringen Stromverbrauch aufweisen, um möglichst lange ohne externe Versorgung durchhalten zu können.

Dazu nun die folgenden Berechnungen:

Ein Satellit sendet zwei Messwerte mit einer Datenrate von 300Bits/S und einer Sendeleistung von 14dBm an die Zentrale. Die Übertragung für die 20 Datenbytes dauert in etwa 0.54 Sekunden.

Betrachten wir zuerst den Energieverbrauch des Funkmoduls:

Bei einem geschätzten Energieverbrauch von 40mAh für das Funkmodul für dauersenden, würde diese Übertragung etwa 20 μ Wh Energie verbrauchen, bei einer Versorgungsspannung von 3.3V. Ein sehr sehr kleiner Akku mit 120mAh beinhaltet bei einer angenommenen Spannung von 3.8V etwa 0.5 Wh Energie. Somit könnten etwa 23'000 solcher Pakete versendet werden bevor der Akku leer ist.

Da jedoch nicht nur das Funkmodul sondern auch der Mikrocontroller und die gesamte Peripherie mit Energie versorgt werden muss, betrachten wir nun auch noch den Controller.

Ein STM32L151 benötigt im Betrieb etwa 5.2 mA.

Nehmen wir für das versendet des Pakets eine Zeit von 3 Sekunden an. Dann benötigt der Controller eine Energie von 14 μ Wh.

Nehmen wir für die restliche Peripherie einen grosszügigen Strom von 20mA an und eine Zeit von 5 Sekunden. Dies ergibt eine Energie von 91 μ Wh.

Dies ergibt im Total eine Energie von $20 \mu\text{Wh} + 14 \mu\text{Wh} + 91 \mu\text{Wh} = 125 \mu\text{Wh}$.

Bei einem Akku mit 0.5Wh ergibt dies eine mögliche Anzahl von 4000 Übertragungen.

Rechnen wir noch mit einem Sicherheitsfaktor von 20% so ergibt dies 3200 Übertragungen.

Wenn nun pro Tag 100 Übertragungen durchgeführt werden, so reicht der Akku des Satelliten für rund 32 Tage ohne auch nur einen einzigen Tag mit Sonnenenergie versorgt worden zu sein.

Es muss berücksichtigt werden, dass mehrere Waagen ebenfalls gespeist werden müssen. Da dies auch Energie braucht, wird ein grösserer Akku eingesetzt. Voraussichtlich 850mAh. Der Preisunterschied ist bei dieser Akkuklasse klein.

5.2 Zentrale

Damit die Zentrale nicht dauern Empfangsbereit sein muss, werden mit den Satelliten definierte Zeitfenster ausgehandelt, in welchem sie Daten senden können.

Dieses Zeitfenster ist dann während einer bestimmten Zeit offen.

Wenn das Funkmodul dauernd aktiv ist und bereit ist, Daten zu empfangen, so benötigt es rund 15mA. Dies ergibt eine Energie von 50mWh. Rechnet man noch den Controller und die Peripherie mit ein so ist man bei etwa 60mWh ohne GSM. Wenn nun die aktive Zeit dank dem Zeitfenster auf 5 Minuten pro

Stunde begrenzt wird, so sinkt er Energieverbrauch auf 5mWh. Damit genügt ein sehr kleiner Akku um gut 100h zu überbrücken.

Die Zentrale weiss grundsätzlich, wie viele Satelliten vorhanden sind.

Damit es während dem Empfangszeitfenster nicht zu einer Kollision der Daten kommt, hat jeder Satellit eine Delaytime. Diese wird dem Satelliten mittels des Zeitprotokolls mitgeteilt.

Während der Konfiguration der Zentrale, kann der Benutzer wählen, in welchem Modus er die Zentrale betreiben möchte. Ob diese ständig Empfangsbereit sein soll, oder ob diese Zeitgesteuert Empfangsslots öffnen soll.

Wenn die Zentrale den Satelliten ein Zeitfenster alle halbe Stunde mitgeteilt hat, so wartet jeder Satellit zuvor noch seine Delaytime ab, bevor er mit dem Senden beginnt.



Der nebenstehende Akku hat eine Kapazität von 850mAh. Dies entspricht in etwa 2700mWh. Damit liesse sich die Zentrale in ständiger Empfangsbereitschaft ca 40h betreiben. Dieser Akku ist zudem sehr klein.

Er misst gerade etwa 80x40x7mm

Wenn man mit diesem Akku das erweiterte Energiemanagement mit der Zeitgesteuerten Kommunikation verwendet, so beläuft sich die überbrückbare Zeit auf Gut 540h oder 22 Tage.

Je nach Solarzelle kann solch ein Akku bereits innert einer Stunde vollständig geladen werden.

Abbildung 11: Akku

5.3 Solarzelle

Die Satelliten benötigen nach der obigen Abschätzung in etwa 125 μ Wh



Abbildung 12: Solarpanel

Nebenstehend ist eine wirklich kleine Solarzelle mit den Abmassen von etwa 70x60mm. Ein solches Modell liefert in etwa 80mA und eine Spannung von 5V. Die wären 400mWh. Da dies jedoch nur unter optimalen Bedingungen möglich ist, nehmen wir für die nachfolgenden Berechnungen einen Ertrag von gerade mal 5% an. Dies wären dann 20mWh

Der Satellit benötigt in einem Tag $24h \times 125\mu Wh \approx 3mWh$ Energie. Nehmen wir an, dass am Aufstellungsort durchschnittlich 4h Sonne zu erwarten ist.

Dann produziert das abgebildete Modul pro Tag etwa $4h \times 20mWh \approx$

80mWh Energie.

Davon wurden 3mWh für die aktive Kommunikation benötigt. Somit verbleiben rund 75mWh welche in den Akku fließen. Dieser hat eine Kapazität von gut 500mWh. Somit ist dieser nach einem Tag bereits zu 15% geladen. Nach gut 7 Tagen ist dieser voll. Mit den erreichten 15% Ladung könnten bereits wieder mindestens 15 Tage überbrückt werden, bei stündlichen Übertragungen.

Man muss beachten, dass dieses Modul extrem klein ist und daher nicht unbedingt die geeignetste Wahl für dieses Vorhaben darstellt. Zudem haben wir mit einem Ertrag von nur 5% gerechnet.

Aber selbst unter diesen sehr schlechten Bedingungen, würde das System problemlos überleben.

6 Kosten

In diesem Kapitel sollen die Kosten grob abgeschätzt werden.

Dies stellt keine abschliessende Berechnung dar und dient lediglich als extrem grober Richtwert.

Die effektiven Kosten können erst in einem späteren Verlauf des Projekts exakter berechnet werden.

6.1 Satelliten

Die Aufschlüsselung findet nach den verschiedenen Bereichen statt.

Bezeichnung	Bemerkung	Geschätzte Kosten in CHF 1 Stück
Gehäuse von OKW	Unbearbeitet	Fr. 54.00
LoRa Modul	Microchip	Fr. 33.75
Akku		Fr. 6.75
Antenne		Fr. 4.05
SMA Anschluss		Fr. 2.70
Leiterplatte	4-Lagig, leer unbestückt	Fr. 10.13
USB Buchse		Fr. 2.70
Controller		Fr. 2.70
Speisung		Fr. 13.50
Dichtung		Fr. 13.50
Restliche Bauteile	Widerstände etc.	Fr. 27.00
M12	Anschlüsse: Power, Bus	Fr. 20.25
Bestückung	Inhouse	Fr. 82.20
Total:		Fr. 273.23

Die Schätzung basiert auf kleinststückzahlen.

Stecker sind oftmals einer der grössten Kostentreiber. Insbesondere wenn diese, wie hier, besonders Robust sein müssen oder Wasserdicht. Dasselbe gilt für Gehäuse.

Unter dem Punkt „restliche Bauteil“ sind jene Bauteile enthalten, welche nur in Summe Kosten verursachen. Die exakten Kosten können erst während dem Design berechnet werden.

Tendenziell könnten diese aber eher sinken als steigen.

Bei einer Stückzahl von 100 Stück, kann davon ausgegangen werden, dass der Preis pro Gerät sich um ca. einen Drittel reduziert.

6.2 Zentrale

Bezeichnung	Bemerkung	Geschätzte Kosten in CHF 1 Stück
Gehäuse von OKW	Unbearbeitet	Fr. 54.00
LoRa Modul	Microchip	Fr. 33.75
Akku		Fr. 6.75
Antenne		Fr. 4.05
SMA Anschluss		Fr. 2.70
Leiterplatte	4-Lagig, leer unbestückt	Fr. 10.13
USB Buchse		Fr. 2.70
Controller		Fr. 2.70
Speisung		Fr. 13.50
Dichtung		Fr. 13.50
Restliche Bauteile	Widerstände etc.	Fr. 27.00
3G Modem	Telit	Fr. 108.00
Bestückung	Inhouse	Fr. 82.20

Total:

Fr. 360.98

6.3 Messsystem

Bezeichnung	Bemerkung	Geschätzte Kosten in CHF 1 Stück
Gehäuse von OKW	Unbearbeitet	Fr. 13.50
3.5mm	Klinkenbuchse	Fr. 5.40
Leiterplatte	4-Lagig, leer unbestückt	Fr. 6.75
Controller		Fr. 2.70
Speisung		Fr. 6.75
ADC		Fr. 4.05
Restliche Bauteile		Fr. 13.50
Bestückung gem. Berechnung	Inhouse	Fr. 50.00
M12 2x		Fr. 33.75

Total:

Fr. 136.40

7 Anhang

7.1 Abbildungsverzeichnis

Abbildung 1: Blockschaltbild Prototyp	6
Abbildung 2: Systemarchitektur Satellit.....	7
Abbildung 3: Systemarchitektur Zentrale	7
Abbildung 4: M12 Stecker Pole	10
Abbildung 5: M12 Stecker Verbindungsart.....	10
Abbildung 6: Schematische Darstellung für Modul Schleifung	10
Abbildung 7: Ruggedized Waterproof Mini-USB connector.....	11
Abbildung 8: Mögliches Gehäuse	12
Abbildung 9: FlowChart Zentrale	23
Abbildung 10: FlowChart Satellit.....	24
Abbildung 11: Akku	26
Abbildung 12: Solarpanel	26