



School of Engineering

InES Institute of
Embedded Systems

Bachelorarbeit

Messstation zur Registrierung von Geschiebe- Bewegungen im Fluss

Autoren	Tobias Keller Tobias Welti
Betreuer	Prof. Hans-Joachim Gelke, Dipl. El. Ing. FH ZHAW Institute for Embedded Systems
Partner	Carlos Rodrigo Wyss Eidg. Forschungsanstalt für Wald, Schnee und Landschaft WSL
Datum	16. Dezember 2014

S

Erklärung betreffend das selbständige Verfassen einer Bachelorarbeit an der School of Engineering

Mit der Abgabe dieser Bachelorarbeit versichert der/die Studierende, dass er/sie die Arbeit selbständig und ohne fremde Hilfe verfasst hat. (Bei Gruppenarbeiten gelten die Leistungen der übrigen Gruppenmitglieder nicht als fremde Hilfe.)

Der/die unterzeichnende Studierende erklärt, dass alle zitierten Quellen (auch Internetseiten) im Text oder Anhang korrekt nachgewiesen sind, d.h. dass die Bachelorarbeit keine Plagiate enthält, also keine Teile, die teilweise oder vollständig aus einem fremden Text oder einer fremden Arbeit unter Vorgabe der eigenen Urheberschaft bzw. ohne Quellenangabe übernommen worden sind.

Bei Verfehlungen aller Art treten die Paragraphen 39 und 40 (Unredlichkeit und Verfahren bei Unredlichkeit) der ZHAW Prüfungsordnung sowie die Bestimmungen der Disziplinarmassnahmen der Hochschulordnung in Kraft.

Ort, Datum:

Unterschriften:

.....

.....

.....

.....

Das Original dieses Formulars ist bei der ZHAW-Version aller abgegebenen Bachelorarbeiten zu Beginn der Dokumentation nach dem Titelblatt mit Original-Unterschriften und -Datum (keine Kopie) einzufügen.

Vorwort

Hier ein bisschen
von dir und mir

Kontakte

Verfasser

Tobias Keller
Schulstrasse 82
CH-8952 Schlieren

E-Mail: kellet01@students.zhaw.ch

Tobias Welti, MSc ETH Zürich
Wissenschaftlicher Assistent ZHAW Institute for Embedded Systems
Technikumstrasse 9
CH-8401 Winterthur

E-Mail: tobias.welti@zhaw.ch
Homepage: <http://www.ines.zhaw.ch>

Betreuer

Prof. Hans-Joachim Gelke, Dipl. El. Ing. FH
ZHAW Institute for Embedded Systems
Technikumstrasse 9
CH-8401 Winterthur

E-Mail: hans.gelke@zhaw.ch
Homepage: <http://www.ines.zhaw.ch>

Partner

Carlos Rodrigo Wyss
Eidg. Forschungsanstalt WSL
Zürcherstrasse 111
8903 Birmensdorf

E-Mail: carlos.wyss@wsl.ch
Homepage: <http://www.wsl.ch>

Inhaltsverzeichnis

1. Einleitung	7
1.1. Ausgangslage	7
1.1.1. Hydrologische Messungen	7
1.2. Projektidee	7
1.3. Ziel	8
2. Aufgabenstellung	9
2.1. Aufgabenstellung	9
2.1.1. Musskriterien	9
2.1.2. Wunschkkriterien	10
2.1.3. Abgrenzungskriterien	10
3. Vorgehen	11
3.1. Überblick	11
3.2. Datenlogger	11
3.2.1. Messdaten sammeln	12
3.2.2. Kontrolle über das Bussystem	12
3.2.3. Steuerung des Betriebs	12
3.2.4. Schnittstelle nach Aussen	12
3.3. Sensoreinheit	12
3.3.1. Messdatenerfassung	12
3.3.2. Ereigniserkennung	13
3.3.3. Datenübertragung	13
3.4. Bussystem	13
4. Funktionale Anforderungen	14
4.1. Datenlogger (F1...)	14
4.1.1. F110 Busmaster	14
4.1.2. F120 Sensorerkennung	14
4.1.3. F130 Uhrzeit	14
4.1.4. F140 Timestamp verteilen	14
4.1.5. F160 Schnittstelle zum Steuerrechner	14
4.1.6. F170 Steuerung Betriebsmodus	14
4.1.7. F180 Daten sammeln	15
4.1.8. F190 Daten speichern	15
4.2. Sensoreinheit (F4...)	15
4.2.1. F410 Ereignisdetektion	15
4.2.2. F430 Datenübertragung	15
4.2.3. F450 Rohdatenaufzeichnung	15
5. Nichtfunktionale Anforderungen	16
6. Hardware-Konzept	17
6.1. Hardware-Architektur	17
6.1.1. Datenlogger	17
6.1.2. Sensoreinheit	17
6.1.3. Bussystem	17

6.2. Komponentenauswahl	17
6.2.1. Mikroprozessor	17
6.2.2. Bus-System	20
6.2.3. Speichermedium	24
6.2.4. Sensor	25
6.2.5. Schnittstelle	25
6.2.6. Gehäuse	25
6.3. Datenlogger	25
6.4. Sensoreinheit	26
7. Software-Konzept	28
7.1. Software-Stack	28
7.1.1. Überblick	28
7.1.2. Messdatenerfassung	28
7.1.3. Ereigniserkennung	30
7.1.4. Detail-Level	33
7.1.5. Timestamp	34
7.2. Busverwaltung	35
7.2.1. Verwaltung des Bussystems	35
7.2.2. Busprotokoll	35
7.2.3. Filesystem	35
7.3. Konfiguration	35
8. Resultate	37
9. Diskussion	38
10. Bedienungsanleitung	39
10.1. Produktbeschrieb	39
10.2. Aufbau der Messstation	39
10.3. Datenlogger	39
10.3.1. Anschlüsse	39
10.4. Sensor	40
10.4.1. Beschleunigungssensor	40
10.4.2. Mikroprozessor	40
10.4.3. Anschlüsse	41
10.5. Bussystem	41
10.5.1. Kabel	41
10.5.2. Abschlusswiderstände	41
10.5.3. Steckverbinder	42
10.6. Ereignis	42
10.6.1. Detail-Level	42
10.7. Konfiguration	47
10.7.1. Anschluss eines Computers	47
10.7.2. Menü	47
10.7.3. Befehle	49
10.7.4. Konfigurationsdatei	56
10.8. Betrieb	56
10.9. Technische Daten	56
11. Verzeichnisse	57
Literaturverzeichnis	58
Abbildungsverzeichnis	59
Tabellenverzeichnis	60

Glossar	61
Abkürzungen	66
A. Anhang	I
A.1. Offizielle Aufgabenstellung	I
A.2. Projektmanagement	III
A.3. Weiteres	III
A.4. Schaltpläne	III
A.5. Datenblätter	VII
A.5.1. NXP LPC4088 32-bit ARM Cortex-M4 microcontroller	VII
A.5.2. Embedded Artists NXP LPC4088 QuickStart Board	IX
A.5.3. Analog Devices ADXL001 Beschleunigungssensor	XI
A.6. Source Code, Daten und Multimedia	XIII

Liste der noch zu erledigenden Punkte

Hier ein bisschen blabla von dir und mir	3
Hardware	17
Sensorauswahl beschreiben	25
software	28
Ereignis-FSM anpassen auf neue Stati	31
Zusammenhänge A/D-Wandlung und Ereigniserkennung und Übertragung beschreiben, dazu ein Kommunikationsdiagramm, wo synchron, wo asynchron.	33
TW: Berechnungen, in welchem Modus wie lange gemessen werden kann, und wie lange ein Sensor mit dem vorhandenen Speicher die Resultate zwischenspeichern kann. Allenfalls ein System erwähnen, das automatisch zwischen verschiedenen Modi hin- und herschalten kann. (Ist allerdings heikel). Wie viele Sensoren können in welchem Modus gleichzeitig am System betrieben werden, bei welcher Ereignisrate ist Schluss mit Busbandbreite.	34
TK: Busverwaltung beschreiben	35
TK: Figur 7.10 neu machen	35
TK: Busprotokoll	35
TK: Kommunikationsdiagramm Bushandler	35
TK: Interrupt-System des Bushandlers aufführen	35
Resultate	37
läuft das zeug?, wie viele sensoren mags leiden?, wie schnell kann fs sein?, wie lange kann man aufzeichnen?, was geht, was nicht?, was haben wir nicht erreicht?, wo haben wir mehr als gefordert erreicht?	37
Diskussion	38
haben wir erfüllt?, wo gabs schwierigkeiten?, worauf sind wir stolz , was könnte man jetzt weiter noch machen? , was ist noch geplant?	38
Bedienungsanleitung	39
bildchen des Aufbaus der Messstation	39
figur über den logger	39
Stromverbrauch logger und pro sensor nennen	40
beispieldaten einfügen	44
beispieldaten einfügen	45
beispieldaten einfügen	45
beispieldaten einfügen	47
logger status listing	50
sensorauswahl listet die sensoren auf, das noch einfügen	51
Länge des Ereignispuffers	53

Liste von Sensor-Stati ins Listing einfügen	54
Beispiel einer Uhrzeit einfügen	54
mehrere konfig-files und file löschen könnte man in den Ausblick nehmen, als zu verbessernde punkte	55
Anhang	I
Zeitplan	
Besprechungsprotokolle oder Journals	III
dieser Abschnitt gehört in den Haupttext.	III
Spezifikationen u. Datenblätter der verwendeten Messgeräte und/oder Komponenten	
Berechnungen, Messwerte, Simulationsresultate	
Grafische Darstellungen, Fotos	XIII
Inhaltsverzeichnis der CD erstellen	
CD mit dem vollständigen Bericht als pdf-File inklusive Film- und Fotomaterial	XIII

1. Einleitung

1.1. Ausgangslage

Die Eidg. Forschungsanstalt für Wald, Schnee und Landschaft (WSL) betreibt Messstationen zur Registrierung von Geschiebe-Bewegungen im Fluss mittels Geophonen, die unter Stahlplatten montiert sind. Diese Platten sind in einer Betonkonstruktion eingelassen, um sie im Flussbett zu fixieren. Die Betonkonstruktion dient gleichzeitig als Kabelkanal. Jedes Geophone ist über ein Kabel mit einem Auswertungs-Rechner (Embedded PC) verbunden. Der Rechner wertet die Signale aller angeschlossenen Geophone kontinuierlich aus, um die Ereignisse zu detektieren. Bei mehreren Geophonen ist hier ein recht leistungsfähiger Rechner nötig, der eine entsprechend hohe Leistungsaufnahme hat. Die baulichen Massnahmen für die Installation der Geophone, der Auswertungsstation sowie der Stromversorgung sind sehr teuer. Da viele dieser Messstationen in Gebirgsbächen installiert werden, fallen hohe Transportkosten für die schweren Materialien an.

1.1.1. Hydrologische Messungen

Diese Messstationen benutzt das WSL, um Daten über Geschiebetransport zu sammeln. Solche Datensammlungen spielen eine zentrale Rolle im Flussbau und der Hydrologie. Die Vorhersage von Menge und Zeitpunkt des Auftretens von Geschiebetransport ist weiterhin schwierig, weshalb intensiv auf diesem Gebiet geforscht wird. Forschungsgruppen aus der ganzen Welt haben unterschiedlichste Sensoren und Messinstallationen entwickelt auf der Suche nach einem Instrument, mehr und bessere Daten sammeln zu können. Dabei kommen Piezoelektrische Sensoren und Geophone zum Einsatz [?], aber auch Beschleunigungssensoren [?]. Hydrophone können ebenfalls eingesetzt werden, um die Geräusche der Geschiebekörper unter Wasser aufzuzeichnen. Dies erfordert aber eine viel höhere Abtastrate und daher mehr Rechenleistung [?]. Um die Daten in Korrelation zu Menge, Korngröße und Fliessgeschwindigkeit setzen zu können, gibt es Messanlagen mit weiteren Sensoren und Fangkörben (z.B. im Erlenbach im Alptal, Kanton Schwyz), die bei hohem Geschiebeaufkommen automatisch ins Flussbett gefahren werden, um das Geschiebe aufzufangen [?]. Auch Fangnetze werden zu diesem Zweck installiert [?].

Die hydrologische Forschung verfolgt unter Anderem das Ziel, bessere Vorhersagen zu Naturereignissen machen zu können. Stauungen in einem Bach durch Geschiebeablagerungen können zu drastischen Verschlimmerungen führen, z.B. Ausuferungen bei einem Hochwasser [?].

1.2. Projektidee

Bruno Fritschi (WSL) ist bestrebt, die momentan sehr aufwändigen baulichen Massnahmen zu vereinfachen und den Sensoraufbau so zu modifizieren, dass die Messstation insgesamt einfacher aufgebaut werden kann.

Die wichtigste Idee sieht vor, die gemessenen Daten direkt am Sensor auszuwerten, und nur jene Daten zu übertragen und zu speichern, an denen die Forschung interessiert ist. Die Reduktion der Datenmenge an der Quelle ermöglicht die Verwendung eines Bussystems, also eines einzigen Kabels für alle Sensoren, statt eines eigenen Kabels für jeden Sensor. Da die Daten über das Bussystem aus dem Bach heraus auf einen Datensammler (Datenlogger) übertragen werden können, würde dies weitere Möglichkeiten eröffnen:

- Stromversorgung der Sensoren über das Kabel des Bussystems. Dies würde die Laufzeit einer solchen Anlage beträchtlich erhöhen, resp. Batterien in den Sensoren unnötig machen.
- Datensammlung an einem zugänglichen Gerät, statt Einsammeln der Sensoren aus dem Bach.
- Überwachung der Anlage zu Laufzeit, Kontrolle über den Zustand der Datensammlung und Änderung der Konfiguration der Sensoren im Betrieb.

Zukünftig sollen die Geophone durch ein- oder mehr-achsige MEMS-Beschleunigungssensoren ersetzt werden, da diese wesentlich kleiner sind. Die Verkleinerung der Sensoren ermöglicht neue Bauformen der Messanlagen:

Die Integration der Sensoren in einer Elastomerplatte statt der bisherigen Montage unter Stahlplatten würde die baulichen Massnahmen und damit die Kosten drastisch senken. Die Verkabelung eines Bussystems ist weit weniger Komplex als für ein sternförmiges System wie bisher. Dies würde die Variante mit Sensoren in Elastomerplatten begünstigen.

1.3. Ziel

In diesem Projekt soll daher ein Messsystem entwickelt werden, das MEMS-Beschleunigungs-Sensoren in einem Bussystem einsetzt und die Signale direkt beim Sensor auswertet. Die Entwicklung zur Serienreife würde den Rahmen einer Bachelorarbeit für zwei Personen sprengen, weshalb nur ein Prototyp entwickelt werden soll. Anhand des Prototyps soll an der WSL untersucht werden können, ob sich die Weiterentwicklung lohnt.

Bei einem positiven Entscheid könnte in einer weiteren Bachelorarbeit die Weiterentwicklung zu einem Serienprodukt versucht werden.

2. Aufgabenstellung

Die offizielle Aufgabenstellung befindet sich im Anhang A.1.

2.1. Aufgabenstellung

Im Rahmen dieser Bachelorarbeit soll eine Lösung erarbeitet werden, um zukünftige Installationen günstiger zu machen. Da solche Messanlagen an sehr vielen Orten auf der ganzen Welt aufgebaut werden, kann durch eine Vereinfachung der Installation viel Aufwand gespart werden.

Die Projektidee stammt von Bruno Fritschi (WSL). Sein Vorschlag sieht vor, die aufgezeichneten Signale direkt am Sensor auszuwerten und nur die gewünschten Ereignis-Daten zu übertragen und zu speichern. Somit könnten die Daten über ein Bussystem übertragen werden und der Rechner für die Sammlung der Daten bräuchte weniger Rechenleistung.

Dank der Bustopologie kommt das Messsystem mit weniger Leitungen aus und kann einfacher installiert werden. Denkbar wäre die Integration in einer Elastomerplatte anstelle der Stahl- und Betonkonstruktion, da viel weniger Leitungen nötig sind. Die Elastomerplatte könnte einfacher im Bachbett verankert werden.

Ziel der Arbeit ist die Entwicklung der Auswertungshardware und des Bussystems. Die Qualität der gemessenen Signale soll mindestens erhalten werden. Die Auswertungsalgorithmen sind nicht Bestandteil der Arbeit und werden vom WSL zur Verfügung gestellt.

Die von der bisherigen Anlage gemachten Messdaten enthalten die Dauer und Intensität jedes Aufschlags (Ereignis) auf der Sensorplatte, sowie die Anzahl Ausschläge (Peaks) pro Aufschlag. Pro Minute wird ein Histogramm über die Intensitäten der Peaks gebildet und abgespeichert.

Denkbar wäre es, einen Prototyp für Vergleichsmessungen im Erlenbach (Alptal, SZ) an einer bestehenden Schwelle zu implementieren.

2.1.1. Musskriterien

- Die Anlage zeichnet den Geschiebetransport im Bachbett auf. Die bisherige Aufzeichnungsrate von 10'000 Messpunkten pro Sekunde soll nicht unterschritten werden.
- Die Anlage liefert eine minütliche Zusammenfassung über die Ereignisse an jedem Sensor. Diese Zusammenfassung enthält die Anzahl Ereignisse, Dauer und Intensität der einzelnen Ereignisse sowie ein Histogramm über die Intensitätsverteilung.
- Die Messstation ist fähig, mindestens zehn Sensoren zu betreiben und ihre Messsignale aufzuzeichnen.
- Es ist möglich, die kompletten Rohdaten von einem Sensor über eine Dauer von 30 Minuten aufzuzeichnen. Während einer solchen Messung dürfen die anderen Sensoren ihre Messung einstellen.
- Die Sensoren können über bis zu fünfzehn Meter im Bachbett verteilt sein.
- Die Leistungsaufnahme der Anlage beim Betrieb von 10 Sensoren ist kleiner als zehn Watt.
- Die Datenaufzeichnung erfolgt in einem eigens entwickelten Datenlogger.

- Am Datenlogger kann ein Laptop angeschlossen werden, um Kontrollparameter der Messanlage zu setzen und um den Status der Anlage abzufragen.
- Die erfassten Messdaten werden im Datenlogger auf einer Speicherkarte gespeichert. Dies ermöglicht ein einfaches Abholen der Daten im Feld, indem die Speicherkarte ausgetauscht wird.

2.1.2. Wunschkriterien

- Die Anlage liefert für jedes Ereignis die Rohdaten in voller zeitlicher Auflösung.
- Der Sensoraufbau ermöglicht es, die Sensoren in einer Elastomerplatte zu verpacken. Die Elastomerplatte kann ohne Betonkonstruktion im Bachbett verankert werden.
- Am Datenlogger kann ein Laptop angeschlossen werden, um die erfassten Messdaten herunterzuladen.

2.1.3. Abgrenzungskriterien

- Es würde den Rahmen dieser Arbeit sprengen, die Messeinheiten zur Produktreife zu bringen. Es wird lediglich aufgezeigt, wie solche Messeinheiten realisiert werden könnten.
- Eine Testinstallation in einem Bach ist nicht möglich. Allenfalls kann ausserhalb der Bachelorarbeit in der Versuchsanstalt für Wasserbau, Hydrologie und Glaziologie der ETH Zürich ein kleiner Testlauf stattfinden.

3. Vorgehen

3.1. Überblick

Das zu entwickelnde Messsystem kann grob in drei Komponenten aufgeteilt werden.

1. Datenlogger
2. Sensoreinheit
3. Bussystem

Der Datenlogger hat die Aufgabe, von mehreren Sensoreinheiten registrierte Ereignisse zu empfangen und zu speichern. Die Sensoreinheiten messen kontinuierlich die Beschleunigung, werten die Signale aus und erkennen Ereignisse, die einer vordefinierten Signalform entsprechen. Alle Sensoreinheiten sind über ein Bussystem mit dem Datenlogger verbunden, um miteinander kommunizieren zu können. Der prinzipielle Aufbau ist in Abbildung 3.1 ersichtlich. Die Stromversorgung der Anlage wird am Datenlogger angeschlossen. Parallel zum Kabel des Datenbusses wird die Stromversorgung der Sensoreinheiten geführt.

Diese drei Einheiten werden im Folgenden genauer definiert.

3.2. Datenlogger

Der Datenlogger hat verschiedene Aufgaben zu erfüllen:

- Sammeln und speichern der Messdaten der Sensoreinheiten.
- Kontrolle über das Bussystem.
- Steuerung des Betriebs der Anlage.
- Schnittstelle für die Konfiguration der Anlage und für das Auslesen der Messdaten.

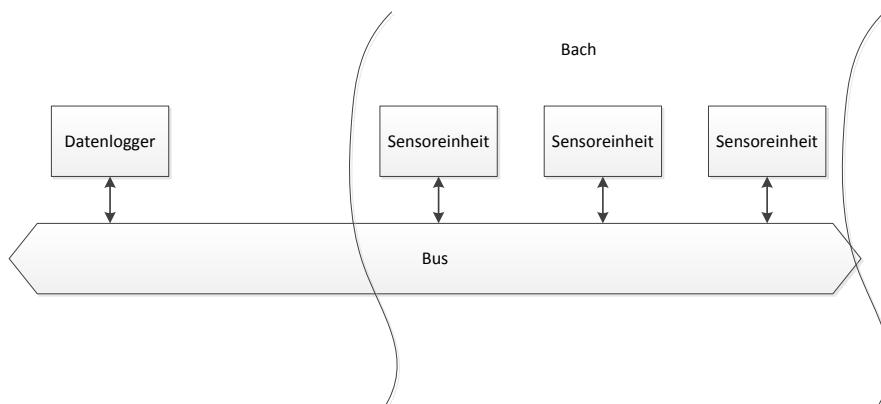


Abbildung 3.1.: Eine Messstation mit einem Datenlogger, der mehrere Sensoreinheiten im Bach steuert.

3.2.1. Messdaten sammeln

Für jede angeschlossene Sensoreinheit führt der Datenlogger eine Datensammlung, in der die registrierten Ereignisse zeitlich sortiert abgespeichert werden. Die Datensammlungen werden in Dateien abgelegt, die auf einem externen, auswechselbaren Medium abgespeichert werden. So können die Messdaten auf einfache Art für die weitere Auswertung abgeholt werden.

3.2.2. Kontrolle über das Bussystem

Als Busmaster hat der Datenlogger die Aufgabe, allen angeschlossenen Einheiten eine eindeutige Identifikationsnummer (ID) zuzuweisen. Über diese ID erkennt der Datenlogger, von welcher Sensoreinheit Daten übertragen werden. Anhand der ID kann der Datenlogger Konfigurationsnachrichten an bestimmte Sensoreinheiten addressieren. Für die Zuordnung der Messdaten zu einem bestimmten Sensor benötigen die Sensoreinheiten ein fixes Erkennungsmerkmal, z.B. eine Seriennummer, die mit den Messdaten abgespeichert werden soll. Wurde einer Sensoreinheit einmal eine ID zugewiesen, wird diese Zugehörigkeit in einer Konfigurationsdatei auf dem Datenlogger abgespeichert, um bei einem Neustart des Systems die gleichen IDs zu vergeben.

3.2.3. Steuerung des Betriebs

Die Messstation hat verschiedene Betriebsmodi, die über den Datenlogger angewählt werden können. Der Datenlogger steuert die einzelnen Sensoreinheiten entsprechend an. Je nach Betriebsmodus werden mehr oder weniger detaillierte Daten über die Ereignisse abgespeichert.

3.2.4. Schnittstelle nach Aussen

Über eine Schnittstelle am Datenlogger kann ein Computer angeschlossen werden. Per Kommandozeile wird die Messstation konfiguriert, der Zustand überprüft und der Betriebsmodus gewählt.

3.3. Sensoreinheit

Die Aufgaben der Sensoreinheit umfassen:

- Erfassung von Messdaten.
- Erkennung von Ereignissen.
- Übertragung der Ereignisdaten an den Datenlogger.

3.3.1. Messdatenerfassung

Der Sensor zur Erfassung der Daten wird mit einer vordefinierten Abtastrate ausgelesen. Die Abtastrate muss so gewählt werden, dass einzelne Ereignisse erkannt werden können, ohne unnötig viel Messdaten zu generieren.

3.3.2. Ereigniserkennung

Im Mikroprozessor werden die Messdaten fortlaufend analysiert. Überschreitet das gemessene Signal einen gewissen Schwellenwert (Threshold), markiert dies den Beginn eines Ereignisses. Das Ereignis ist beendet, wenn der Signalpegel für eine gewisse Zeit (Timeout) unterhalb des Threshold bleibt. Für jedes Ereignis wird abgespeichert, wann es aufgetreten ist (Timestamp), wie lange es gedauert hat, wie hoch der Signalpegel maximal ausschlug und wie viele Signalspitzen (Peaks) aufgetreten sind. Allenfalls können auch die Höhen und Timestamps aller Peaks übertragen werden.

3.3.3. Datenübertragung

Die Sensoreinheit sendet die Messdaten regelmässig über das Bussystem an den Datenlogger. Nach Bestätigung des Erhalts werden die Messdaten aus dem Speicher der Sensoreinheit gelöscht.

3.4. Bussystem

Das Bussystem verbindet die Einheiten der Messstation miteinander. Die gesamten Messdaten und Steuerkommandos werden über den Bus übertragen. Das Bussystem muss die Datenmenge der angeschlossenen Sensoren bewältigen können, über die geforderte Distanz funktionieren und möglichst robust gegenüber äusseren Einflüssen sein. Der Busmaster hat die Möglichkeit, laufende Übertragungen von Sensoreinheiten zu unterbrechen, um Steuerkommandos zu senden.

4. Funktionale Anforderungen

4.1. Datenlogger (F1...)

4.1.1. F110 Busmaster

Der Datenlogger übernimmt die Kontrolle des Bussystem. Bei Inbetriebnahme des Systems tastet der Datenlogger den Bus nach Sensoreinheiten ab und erteilt jeder Sensoreinheit eine eindeutige Identifikationsnummer (ID). Die ID des Datenloggers soll so gewählt werden, dass er jederzeit Priorität hat, auf den Bus zu schreiben.

4.1.2. F120 Sensorerkennung

Die angeschlossenen Sensoren werden vom Datenlogger erkannt und mit einer ID versehen. Anhand der ID wird die Priorität bei der Datenübertragung festgelegt und der Sensor identifiziert. Ein Sensor, der bereits am System angeschlossen war, erhält wieder die gleiche ID, sofern die Konfigurationsdatei nicht gelöscht wurde.

4.1.3. F130 Uhrzeit

Der Datenlogger verfügt über eine interne Uhr, um die Ereignisse in den Dateien mit einem lesbaren Zeitstempel zu versehen.

4.1.4. F140 Timestamp verteilen

Der Datenlogger sendet ein Signal an alle Sensoreinheiten, dass der Zeitstempel (Timestamp) neu gestellt werden soll. Ab dann beziehen sich die Timestamps auf die Dauer seit dem jetzigen Zeitpunkt.

4.1.5. F160 Schnittstelle zum Steuerrechner

Der Datenlogger bietet eine Schnittstelle, an der ein Steuerrechner (Laptop, Personal Computer (PC)) angeschlossen werden kann. Über diese Schnittstelle kann der Betrieb der ganzen Anlage gesteuert werden.

4.1.6. F170 Steuerung Betriebsmodus

Der Betriebsmodus der Sensoren wird vom Datenlogger aus gesteuert: Wie viele und welche Art von Daten gesammelt werden soll und ob alle Sensoren oder nur bestimmte aktiv sein sollen.

Folgende Betriebsmodi sind verfügbar:

- Normaler Modus: Alle Sensoren übermitteln die verarbeiteten Ereignisdaten. Zeitpunkt, Intensität, Dauer und Anzahl Ausschläge jedes Ereignis werden gespeichert.
- Detaillierter Modus: Alle Sensoren übermitteln die verarbeiteten Ereignisdaten sowie die gesamten Messdaten für die Dauer des Ereignis.

- Rohdatenmodus: Ein Sensor übermittelt kontinuierlich Rohdaten, die anderen Sensoren werden vorübergehend abgeschaltet.

4.1.7. F180 Daten sammeln

Der Datenlogger fragt in regelmässigen Abständen bei den Sensoreinheiten an, ob Ereignisdaten zur Übertragung bereit sind. Diese übermitteln die vorliegenden Ereignisdaten.

4.1.8. F190 Daten speichern

Die Daten werden vom Datenlogger auf einer Speicherplatte in Dateien abgelegt. Nach entsprechenden Befehlen vom Steuerrechner kann die Karte entfernt und ausgetauscht werden, um die Daten abzuholen.

4.2. Sensoreinheit (F4...)

4.2.1. F410 Ereignisdetektion

Die Sensoreinheit liest den Sensor mit einer definierten Abtastrate aus und wertet die Messdaten aus. Der Prozessor erkennt Ereignisse anhand definierter Kriterien. Zu jedem Ereignis werden folgende Daten gespeichert: Zeitpunkt (Timestamp), Dauer, Anzahl Peaks und höchster Peak. In einem zweiten Betriebsmodus können alle Messpunkte während einem Ereignis gespeichert werden.

4.2.2. F430 Datenübertragung

Die Sensoreinheit übermittelt die Ereignisdaten über das Bussystem an den Datenlogger.

4.2.3. F450 Rohdatenaufzeichnung

In einem Sondermodus werden alle Messpunkte gespeichert und über das Bussystem an den Datenlogger übertragen. In diesem Betriebsmodus darf auch nur eine Sensoreinheit aktiv sein, die anderen werden auf Standby geschaltet.

5. Nichtfunktionale Anforderungen

- Die gesamte Messstation soll eine geringere Leistungsaufnahme haben als eine aktuelle Messstation mit Geophonen. Für zehn Geophone sind dies zur Zeit ungefähr zehn Watt.
- Die Installation soll weniger bauliche Massnahmen erfordern als eine aktuelle Messstation mit Geophonen.
- Die erfassten Ereignisdaten sollen mindestens so detailliert sein wie von den bisherigen Installationen.
- Sensoreinheiten müssen wasserdicht verpackt werden können.

6. Hardware-Konzept

6.1. Hardware-Architektur

Anhand der funktionalen Vorgaben für die Messstation werden der Datenlogger, die Sensoreinheit und das Bussystem im folgenden genauer spezifiziert und die Komponenten ausgewählt.

6.1.1. Datenlogger

Das Hardware-Konzept des Datenloggers ist in Abbildung 6.1 dargestellt. Der Datenlogger sammelt die Daten der Sensoreinheiten über das Bussystem ein und speichert sie ab. Dafür benötigt er das Bussystem, einen Mikrokontroller, internen Speicher und ein leicht auswechselbares Speichermedium. Außerdem soll über eine Schnittstelle ein Computer angeschlossen werden können, um den Betrieb der Messstation zu steuern. Der Datenlogger wird in einem wasserdichten Gehäuse untergebracht. Für den Austausch des Speichermediums wäre eine verschraubbare Öffnung denkbar.

6.1.2. Sensoreinheit

Die Sensoreinheit benötigt einen Beschleunigungssensor, um die Einschläge von Geschiebe zu messen. Über einen Analog-Digital-Wandler (A/D-Wandler, Englisch Analog Digital Converter (ADC)) werden die Messsignale digitalisiert. Die gemessenen Signale werden von einem Mikrokontroller verarbeitet, im internen Speicher zwischengespeichert und über das Bussystem an den Datenlogger übertragen. Abbildung 6.2 zeigt das Hardware-Konzept der Sensoreinheit.

6.1.3. Bussystem

Das Bussystem muss die Daten und Befehle zwischen Datenlogger und Sensoreinheiten übertragen. Die Reichweite des Bussystems muss genügen, um alle Komponenten der Messinstallation zu verbinden. Die Datenbandbreite muss die Übertragung der Messresultate aller Sensoren erlauben.

6.2. Komponentenauswahl

6.2.1. Mikroprozessor

Bei der Auswahl des Mikrocontrollers werden folgende Kriterien berücksichtigt:

- Genügend Rechenleistung für allfällige zusätzliche Anforderungen.
- A/D-Wandler mit genügender Abtastrate und Auflösung (Bit-Breite).
- DSP integriert für die schnelle Verarbeitung der Messdaten.
- Nested Vectored Interrupt Controller (NVIC).

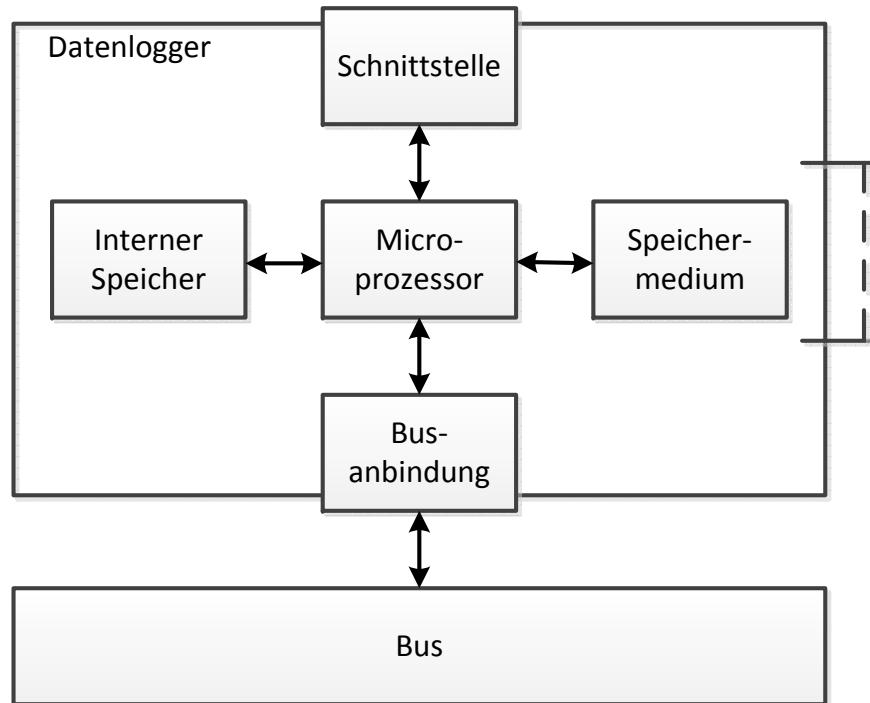


Abbildung 6.1.: Hardwarekonzept des Datenloggers.

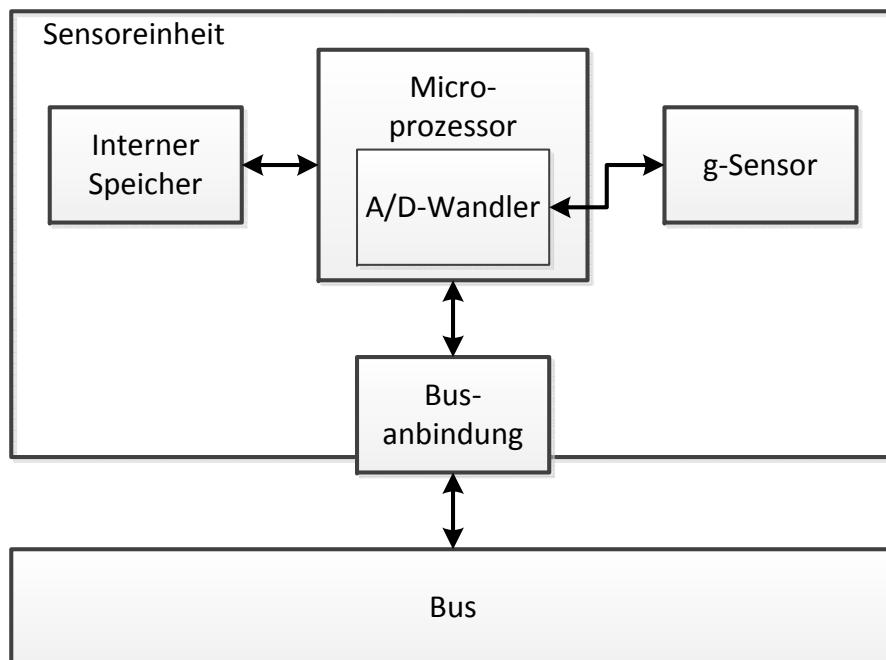


Abbildung 6.2.: Hardwarekonzept der Sensoreinheit.

- Ein-/Ausgänge für das Bussystem.
- Ein-/Ausgänge für den externen Speicher.
- möglichst geringer Stromverbrauch.

Für die vorliegende Anwendung eignen sich Mobile-Prozessoren sehr gut. Sie sind für den Einsatz in mobilen Geräten konzipiert, d.h. für den Batteriebetrieb, sind aber trotzdem sehr leistungsfähig. Die *ARM Cortex*-Reihe bietet ein breites Spektrum an Prozessoren an. Im Mobile-Segment der *ARM Cortex*-Reihe sind vier Prozessoren erhältlich. Da nur ein Modell einen Digitalen Signal-Prozessor (DSP) aufweist, ist die Entscheidung einfach. Der *ARM Cortex-M4* ist auch der neueste Prozessor aus dem Mobile-Segment. Mit dem Ausblick, das Messsystem in einem zukünftigen Projekt zur Serienreife zu bringen, macht es nur Sinn, den neuesten Prozessor zu verwenden. Ein Auszug aus dem Datenblatt des *ARM Cortex-M4* befindet sich im Anhang A.5.1. Abbildung 6.3 zeigt die Fähigkeiten des *ARM Cortex-M4*.

Nested Vectored Interrupt Controller Ein Prozessor mit NVIC kann auf verschiedene Ereignisse reagieren, indem Interrupts ausgelöst werden. Jedem Interrupt kann eine Priorität zugewiesen werden, um festzulegen, ob ein Interrupt einen anderen unterbrechen darf, der gerade vom Prozessor abgearbeitet wird. Für das Messsystem wird ein NVIC benötigt, da mehrere zeitkritische Prozesse parallel ablaufen sollen. Einerseits muss die Abtastrate der Messwerterfassung genau eingehalten werden. Andererseits darf die Verarbeitung der Messwerte nicht zu lange unterbrochen werden, um einen Überlauf der Queue zu vermeiden. Die Daten der Ereignisse müssen parallel dazu an den Datenlogger übermittelt werden. Die Prioritäten dieser Prozesse müssen richtig gewählt werden. Im Abschnitt ?? wird näher darauf eingegangen.

DSP Der *ARM Cortex-M4* verfügt über DSP-Funktionen, eine sog. single-cycle multiply-accumulate unit (MAC). In einer single-cycle MAC können Multiplikationen in einem einzigen Prozessorzyklus ausgeführt werden. Normalerweise benötigt ein Prozessor für eine Multiplikation bis zu mehreren zehn Zyklen, bis das Resultat vorliegt. Mit einem DSP ist es daher möglich, z.B. eine Filterfunktion viel effizienter zu berechnen als mit einem normalen Prozessor, da für diese viele Multiplikationen und Additionen ausgeführt werden müssen.

Floating Point Unit (FPU) Eine FPU führt Berechnungen mit Dezimalbrüchen sehr rasch aus. Für unsere Anwendung ist eine FPU keine Voraussetzung. Falls in einem späteren Projekt komplexere Filter oder andere Anwendungen berechnet werden müssen, könnte die FPU aber ein Vorteil sein.

Wake Up Interrupt Controller Interface Ein Wake Up Interrupt Controller Interface ermöglicht es, einen Prozessor in einen Stromsparmodus zu versetzen und ihn durch ein definiertes Signal wieder zu wecken. Damit ist es möglich, eine Sensoreinheit praktisch ganz abzuschalten, wenn sie keine Messungen durchführt. Durch eine Nachricht über das Bussystem kann die Sensoreinheit wieder eingeschaltet werden. Der Cortex-M4 verfügt über 240 mögliche Wake Up Interrupts, kann also für 240 Ereignisse programmiert werden, die ihn aufwecken oder in einen Stromsparmodus senden können (vgl. [3]).

Rechenleistung Die Rechenleistung des *ARM Cortex-M4* hängt von der Implementation ab. Die Firma ARM produziert den *ARM Cortex-M4* nicht selbst, sondern lizenziert Chip-Hersteller für die Verwendung der Architektur in ihren Prozessoren. Vom *ARM Cortex-M4* sind mehrere Ausführungen erhältlich. Verschiedene Chip-Hersteller implementieren eine Architektur des *ARM Cortex-M4* in ihren Prozessoren.

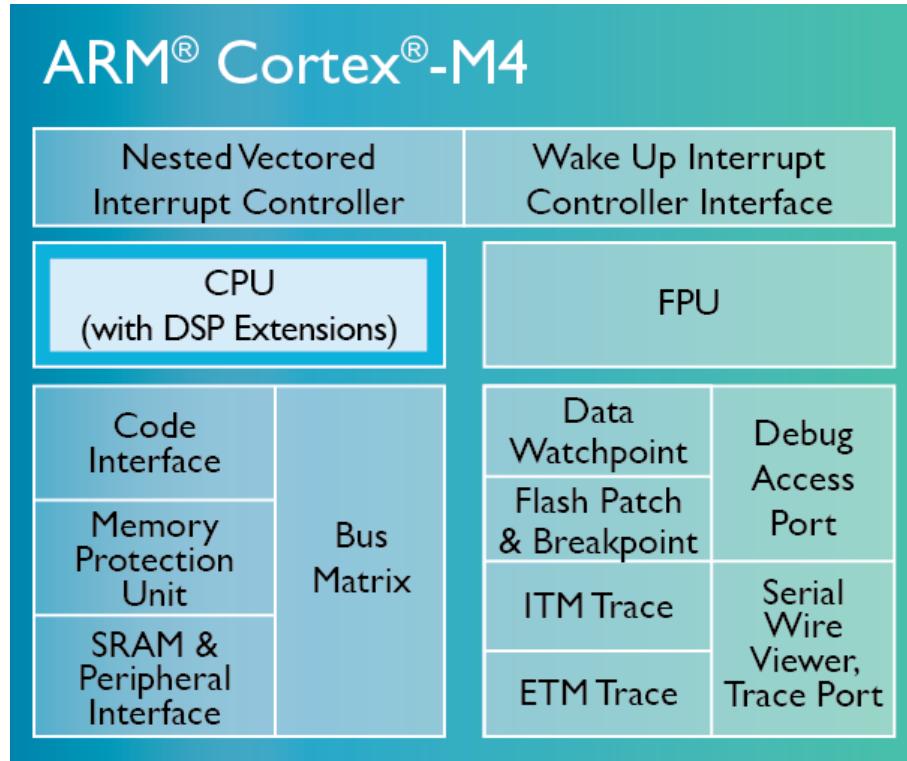


Abbildung 6.3.: Chipdiagramm der *ARM Cortex-M4* Architektur [3].

A/D-Wandler Bei den bestehenden Messstationen wird die Datenmenge stark reduziert, indem über eine Minute ein Histogramm mit den Peakintensitäten als Klassen berechnet wird. Die Intensitäten werden in 18 Klassen logarithmischer Abstufung eingeteilt. Das entspricht einer Bit-Breite von etwas mehr als vier Bit (4 Bit ermöglichen 16 Werte). Da für die Klasseneinteilung eine logarithmische Abstufung gewählt wurde, muss ein linearer A/D-Wandler trotzdem eine höhere Bit-Breite als vier aufweisen, um die gleiche Auflösung wie in den unteren logarithmischen Klassen zu erreichen. Mit 12 Bit Auflösung sind 4096 Stufen unterscheidbar, was für diese Anwendung genügt.

Peripherie Damit der Prozessor über das Bussystem kommunizieren und auf ein externes Speichermedium schreiben kann, sind genügend Ein- und Ausgabe-Pins nötig.

Wahl eines Prozessors Da der Entscheid für eine Hardware schon zu Beginn des Projekts gefällt werden musste, wurde auf eine grosszügige Sicherheitsmarge in Sachen Rechenleistung und Bit-Breite geachtet. Um Kosten und Baugröße der Sensoreinheit klein zu halten, suchten wir nach einem Evaluationsboard mit *ARM Cortex-M4* Prozessor. Das *LPC4088 QuickStart Board* von *NXP Semiconductors* hat genügend Arbeitsspeicher für den Prozessor, verfügt über die benötigten Pins für die Peripherie und hat bei weitem genügend Rechenleistung. Die Fähigkeiten des NXP *LPC4088FET208* Prozessors sind in Tabelle 6.1 dargestellt.

Auf dem *NXP LPC4088 QuickStart Board* sind zusätzliche Bauteile verbaut, z.B. Speicher (SDRAM und Flash-Speicher und Pin-Steckleisten, um CAN-Bus-Pins oder A/D-Eingänge anzuschliessen. Tabelle 6.2 listet die für dieses Projekt relevanten, zusätzlichen Eigenschaften auf.

6.2.2. Bus-System

Das Bussystem für die Messanlage muss die folgenden Kriterien erfüllen:

Taktfrequenz	bis 120 MHz
NVIC	vorhanden
FPU	vorhanden
Programmspeicher	512 kByte
Arbeitsspeicher (intern)	96 kByte
CAN-Bus	2
USB	2
SD-Card	Anschlüsse vorhanden
A/D-Wandler	8 Eingänge, 12 Bit

Tabelle 6.1.: Fähigkeiten des NXP LPC4088 Prozessors [5].

Prozessor	NXP LPC4088FET208
Taktfrequenz	bis 120 MHz
Flash-Speicher	8 MByte
SDRAM	32 MByte
A/D-Wandler	6 Eingänge nutzbar, 12 Bit

Tabelle 6.2.: Zusätzliche Fähigkeiten des NXP LPC4088 QuickStart Boards von *Embedded Artists* [6].

- Übertragungsbandbreite genügend für fortlaufende Übertragung von Rohdaten einer Sensoreinheit.
- Reichweite mindestens 20 Meter.
- Robust gegenüber äusseren Einflüssen.
- Mindestens zwanzig Busteilnehmer möglich.

In Tabelle 6.3 sind die Eigenschaften diverser Bussysteme aufgeführt.

Kommentare SPI und I2C sind nur für kurze Distanzen geeignet und sind deshalb keine Option. Die Verwendung von Ethernet zur Datenübertragung würde zwei Schnittstellen auf jeder Sensoreinheit voraussetzen, um die Sensoren hintereinander zusammenzuhängen (Daisy Chain). Jedes Paket müsste vom Mikrokontroller weitergeleitet werden, wenn es für einen anderen Empfänger bestimmt ist. Dies führt zu einer zusätzlichen Belastung der Microcontroller. Stromversorgung über Ethernet ist mit PowerOverEthernet (PoE) zwar möglich, erfordert aber spezielle Geräte zur Speisung über den Stecker des Datenkabels. Dies verunmöglicht eine Daisy Chain mit PoE, neben dem Datenkabel wäre noch ein Kabel für die Stromversorgung notwendig.

Vergleich CAN-Bus und RS485 Die beiden Bussysteme CAN und RS485 sind nicht einfach zu vergleichen, da der RS485-Standard nur die elektrischen Eigenschaften des Systems beschreibt (OSI Layer 1). Der CAN-Standard beschreibt auch den Data Link Layer (OSI Layer 2). Der Data Link Layer beschreibt Methoden, die die Übertragung zuverlässig machen. Ein Beispiel dafür ist eine Prüfsumme, die es ermöglicht, eine fehlerhafte Übertragung im Empfänger festzustellen. Der Empfänger bestätigt bei korrekter Prüfsumme den Empfang. Stellt einer der Empfänger einen Fehler fest, sendet er eine Fehlermeldung über den Bus und stört damit die Übertragung. So ist es nicht möglich, dass einige Empfänger die Nachricht lesen konnten und andere nicht. Der Sender ist bei erfolgter Bestätigung sicher, dass seine Nachricht erfolgreich an alle Empfänger übertragen wurde (vgl. [10]).

CAN-Bus definiert auch eine Adressierung, Kollisions-Erkennung und -Auflösung, Prioritäten der Busteilnehmer und ein Nachrichtenformat. Bei RS485 besteht eine Nachricht aus einem einzelnen Zeichen. Der Data Link Layer muss gänzlich in Software gelöst werden. Dafür ist es möglich, das Protokoll komplett selbst zu definieren. Dies erlaubt beliebig lange Übertragungen von einem Busteilnehmer.

	Bitrate	Distanz	Clients	Besonderheiten
CAN	1 MBit/s 125 kBit/s	40 m 500 m	> 20	+ Collision Detection (CD) umgehen mit Polling durch Master. + Bei synchronem CAN wird CD durch ID gelöst. + CAN Controller sendet Interrupt Request bei erhaltener Nachricht.
SPI	..100 MBit/s	< 1 m	slave select	- Pro Client eine Slave Select Leitung - alternativ: Daisy Chain ⇒ alle Mikrocontroller (MC) beschäftigt. - Bei Ausfall eines MC ganzer Bus unterbrochen.
RS485	35 MBit/s 100 kBit/s	10 m 1200 m	>32	- Master am besten in der Mitte des Bus ⇒ ungünstig. - Braucht 2..4 Drähte (bei Full Duplex) - braucht pull-up und pull-down Widerstände ⇒ mehr Leistungsaufnahme.
Ethernet	100 MBit/s	100 m	> 20	+ Stromversorgung bei Power over Ethernet (PoE) integriert. - kein Bus sondern allenfalls Daisy Chain. - bei Daisy Chain kein PoE möglich.
Feldbus				ist eine Familie von Bussen, z.B. CAN-Bus
I2C	0.4..5 Mbit/s	wenige Meter	< 20	nur für kurze Distanzen, Bitrate nimmt mit zunehmender Distanz rasch ab.

Tabelle 6.3.: Entscheidungsmatrix für die Auswahl des Bussystems.

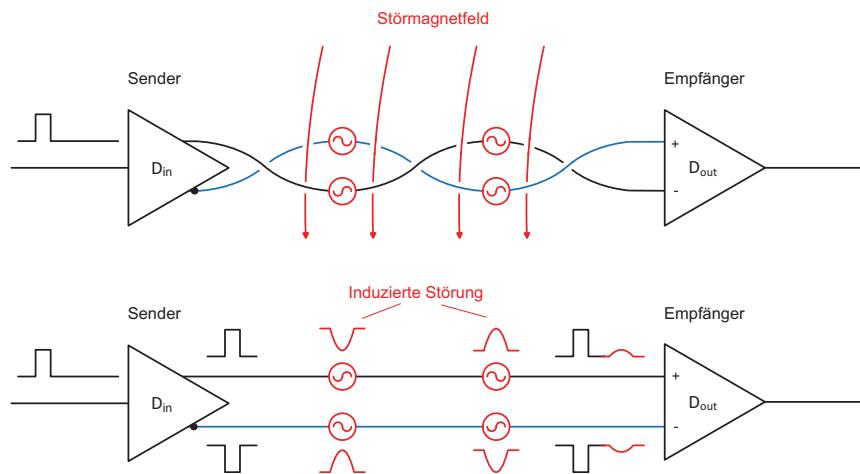


Abbildung 6.4.: Differentielle Leitung [4].

CAN-Bus limitiert die Nachrichtenlänge auf 8 Bytes. Die Übertragung längerer Nachrichten muss über Software geregelt werden (vgl. [10]).

Bei beiden Standards ist der Stecker nicht definiert. Das lässt die komplette Freiheit für die Wahl eines wasserdichten Steckverbinders.

Da der CAN-Bus bereits mit dem Standard viele benötigte Merkmale mitbringt, fällt die Entscheidung nicht schwer.

Entscheidung CAN-Bus erfüllt alle Kriterien und erlaubt es, den Busmaster am Ende des Bus zu platzieren. Dies ist ein weiterer Vorteil gegenüber RS485, wo der Master in der Mitte platziert werden sollte. CAN-Bus bietet bereits Collision Detection und Fehlererkennung, während dies bei RS485 in der Software gelöst werden muss. Für CAN-Bus sind Bus-Treiber (Transceiver) erhältlich, die mit hohen Spannungen umgehen können, was das Bussystem robuster gegenüber Umwelteinflüssen macht. Die Grösse der Datenpakete ist bei CAN-Bus auf 8 Byte begrenzt, bei RS485 werden die Datenpakete über die Software frei definiert, was einen Vorteil von RS485 darstellt. Insgesamt überwiegen die Vorteile von CAN-Bus klar.

CAN-Bus CAN-Bus ist ein Bussystem über eine differentielle Leitung. Auf zwei Drähten liegt eine bestimmte Spannung an. Der Empfänger misst den Spannungsdifferenz zwischen den beiden Drähten. Die Differenz wird vom Sender entweder klein oder gross gehalten, um ein 'dominantes' oder ein 'rezessives' Bit zu senden. 'Rezessiv' bedeutet dabei, dass ein anderer Busteilnehmer durch anlegen eines 'dominannten' Bits das 'rezessive' Bit überschreiben kann. Welche Spannungsdifferenzen 'dominant' und 'rezessiv' darstellen, ist vom Standard nicht definiert und lässt dem Entwickler damit freie Hand, die elektrischen Eigenschaften seiner CAN-Implementation seinen Bedürfnissen entsprechend zu definieren. Um Signalreflexionen am Ende der Leitungen zu vermeiden, wird ein Terminierungswiderstand an beiden Enden benötigt, der die beiden Leitungen abschliesst. (vgl. [1]).

Die beiden Drähte der differentiellen Leitung sind verdrillt. Dadurch wird das Signal der Leitung besser vor äusseren Einflüssen geschützt. Elektrische und magnetische Felder induzieren in einem Draht einen Strom. Da das äussere Feld aber in beiden Drähten praktisch den gleichen Strom induziert, ändert sich an der Spannungsdifferenz in den Drähten kaum etwas (siehe Abbildung 6.4). Die beiden Drähte stellen auch eine Spule dar. Durch das Verdrillen der Drähte ändert sich die Ausrichtung der Spule im äusseren Feld auf kurzen Abständen, induzierte Störungen heben sich so gegenseitig auf (vgl. [4, Kap. 2, S. 14]).

Die Kollisions-Erkennung und -Auflösung wird über ID-Nummern der Teilnehmer gelöst. Vor dem Start einer Übertragung prüft der Teilnehmer, dass der Bus zur Zeit frei ist. Dann sendet der Teilnehmer seine ID-Nummer. Falls zwei Teilnehmer gleichzeitig zu senden beginnen, werden sie irgendwann unterschiedliche Bits senden. Der Teilnehmer, der dann das 'dominante' Bit sendet, liest vom Bus den gleichen Wert, den er gerade sendet. Für ihn ist die Übertragung nicht gestört. Der Teilnehmer mit dem 'reziproken' Bit liest aber das 'dominante' Bit des anderen Senders und muss seine Übertragung sofort abbrechen. Diese Art der Kollisionsauflösung hat den Vorteil, dass keine Übertragungszeit verloren geht, da einer der Teilnehmer seine Nachricht senden darf. Der Verlierer wartet, bis der Bus wieder frei ist und probiert dann erneut, die Nachricht zu senden (vgl. [1]).

Die korrekte Übertragung wird mittels eines Cyclic Redundancy Check (CRC)-Codes überprüft. Der Empfänger berechnet bereits während dem Empfang der Nachricht die CRC-Prüfsumme. Sobald die vom Sender mitgeschickte CRC-Prüfsumme übertragen ist, kann der Empfänger diese mit der berechneten CRC vergleichen. Stimmen die Prüfsummen überein, bestätigt der Empfänger die korrekte Übertragung (vgl. [1]).

6.2.3. Speichermedium

Kriterien Das externe Speichermedium soll möglichst klein sein, wenig Stromverbrauch haben und einfach auswechselbar sein. Bei Inaktivität sollte das Medium wenn möglich keinen Strom verbrauchen. Für einen mehrwöchigen unabhängigen Betrieb einer Messstation muss genügend Speicherkapazität bereitgestellt werden.

Datenmenge Pro Sensor werden bei hohem Geschiebeaufkommen maximal hundert Ereignisse pro Sekunde erwartet. Ein solches Geschiebeaufkommen stellt jedoch die Ausnahme dar. Ein Ereignis benötigt je nach verlangtem Detailgrad und Dauer des Ereignisses 10–200 Byte Speicherplatz. Für den normalen Betriebsmodus werden 50 Byte/Ereignis gerechnet, bei 5 Ereignissen pro Sekunde. Damit ergibt sich eine Datenrate von 250 Byte/s, die es pro Sensor abzuspeichern gilt. Mit zehn Sensoren im Einsatz müssen 2.5 kByte/s gespeichert werden.

Unabhängige Betriebsdauer Pro Gigabyte Speicherplatz können 111 Stunden Daten für zehn Sensoren gespeichert werden. Bei hohem Geschiebeaufkommen mit zwanzig mal mehr Ereignissen bleiben immer noch 5 Stunden Aufzeichnungszeit pro Gigabyte. Begrüßt man sich mit weniger Details, fallen pro Sensor in zehn Sekunden rund 400 Byte Daten an. Bei dieser Datenrate reicht ein Gigabyte für rund 700 Stunden. Auch bei hohem Geschiebeaufkommen kann die Anlage mehrere Tage an Daten speichern.

Kapazität Heute sind Speichermedien mit Kapazitäten bis über 128 GB erhältlich, so dass die Detailrate kein entscheidendes Kriterium mehr darstellt.

Datentransfer Für den Transfer der Daten aus dem Datenlogger auf einen Computer gibt es grundsätzlich zwei Varianten. Entweder man liest die Daten über eine Schnittstelle auf den Computer aus, oder man tauscht das Speichermedium aus. Das Auslesen via Schnittstelle benötigt zusätzlich Strom, das Wechseln des Speichermediums setzt einen mehr oder weniger komfortablen und trotzdem wasserdichten Zugang zum Medium voraus. Da heute Speichermedien mit kleinem Platzbedarf erhältlich sind, kann ein solcher Zugang recht einfach mit einem Schraubverschluss realisiert werden.

Vergleich In Tabelle 6.4 werden verschiedene Speichermedien miteinander verglichen. In der Spalte 'Breite' ist aufgelistet, wie gross eine Öffnung mindestens sein muss, um das Speichermedium wechseln zu können. 'Pins' gibt an, wie viele Leitungen für den Anschluss des Mediums am Microcontroller nötig sind. Der Stromverbrauch in Klammern ist für den Standby-Modus des Speichermediums.

	Breite	Pins	Stromverbrauch	Bemerkungen
SD-Card	24 mm	9	20..100 mA (0.2 mA)	4 bit breiter serieller Bus
CompactFlash	43 mm	50	max. 70 mA (k.A.)	paralleler Bus
USB-Stick	min. 12 mm	4	typ. 70 mA (k.A.)	

Tabelle 6.4.: Entscheidungsmatrix zur Auswahl des Speichermediums [2, 7, 9].

Entscheid Für einen verschraubbaren Verschluss ist die CompactFlash-Karte zu breit, das Gehäuse würde dadurch sehr gross werden. Die SD-Karte und der USB-Stick sind vergleichbar in der Grösse. Von der SD-Karte sind auch kleinere Varianten erhältlich. Eine Öffnung für den Austausch des Speichermediums kann eine gewisse Grösse ohnehin nicht unterschreiten, damit hineingegriffen werden kann. Da die SD-Karte im Standby den geringeren Stromverbrauch hat, wird der Datenlogger mit einem SD-Kartenleser ausgestattet.

6.2.4. Sensor

Sensorauswahl b
ben

6.2.5. Schnittstelle

Das gewählte *NXP LPC4088 QuickStart Board* verfügt über einen Universal Serial Bus (USB)-Anschluss, über den eine serielle Schnittstelle angesprochen werden kann. Für eine einfache Kommandozeile oder ein Konfigurationsmenü ist diese Schnittstelle ausreichend. Die Schnittstelle kann mit einer Übertragsrate von 9600 bis 115200 Baud betrieben werden.

6.2.6. Gehäuse

Um den Datenlogger und die Sensoreinheiten wasserdicht zu verpacken, wurden Gehäuse und Komponenten mit der Schutzklasse IP68 gesucht. Die wasserdichten Gehäuse der Firma *FIBOX* sind in verschiedenen Ausführungen erhältlich. Da diese Arbeit kein serienreifes Produkt zum Ziel hatte, wurden Gehäuse aus ABS-Kunststoff mit transparenten Deckeln gewählt.

Die Stecker wurden ebenfalls mit Schutzklasse IP68 gewählt. Als zusätzliches Kriterium galt der Durchmesser des Kabelmantels. Die Stecker *Lumberg 0332 05-1* verfügen über fünf Pole, die vierpolige Variante war zur Zeit nicht lieferbar.

Für den USB-Anschluss gibt es wasserdichte Durchführungen, an denen sowohl innen als auch aussen Standard-USB-Kabel angeschlossen werden können. Die *Buccaneer PCXP6043/B*-Gerätebuchse von *Bulgin* erfüllt die Anforderungen. Für diese Buchse ist auch eine wasserdichte Kappe erhältlich.

Die verschraubbare Öffnung für die SD-Karte besteht aus einer M36/M32-Gewindebuchse, in die ein M32-Schraubdeckel eingeschraubt wird.

6.3. Datenlogger

Die Hardware-Architektur des Datenloggers ist in Abbildung 6.5 schematisch dargestellt. Kernstück ist der Mikrokontroller *NXP LPC4088*. Der Mikrokontroller ist über eine serielle Schnittstelle mit dem CAN-Transceiver verbunden. Der Transceiver sendet Konfigurations- und Kontrolldaten und empfängt Messdaten über den CAN-Bus.

Dem Mikrokontroller steht sowohl interner Programm- und Datenspeicher zur Verfügung, dieser ist aber sehr beschränkt. Damit Konfigurations- und Messdaten zwischengespeichert werden können, steht dem Mikrokontroller auf dem *NXP LPC4088 QuickStartBoard* (gestrichelter Kasten) mehr Speicher

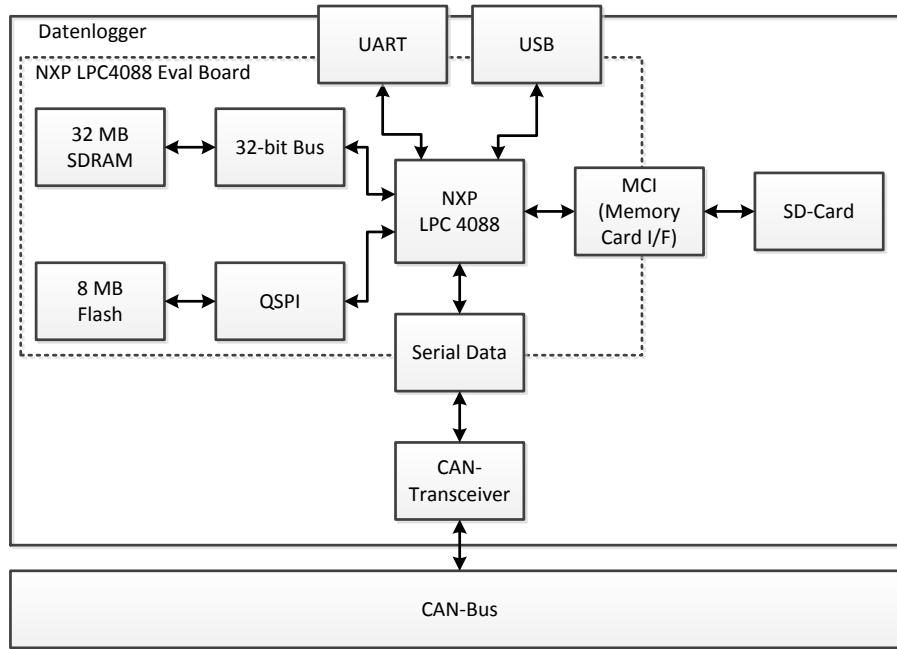


Abbildung 6.5.: Schematischer Hardware-Aufbau des Datenloggers.

zur Verfügung: 8 MByte Flash-Speicher und 32 MByte SDRAM. Dieser Speicher genügt, um Daten von mehreren Sensoreinheiten zwischenzuspeichern.

Über das Memory Card Interface (MCI) ist eine SD-Karte an den Mikrokontroller angebunden. Auf der SD-Karte werden sowohl die Konfigurationsdaten als auch die Messdaten gespeichert. Heute sind SD-Karten mit bis zu 256 GByte Speicherplatz erhältlich (SDXC-Karten). Zur Zeit unterstützt die verwendete Bibliothek zur Verwendung der SD-Karten nur SDHC-Karten mit bis zu 32 GByte Speicherplatz.

Für die Kommunikation mit einem Computer verfügt der Datenlogger über einen USB-Anschluss. Über einen Terminal-Emulator wie *PuTTY* kann via serielle Schnittstelle auf ein Konfigurationsmenü zugegriffen werden, um die Messanlage zu überwachen und zu steuern.

6.4. Sensoreinheit

Die Sensoreinheit ist ähnlich aufgebaut wie der Datenlogger. Das Schema in Abbildung 6.6 zeigt die Hardware. Wie der Datenlogger ist der NXP LPC4088 Mikrokontroller mit zusätzlichem SDRAM und Flash-Speicher ausgerüstet. Der Anschluss an den CAN-Bus ist identisch. Die SD-Karte wird in der Sensoreinheit nicht benötigt und wurde deshalb weggelassen. Auch eine USB-Schnittstelle für die Konfiguration ist nicht nötig, da die Konfiguration über den CAN-Bus erfolgt.

Der Beschleunigungs-Sensor wird vom *QuickStart Board* mit Spannung versorgt und gibt die gemessene Beschleunigung als analoge Spannung aus. Diese Spannung wird vom A/D-Wandler des NXP LPC4088 Mikrokontroller gemessen.

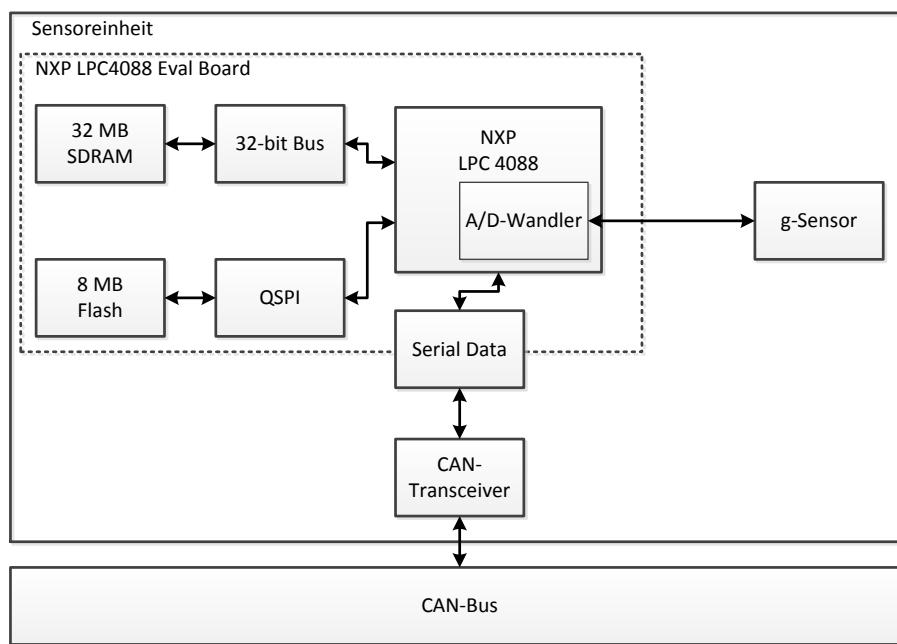


Abbildung 6.6.: Schematischer Hardware-Aufbau der Sensoreinheit.

7. Software-Konzept

7.1. Software-Stack

7.1.1. Überblick

7.1.2. Messdatenerfassung

Der NXP LPC4088 Mikroprozessor verfügt über einen 12-bit A/D-Wandler, der über einen Multiplexer auf acht Pins messen kann. Auf dem verwendeten Quickstart-Board stehen sechs Pins für A/D-Wandlung zur Verfügung. Für die geplante Anwendung reicht ein A/D-Pin, da der Beschleunigungs-Sensor die Beschleunigung nur auf einer Achse misst. Der A/D-Wandler des NXP LPC4088 wird mit einer Abtastrate von 10 kHz betrieben. Falls höhere Abtastraten nötig sind, kann der A/D-Wandler theoretisch mit bis zu 400 kHz betrieben werden.

Abtastrate Um die Abtastrate genau einzuhalten wird ein Timer des *NXP LPC4088* Mikrocontrollers eingesetzt, der die Messung des A/D-Wandlers anstößt (Listing 7.1). Die A/D-Wandlung nimmt immer genau 31 Taktzyklen in Anspruch. Nach abgeschlossener Messung schreibt der A/D-Wandler den Messwert in ein eigenes Register und setzt einen Interrupt Request (IRQ). Der IRQ signalisiert dem Mikrocontroller, dass ein Messwert zur Abholung bereit liegt. Damit der Messwert so rasch wie möglich, auf jeden Fall aber vor Ablauf der nächsten Messperiode, für die weitere Verarbeitung abgeholt werden kann, hat der IRQ die höchste mögliche Priorität. Der Interrupt des A/D-Wandlers unterbricht also jeden laufenden Prozess.

```
1 // ISR
2 void start_ADC_Conversion(){
3     // start conversion when the ticker fires
4     LPC_ADC->CR |=(1 << 24);
5 }
6
7 // ...
8 // register Interrupt Handler and attach ticker
9 int register_ADC_interrupt(analogin_s *obj, PinName pin, uint32_t
    ADC_IRQHandler, uint32_t interval){
10    // ...
11    ticker.attach_us(&start_ADC_Conversion, interval);
12    // ...
13 }
```

Listing 7.1: Timer mit Aufruf der A/D-Wandler-Funktion (ADC_4088.cpp)

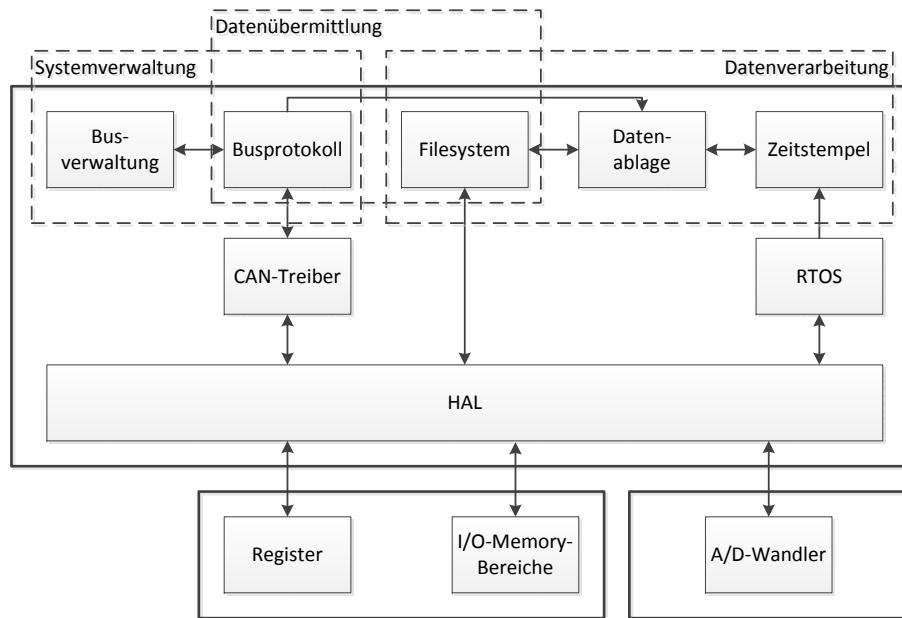


Abbildung 7.1.: Softwarestack des Datenloggers.

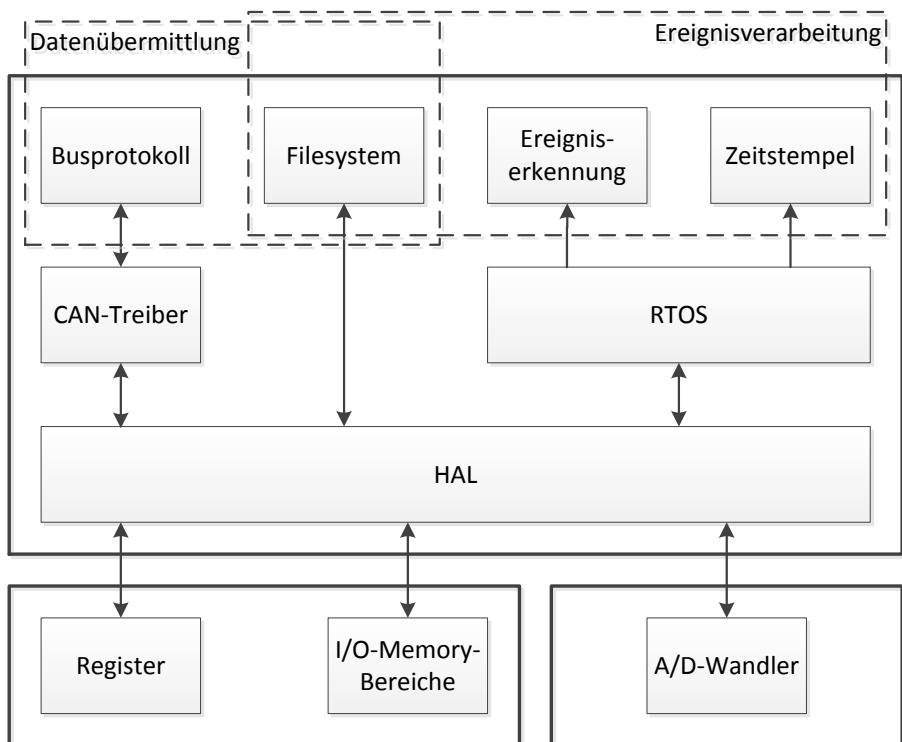


Abbildung 7.2.: Softwarestack der Sensoreinheit.

ISR Der Mikrokontroller ruft die Interrupt Service Routine (ISR) (Listing 7.2) auf, um den IRQ abzuhandeln. Die ISR holt den Messwert aus dem Register und speichert ihn zusammen mit dem Timestamp in einer FIFO-Queue ab. Die ISR erhöht den Timestamp um eins für den nächsten Messwert und gibt dann die Kontrolle an den unterbrochenen Prozess zurück.

```

1 void isr_nextMeasurement(){
2     // read ADC measurement from Register, automatically resets IRQ
3     value = LPC_ADC->GDR;
4     value = (value >> 4) & 0xFFFF;
5     timestamp++;
6
7     enqueue_impact_input(timestamp, value);
8 }
```

Listing 7.2: ISR zur Abhandlung des ADC-Interrupt Requests (impact_event.cpp)

Timestamp Der Timestamp muss zwingend in der ISR (Listing 7.2) erhöht werden, damit er immer sofort nach der Erfassung eines neuen Messwerts aktualisiert wird. Würde der Timestamp mittels einem zweiten Timer und einer zweiten ISR erhöht, bestünde die Gefahr einer ungeregelten Reihenfolge der Abarbeitung der IRQs. Einmal würde der Timestamp vor dem Kopieren der Messdaten erhöht, ein anderes mal erst danach.

Datenauswertung Die Datenauswertung wird in regelmässigen Abständen aufgerufen und arbeitet die FIFO-Queue mit den neuen Messwerten ab, um Ereignisse zu erkennen. Der Aufruf der Ereigniserkennung ist nicht so stark an einen genauen Takt gebunden wie die A/D-Wandlung, da die FIFO-Queue genügend gross ist, um mehrere hundert Messwerte zwischenspeichern. Ein Thread ruft die Ereigniserkennung auf und wartet nach Beendigung der Subroutine 1 ms.

7.1.3. Ereigniserkennung

Hilbert-Transformation

Von der WSL wurde die Ereigniserkennung bisher mittels Hilbert-Transformation gelöst. Die Hilbert-Transformation liefert die umhüllende Kurve des gemessenen Signals. Überschreitet die Umhüllende den Threshold, markiert dies den Start eines neuen Ereignisses. Fällt die Umhüllende unter den Threshold, ist das Ereignis beendet.

Die Berechnung der Hilbert-Transformation erfordert einen Aufwand. Mittels diskrete Fourier-Transformation (DFT) wird das Spektrum des Signals berechnet. Negative Frequenzanteile werden auf null gesetzt und das resultierende Spektrum mittels inverse diskrete Fourier-Transformation (IDFT) wieder in ein Signal umgerechnet (vgl. [8]). Das resultierende Signal umhüllt das Eingangssignal.

Für die DFT und die IDFT ist der Rechenaufwand je $N \cdot \log_2(N)$. Je mehr Datenpunkte in einem Schritt verrechnet werden (Blockgrösse), desto höher ist der Aufwand, aber desto genauer ist das Resultat. Mit einer Blockgrösse von 128 Messwerten benötigt die DFT und die IDFT je 896 komplexe Multiplikationen und Additionen (vgl. [12, Kap. 3, S. 48]). Pro Messwert sind das 7 komplexe Multiplikationen und Additionen. Dank der DSP-Fähigkeiten des gewählten Cortex™-M4 Prozessors liegt die zu erwartende Prozessorauslastung für die Hilbert-Transformation bei einer Abtastrate von 10 kHz bei wenigen Prozent.

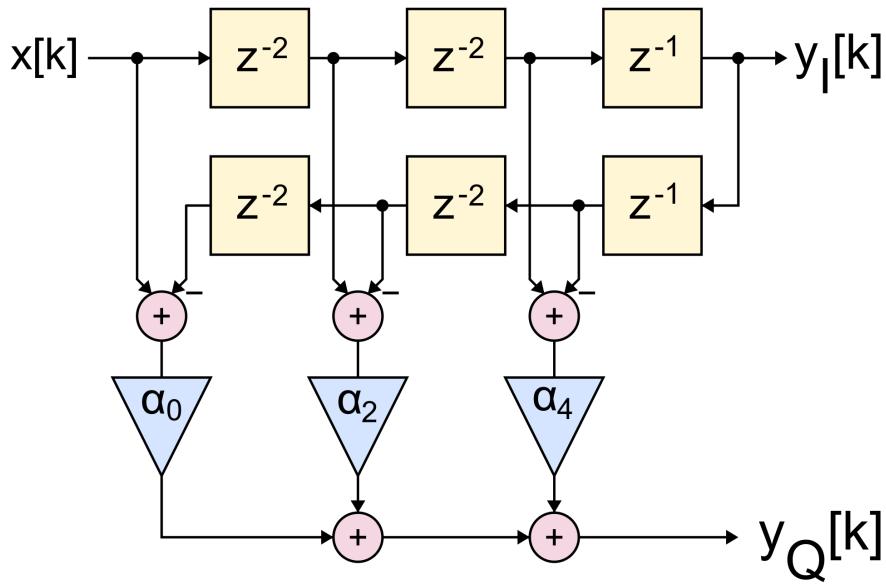


Abbildung 7.3.: Hilbert-Transformation als FIR-Filter [11].

Hilbert-Transformation als FIR-Filter

Die Hilbert-Transformation kann mittels eines FIR-Filters angenähert werden. Ein Allpass mit gerader Filter-Ordnung und geeignete gewählten Koeffizienten (Abbildung 7.3) liefert eine gute Näherung. Je nach gewählter Ordnung des Filters ist der Rechenaufwand aber in ähnlicher Größenordnung wie mit DFT und IDFT.

Zustandsmaschine

Um den Rechenaufwand der Hilbert-Transformation zu umgehen, lösen wir die Ereigniserkennung mittels einer Finite State Machine (FSM). Das Zustandsdiagramm der Finite State Machine in Abbildung 7.4 zeigt alle möglichen Zustände der FSM und welche Ereignisse einen Übergang in einen anderen Zustand auslösen.

Ereignis-FSM an auf neue Stati

Konfiguration der Zustandsmaschine Über Parameter wird definiert, welche Signalform als Ereignis erkannt werden soll. In Abbildung 7.5 sind die Parameter dargestellt.

Nullpegel Der Nullpegel kann angepasst werden, um die Erdanziehung, die als Beschleunigung auf den Sensor wirkt, zu kompensieren. Je nachdem wie der Sensor orientiert ist, ist die Erdanziehungskraft nicht parallel zur Mess-Achse des Sensors. Die vom Sensor gemessene Erdanziehungskraft in Richtung der Mess-Achse ist damit nicht immer gleich gross. Deshalb muss der Nullpegel angepasst werden können. Die mittlere gestrichelte schwarze Linie in Abbildung 7.5 stellt den Nullpegel dar.

Threshold Der Threshold definiert, ab welcher Abweichung des Signalpegels vom Nullpegel die FSM einen Messwert als 'hoch' betrachten soll. Zu beachten ist, dass der Threshold auf beide Seiten des Nullpegels gilt. Da der Sensor sowohl Beschleunigungen nach oben wie auch nach unten erfährt, unterscheidet die FSM dies mit den Ereignissen 'hoch_positiv' resp. 'hoch_negativ'. Signalpegel, die den Threshold nicht überschreiten, werden als 'niedrig' eingestuft. In Abbildung 7.5 ist das erste Erreichen des (negativen) Thresholds mit einer vertikalen roten Linie markiert.

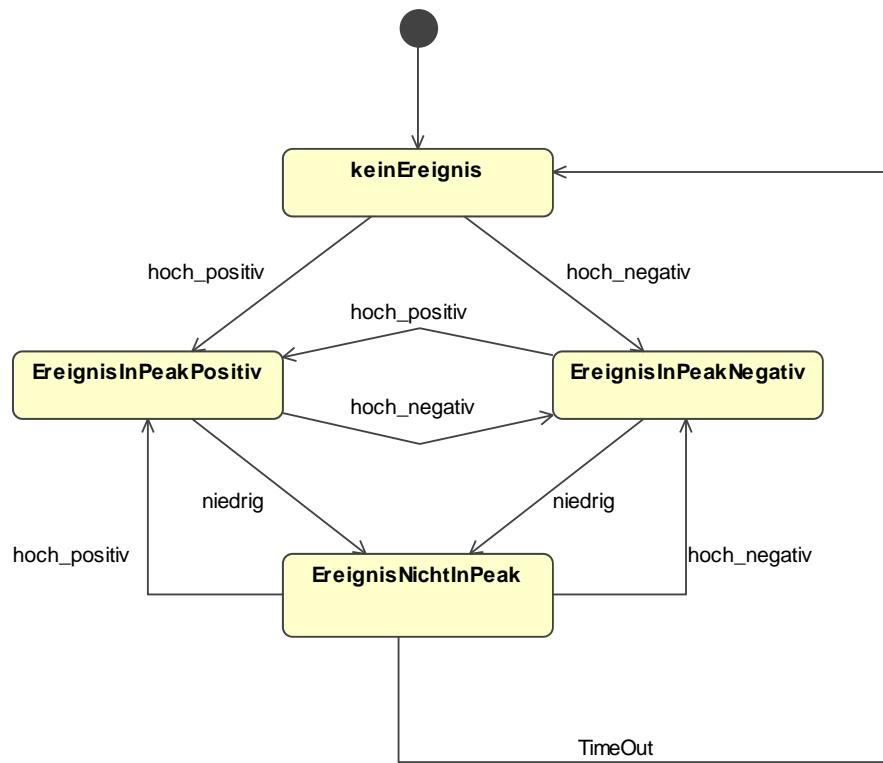


Abbildung 7.4.: Zustandsmaschine der Ereigniserkennung.

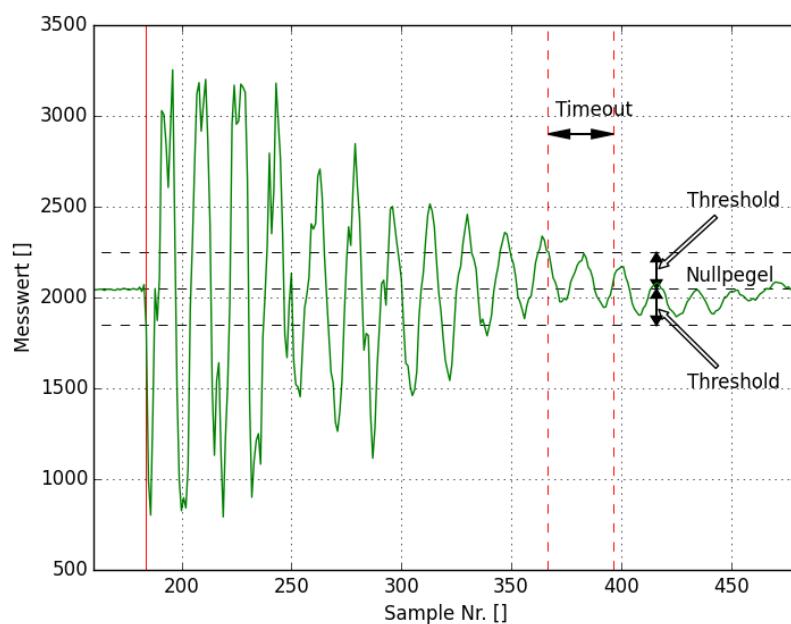


Abbildung 7.5.: Parameter der Ereigniserkennung.

Timeout Da ein Ereignis aus mehr als einem Peak besteht, muss eine Dauer (Timeout) definiert werden können, während der die FSM auf den Beginn eines neuen Peaks wartet. Tritt während des Timeouts kein neuer Peak auf, gilt das Ereignis als beendet. Die gestrichelten roten Linien in Abbildung 7.5 zeigen den Timeout.

Ablauf der Zustandsmaschine

Im Folgenden wird der Ablauf in der Zustandsmaschine genauer erklärt. Im Zustandsdiagramm in Abbildung 7.4 sind die Namen der Zustände und Ereignisse ersichtlich. Der Übersichtlichkeit halber wurde auf die Auflistung der Aktionen im Diagramm verzichtet.

Die FSM wird im Zustand 'keinEreignis' initialisiert. Tritt ein Messwert auf, der als 'hoch_positiv' klassiert wird, wechselt die FSM in den Zustand 'EreignisInPeakPositiv'. In diesem Zustand verbleibt die FSM, bis ein anders klassifizierter Messwert eintrifft.

Ein Messwert 'niedrig', also unterhalb des Thresholds, führt zu einem Übergang in den Zustand 'Ereignis_NichtInPeak'. Dieser Übergang startet einen Timer, der während der im Parameter 'Timeout' definierten Anzahl Messwerte läuft. Falls die FSM bis zum Ablauf des Timers keinen Messwert 'hoch_positiv' oder 'hoch_negativ' erhält, wechselt sie wieder in den Zustand 'keinEreignis' und übergibt die Ereignisdaten dem Prozess, der für die Übertragung zum Datenlogger zuständig ist.

Der Timer läuft nicht in Echtzeit, sondern zählt die Anzahl Messwerte seit seinem Start, da die Verarbeitung asynchron zur Erfassung der Messwerte läuft. Das bedeutet, dass die Verarbeitung problemlos während mehreren Messwerten stillstehen kann, ohne dass Messwerte verloren gehen. Dies ist möglich, da die Messwerte in eine Warteschlange (FIFO-Queue) geschrieben werden, von wo sie von der FSM abgeholt werden. So lange die FIFO-Queue nicht überfüllt wird, gehen keine Messwerte verloren. Die Verarbeitung der Messwerte in der FSM erfolgt im *NXP LPC4088* Mikrokontroller schnell genug, um theoretisch mit einer Abtastrate bis 200 kHz messen zu können.

Falls die FIFO-Queue doch einmal überlaufen sollte, wird eine Nachricht an den Datenlogger übertragen, der eine entsprechende Meldung in die Datendatei der Sensoreinheit einträgt. Dies bedeutet, dass zu diesem Zeitpunkt einige Messwerte verloren gegangen sind. Da der Timestamp in der Sensoreinheit trotzdem mit jedem Sample erhöht wird, stimmen die Zeitangaben in der Datendatei weiterhin. Es besteht lediglich eine Lücke im Datensatz.

Zusammenhänge
Wandlung und Erkennung und
tragung beschreibt
zu ein Kommunikations-
diagramm, wo sy-
wo asynchron.

7.1.4. Detail-Level

Je nach Einsatz der Messstation variiert die benötigte unbeaufsichtigte Messdauer von einigen Tagen bis zu mehreren Monaten. Die zu speichernde Datenmenge muss an diese Dauer angepasst werden können. Für einen langen Einsatz werden von Vorteil weniger detaillierte Messdaten abgespeichert, während für einen eher kurzen Einsatz unter Umständen sogar unkomprimierte Daten abgespeichert werden können.

Für die Wahl einer geeigneten Datenrate stehen vier verschiedene Modi, sog. Detail-Level zur Verfügung.

Im 'raw'-Modus werden Rohdaten gespeichert (Abbildung 7.6). Dieser Modus soll hauptsächlich der Gewinnung von Kalibrierungsdaten dienen, da hier sehr viele Daten anfallen. Es wird empfohlen, nur wenige Sensoren in diesen Modus zu versetzen.

Im 'detailed'-Modus werden von jedem Ereignis alle Datenpunkte gespeichert (Abbildung 7.7). Die uninteressanten Datenpunkte zwischen den Ereignissen werden nicht übertragen und aufgezeichnet. Dieser Modus ist sowohl für kurze als auch für längere Messperioden interessant, da er nur interessante Daten aufzeichnet. Die Menge anfallender Daten hängt direkt von der Häufigkeit der Ereignisse ab.

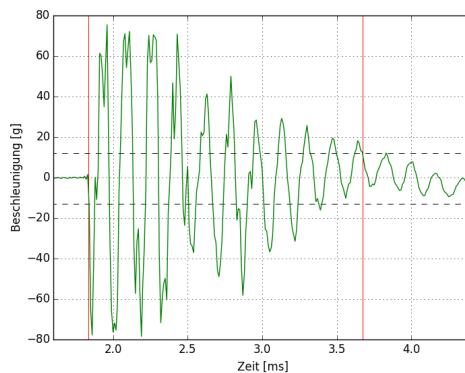


Abbildung 7.6.: Detail-Level 'raw'.

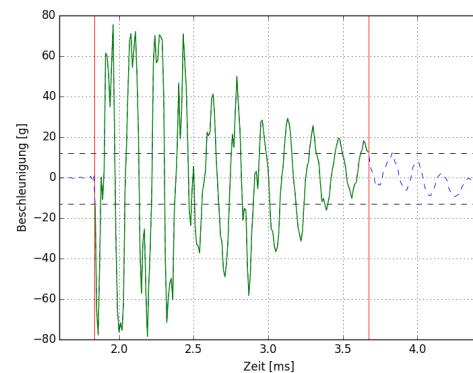


Abbildung 7.7.: Detail-Level 'detailed'.

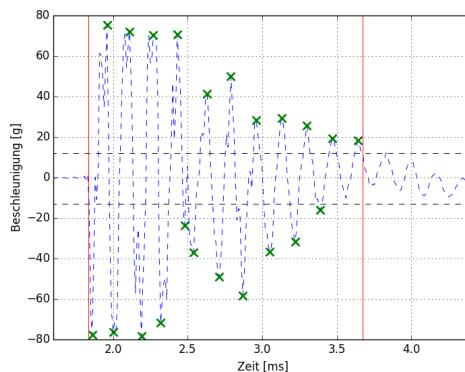


Abbildung 7.8.: Detail-Level 'peaks only'.

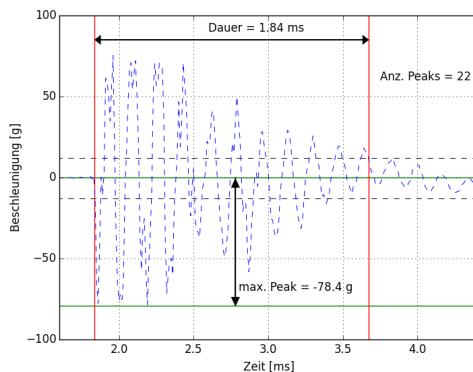


Abbildung 7.9.: Detail-Level 'sparse'.

Der 'peaks only'-Modus zeichnet die Dauer des Ereignisses sowie Zeitpunkt und Intensität aller Peaks eines Ereignisses auf. Damit ist ein grosser Teil der Information des 'detailed'-Modus vorhanden, aber in geringerer zeitlicher Auflösung.

Der 'sparse'-Modus liefert lediglich die Dauer des Ereignisses, die Anzahl Peaks und den maximalen Ausschlag. Dies ist eine minimale Informationsmenge, die dennoch eine Aussage über das Geschiebkon zulässt.

Eine ausführliche Beschreibung der Detail-Level ist in der Bedienungsanleitung ab Abschnitt ??, Seite ?? zu finden.

rechnungen, in Modus wie lan ssen werden d wie lange ein mit dem vor en Speicher die zwischenspei nn. Allenfalls em erwähnen, matisch zw erschiedenen n- und herschal . (Ist allerdings Wie viele Senso en in welchem gleichzeitig am betrieben wer welcher Ereig st Schluss mit breite.

7.1.5. Timestamp

Der Timestamp ist ein Zähler, der in jeder Sensoreinheit die gemessenen Samples zählt. Die übertragenen Ereignisse werden mit dem Timestamp versehen. Der Datenlogger kann anhand des Timestamps, der Abtastrate der betreffenden Sensoreinheit und des Zeitpunkts des letzten Zurücksetzens der Timestamps den Zeitpunkt der Aufnahme des Ereignisses berechnen (Gleichung 7.1).

$$t_{Ereignis} = t_{reset} + (timestamp/fs) \quad (7.1)$$

7.2. Busverwaltung

7.2.1. Verwaltung des Bussystems

TK: Busverwaltung schreiben

7.2.2. Busprotokoll

TK: Figur 7.10 r
chen

7.2.3. Filesystem

TK: Busprotokoll

TK: Kommunikationsdiagramm Bushandler

TK: Interrupt-Sy...
des Bushandlers
ren

Der Datenlogger legt die Messdaten für jede Sensoreinheit in eine eigene Datei ab. Die Filepointer werden zusammen mit den Konfigurationsdaten im Arbeitsspeicher gehalten. Beim Stoppen einer Sensoreinheit wird die Datei geschlossen und beim erneuten Start der Sensoreinheit eine neue Datei eröffnet. Beim Stopp/Start des Datenloggers erfolgt dies für alle Dateien.

Vor dem Entfernen der SD-Karte muss deshalb der Datenlogger gestoppt werden. Zu diesem Zweck gibt es im Konfigurationsmenü einen eigenen Befehl 'umount SD card'.

Die Dateinamen haben die Form 'sxx_MMDD_hhmm.dat', wobei 'xx' der CAN-ID entspricht, 'MMDD' dem Monat/Tag und 'hhmm' der Stunde/Minute des Starts.

Bei Konfigurationsänderungen schreibt der Datenlogger einen entsprechenden Eintrag in die Sensordatei. Der Eintrag wird auf eine neue Zeile geschrieben und mit dem Symbol '#' als Log-Eintrag markiert. Der Eintrag enthält die Uhrzeit und den neuen Wert des Parameters. Am '#' -Symbol erkennt ein Auswertungsprogramm eine solche Informationszeile und kann diese ignorieren oder bei Bedarf auch entsprechend verarbeiten.

7.3. Konfiguration

Die Konfiguration der Messstation erfolgt einerseits über die Datei 'config.txt', die sich auf der SD-Karte befindet. Andererseits steht über einen USB-Anschluss eine serielle Schnittstelle zur Verfügung. Mit einem Terminal-Emulator kann auf ein Konfigurationsmenü zugegriffen werden, um die Funktionen der Messstation zu steuern.

Die Verwendung der seriellen Schnittstelle über USB ist in Abschnitt 10.7.1 der Bedienungsanleitung (ab Seite 47) beschrieben.

Die komplette Beschreibung des Konfigurationsmenüs befindet sich in der Bedienungsanleitung ab Abschnitt 10.7.2, Seite 47.

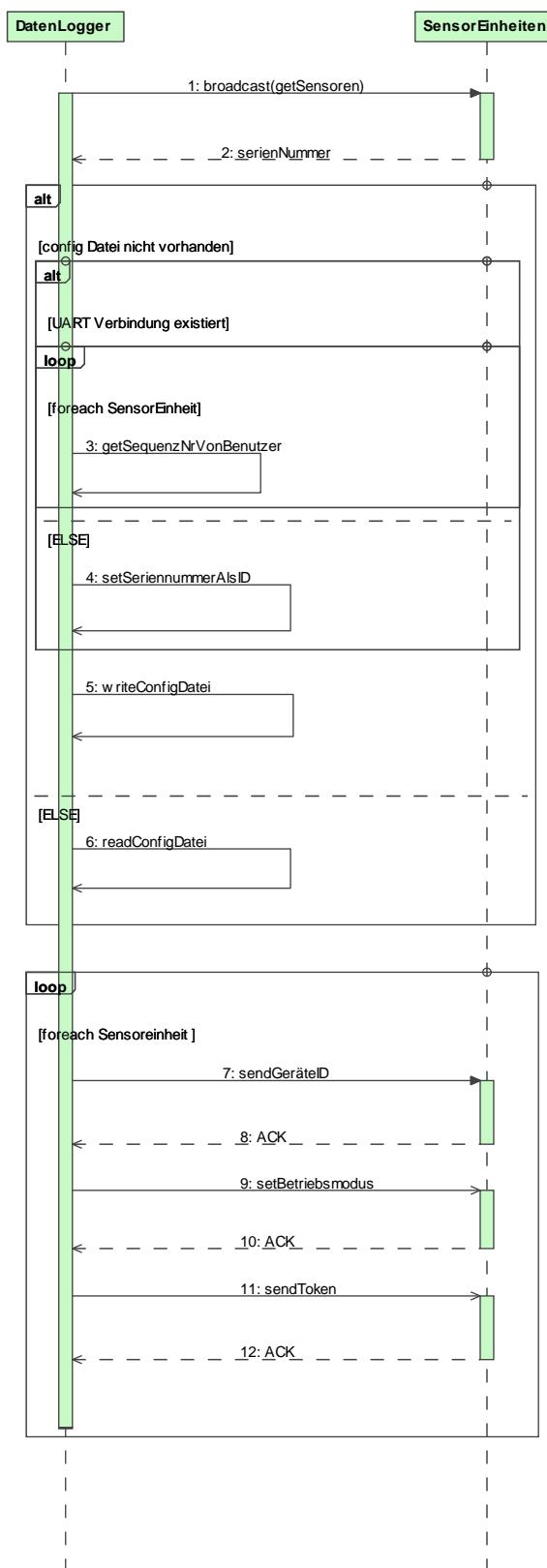


Abbildung 7.10.: Sequenzdiagramm des Startupvorgangs der Messstation.

8. Resultate

Resultate

läuft das zeug?,
le sensoren mags
den?, wie schnell
fs sein?, wie lang
man aufzeichnen
geht, was nicht?
haben wir nicht
reicht?, wo habe
mehr als geforde
reicht?

9. Diskussion

on
ir erfüllt?, wo
wierigkeiten?,
ind wir stolz,
nte man jetzt
och machen? ,
och geplant?

10. Bedienungsanleitung

10.1. Produktbeschrieb

Die Messstation wurde entwickelt, um Geschiebemessungen in einem Bach oder Fluss zu machen. Als Vorbild hat eine Messstation mit Geophone als Sensoren gedient, die einen Embedded-PC als Auswertungsrechner einsetzt. Der bauliche Aufwand für eine solche Messstation ist ziemlich gross, da viele Kabel verlegt und vor dem Geschiebe geschützt werden müssen. Mit dem Ziel, den Aufwand für die Konstruktion und die Verkabelung zu reduzieren, wurde eine neue Messstation entwickelt, die über ein Bussystem kommuniziert und die Messdaten gleich am Sensor auswertet. Auf diese Weise müssen nur noch die gewünschten, vorher spezifizierten Messdaten übertragen und gespeichert werden.

Mit der neuen Messstation können bis zu 20 Sensoren an einer Kontrolleinheit angeschlossen werden, die alle Messdaten aufzeichnet. Diese Anzahl kann nach eingehenden Tests wahrscheinlich noch erhöht werden. Die Obergrenze wird auch von der gewünschten Messdatenqualität abhängen, da das Bussystem eine begrenzte Übertragungskapazität hat.

10.2. Aufbau der Messstation

Die Messstation besteht aus einem Datenlogger der über ein Bussystem mit den Sensoreinheiten verbunden ist. Die Stromversorgung (12 V Gleichspannung) erfolgt über zwei zusätzliche Leitungen im gleichen Kabel wie für das Bussystem verwendet wird. Das Kabel verläuft vom Datenlogger zu Sensoreinheit zu Sensoreinheit.

Stromversorgung, Verdrahtung, Can-Bus, Terminator, R2D2, C3PO

bildchen des Auf
der Messstation

Die Sensoreinheiten sind in einem wasserdichten Gehäuse verbaut und über wasserdichte Steckverbinder (IP68) untereinander verbunden.

10.3. Datenlogger

Das Herzstück der Anlage ist der Datenlogger, der die Kontrolle über die Kommunikation hat, die Messdaten speichert und den Anschluss eines Computers für die Konfiguration der Anlage ermöglicht. Im Datenlogger arbeitet ein *ARM Cortex-M4* Prozessor mit 120 MHz Taktgeschwindigkeit. Dieser Prozessor steuert die Kommunikation und wandelt die Messdaten für die Speicherung in lesbare Information um.

10.3.1. Anschlüsse

Der Datenlogger verfügt über vier Anschlüsse.

figur über den lo

Stromversorgung

Ein Anschluss ist für die Stromversorgung. Hier werden 12V Gleichspannung angeschlossen. Der Stromverbrauch ist abhängig von der Anzahl angeschlossener Sensoren.

Verbrauch logger
sensor nennen

Busanschluss

Über den zweiten Anschluss wird das Kabel des Bussystems angeschlossen. Der Steckverbinder ist wassererdicht (IP68, Binder), damit kein Regen- oder Kondenswasser in den Datenlogger eindringen kann. Die Messstation verwendet CAN-Bus für die Kommunikation, da dieser Standard ein sehr robustes Protokoll für die Sicherstellung der korrekten und fehlerfreien Übertragung hat.

Schnittstelle zum Computer

Ein wassererdichter Steckverbinder für ein USB-Kabel erlaubt den Anschluss eines Computers oder auch einen Smartphones. Über eine serielle Schnittstelle kann dann die Messstation konfiguriert werden. Der Anschluss eines Computers und die Konfiguration ist in den Abschnitten 10.7.1 bis 10.7.3 ab Seite 47 im Detail beschrieben.

Speicherkarte

Die Speicherkarte kann über einen wassererdichten Schraubdeckel ausgewechselt werden, ohne dass das ganze Gehäuse des Datenloggers geöffnet werden muss. Der Umgang mit der Speicherkarte ist im Abschnitt 10.7.3 ab Seite 49 erklärt.

10.4. Sensor

Die Sensoreinheit der Anlage verfügen über einen Beschleunigungssensor und einen Mikroprozessor. Der Sensor misst über die Beschleunigung die Vibrationen, die vom Einschlag des Geschiebes verursacht werden. Der Mikroprozessor wertet die Messdaten aus und erkennt nach vorher definierten Kriterien die Ereignisse, deren Daten gesammelt werden sollen. Diese bereinigten Daten werden dann über das Bussystem an den Datenlogger übertragen.

10.4.1. Beschleunigungssensor

Der Beschleunigungssensor (*Analog Devices ADXL001-70*) misst Beschleunigungen zwischen -70 g und +70 g. Ein g entspricht der Beschleunigung durch die Erdanziehungskraft, ungefähr 10 m/s^2 . Sollten die Vibrationen stärkere Beschleunigungen erzeugen, kann entweder die Konstruktion der Messanlage angepasst werden um die Vibrationen schwächer zu machen, oder der Sensor ausgetauscht werden. Aus der Baureihe *ADXL001* von *Analog Devices* sind auch Sensoren mit Messbereichen von $\pm 250 \text{ g}$ und $\pm 500 \text{ g}$ erhältlich. Diese Modelle geben Messsignale im gleichen Spannungsbereich aus wie das verwendete Modell ADXL001-70 und können daher ohne Umkonfiguration oder Anpassungen an der Software verwendet werden.

10.4.2. Mikroprozessor

Der *ARM Cortex-M4* mit 120 MHz Taktfrequenz hat weit mehr Rechenleistung, als für die vorliegende Anwendung benötigt wird. Sollten komplexere Algorithmen für die Erkennung von Ereignissen, für die Filterung der Messdaten oder für die Verarbeitung der Signale benötigt werden, wäre dies dank der bereits vorhandenen Rechenleistung möglich.

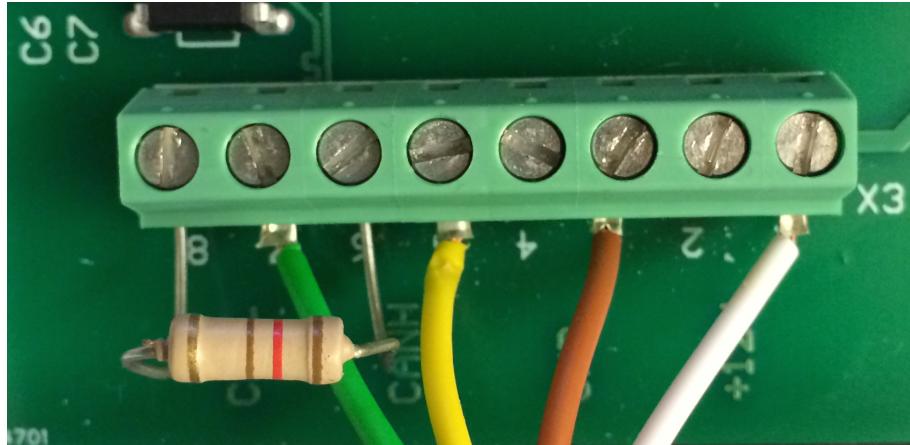


Abbildung 10.1.: Anbringung des Abschlusswiderstands.

10.4.3. Anschlüsse

Die Sensoreinheit hat zwei Anschlüsse für das Bussystem. Mit dieser Bauweise ist es möglich, die Stichleitung zum Transceiver möglichst kurz zu halten, um die Signalqualität möglichst wenig negativ zu beeinflussen. Die Anschlüsse sind wasserfest (IP68).

10.5. Bussystem

Als Bussystem wird CAN-Bus verwendet. Mit einer Datenübertragungsrate von 1 MBit/s bis zu einer Kabelgesamtlänge von 40 m, resp 125 kBit/s bis zu 500 m eignet sich CAN-Bus für diese Messstation sehr gut. Es sind keine Kabellängen zu erwarten, die weit über 40 m hinausgehen. CAN-Bus ist des Weiteren sehr robust gegenüber Umwelteinflüssen, die die Übertragung stören könnten.

10.5.1. Kabel

Die Kabel der Messstation verfügen über 4 Leitungen, die jeweils paarweise verdrillt sind. Zusätzlich zu den Leitungen ist eine Zugentlastung zum Schutz vor mechanischer Beschädigung im Kabel enthalten. Um die zwei Aderpaare und die Zugentlastung ist ein metallisches Geflecht als Schild angebracht, das sowohl gegen mechanische Beschädigung als auch gegen elektrische Störeinflüsse schützt. Die PVC-Hülle des Kabels ist noch ein zusätzlicher mechanischer Schutz.

Ein Aderpaar wird für die Spannungsversorgung verwendet, das zweite Aderpaar für die Kommunikation auf dem Bussystem.

10.5.2. Abschlusswiderstände

Damit die Bus-Signale an den Enden der Leitung nicht reflektiert werden und die Übertragung stören, müssen an beiden Enden des Bussystems Abschlusswiderstände eingesetzt werden. Die Abschlusswiderstände (120Ω) verbinden die Anschlüsse CANH und CANL auf der Leiterplatte und werden wie in Abbildung 10.1 angebracht.

Pol Nr.	Leitung	Anschluss Schraubklemme	Aderfarbe
1	Spannungsversorgung +12 V	+12V	weiss
2	Spannungsversorgung 0 V	GND	braun
3	unbenutzt		
4	CAN-Bus High	CANH	gelb
5	CAN-Bus Low	CANL	grün

Tabelle 10.1.: Steckerbestückung des Bussystems.

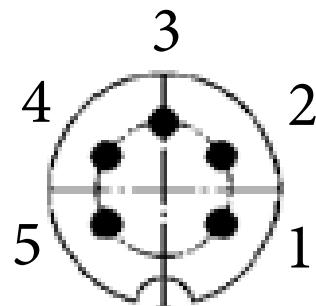
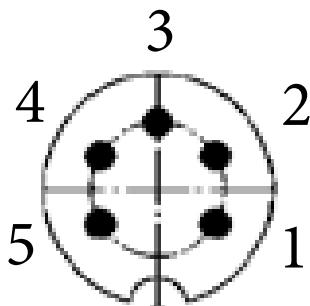


Abbildung 10.2.: Polnummerierung in der Buchse. Abbildung 10.3.: Polnummerierung im Stecker.

10.5.3. Steckverbinder

Die Steckverbinder sind fünfpolig und wasserdicht (IP68). Die Pole sind gemäss Tabelle 10.1 bestückt, die Polnummerierung in Stecker und Buchse sind in den Abbildungen 10.2 und 10.3 dargestellt.

Auf der Leiterplatte der Sensoreinheit ist eine Schraubklemmenleiste mit 8 Anschlüssen angebracht. Hier werden die von den Buchsen kommenden Leitungen angeschlossen. Jeweils zwei benachbarte Anschlüsse sind zusammengeschaltet, damit jede Leitung einzeln fest verschraubt werden kann (Abbildung 10.4). Es wird nicht empfohlen, zwei gleiche Leitungen in einer einzelnen Schraubklemme zusammen zu verschrauben.

10.6. Ereignis

Als Ereignis wird eine Signalform bezeichnet, die einem Einschlag eines Steins auf dem Sensor entspricht. Um Ereignisse zu erkennen, wird ein Threshold für den Signalpegel definiert. Wird dieser Wert überschritten, beginnt ein Ereignis. Sobald der Signalpegel für eine gewisse Zeit unterhalb des Thresholds geblieben ist, ist das Ereignis beendet. Das Verfahren der Ereigniserkennung wird im Abschnitt 7.1.3 ab Seite 30 im Detail erklärt.

Ein Beispiel eines Ereignisses ist in Abbildung 10.5 gegeben. Es handelt sich um den Einschlag eines Golfballs auf einer Aluminiumplatte, unter der der Beschleunigungssensor montiert ist. Die schwarz gestrichelten Geraden zeigen den Threshold, die roten Geraden zeigen den erkannten Anfang und Ende des Ereignisses. Das Ende des Ereignisses wird von der Ereigniserkennung auf jenen Messpunkt gelegt, an dem der Signalpegel den Threshold vor dem Timeout das letzte Mal unterschritten hat. Die Messwerte vor der dem Timeout werden je nach Detail-Level abgeschnitten. Die folgende Beschreibung der Detail-Level bezieht sich auf dieses Beispiel-Ereignis.

10.6.1. Detail-Level

Die Messstation kann Daten mit unterschiedlichen Detailgraden speichern. Jede Sensoreinheit kann individuell auf einen Detail-Level konfiguriert werden.

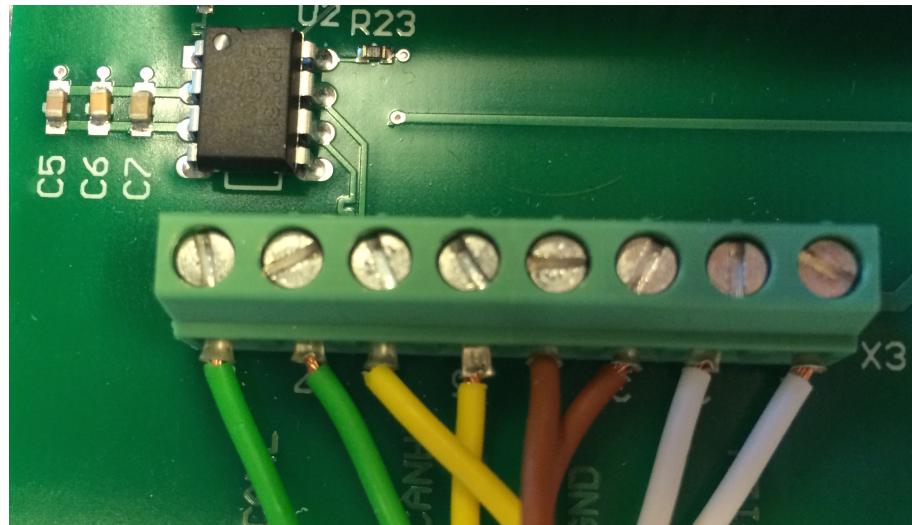


Abbildung 10.4.: Anschluss der Leitungen an den Schraubklemmen.

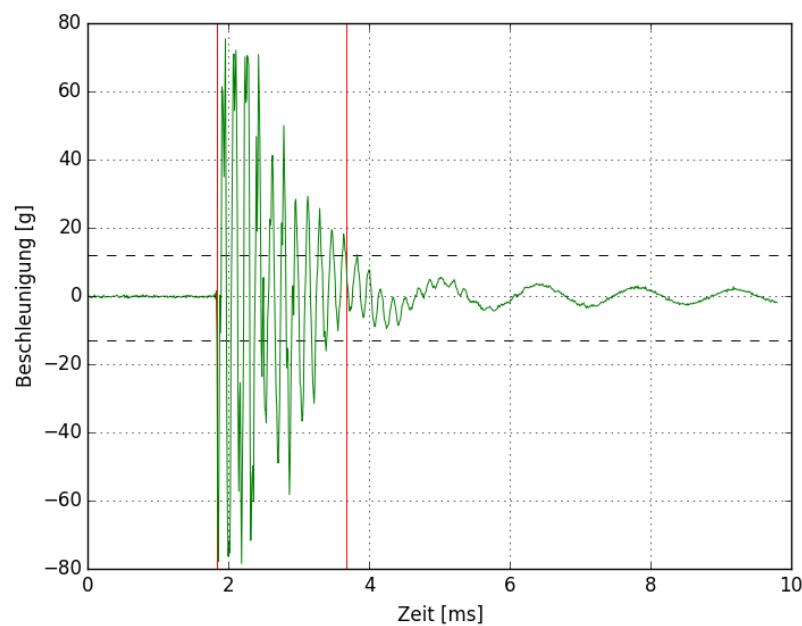


Abbildung 10.5.: Beispiel von Rohdaten.

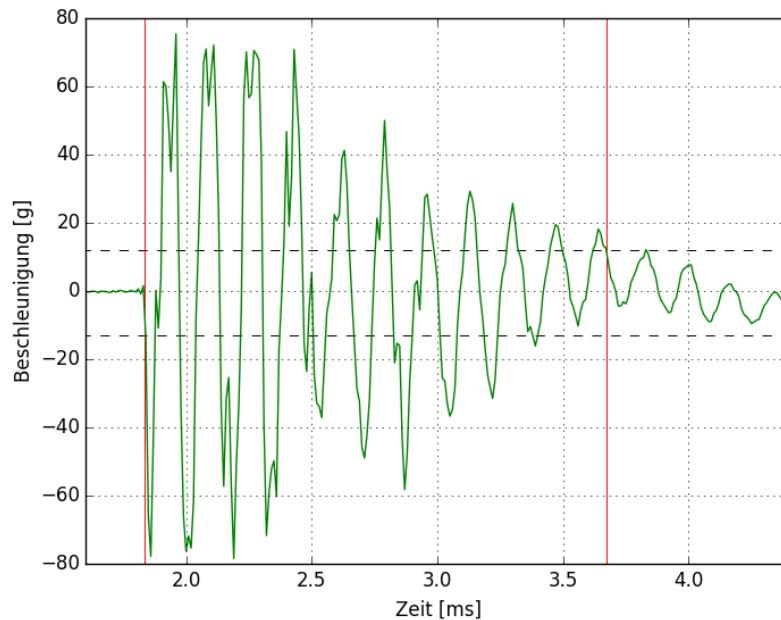


Abbildung 10.6.: Detail-Level 'raw'.

- raw
- detailed
- peaks only
- sparse
- off

Rohdaten (raw)

Im Modus 'raw' werden Rohdaten ohne Ereigniserkennung übertragen und gespeichert. Es resultiert eine lückenlose Erfassung aller Messpunkte. Abbildung 10.6 zeigt in Grün, welche Daten des Beispieleignisses im Modus 'raw' übertragen werden.

Datenaufkommen Das Datenaufkommen ist in diesem Modus am grössten. Für jedes Sample wird ein 8-bit Wert abgespeichert. Bei einer Abtastrate von 10000 Hz fallen also 10000 Byte Daten pro Sensor und Sekunde an.

aten einfügen

Beispieldaten

Detaillierte Ereignisdaten (detailed)

Die Ereigniserkennung sucht den Beginn und das Ende des Ereignisses. Es werden sämtliche Samples des Ereignisses übertragen. Im Konfigurationsmenü (Listing ??) wird dieser Modus als 'detailed' bezeichnet. In Abbildung 10.7 entspricht die grüne Kurve den gespeicherten Daten.

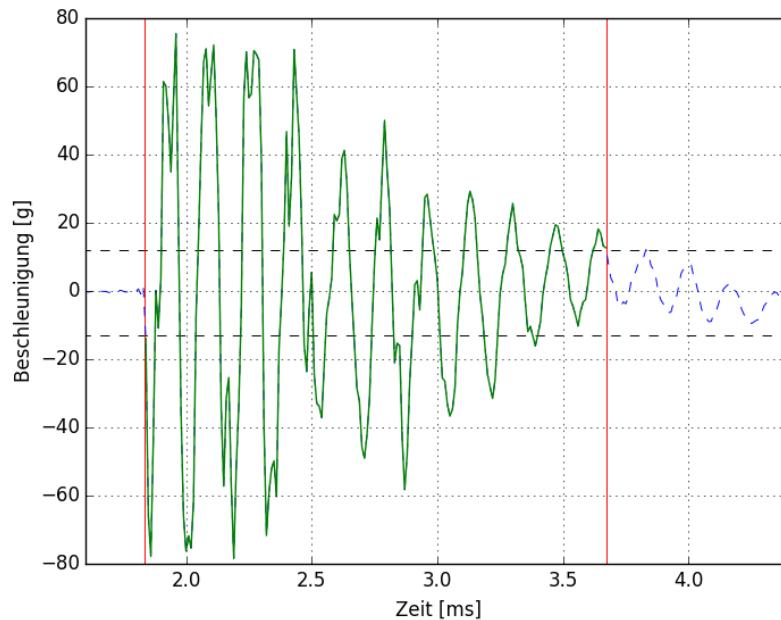


Abbildung 10.7.: Detail-Level 'detailed'.

Datenaufkommen Pro Sample wird ein Byte abgespeichert, der Startzeitpunkt und die Dauer belegen zusammen sechs Byte pro Ereignis. Die Datenmenge variiert mit der Dauer der Ereignisse. Es wird geschätzt, dass bei hohem Ereignisaufkommen während etwa 10 % der Messzeit ein Ereignis vorliegt. Somit wird die Datenrate ungefähr ein Zehntel der Abtastrate in Byte/Sekunde sein.

Beispieldaten

beispieldaten ein

Peaks (peaks only)

Die Ereigniserkennung sucht im Modus 'peaks only' den Beginn und das Ende und somit die Dauer des Ereignisses. Außerdem wird der maximale Ausschlag und alle Peakspitzen mit dem Timestamp und Höhe des Ausschlags gespeichert.

Datenaufkommen Für die Eckdaten (Beginn, Dauer, Anzahl Peaks, maximaler Ausschlag) fallen 8 Byte Daten an. Jeder Peak benötigt zwei Byte: ein Byte für die Anzahl Samples seit dem letzten Peak und ein Byte für die Höhe des Ausschlags. In der Abbildung 10.9 zeigt die blau gestrichelte Linie den Verlauf der Messkurve, die roten Geraden markieren Beginn und Ende des Ereignisses. Die grünen 'X' markieren die Peakspitzen, die übertragen werden.

Beispieldaten

beispieldaten ein

Minimale Daten (sparse)

Im Modus 'sparse' werden nur Eckdaten des Ereignisses gespeichert: Beginn und Dauer, Anzahl Peaks und maximaler Ausschlag. Dafür genügen acht Byte pro Ereignis. Die Eckdaten sind in Abbildung 10.9 dargestellt.

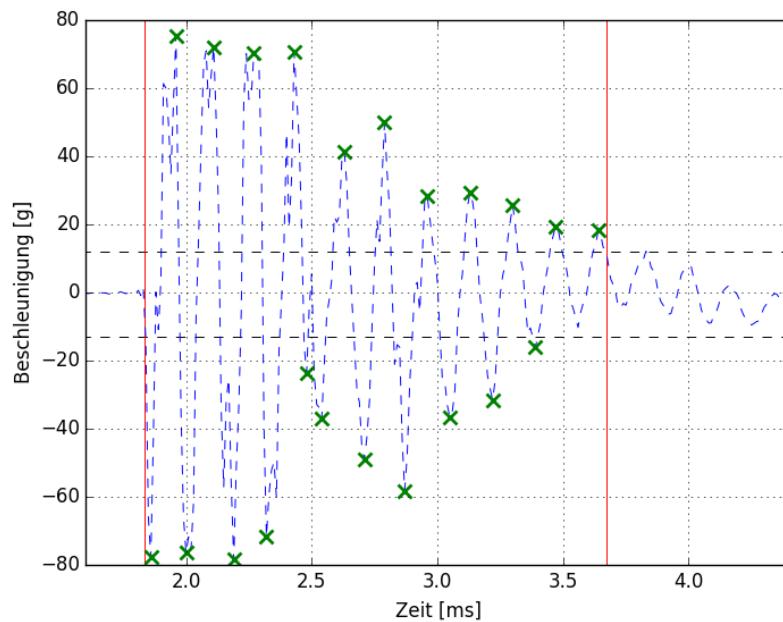


Abbildung 10.8.: Detail-Level 'peaks'.

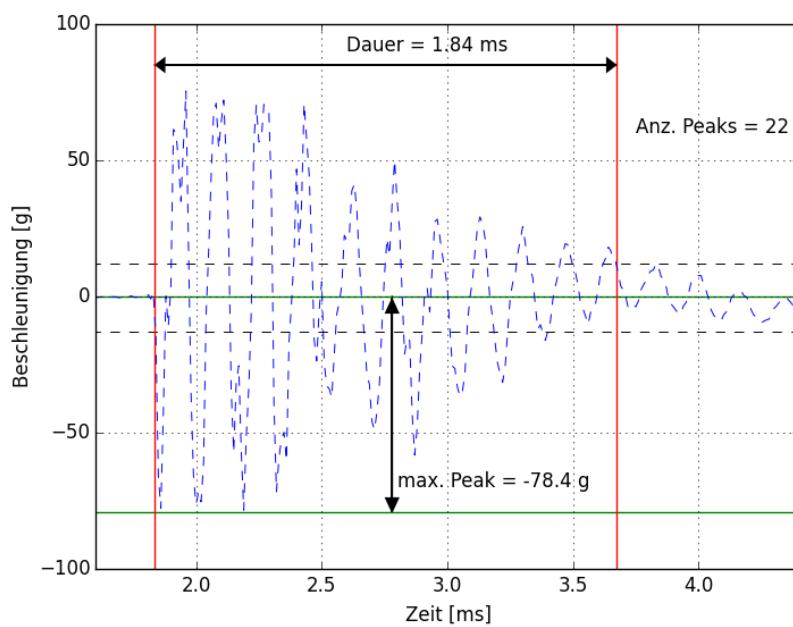


Abbildung 10.9.: Detail-Level 'sparse'.

Datenaufkommen Der Timestamp für den Beginn des Ereignisses benötigt vier Byte, die Dauer (Anzahl Samples) zwei Byte. Die Anzahl Peaks und der maximale Ausschlag belegen je ein Byte.

Beispieldaten

beispieldaten ein

Inaktiv (off)

Die Sensoreinheit kann in den Detail-Level 'off' gesetzt werden. In diesem Fall startet sie auch keine Messung, wenn der Datenlogger alle Sensoreinheiten startet.

In diesem Detail-Level fallen keine Daten an.

10.7. Konfiguration

10.7.1. Anschluss eines Computers

Am USB-Anschluss des Datenloggers kann ein Computer angeschlossen werden, um auf die serielle Schnittstelle des Datenloggers zuzugreifen. Um die serielle Schnittstelle zu verwenden, wird ein Terminal-Emulator wie *PuTTY* oder *minicom* benötigt. Um mit *PuTTY* eine Verbindung aufzubauen, muss die Schnittstelle und die Übertragungsrate (Baud) angegeben werden. Die Übertragungsrate ist 9600 baud, die Schnittstelle kann variieren.

Windows Unter Windows erfolgt die Verbindung auf eine der COMx-Schnittstellen. Die Nummer der COM-Schnittstelle kann im Gerät-Manager herausgesucht werden, die Bezeichnung lautet 'mbed Serial Port (COMx)', wobei 'x' eine Nummer ist. In *PuTTY* muss nur 'COMx' angegeben werden.

Linux Unter Linux findet man die Schnittstellenbezeichnung mit dem Befehl 'ls /dev/ttyACM*' heraus, in *PuTTY* wird dann '/dev/ttyACMx' angegeben.

Mac OS X Unter Mac OS X lautet der Befehl 'ls /dev/tty.usbmodem*', der in einem Terminal eingegeben werden muss. Als Terminal-Emulator kann 'screen' verwendet werden. Auf Apple Mac Computern mit USB 3.0 kann es zu Schwierigkeiten mit der Verbindung kommen. Den Herstellern des Prozessorboards ist dies bekannt, sie arbeiten an einer Lösung.

Die Einstellungen für die serielle Schnittstelle sind normalerweise bereits korrekt gesetzt. Es werden 8 Datenbits verwendet, 1 Stopbit und keine Parität (parity).

Weitere Hilfe für die Verwendung eines Terminal-Emulators findet man unter <http://developer.mbed.org/handbook/Terminals>.

10.7.2. Menü

Beim Herstellen der Verbindung über einen Terminal-Emulator wird das Basis-Menü angezeigt. Durch Eingabe der Zahl wird der entsprechende Menü-Eintrag gewählt. Im Folgenden wird das gesamte Menü im Detail beschrieben.

Das Basis-Menü (siehe Listing 10.7.2) listet alle Überwachungs- und Konfigurations-Möglichkeiten auf.

```
1) list files
2) format SD card
3) mount SD card
4) unmount SD card
5) logger status
6) start/stop logging
7) sensor parameters
8) sensor states
9) reset timestamp
10) internal clock
11) config file
```

Dateien auflisten Mit dem Befehl 'list files' wird eine Liste aller Dateien auf der SD-Karte angezeigt. Die Liste enthält die Dateigröße sowie den Dateinamen, siehe Abschnitt 10.7.3.

SD-Karte formatieren Um die SD-Karte für den ersten Gebrauch vorzubereiten, sollte sie formatiert werden. Dies erfolgt von Vorteil auf einem Computer, kann aber auch im Datenlogger mit dem Befehl 'format SD card' gemacht werden, siehe Abschnitt 10.7.3.

SD-Karte anmelden Nach dem Einsetzen einer SD-Karte erkennt der Datenlogger dies normalerweise automatisch. Es kann jedoch vorkommen, dass der Datenlogger auf die neue Karte aufmerksam gemacht werden muss. Dies erfolgt mit dem Befehl 'mount SD card', siehe Abschnitt 10.7.3.

SD-Karte abmelden Vor dem Entfernen der SD-Karte müssen alle Dateien geschlossen werden. Dies erfolgt mit dem Befehl 'unmount SD card', siehe Abschnitt 10.7.3.

Status des Datenloggers Mit dem Befehl 'logger status' werden einige Betriebszustandsdaten des Datenloggers angezeigt, siehe Abschnitt 10.7.3.

Aufzeichnung starten/stoppen Um die Aufzeichnung im ganzen System zu starten oder zu stoppen wird der Befehl 'start/stop logging' verwendet, siehe Abschnitt 10.7.3.

Sensor-Einstellungen Mit dem Befehl 'sensor parameters' kann eine einzelne Sensoreinheit oder alle Sensoreinheiten zusammen konfiguriert werden. Siehe Abschnitt 10.7.3.

Status der Sensoreinheiten Der Betriebszustand aller angeschlossenen Sensoreinheiten kann mit dem Befehl 'sensor state' (siehe 10.7.3) aufgelistet werden.

Timestamp zurücksetzen Um den Timestamp in allen Sensoreinheiten auf Null zurückzustellen, wird der Befehl 'reset timestamp' verwendet. Siehe Abschnitt 10.7.3.

Interne Uhr Die interne Uhr wird mit dem Befehl 'internal clock' eingestellt, Abschnitt 10.7.3 beschreibt dies im Detail.

Konfigurations-Datei Mit dem Befehl 'config file' wird die Konfiguration der Sensoren abgespeichert oder aus einer Datei eingelesen, siehe Abschnitt 10.7.3.

10.7.3. Befehle

dateiliste einfüge

Dateiliste

```
1 HIER LISTE DER FILES EINFÜGEN
2 0) exit
```

SD-Karte formatieren

```
1 1) confirm formatting of SD card.
2 All data will be erased.
3 0) cancel
```

Listing 10.1: Untermenü SD-Karte formatieren

Beim Formattieren werden alle Dateien auf der SD-Karte gelöscht, inklusive der Konfigurationsdatei mit allen Sensor-Einstellungen. Der Befehl 'format SD card' holt vor der Ausführung nochmals eine Bestätigung ein, ob sich der Benutzer wirklich sicher ist, dass er alle Dateien löschen will (Listing 10.1). Während dem Formatieren wird die Meldung 10.2 angezeigt.

```
1 formatting SD Card
```

Listing 10.2: Statusmeldung SD formatieren

Sind beim Formatieren Fehler aufgetreten, erhält man die Fehlermeldung 10.3. In diesem Fall sollte die Karte in einem Computer geprüft und formatiert werden.

```
1 Formatting SD card FAILED. Please use a Computer to format the card.
```

Listing 10.3: Fehlermeldung SD formatieren

Bei erfolgreicher Formatierung wird die Meldung 10.4 ausgegeben.

```
1 Formatting done
2 Returning to base menu.
```

Listing 10.4: Erfolgsmeldung SD formatieren

SD-Karte anmelden

Damit Dateien auf die SD-Karte geschrieben werden können, muss sie vorher erkannt werden. Normalerweise geschieht dies, sobald die Karte eingesetzt wird. Wenn keine SD-Karte erkannt wird, wird dies im Basismenü angezeigt wie im Listing 10.5. Durch den Aufruf des Befehls 'mount SD card' im Basismenü (Listing 10.7.2) kann die eingesetzte SD-Karte angemeldet werden.

Nach erfolgreicher Anmeldung der SD-Karte wird das Basismenü angezeigt.

Wenn keine SD-Karte erkannt werden kann, wird eine Fehlermeldung 10.5 ausgegeben.

```
1 No SD card detected! Please insert card and try again!
```

Listing 10.5: Fehlermeldung SD-Karte anmelden

SD-Karte abmelden

Bevor die SD-Karte aus dem Datenlogger entfernt wird, sollte sie abgemeldet werden. Der Datenlogger schliesst bei diesem Vorgang alle geöffneten Dateien, um Datenverlust zu vermeiden. Da beim Abmelden der Karte die Aufzeichnung der Daten gestoppt wird, wird vorher eine Bestätigung verlangt (Listing 10.6).

```

1) unmount SD card
2) This will stop logging and close all data files.
3) cancel

```

Listing 10.6: Untermenü SD-Karte abmelden

Falls die Konfiguration der Sensoren verändert, aber noch nicht in die Konfigurationsdatei geschrieben wurde, wird eine Warnung angezeigt (Listing 10.7). Die Konfiguration bleibt im Speicher des Datenloggers erhalten, so lange die Spannungsversorgung angeschlossen ist und kann auch auf der neuen SD-Karte gespeichert werden.

```

*****
* WARNING: sensor configuration data has not been saved to file! *
* If you want to save config to file, cancel now. *
*****

```

Listing 10.7: Warnung vor SD-Karte abmelden bei ungespeicherter Konfiguration

Logger-Status

Mit dem Befehl 'logger status' können einige Information über den Datenlogger angezeigt werden.

```

1 Time: Sat Dec 6 18:00:00 2014
2 Started?: 1

```

Listing 10.8: Untermenü Logger-Status

Starten und stoppen der Aufzeichnung

Um die Datenspeicherung im Datenlogger zu unterbrechen oder wieder zu starten wird der Befehl 'start/stop logger' verwendet. Beim Aufruf des Befehls wird ein Untermenü gemäss Listing 10.9 oder Listing 10.10 angezeigt.

```

1 Logger is running.
2 1) stop the logging .
3 0) cancel

```

Listing 10.9: Untermenü Stoppen der Aufzeichnung

Da sich das Menü dem gegenwärtigen Zustand anpasst, sieht es bei gestoppter Aufzeichnung aus wie in Listing 10.10.

```

1 Logger is stopped.
2 1) start the logging .
3 0) cancel

```

Listing 10.10: Untermenü Starten der Aufzeichnung

Wenn die Aufzeichnung am Logger gestoppt wird, wird an alle Sensoreinheiten der Befehl zum Aufzeichnungsstopp gesendet. Die Einstellungen zum Detailmodus bleiben in der Sensoreinheit aber erhalten. Beim erneuten Starten der Aufzeichnung im Logger werden auch die Sensoreinheiten wieder gestartet. Es besteht auch die Möglichkeit, einzelne Sensoreinheiten zu stoppen (siehe Abschnitt 10.7.3). Eine gestoppte Sensoreinheit bleibt auch beim Starten der Aufzeichnung am Datenlogger gestoppt, da sie schon vor dem Stoppen in diesem Zustand war.

Sensor-Parameter

Die Parameter der Datenerfassung und Ereigniserkennung können für alle Sensoreinheiten gemeinsam oder für jede Sensoreinheit individuell eingestellt werden. Die Auswahl einer einzelnen oder aller Sensorseinheiten erfolgt beim Einstieg in das Untermenü der Sensor-Parameter. Listing 10.11 zeigt die Auswahl der Sensoren. Die Auswahlliste enthält gleich die aktuellen Werte der Parameter, damit man eine Übersicht hat.

```

1  Nr      SID    serial          fs threshold baseline timeout   detail
2  1)      2     461bfdf6    10000       200     2047      30      raw
3  2)      3     361509a5    10000       150     2000      20      peaks
4  ...
5  7)      8     1562a010    20000      400     2047      15      detailed
6
7  #) Select a sensor from the list.
8  99) Select all sensors.
9  0) cancel

```

Listing 10.11: Untermenü Sensor-Auswahl

Nach der Auswahl einer Sensoreinheit gelangt man zur Auswahl des anzupassenden Parameters, Listing 10.12. Ist ein einzelner Sensor ausgewählt, werden hier noch einmal die aktuellen Werte der Parameter angezeigt.

```

1  1) set sampling rate (current: 10000 Hz)
2  2) set threshold value (current: 200)
3  3) set baseline value (current: 2047)
4  4) set timeout (current: 30)
5  5) set detail level (current: peaks only)
6  6) start or stop recording (current: started)
7  0) exit

```

Listing 10.12: Untermenü Sensor-Parameter

Abtastrate Die Abtastrate legt fest, wie oft pro Sekunde ein Messwert vom Beschleunigungssensor eingelesen werden soll. Die Sensoreinheiten können in einem Bereich zwischen 100Hz und 200000Hz messen. Die Abtastrate kann in Schritten von 100Hz eingestellt werden, Listing 10.13.

Die Abtastrate hat einen wesentlichen Einfluss auf die zu übertragende und zu speichernde Datenmenge, die benötigte Rechenleistung. Davon wiederum hängt die Zeitspanne ab, wie lange die Messstation ohne Wartung betrieben werden kann. Es wird deshalb empfohlen, die Abtastrate nur so hoch einzustellen, wie es wirklich benötigt wird. Wie hoch dieser Wert ist, hängt stark von den geplanten Auswertungen ab.

```

1  #) Enter sampling rate in Hz. (multiple of 100 Hz in range
   100..200'000 Hz)
2  0) cancel

```

Listing 10.13: Untermenü Abtastrate

Bei Eingabe einer ungültigen oder nicht unterstützten Abtastrate wird eine Fehlermeldung ähnlich Listing 10.14 angezeigt.

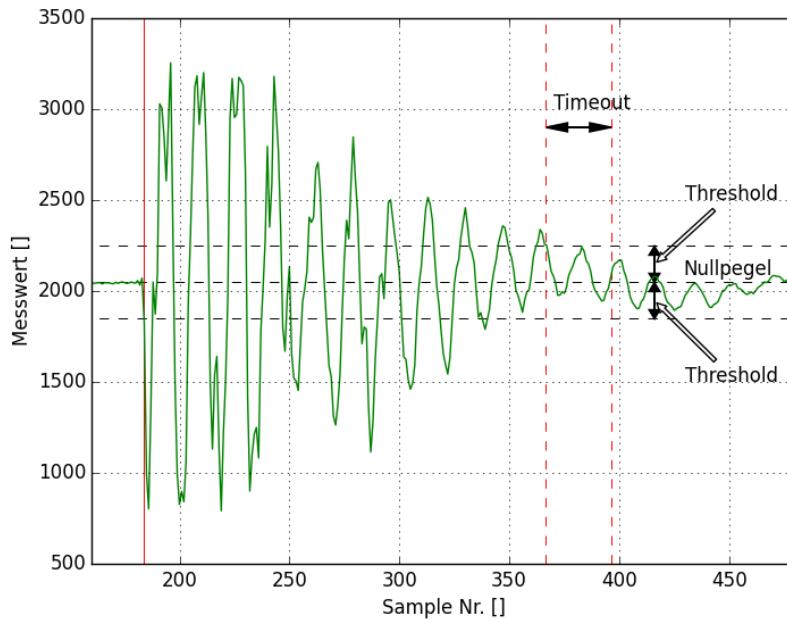


Abbildung 10.10.: Zusammenhänge der Parameter der Ereigniserkennung.

```
1 Sampling rate 220000 Hz not supported, too high.
```

Listing 10.14: Fehlermeldung bei ungültiger Abtastrate

Bei einer Änderung der Abtastrate wird automatisch der Timestamp zurückgesetzt, damit die Zuweisung der Messdaten zu einem Zeitpunkt weiterhin erfolgen kann.

Ereigniserkennung Die Ereigniserkennung hat drei Parameter, die die Form der gesuchten Ereignisse bestimmen. Abbildung 10.10 illustriert die Zusammenhänge zwischen Threshold und Nullpegel sowie die Funktionsweise desTimeouts.

Threshold Der Threshold (Schwellenwert) ist ein Parameter der Ereigniserkennung. Er bestimmt, ab welcher Abweichung vom Nullwert ein Signal als Peak betrachtet werden soll. Bei der Wahl des Thresholds ist zu beachten, dass der Threshold auf beide Seiten des Nullwerts gilt. Daher darf die Summe des Nullpegels und des Thresholds nicht den maximalen Wert (4096) des A/D-Wandlers überschreiten. Ebenso muss der Wert des Thresholds kleiner sein als der Nullpegel, damit kein negativer Messwert anliegen müsste, um einen Peak zu erzeugen. Diese Einschränkungen werden bei der Eingabe noch einmal angezeigt, Listing 10.15.

```
1 #) Enter threshold value.
2   baseline + threshold must not exceed 4096
3   and
4   baseline - threshold must not be below 0
5  0) cancel
```

Listing 10.15: Untermenü Threshold

Bei Verletzung der Kriterien für den Threshold wird eine entsprechende Fehlermeldung angezeigt, Listing 10.16. Da die gleichen Kriterien auch bei der Einstellung des Nullpegels gelten, empfiehlt es sich, zuerst einen kleinen Wert für den Threshold zu wählen. Dann kann der Nullpegel ohne grosse Einschränkung eingestellt werden. Danach setzt man den passenden Threshold.

```

1 Invalid threshold value:
2 threshold + baseline must not exceed 4096
3 and
4 threshold must be smaller than baseline value.

```

Listing 10.16: Fehlermeldung ungültiger Threshold

Nullpegel Der Nullpegel wird mit einer ähnlichen Maske (Listing 10.17) wie der Threshold eingestellt, auch die Einschränkungen für den Wertebereich sind die Gleichen.

```

1 #) Enter baseline value (default: 2047).
2 0) cancel

```

Listing 10.17: Untermenü Null-Level

Die Fehlermeldung bei ungültigen Werten für den Nullpegel ist in Listing 10.18 aufgeführt.

```

1 Invalid baseline value:
2 threshold + baseline must not exceed 4096.
3 and
4 threshold - baseline must not be below 0 value

```

Listing 10.18: Fehlermeldung ungültiger Nullpegel

Timeout Der Timeout definiert, wie viele Samples der Signalwert unterhalb des Thresholds liegen kann, bevor das Ereignis als beendet betrachtet wird (Listing 10.19). Die einzige Einschränkung an den Timeout ist, dass er die Länge des Ereignispuffers nicht überschreiten darf.

```

1 #) Enter timeout in samples.
2 0) cancel

```

Länge des Ereignisfegers

Listing 10.19: Untermenü Timeout

Bei zu langem (Listing 10.20) oder sehr kurzem (Listing 10.21) Timeout wird eine Fehlermeldung resp. Warnung angezeigt.

```
1 Timeout too long, can not exceed 512.
```

Listing 10.20: Fehlermeldung zu langer Timeout

```

1 Timeout 0 will end impact after each peak.
2 Timeout 0 in effect.

```

Listing 10.21: Warnung kurzer Timeout

Detaillevel Über die Wahl des Detaillevels wird bestimmt, wie viele und welche Daten von jedem Ereignis übertragen und gespeichert werden sollen (Listing 10.22). Die Detaillevel sind geordnet nach anfallender Datenmenge, beginnend mit dem grössten Aufwand. Die Detaillevel sind in Abschnitt 10.6, Seite 42 beschrieben.

```

1 1) raw (continuous data)
2 2) detailed (start time, all samples of impact)
3 3) peaks only (start time, nr of peaks, peaks
4 4) sparse (only start time, duration, nr of peaks, max peak)
5 5) off
6 0) cancel

```

Listing 10.22: Untermenü Detail-Level

Start/Stop Sensor Jede Sensoreinheit kann einzeln gestartet oder gestoppt werden, vorausgesetzt der Datenlogger ist gestartet. Im Untermenü ist ersichtlich, in welchem Zustand die ausgewählte Sensoreinheit gerade ist (listing 10.23). Falls die Konfigurationsänderung alle Sensoreinheiten betreffen soll, wird die Anzahl gestarteter und gestoppter Sensoren angezeigt (listing 10.24).

```
1 Selected sensor is currently stopped.
2 1) start
3 2) stop
4 0) cancel
```

Listing 10.23: Untermenü Start/Stop einzeln

```
1 Started sensors: 3
2 Stopped sensors: 0
3 1) start
4 2) stop
5 0) cancel
```

Listing 10.24: Untermenü Start/Stop alle Sensoren

Wenn ein Sensor gestartet wird, muss für die anfallenden Daten eine Datei erzeugt werden. Schlägt dies fehl, wird dies mit der Fehlermeldung 10.25 angezeigt. Der Sensor wird dann nicht gestartet. Es wird empfohlen, in diesem Fall die SD-Karte zu überprüfen. Möglicherweise verfügt die SD-Karte nicht mehr über genügend Speicherplatz.

```
1 Could not create or open file. Please check SD card for free space.
```

Listing 10.25: Fehlermeldung beim Starten eines Sensors

Sensor-Status

Um die Einstellungen aller Sensoren auf einen Blick zu überprüfen, kann mit dem Befehl 'sensor states' die Liste aller Einstellungen aufgerufen werden, Listing 10.26. In der Liste sind eine interne Nummer, die CAN-Bus-ID, die Seriennummer der Sensoreinheit sowie alle Einstellungen aufgeführt.

Zeige von Sensor-
Listing einfü-

```
1 Listing sensor config:
2 Nr SID serial      fs threshold baseline timeout      detail
3 1)   2 461bfdf6  10000      200    2047      30      raw
4 2)   3 361509a5  10000      150    2000      20 peaks only
5
6 0) continue
```

Listing 10.26: Untermenü Sensor-Status

Timestamp zurücksetzen

Der Timestamp kann manuell zurückgesetzt werden, dafür wird eine Bestätigung eingeholt: Listing 10.27. Das Zurücksetzen des Timestamps betrifft immer alle Sensoreinheiten.

```
1 1) re-synchronize timestamp
2 0) cancel
```

Listing 10.27: Untermenü Timestamp zurücksetzen

Interne Uhr

Anhand der internen Uhr werden die Timestamps vom Datenlogger in einen realen Zeitpunkt umgerechnet. Deshalb wird empfohlen, die interne Uhr auf die korrekte Uhrzeit einzustellen. Dies erfolgt über den Befehl 'internal clock', Listing 10.28. Hier kann das Datum (Untermenü 10.29) und die Tageszeit (Untermenü 10.30) eingestellt werden.

```

1 1) adjust date
2 2) adjust time
3 0) exit
4
5 current time: Sat Dec 6 18:00:00 2014

```

Beispiel einer Uhr einfügen

Listing 10.28: Untermenü interne Uhr

Das Einstellung des Datums erfolgt mit Inkrementen resp. Dekrementen von 365, 30, 10 Tagen oder 1 Tag.

```

1 adjust internal date
2 1) adjust date +365 days
3 2) adjust date -365 days
4 3) adjust date + 30 days
5 4) adjust date - 30 days
6 5) adjust date + 10 days
7 6) adjust date - 10 days
8 7) adjust date + 1 day
9 8) adjust date - 1 day
10 0) exit
11
12 current time: Sat Dec 6 18:00:00 2014

```

Listing 10.29: Untermenü Datum einstellen

Die Uhrzeit wird mittels Inkrementen resp. Dekrementen von 1 Stunde, 10 oder 1 Minute oder 1 Sekunde eingestellt. Die aktuelle Uhrzeit wird jeweils unterhalb des Menüs angezeigt.

```

1 adjust internal time
2 1) adjust time +1 hour
3 2) adjust time -1 hour
4 3) adjust time +10 minute
5 4) adjust time -10 minute
6 5) adjust time +1 minute
7 6) adjust time -1 minute
8 7) adjust time +1 second
9 8) adjust time -1 second
10 0) exit
11
12 current time: Sat Dec 6 18:00:00 2014

```

Listing 10.30: Untermenü Uhrzeit einstellen

Konfigurations-Datei

Über den Befehl 'config file' (Listing 10.33) kann die aktuelle Konfiguration der Sensoreinheiten in die Konfigurationsdatei 'config.txt' abgespeichert werden, oder eine neue Konfiguration aus der Datei eingelesen und an die Sensoreinheiten gesendet werden. Es ist nicht möglich, aus mehreren Konfigurationsdateien auszuwählen.

mehrere config-file löschen können in den Ausblick kommen, als zu verbreite Punkte

```

1) read configuration from file and set up all sensors accordingly.
2) store current configuration in file. Old config file will be
   overwritten.
3) cancel

```

Listing 10.31: Untermenü Konfigurationsdatei

Falls die Konfigurationsdatei nicht gespeichert werden kann, wird die Fehlermeldung in Listing 10.32 angezeigt. Um die Konfigurationsdatei trotzdem speichern zu können, kann die SD-Karte abgemeldet und ausgetauscht werden. Die Konfiguration bleibt erhalten, so lange die Spannungsversorgung besteht.

Um einem möglichen Konfigurationsdatenverlust vorzubeugen empfiehlt es sich, die Konfigurationsdatei zu speichern, sobald alle Einstellungen wie gewünscht gemacht wurden. Nach einem Unterbruch in der Spannungsversorgung wird nach automatisch die Konfiguration aus der

```

1 The config file could not be written. Please check the SD card in a
   computer.
2 The configuration data will remain stored in the logger unless you
   turn off power.

```

Listing 10.32: Fehlermeldung beim Speichern der Konfigurationsdatei

10.7.4. Konfigurationsdatei

In der Konfigurationsdatei werden die Einstellungen der Sensoreinheiten gespeichert. Die Datei kann auch auf einem Computer erstellt werden und über eine SD-Karte auf den Datenlogger übertragen werden. Ein Beispiel einer Konfigurationsdatei zeigt Listing 10.33. Die erste Zeile enthält die Anzahl Datensätze. Jeder Datensatz enthält die Konfiguration einer Sensoreinheit. Auf der zweiten und den folgenden Zeilen folgen die Datensätze.

Ein Datensatz enthält die CAN-Bus-ID, die Seriennummer der Sensoreinheit als Hexadezimalzahl, die Abtastrate (ohne die hintersten zwei Nullen), den Threshold, den Nullpegel und den Timeout sowie den Detail-Level und ob der Sensor Daten aufzeichnen soll (1) oder nicht (0).

Der Datenlogger versucht nach dem Einschalten der Spannungsversorgung die Konfigurationsdatei von der SD-Karte zu lesen. Falls keine SD-Karte vorhanden ist, startet der Datenlogger den Betrieb nicht, da keine Messdaten gespeichert werden können. Falls die SD-Karte vorhanden ist, aber keine Konfigurationsdatei 'config.txt' enthält, startet der Betrieb der Messstation im Standardmodus.

Standardmodus Im Standardmodus arbeiten alle Sensoreinheiten mit einer Abtastrate von 10000Hz , Threshold 200, Nullpegel 2047, Timeout 30. Der Detail-Level wird auf 'peaks only' gesetzt, siehe Abschnitt 10.6, Seite 42. Alle Sensoreinheiten werden gestartet.

```

1 {3,
2 {2, 461bfdf6, 100, 200, 2047, 30, 3, 1},
3 {3, 15047b39, 100, 150, 2040, 30, 2, 1},
4 {4, b78d6dca, 100, 100, 2049, 30, 4, 1},
5 }

```

Listing 10.33: Beispiel einer Konfigurationsdatei

Falls Änderungen an der Konfiguration gemacht, aber nicht in die Konfigurationsdatei geschrieben wurden, erscheint im Basismenü die Information gemäß Listing 10.34.

```
1 Config has been modified but not saved to SD card.
```

Listing 10.34: Information bei ungespeicherter Konfiguration

10.8. Betrieb

10.9. Technische Daten

Cortex-M4 CAN-Bus/Transceiver Spannungen Stromverbrauch Kabel Stecker Sensoren SD-Karte g-Bereiche Messzeiten Puffergrößen logger und sensor anzahl sensoren (bandbreite/signalstärke/rechenleistungLogger)

11. Verzeichnisse

Literaturverzeichnis

- [1] CAN Specification 2.0, 09 1991.
- [2] SD Specifications, Physical Layer Simplified Specification, Version 4.10, o. J.
- [3] ARM Ltd. Cortex-M4 Processor, o. J.
<http://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php>, Stand: 7.12.2014.
- [4] Thomas Müller. Vorlesung Kommunikationstechnik 1, 2013. Kap. 2, S. 14.
- [5] NXP Semiconductors. LPC4088FBD208 Processor Datasheet, o. J.
http://www.nxp.com/products/microcontrollers/cortex_m4/lpc4000/LPC4088FBD208.html, Stand: 7.12.2014.
- [6] NXP Semiconductors. NXP LPC4088 QuickStart Board, o. J.
<http://www.nxp.com/demoboard/OM13063.html>, Stand: 7.12.2014.
- [7] o. V. Wikipedia: CompactFlash, 12 2014. <http://en.wikipedia.org/wiki/CompactFlash>, Stand: 8.12.2014.
- [8] o. V. Wikipedia: Hilbert-Transformation, 11 2014.
<https://de.wikipedia.org/wiki/Hilbert-Transformation>, Stand: 6.12.2014.
- [9] o. V. Wikipedia: USB, 12 2014. <http://en.wikipedia.org/wiki/USB>, Stand: 8.12.2014.
- [10] o. V. CAN vs. RS485. White paper, IXXAT Automation GmbH, o. J.
- [11] 'wdwd'. FIR Hilbert Transform Filter. Licensed under the Creative Commons Attribution-Share Alike 3.0 Unported., 3 2013. URL:
https://commons.wikimedia.org/wiki/File:FIR_Hilbert_Transform_Filter.svg License:
<https://creativecommons.org/licenses/by-sa/3.0/deed.en>.
- [12] Sigisbert Wyrsch. Vorlesung Digitale Signalverarbeitung 1, 2013. Kap. 3, DFT und FFT, S. 48.

Abbildungsverzeichnis

3.1. Eine Messstation mit einem Datenlogger, der mehrere Sensoreinheiten im Bach steuert.	11
6.1. Hardwarekonzept des Datenloggers.	18
6.2. Hardwarekonzept der Sensoreinheit.	18
6.3. Chipdiagramm der <i>ARM Cortex-M4</i> Architektur [3].	20
6.4. Differentielle Leitung [4].	23
6.5. Schematischer Hardware-Aufbau des Datenloggers.	26
6.6. Schematischer Hardware-Aufbau der Sensoreinheit.	27
7.1. Softwarestack des Datenloggers.	29
7.2. Softwarestack der Sensoreinheit.	29
7.3. Hilbert-Transformation als FIR-Filter [11].	31
7.4. Zustandsmaschine der Ereigniserkennung.	32
7.5. Parameter der Ereigniserkennung.	32
7.6. Detail-Level 'raw'.	34
7.7. Detail-Level 'detailed'.	34
7.8. Detail-Level 'peaks only'.	34
7.9. Detail-Level 'sparse'.	34
7.10. Sequenzdiagramm des Startupvorgangs der Messstation.	36
10.1. Anbringung des Abschlusswiderstands.	41
10.2. Polnummerierung in der Buchse.	42
10.3. Polnummerierung im Stecker.	42
10.4. Anschluss der Leitungen an den Schraubklemmen.	43
10.5. Beispiel von Rohdaten.	43
10.6. Detail-Level 'raw'.	44
10.7. Detail-Level 'detailed'.	45
10.8. Detail-Level 'peaks'.	46
10.9. Detail-Level 'sparse'.	46
10.10. Zusammenhänge der Parameter der Ereigniserkennung.	52
A.1. Hauptkomponenten des NXP LPC4088 QuickStart Boards von Embedded Artists.	IX
A.2. Pins des NXP LPC4088 QuickStart Boards von Embedded Artists.	IX

Tabellenverzeichnis

6.1. Fähigkeiten des NXP LPC4088 Prozessors [5].	21
6.2. Zusätzliche Fähigkeiten des NXP LPC4088 QuickStart Boards von <i>Embedded Artists</i> [6].	21
6.3. Entscheidungsmatrix für die Auswahl des Bussystems.	22
6.4. Entscheidungsmatrix zur Auswahl des Speichermediums [2, 7, 9].	25
10.1. Steckerbestückung des Bussystems.	42

Glossar

A/D-Wandler Analog/Digital-Wandler, (Englisch: ADC). Ein A/D-Wandler misst die Spannung, die an einem Pin anliegt und gibt einen digitalen Wert aus, der die Höhe der Spannung angibt. Bei der Umwandlung in einen digitalen Wert erfolgt eine Quantisierung. Je grösser die Bit-Breite des A/D-Wandlers ist, umso kleiner wird die Schrittgrösse von einem digitalen Wert zum nächst-höheren Wert. 17, 20, 21, 27, 49, *siehe* Quantisierung & Bit-Breite

Abschlusswiderstand Bei elektrischen Signalübertragungen treten am Ende der Leitung Signalreflexionen auf. Die Reflexionen pflanzen sich in der Leitung in entgegengesetzter Richtung fort und stören so die Übertragung. Um die Reflexionen zu verhindern, können Widerstände eingesetzt werden, die die elektrische Energie in Wärme umzuwandeln. Dadurch treten keine Reflexionen mehr auf. 38, 56

Abtastrate Definiert, in welchen zeitlichen Abständen ein Messwert erfasst werden soll. Üblicherweise wird dieser Wert in Hz oder s^{-1} angegeben. 14, 27, 31, 41, 42, 53

Betriebsmodus Ein Modus bestimmt die Verhaltensweise eines Systems. Je nach gewähltem Modus können mit den gleichen Eingaben und Befehlen andere Aktionen ausgeführt und andere Resultate ausgegeben werden. 11

Bit-Breite Die Bit-Breite gibt an, wie viele Bit für die Darstellung eines Wertes verwendet werden. Je grösser die Bit-Breite, desto mehr unterschiedliche Werte können dargestellt werden. Mit einer Bit-Breite von n können 2^n Werte dargestellt werden. 17, 20

Blockgrösse Anzahl Werte, die in einer DFT oder IDFT verrechnet wird. 27

Busmaster Ein Gerät, das die Kontrolle über ein Bussystem hat. Der Busmaster kann den anderen Busteilnehmern (Slaves) eine Genehmigung erteilen, Daten über das Bussystem zu übertragen. Der Busmaster hat aber jederzeit die Möglichkeit, einen Slave in der Übertragung zu unterbrechen. *siehe* Slave

Bussystem Ein elektrisches System für die Kommunikation zwischen mehreren Geräten. Ein Bussystem besteht aus Datenleitungen, über welche Signale gesendet werden, und aus Schnittstellen, an denen die Busteilnehmer angeschlossen werden. Die Besonderheit liegt darin, dass über ein Leitungssystem mehr als zwei Geräte miteinander kommunizieren können. Mittels eines Adressierungsschemas kann der/die Empfänger ausgewählt werden. 8, 10, 12–14, 17, 19–23, 36, 37, 39, 57, *siehe* Signal

Collision Detection Kollisionserkennung. Als Kollision bezeichnet man den gleichzeitigen Versuch mehrerer Busteilnehmer, eine Nachricht zu übermitteln. Dies führt zu einer Überlagerung der Nachrichten und macht diese unlesbar. Gleichzeitig mit dem Schreiben liest der Transceiver die Signale vom Bus. Stimmen die gelesenen Signale nicht mit den Geschriebenen überein, bedeutet dies, dass ein anderer Teilnehmer ebenfalls auf den Bus schreibt und eine Kollision vorliegt. Das Verhalten bei einer Kollision ist vom Bussystem abhängig. 23, *siehe* Transceiver

Computer Englisch für Elektronenrechner. Unter einem Computer versteht man umgangssprachlich einen Personal Computer (PC) oder einen tragbaren Computer (Laptop). Heute sind Computer so leistungsfähig und so stark miniaturisiert, dass sie mühelos in einer Aktentasche Platz finden. Weniger leistungsfähig, dafür noch kleiner sind Embedded Systems, eingebettete Systeme, die Laien nicht als Computer erkannt werden. 11, 36, 37, 43, 45, 46, 53, *siehe* PC & Embedded System

CRC Cyclic Redundancy Check. Ein Codeverfahren, das eine Prüfsumme über eine Nachricht berechnet. Durch Nachrechnen der Prüfsumme im Empfänger kann die Nachricht auf Fehlerfreiheit überprüft werden. 24

Daisy Chain Englisch für Gänseblümchenkette. Gemeint ist das aneinanderreihen mehrerer Geräte. Im Unterschied zu einem Bussystem müssen die Geräte alle Nachrichten, die für andere Empfänger bestimmt sind, aktiv weiterleiten. Dies braucht Rechenleistung in den Geräten. Wenn ein Gerät ausfällt, gehen alle Nachrichten in diesem Gerät verloren. 21, 22

Datenlogger Ein Gerät zur Sammlung und Speicherung von Messdaten von mehreren Sensoreinheiten. 8–14, 17–19, 24–26, 28, 31, 36, 37, 43, 45, 47, 50, 51, 53, 56, *siehe* Sensoreinheit

DSP Digitaler Signal-Prozessor. Ein Mikroprozessor oder ein Bestandteil eines solchen, der dank spezieller Hardware fähig ist, Multiplikationen und Additionen extrem schnell auszuführen. Außerdem verfügt ein DSP über Schieberegister, um Rechenoperationen über eine Serie der aktuellsten Messwerte auszuführen. Da bei der digitalen Signalverarbeitung oft für jeden Messwert viele Multiplikations- und Additions-Schritte ausgeführt werden müssen, ist es wichtig, dass Multiplikations-Operationen in einem Taktzyklus ausgeführt werden können. In einem normalen Prozessor werden für eine Multiplikation mehrere Taktzyklen benötigt. 17, 19, *siehe* Mikrocontroller

Embedded System Deutsch: eingebettetes System. Ein Computer, der nicht über die üblichen Ein- und Ausgabemöglichkeiten eines Computers wie Bildschirm und Tastatur verfügt. Oft werden deshalb Embedded Systems nicht als Computer wahrgenommen. Sie sind im Gegensatz zu PCs, die als Alleskönnner konzipiert sind, auf eine bestimmte Aufgabe zugeschnitten und deshalb nur mit den nötigen Bedien-Elementen versehen. Oft genügen für die Aufgaben weniger leistungsfähige Prozessoren als in einem PC, sogenannte Mikroprozessoren. *siehe* Computer & Mikrocontroller

Ereignis Eine Abfolge von Messwerten, die einer vordefinierten Form entspricht. Es kann zum Beispiel ein Schwellenwert (engl. threshold) definiert sein. Das Überschreiten dieses Wertes kann dann den Beginn, das Unterschreiten des Schwellenwertes das Ende eines Ereignisses markieren. 8–14, 19, 24, 27, 29, 39, 42, 43, 48, 50, 59

Ereigniserkennung Auswertung von Messdaten um definierte Signalformen (Ereignisse) zu erkennen. 27, 29, 39, 42, 48, 49, 56, *siehe* Ereignis & Signal

Finite State Machine Englisch für Zustandsmaschine. Eine Finite State Machine definiert eine endliche Anzahl Zustände, die die Maschine einnehmen kann. Die FSM reagiert auf Ereignisse, indem sie Aktionen auslöst und allenfalls in einen anderen Zustand wechselt. Für jeden Zustand ist definiert, welche möglichen Ereignisse welche Aktionen auslösen, und in welchen Folgezustand gewechselt werden soll. 29

FIR-Filter Finite Impulse Response, Englisch für Filter mit endlicher Impuls-Antwort. Ein FIR-Filter hat immer eine endliche Impulsantwort, d.h. auf einen kurzen Impuls am Eingang des Filters folgt am Ausgang eine Antwort des Filters, die garantiert endet. Die Ordnung des FIR-Filters gibt an, wie lange die Antwort dauern wird. 27, 29, 56

Flash-Speicher Nicht-flüchtiger Speicher. Der Inhalt des Speichers bleibt auch bestehen, wenn keine Versorgungsspannung anliegt. 20

FPU Floating Point Unit, ein Rechenwerk in der CPU, die Berechnungen mit Dezimalbrüchen sehr schnell ausführen kann. 19

Geophon Ein Messgerät für Vibrationen des Bodens. Ein Geophon misst Bewegungen mittels einer magnetischen Masse, die beweglich in einer Spule aufgehängt ist. Wird das Geophon in Bewegung versetzt, schwingt die magnetische Masse aufgrund ihrer Trägheit und induziert dadurch einen Strom in der Spule. Durch Messung dieses Stroms kann die Bewegung registriert werden. 7, 15, 36

Hardware Die Hardware ist das eigentliche Rechenwerk eines Computers, worauf die Software ausgeführt wird. Dazu gehören alle elektrischen, elektronischen und mechanischen Bauteile eines Computers. Die Central Processing Unit (CPU) eines Computers könnte mit dem Hirn verglichen werden, hier laufen fast sämtliche Informationen und Instruktionen zusammen. 17

Hilbert-Transformation Mathematische Umrechnung einer Wertfolge, um die umhüllende Kurve zu erhalten. 27, 29, 56

Interrupt Englisch für Unterbrechen. Durch ein spezielles Signal wird dem Prozessor mitgeteilt, dass ein Ereignis eingetreten ist. Der Prozessor unterbricht die laufende Funktion und ruft eine spezielle Routine, die ISR auf. Die ISR verarbeitet das Ereignis und gibt danach die Kontrolle an die vorher unterbrochene Funktion zurück. Mit einem Interrupt wird zwar die Abarbeitung einer Funktion kurzzeitig unterbrochen, dafür verliert der Prozessor keine Rechenzeit mit Polling. *siehe* Polling

Kommandozeile Eine Eingabeaufforderung auf dem Bildschirm, wo der Benutzer über eine Tastatur Befehle eingeben kann, die vom Computer interpretiert und ausgeführt werden. 11, *siehe* Computer

MAC Multiply-ACcumulate unit: Ein Rechenwerk in der CPU, wo Multiplikationen und Additionen ausgeführt werden. 19

MEMS Microelectromechanical System, Englisch für Mikroelektromechanisches System. Bezeichnung für Elektromechanische Systeme, die sehr stark miniaturisiert worden sind. MEMS-Geräte verwenden Bauteile, die nur wenige Mikrometer bis einen Millimeter gross [?]. 7

Mikrokontroller Englisch Microcontroller (MC). Ein Prozessor mit weniger universellen Fähigkeiten als eine CPU, dafür mit weniger Stromverbrauch. Ein Mikrokontroller verfügt meistens über spezielle Ein- und Ausgänge (Pins), über die zum Beispiel die anliegende Spannung gemessen oder eine bestimmte Spannung ausgegeben werden kann. Durch die kleinere Bauform und die geringere Leistungsaufnahme eignen sich diese Prozessoren besonders für den Einsatz in Embedded Systems, die oft längere Zeit unabhängig vom Stromnetz funktionieren müssen. 17, 21, *siehe* Pin

Nested Vectored Interrupt Controller Wird benötigt, um möglichst rasch auf mehrere asynchron eintreffende Ereignisse reagieren zu können. Ein Prozessor mit NVIC kann auf verschiedene Ereignisse reagieren, indem Interrupts ausgelöst werden. Jedem Interrupt kann eine Priorität zugewiesen werden, um festzulegen, ob ein Interrupt einen anderen unterbrechen darf, der gerade vom Prozessor abgearbeitet wird. 19, *siehe* Interrupt

Nullpegel Signalpegel, wenn keine Beschleunigung gemessen wird. 29, 48–50, 53

PC PC oder umgangssprachlich Computer. Eine elektronische Rechenmaschine mit der sehr viele Rechenschritte in sehr kurzer Zeit ausgeführt werden können. Mit der richtigen Programmierung (Software) können PCs sehr unterschiedliche, umfangreiche Aufgaben lösen. *siehe* Software

Peak Englisch für Spitzenwert oder Signalspitze. 12, 29

Pin Ein Anschluss an einem Chip oder einer Leiterplatte. Über Pins werden elektronische Bauteile miteinander verbunden und Peripheriegeräte wie z.B. Sensoren angeschlossen. 20, 27, *siehe* Sensor

Polling Englisch für Abfragen. Bezeichnet das wiederholte Abfragen einer Funktion oder eines Wertes, bis ein bestimmtes Ereignis auftritt. Der Prozessor ist mit der Abfrage beschäftigt und verliert dadurch Rechenzeit. 22

Quantisierung Einteilung eines Werts aus einer kontinuierlichen Skala in eine abgestufte Skala. Bei der Quantisierung wird der nächstgelegene Wert auf der abgestuften Skala ausgewählt. Je nach Anwendung wird die abgestufte Skala mit konstanter Stufengröße (linear) gewählt, oder mit unterschiedlichen Stufengrößen je nach absolutem Wert (z.B. exponentiell). Der Quantisierungsfehler entspricht der Differenz zwischen dem analogen und dem quantisierten, digitalen Wert. Je grösser die Bit-Breite der Quantisierung ist, desto kleiner wird der Quantisierungsfehler. *siehe* Bit-Breite

Sample Englisch für Messwert, Messpunkt. 41, 42

SDRAM Synchronous Dynamic RAM. Flüchtiger Arbeitsspeicher. Der Inhalt des Arbeitsspeichers muss regelmässig aufgefrischt werden, sonst geht die Information verloren. 20, *siehe*

Sensor Ein Messgerät für physikalische Größen wie Temperatur, Feuchtigkeit, Luftdruck oder Beschleunigung. 5, 6, 8, 9, 11–14, 17, 21, 24, 27, 31, 33, 36

Sensoreinheit Ein kombiniertes elektronisches Gerät zur Messung von physikalischen Daten und der Verarbeitung dieser Daten. Das Gerät verfügt über einen Mikroprozessor und einen Sensor. Optional kann auch eine Schnittstelle für die Kommunikation mit einem Datenlogger vorhanden sein. 10–15, 17–21, 25, 26, 28, 36–39, 41, 43, 45, 47, 48, 50–53, 56, *siehe* Mikrokontroller, Sensor & Datenlogger

serielle Schnittstelle Eine Schnittstelle, über die Daten bitweise übertragen werden, im Gegensatz zu paralleler Übertragung, wo auf mehreren Leitungen mehrere Bits gleichzeitig übertragen werden. 61

Signal Ein Signal ist ein Informationsträger. Die Information wird einem Signalwert zugeordnet. Ein einfaches Beispiel ist die Spannung am Ausgang eines Beschleunigungs-Sensors. Der Sensor gibt über die Höhe der Spannung an, wie stark die Beschleunigung von einem definierten Referenzwert abweicht. Die Spannung trägt also eine Information über den Messwert und ist daher ein Signal. Aus dem Datenblatt kann herausgelesen werden, wie hoch die ausgegebene Spannung bei einer bestimmten Beschleunigung ist. So kann das Signal in Information umgewandelt werden. 27, *siehe* Sensor

Slave Ein Busteilnehmer, der nur Daten über den Busübertragen darf, wenn ihm der Busmaster die Genehmigung dafür erteilt. Dies kann z.B. in Form eines sog. Tokens geschehen. *siehe* Busmaster

Software Programm zur Steuerung der Rechenabläufe in einem Computer. Durch die Software wird einem Computer die Fähigkeit gegeben, bestimmte Aufgaben zu lösen und Resultate in einer für Menschen lesbaren Form darzustellen. Die Software enthält Instruktionen, die in der CPU des Computers ausgeführt werden. 37

Speichermedium Ein Stück Hardware, auf dem Daten gespeichert werden können. 17

Stichleitung Abzweigungsleitung an einer Busleitung. Um das Signal vom Bus zum Transceiver zu führen, wird eine Stichleitung benötigt, die wie eine T-Kreuzung von der Busleitung wegführt. Je länger die Stichleitung ist, desto stärker leider die Signalqualität auf der Busleitung. 38, *siehe* Transceiver

Terminal-Emulator Ein Programm, das ein Terminal emuliert. Der Benutzer sieht ein Fenster mit einer Kommandozeile zur Befehlseingabe, das aussieht wie auf einem Terminal. 43, 44, *siehe* Terminal

Threshold Englisch für Schwellenwert. 12, 27, 29, 31, 39, 48–50, 53

Timeout Englisch für Zeitüberschreitung. 12, 29, 39, 48, 50, 53

Timestamp Englisch für Zeitmarke. Mittels eines Timestamps kann ein Ereignis oder ein Messwert einem genauen Zeitpunkt zugeordnet werden. Der Timestamp wird dafür zu einem bestimmten Zeitpunkt auf null gesetzt (Reset) und in allen Messgeräten in vordefiniertem Takt erhöht. Der Timestamp gibt an, wie viel Zeit seit dem Reset vergangen ist. Durch die Wahl des Takts wird die zeitliche Auflösung definiert. 12–14, 42, 43

Transceiver Ein Gerät das gleichzeitig Transmitter und Receiver, also Sender und Empfänger ist. Der Prozessor sendet die zu übertragenden Daten über eine serielle Schnittstelle an den Transceiver. Der Transceiver erzeugt dann die notwendigen elektrischen Signale auf dem Bus, um die Daten zu übertragen. Beim Empfangen einer Meldung liest der Transceiver die Signale vom Bus und übersetzt sie für die serielle Übertragung zum Prozessor. Ein Transceiver ist notwendig, wenn der Prozessor die vom Bus erwarteten elektrischen Signale nicht selbst erzeugen kann. 23, 38, *siehe* serielle Schnittstelle

Abkürzungen

ADC Analog Digital Converter 17, 58

CD Collision Detection 22

CD Compact Disc VII, XIII

CRC Cyclic Redundancy Check 24

DFT diskrete Fourier-Transformation 27, 58

DSP Digitaler Signal-Prozessor 19, 59

FPU Floating Point Unit 19

FSM Finite State Machine 29, 31, 59

ID Identifikationsnummer 11, 13

IDFT inverse diskrete Fourier-Transformation 27, 58

ISR Interrupt Service Routine 60

MAC multiply-accumulate unit 19

MC Mikrokontroller 22

NVIC Nested Vectored Interrupt Controller 17, 19

PC Personal Computer 13, 60

USB Universal Serial Bus 25, 37, 43, 44

WSL Eidg. Forschungsanstalt für Wald, Schnee und Landschaft 7, 27

Listings

7.1. Timer mit Aufruf der A/D-Wandler-Funktion (ADC_4088.cpp)	28
7.2. ISR zur Abhandlung des ADC-Interrupt Requests (impact_event.cpp)	30
10.1. Untermenü SD-Karte formatieren	49
10.2. Statusmeldung SD formatieren	49
10.3. Fehlermeldung SD formatieren	49
10.4. Erfolgsmeldung SD formatieren	49
10.5. Fehlermeldung SD-Karte anmelden	49
10.6. Untermenü SD-Karte abmelden	50
10.7. Warnung vor SD-Karte abmelden bei ungespeicherter Konfiguration	50
10.8. Untermenü Logger-Status	50
10.9. Untermenü Stoppen der Aufzeichnung	50
10.10Untermenü Starten der Aufzeichnung	50
10.11Untermenü Sensor-Auswahl	51
10.12Untermenü Sensor-Parameter	51
10.13Untermenü Abtastrate	51
10.14Fehlermeldung bei ungültiger Abtastrate	51
10.15Untermenü Threshold	52
10.16Fehlermeldung ungültiger Threshold	52
10.17Untermenü Null-Level	52
10.18Fehlermeldung ungültiger Nullpegel	53
10.19Untermenü Timeout	53
10.20Fehlermeldung zu langer Timeout	53
10.21Warnung kurzer Timeout	53
10.22Untermenü Detail-Level	53
10.23Untermenü Start/Stop einzeln	53
10.24Untermenü Start/Stop alle Sensoren	54
10.25Fehlermeldung beim Starten eines Sensors	54
10.26Untermenü Sensor-Status	54
10.27Untermenü Timestamp zurücksetzen	54
10.28Untermenü interne Uhr	54
10.29Untermenü Datum einstellen	55
10.30Untermenü Uhrzeit einstellen	55
10.31Untermenü Konfigurationsdatei	55
10.32Fehlermeldung beim Speichern der Konfigurationsdatei	56
10.33Beispiel einer Konfigurationsdatei	56
10.34Information bei ungespeicherter Konfiguration	56

A. Anhang

Anhang

A.1. Offizielle Aufgabenstellung

Titel: Messstation zur Registrierung von Geschiebe-Bewegungen im Fluss

Betreuer: Hans-Joachim Gelke, gelk

Fachgebiet: Mikroelektronik (ME)

Studiengang: IT

Zuordnung: Institute of Embedded Systems (InES)

Industriepartner:

Eidg. Forschungsanstalt für Wald, Schnee und Landschaft WSL,

8903 Birmensdorf (<http://www.wsl.ch/>)

Bruno Fritschi, bruno.fritschi@wsl.ch (Projektidee)

Dieter Rickenmann, dieter.rickenmann@wsl.ch (Betreuung)

Gruppengrösse: 2,

Reserviert für: Tobias Welti (weltitob), Tobias Keller (kellet01)

Ausgangslage:

Das WSL betreibt eine Messstation zur Registrierung von Geschiebe-Bewegungen im Fluss mittels Geophonen, die unter Stahlplatten montiert sind. Diese Platten sind in einer Betonkonstruktion eingelassen, um sie im Flussbett zu fixieren. Die Geophone sind über Kabel mit einem Auswertungs-Rechner (Embedded PC) verbunden, der die Signale auswertet. Die baulichen Massnahmen für die Installation der Sensoren, der Auswertungsstation sowie der Stromversorgung sind sehr teuer. Zukünftig sollen die Geophone durch eindimensionale MEMS Beschleunigungssensoren ersetzt werden, da diese kleiner sind.

Aufgabenstellung:

Um die Kosten zu senken und die zu übertragende Datenmenge zu reduzieren, soll die Auswertung der Daten direkt am Sensor erfolgen. Somit könnten die Daten über ein Bussystem übertragen werden und der Auswertungsrechner bräuchte weniger Leistung.

Dank der Bustopologie ist das Messsystem weniger komplex und kann einfacher installiert werden. Denkbar wäre die Integration in einer Gummimatte anstelle der Stahl- und Betonkonstruktion, da viel weniger Leitungen nötig sind.

Ziel der Arbeit ist die Entwicklung der Auswertungshardware und des Bussystems. Die Auswertungsalgorithmen sind nicht Bestandteil der Arbeit und werden vom WSL zur Verfügung gestellt.

Denkbar wäre es, einen Prototyp für Vergleichsmessungen im Erlenbach (Alptal, SZ) an einer bestehenden Schwelle zu implementieren.

A.2. Projektmanagement

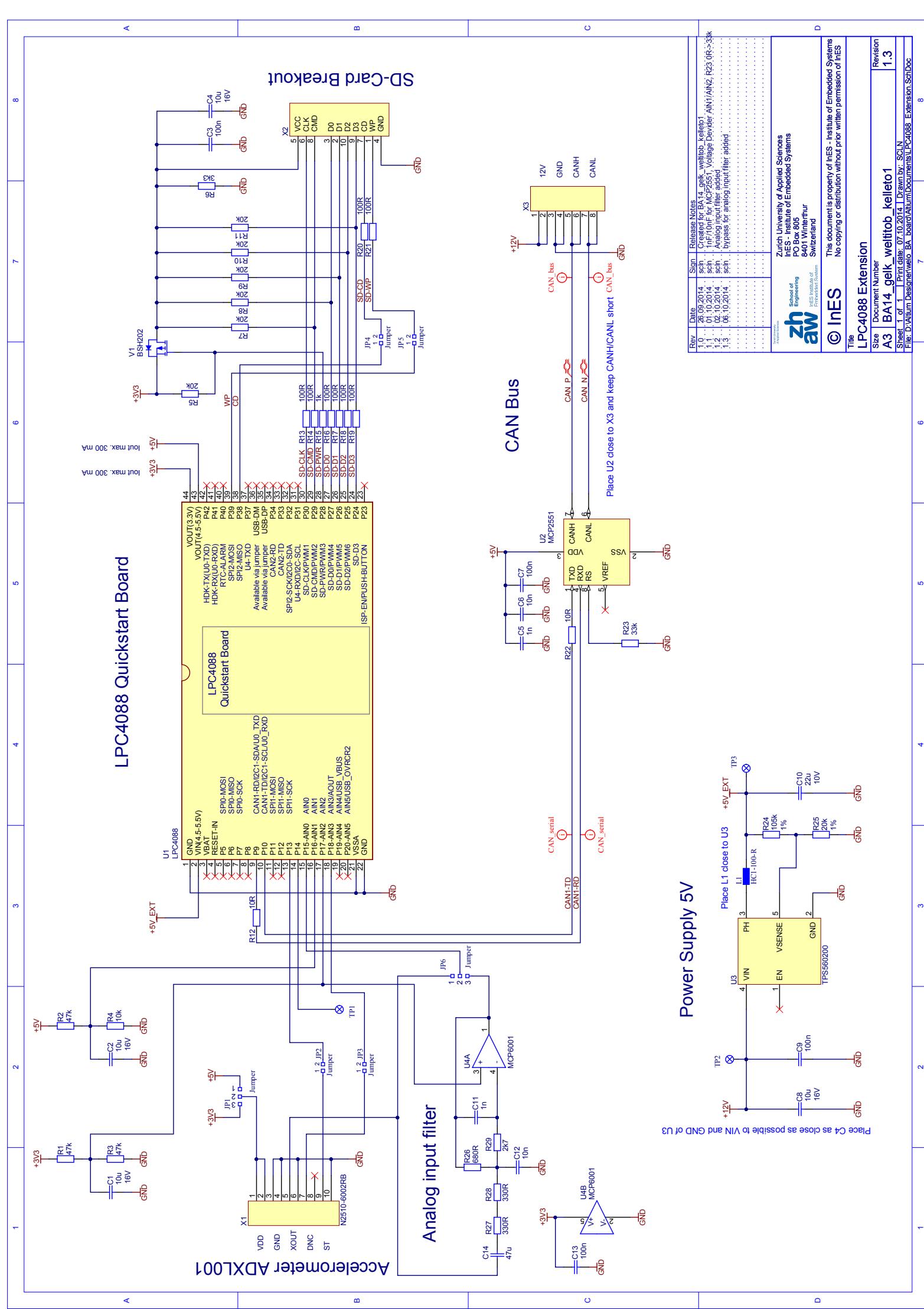
Zeitplan
Besprechungspro
oder Journals

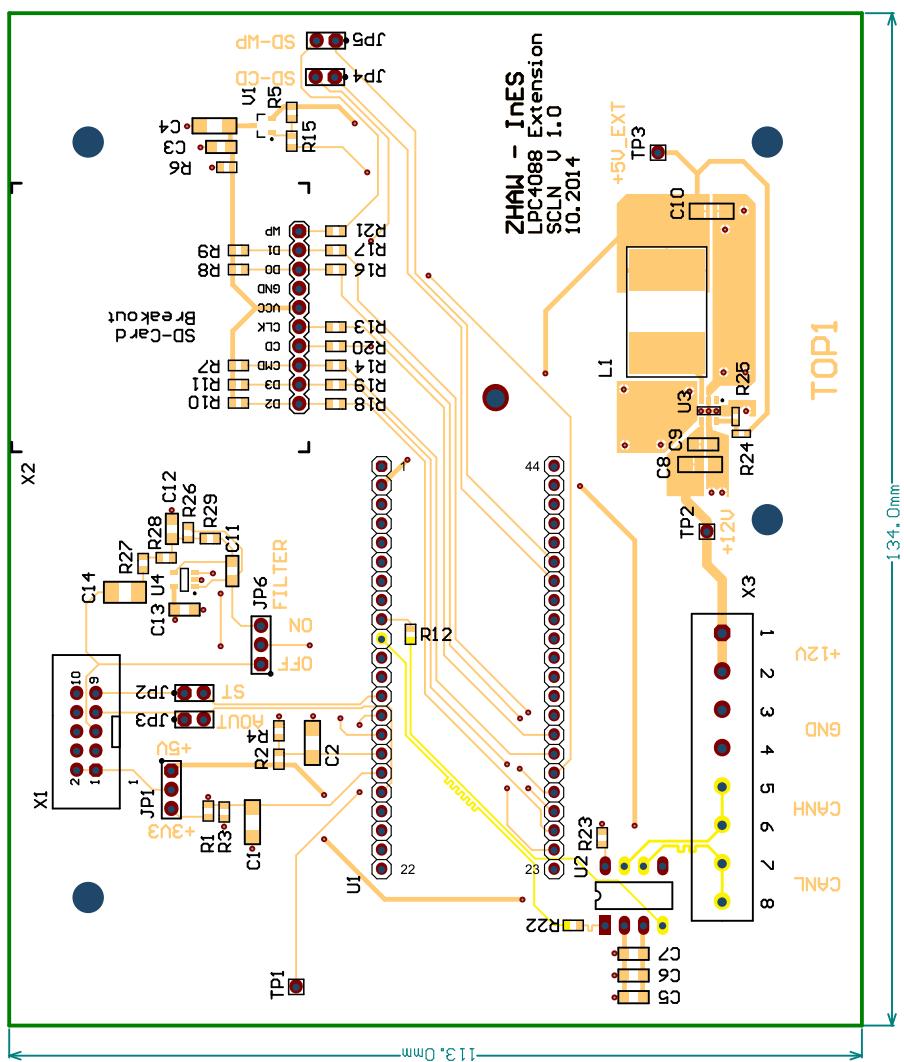
A.3. Weiteres

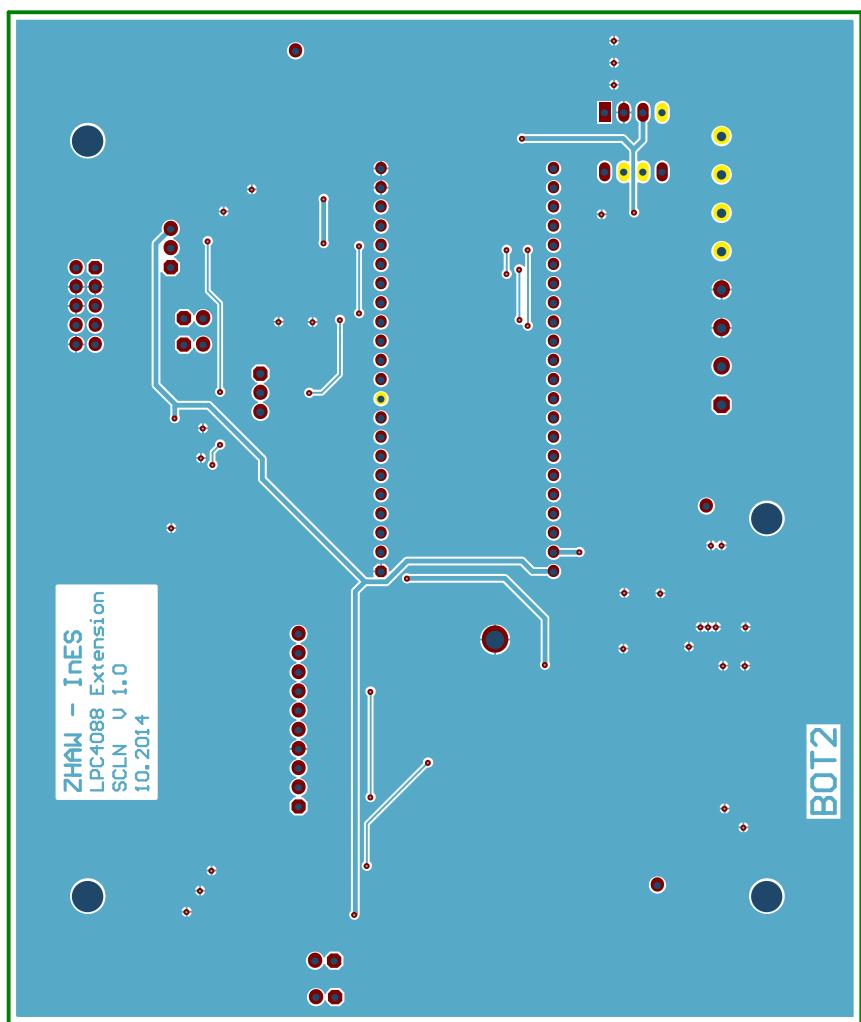
A.4. Schaltpläne

Für den Datenlogger und die Sensoreinheiten wurde mit der kompetenten Hilfe von Erich Ruff (ZHAW InES) und Valentin Schlatter (ZHAW InES) eine Leiterplatte entworfen sowie Gehäuse gebaut. Die Leiterplatte wurde so entworfen, dass über die Bestückung entschieden werden kann, ob ein Datenlogger oder eine Sensoreinheit gebaut wird. Für einen Datenlogger wird die Leiterplatte mit einem SD-Karten-Slot bestückt. Für die Sensoreinheit wird ein Tiefpassfilter und der Anschluss für den Sensor bestückt. Beide Varianten enthalten die Spannungsversorgung (12 V auf 5 V), einen CAN-Transceiver und die Anschlüsse für die Kabel. Der Schaltplan sowie das Leiterplattenlayout befindet sich im Anhang A.4

dieser Abschnitt
in den Haupttext





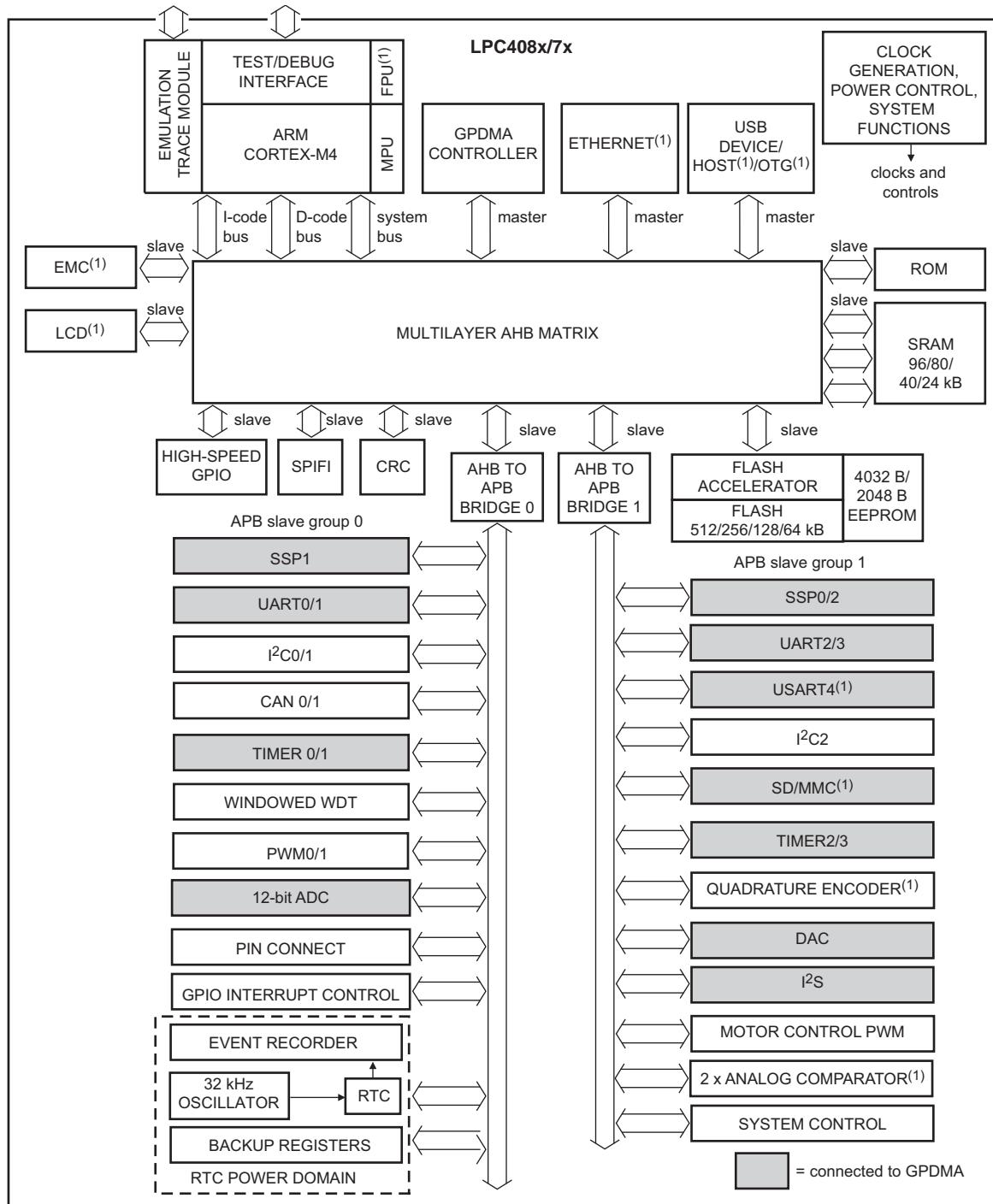


A.5. Datenblätter

Um die Dokumentation übersichtlich zu halten, wird der Grossteil der Datenblätter nicht mit der Dokumentation ausgedruckt, sondern auf der beiliegenden Compact Disc (CD) mitgeliefert.

A.5.1. NXP LPC4088 32-bit ARM Cortex-M4 microcontroller

5. Block diagram



002aag491

(1) Not available on all parts.

Fig 1. Block diagram

A.5.2. Embedded Artists NXP LPC4088 QuickStart Board

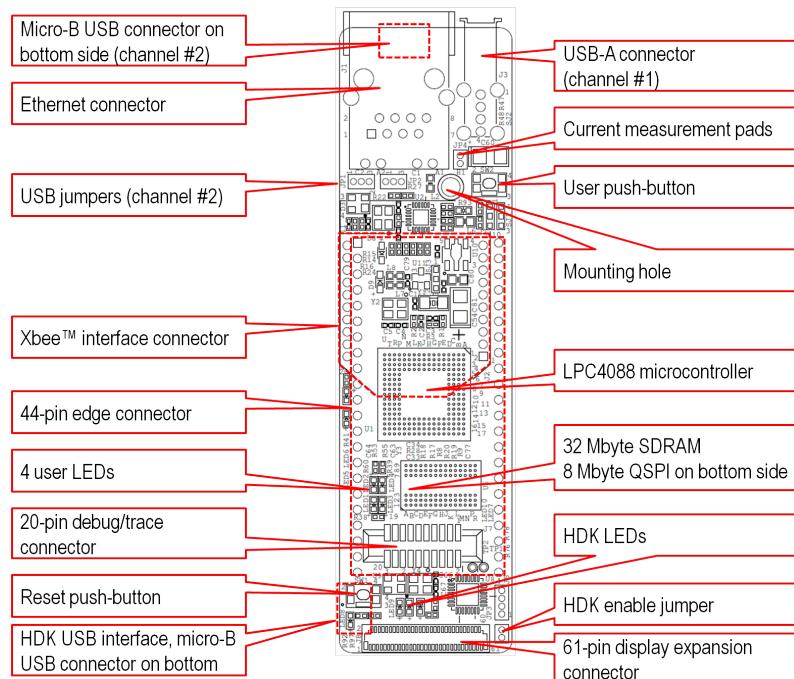


Abbildung A.1.: Hauptkomponenten des NXP LPC4088 QuickStart Boards von Embedded Artists.

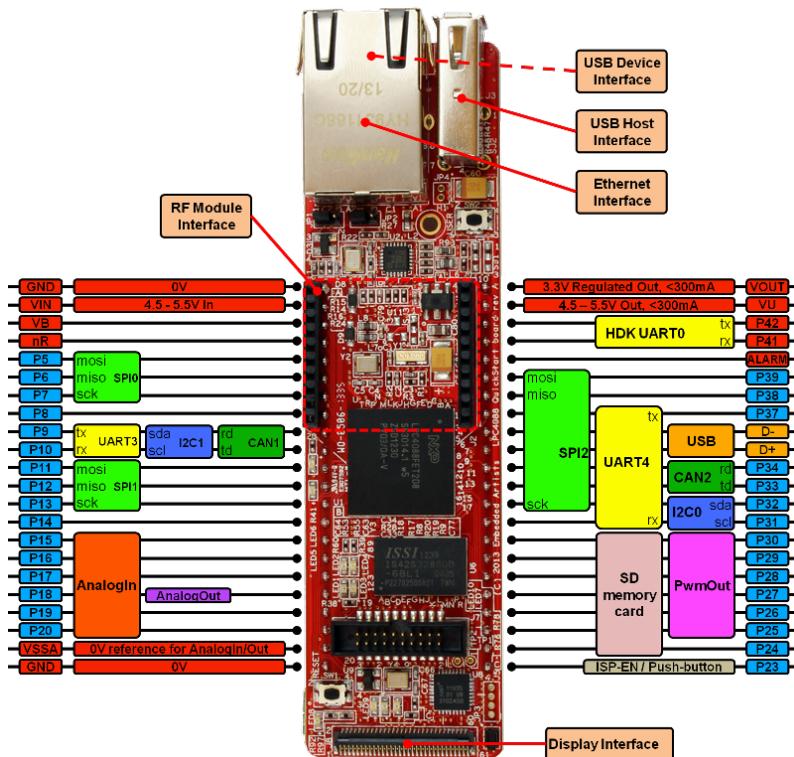


Abbildung A.2.: Pins des NXP LPC4088 QuickStart Boards von Embedded Artists.

[Users\(/activity/\)](#) » [embeddedartists\(/users/embeddedartists/\)](#) » [Notebook\(/users/embeddedartists/notebook\)](#) » LPC4088 QuickStart Board - Hardware Information

LPC4088 QuickStart Board - Hardware Information

Page last updated 14 Nov 2013(14 Nov 2013), by [EmbeddedArtists AB](#)(<https://developer.mbed.org/users/embeddedartists/>). 17 replies ([/users/embeddedartists/notebook/lpc4088-quickstart-board---hardware-information/#commentform](#))

LPC4088 QuickStart Board Hardware Features

MCU

- LPC4088, Cortex-M4F core running at up to 120MHz

Memory

- 512 KByte on-chip FLASH
- 96 KByte on-chip SRAM
- 4 KByte on-chip E2PROM
- 8 MByte QSPI FLASH (can execute program code and/or contain a file system)
- 32 MByte SDRAM with 32-bit databus access
- On-board globally unique MAC address.

Connectors

- 2x22 pin edge pins that are very compatible with the original 2x20 pin LPC1768 mbed pinning
- 10/100Mbps Ethernet (RJ45)
- USB-A (USB Host interface)
- USB-micro B (USB Device interface)
- USB-micro B (mbed HDK debug interface)
- 20 pos SWD/Trace connector (ARM standard debug connector)
- 61 pos 0.3mm pitch FPC connector for **display** expansion
- 20 pos **XBee** compatible connector for RF module add-on

Display expansion

- Up to 24-bit pixel data (16-bit most common) via 61 pos, 0.3mm pitch FPC connector.

Other

- Proper **ESD protection** on communication interfaces.
- On-board HDK (debug interface functions)
- Supported by the on-line mbed SDK.
- Supported by a lot of professional quality software and examples by EA.
- Industrial temperature specified (-40 to +85 degrees Celsius).
- **ISO 9001** produced.
- Production compensated for carbon dioxide emission.
- Current consumption down to about **5 mA**.

Pin Usage

LPC4088 QuickStart Board - Hardware Information

Page owner: [EmbeddedArtists AB](#)(<https://developer.mbed.org/users/embeddedartists/>)

Created 05 Sep 2013(05 Sep 2013).

Last updated 14 Nov 2013(14 Nov 2013)

A.5.3. Analog Devices ADXL001 Beschleunigungssensor

SPECIFICATIONS

SPECIFICATIONS FOR 3.3 V OPERATION

$T_A = -40^\circ\text{C}$ to $+125^\circ\text{C}$, $V_S = 3.3 \text{ V} \pm 5\%$ dc, acceleration = 0 g, unless otherwise noted.

Table 1.

Parameter	Conditions	ADXL001-70			ADXL001-250			ADXL001-500			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
SENSOR											
Nonlinearity		0.2	2		0.2	2		0.2	2		%
Cross-Axis Sensitivity	Includes package alignment	2			2			2			%
Resonant Frequency		22			22			22			kHz
Quality Factor		2.5			2.5			2.5			
SENSITIVITY											
Full-Scale Range Sensitivity	$I_{\text{OUT}} \leq \pm 100 \mu\text{A}$ 100 Hz	-70		+70	-250		+250	-500		+500	g mV/g
16.0					4.4			2.2			
OFFSET	Ratiometric	1.35	1.65	1.95	1.35	1.65	1.95	1.35	1.65	1.95	V
Zero-g Output											
NOISE											
Noise	10 Hz to 400 Hz	85			95			105			mg rms
Noise Density	10 Hz to 400 Hz	3.3			3.65			4.25			$\text{mg}/\sqrt{\text{Hz}}$
FREQUENCY RESPONSE											
-3 dB Frequency		32			32			32			kHz
-3 dB Frequency Drift Over Temperature		2			2			2			%
SELF-TEST											
Output Voltage Change		400			125			62			mV
Logic Input High		2.1			2.1			2.1			V
Logic Input Low			0.66			0.66			0.66		V
Input Resistance	To ground	30	50		30	50		30	50		$\text{k}\Omega$
OUTPUT AMPLIFIER											
Output Swing	$I_{\text{OUT}} = \pm 100 \mu\text{A}$	0.2		$V_S - 0.2$	0.2		$V_S - 0.2$	0.2		$V_S - 0.2$	V
Capacitive Load		1000			1000			1000			pF
PSRR (CFSR)	DC to 1 MHz	0.9			0.9			0.9			V/V
POWER SUPPLY (V_S)											
Functional Range		3.135	6		3.135	6		3.135	6		V
I_{SUPPLY}		2.5	5		2.5	5		2.5	5		mA
Turn-On Time		10			10			10			ms

A.6. Source Code, Daten und Multimedia

Da der Source Code sehr umfangreich ist, wird darauf verzichtet, ihn ausgedruckt zur Verfügung zu stellen. Er befindet sich auf der beiliegenden CD.

Spezifikationen
tenblätter der ver-
ten Messgeräte u.
der Komponente
Berechnungen, M
te, Simulationsre
Grafische Darste
Fotos

Inhaltsverzeichni
CD erstellen
CD mit dem voll
gen Bericht als p
inklusive Film- u
tomaterial