



**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**

8. Dezember 2014

## 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 Disziplarmassnahmen 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: [häsch.überhaupt@privati.adresse?](mailto:häsch.überhaupt@privati.adresse?)

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

E-Mail: [tobias.welti@zhaw.ch](mailto: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](mailto: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](mailto:carlos.wyss@wsl.ch)  
Homepage: <http://www.wsl.ch>

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>6</b>
1.1. Ausgangslage . . . . .	6
<b>2. Aufgabenstellung</b>	<b>7</b>
2.1. Aufgabenstellung . . . . .	7
2.1.1. Musskriterien . . . . .	7
2.1.2. Wunschkriterien . . . . .	8
2.1.3. Abgrenzungskriterien . . . . .	8
<b>3. Vorgehen</b>	<b>9</b>
3.1. Überblick . . . . .	9
3.2. Datenlogger . . . . .	9
3.2.1. Messdaten sammeln . . . . .	10
3.2.2. Kontrolle über das Bussystem . . . . .	10
3.2.3. Steuerung des Betriebs . . . . .	10
3.2.4. Schnittstelle nach Aussen . . . . .	10
3.3. Sensoreinheit . . . . .	10
3.3.1. Messdatenerfassung . . . . .	10
3.3.2. Ereigniserkennung . . . . .	11
3.3.3. Datenübertragung . . . . .	11
3.4. Bussystem . . . . .	11
<b>4. Funktionale Anforderungen</b>	<b>12</b>
4.1. Datenlogger (F1...) . . . . .	12
4.1.1. F110 Busmaster . . . . .	12
4.1.2. F120 Sensorerkennung . . . . .	12
4.1.3. F130 Uhrzeit . . . . .	12
4.1.4. F140 Timestamp verteilen . . . . .	12
4.1.5. F160 Schnittstelle zum Steuerrechner . . . . .	12
4.1.6. F170 Steuerung Betriebsmodus . . . . .	12
4.1.7. F180 Daten sammeln . . . . .	13
4.1.8. F190 Daten speichern . . . . .	13
4.2. Sensoreinheit (F4...) . . . . .	13
4.2.1. F410 Ereignisdetektion . . . . .	13
4.2.2. F430 Datenübertragung . . . . .	13
4.2.3. F450 Rohdatenaufzeichnung . . . . .	13
<b>5. Nichtfunktionale Anforderungen</b>	<b>14</b>
<b>6. Grundlagen</b>	<b>15</b>
<b>7. Hardware-Konzept</b>	<b>16</b>
7.1. Hardware-Architektur . . . . .	16
7.1.1. Datenlogger . . . . .	16
7.1.2. Sensoreinheit . . . . .	16
7.1.3. Bussystem . . . . .	17
7.2. Komponentenauswahl . . . . .	17
7.2.1. Mikroprozessor . . . . .	17
7.2.2. Bus-System . . . . .	20

7.2.3. Speichermedium . . . . .	22
7.2.4. Sensor . . . . .	23
7.2.5. Schnittstelle . . . . .	23
7.3. Komponenten . . . . .	23
7.3.1. Cortex M4 Mikroprozessor . . . . .	23
7.3.2. Beschleunigungs-Sensor . . . . .	23
7.3.3. CAN Bus . . . . .	23
7.3.4. SD Karte . . . . .	24
7.3.5. UART Schnittstelle . . . . .	24
7.3.6. Gehäuse . . . . .	24
7.4. Datenlogger . . . . .	24
7.5. Sensoreinheit . . . . .	24
<b>8. Software-Konzept</b>	<b>26</b>
8.1. Software-Stack . . . . .	26
8.1.1. Überblick . . . . .	26
8.1.2. Messdatenerfassung . . . . .	27
8.1.3. Ereigniserkennung . . . . .	27
8.1.4. Timestamp . . . . .	31
8.1.5. Verwaltung der Messstation . . . . .	31
8.1.6. Busprotokoll . . . . .	33
8.1.7. Filesystem . . . . .	33
8.1.8. UART-Kommandozeile . . . . .	33
8.2. Funktionalität . . . . .	33
8.3. Konfiguration . . . . .	33
8.4. Parallele Prozesse . . . . .	33
<b>9. Resultate</b>	<b>34</b>
<b>10. Diskussion</b>	<b>35</b>
<b>11. Verzeichnisse</b>	<b>36</b>
<b>Literaturverzeichnis</b>	<b>37</b>
<b>Abbildungsverzeichnis</b>	<b>38</b>
<b>Tabellenverzeichnis</b>	<b>39</b>
<b>Glossar</b>	<b>40</b>
<b>Akronyme</b>	<b>43</b>
<b>A. Anhang</b>	<b>I</b>
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
was gibt es bis jetzt? wie wirds gemacht? was gäbe es für alternativen?Mu LITERATURVERWEISE! . . . . .	6
etwas über signalerfassung. . . . .	15
etwas über signalverarbeitung (aufwand hilbert etc.) . . . . .	15
Kriterienliste RS485/CAN einfügen, Lit-Referenz auf White Paper von IXXAT . . . . .	20
Literatur-Referenzen in Tabelle 7.4 . . . . .	23
Sensorauswahl beschreiben . . . . .	23
Schnittstellenauswahl beschreiben . . . . .	23
Texten . . . . .	23
Cortex M4 beschreiben . . . . .	23
Flash Speicher beschreiben . . . . .	23
SDRAM beschreiben . . . . .	23
Sensor beschreiben . . . . .	23
CAN Bus beschreiben . . . . .	23
CAN Transceiver beschreiben . . . . .	24
SD-Karte/MCI beschreiben . . . . .	24
UART beschreiben . . . . .	24
Gehäuse beschreiben . . . . .	24
Datenlogger übersicht . . . . .	24
Sensoreinheit übersicht . . . . .	24
Prioritäten des Prozesse Messung, Erkennung, Übermittlung erklären, und wie wir das gelöst haben.	27
figure: umhüllende (matlab: hilbert()) . . . . .	27
Ereignis-FSM anpassen auf neue Stati . . . . .	28
Meldung ins Busprotokoll für den Verlust von Messwerten! . . . . .	31
Zusammenhänge A/D-Wandlung und Ereigniserkennung und Übertragung beschreiben, dazu ein Kommunikationsdiagramm, wo synchron, wo asynchron. . . . .	31
Verschiedene Betriebsmodi mit Grafiken beschreiben . . . . .	31
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. . . . .	31
Timestamp beschreiben, Rechnung über die Dauer der eindeutigen Zuweisung. . . . .	31

Busverwaltung beschreiben . . . . .	31
Figur 8.6 aufteilen auf zwei Seiten. (PDF-crop) . . . . .	32
Busprotokoll austüfteln. Darstellung siehe HW-Konzept Rioxo, genaue Beschreibung der Nachrichtentypen. Timestamp der einzelnen Peaks bezieht sich auf Offset vom Beginn des Impacts. . . . .	33
Kommunikationsdiagramm Bushandler . . . . .	33
Interrupt-System des Bushandlers aufführen . . . . .	33
Texten . . . . .	33
Frage: wird für jeden Sensor ein eigenes File geführt? Kann man alle Files offen lassen oder ist das keine gute Idee? Was ist besser, jedes mal das File-Ende zu suchen um neues anzuhängen? . . . . .	33
Dokumentation über die Kommandos, wird später für die Bedienungsanleitung gebraucht . . . . .	33
Texten . . . . .	33
Hier eine Art Bedienungsanleitung zur Konfiguration geben. Welches Kommando hat was zur Folge? (Wird Datenerfassung neu gestartet, werden allenfalls andere Sensoren deaktiviert etc. . . . .	33
läuft das Zeug? . . . . .	34
wie viele Sensoren mag's leiden? . . . . .	34
wie schnell kann es sein? . . . . .	34
wie lange kann man aufzeichnen? . . . . .	34
was geht, was nicht? . . . . .	34
was haben wir nicht erreicht? . . . . .	34
wo haben wir mehr als gefordert erreicht? . . . . .	34
haben wir erfüllt? . . . . .	35
wo gabs Schwierigkeiten? . . . . .	35
worauf sind wir stolz . . . . .	35
was könnte man jetzt weiter noch machen? . . . . .	35
was ist noch geplant? . . . . .	35
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 Geophone sind über Kabel mit einem Auswertungs-Rechner (Embedded PC) verbunden, der die Signale auswertet. Die baulichen Massnahmen für die Installation der Geophone, der Auswertungsstation sowie der Stromversorgung sind sehr teuer.

es bis jetzt?  
s gemacht? was  
für alternati-

TURVERWEISE!

## 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, 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 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.

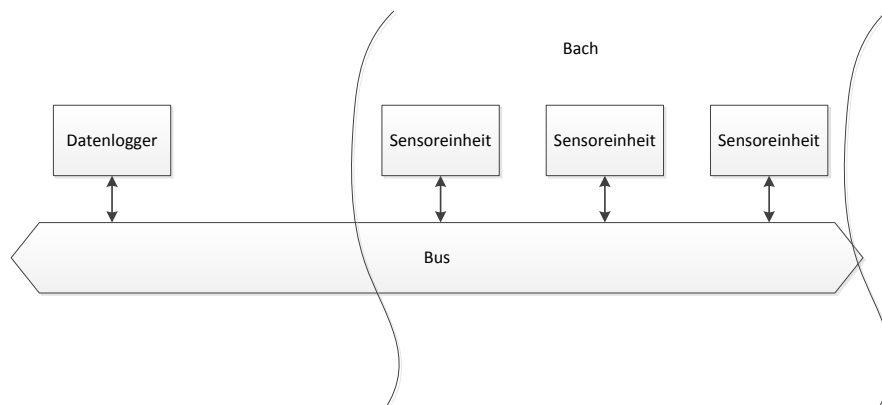


Abbildung 3.1.: Eine Messstation mit einem Datenlogger, der mehrere Sensoreinheiten im Bach steuert.

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.

### 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 adressieren. 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 zugeteilt, 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 detailreiche 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 Speicherkarte 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 kann 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. Grundlagen

etwas über mems

etwas über signa  
sung.

etwas über signa  
beitung (aufwan  
etc.)

## 7. Hardware-Konzept

### 7.1. Hardware-Architektur

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

#### 7.1.1. Datenlogger

Das Hardware-Konzept des Datenloggers ist in Abbildung 7.1 dargestellt. Der Datenlogger sammelt die Daten der Sensoreinheiten über das Bussytem ein und speichert sie ab. Dafür benötigt er das Bussytem, einen Mikroprozessor, 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.

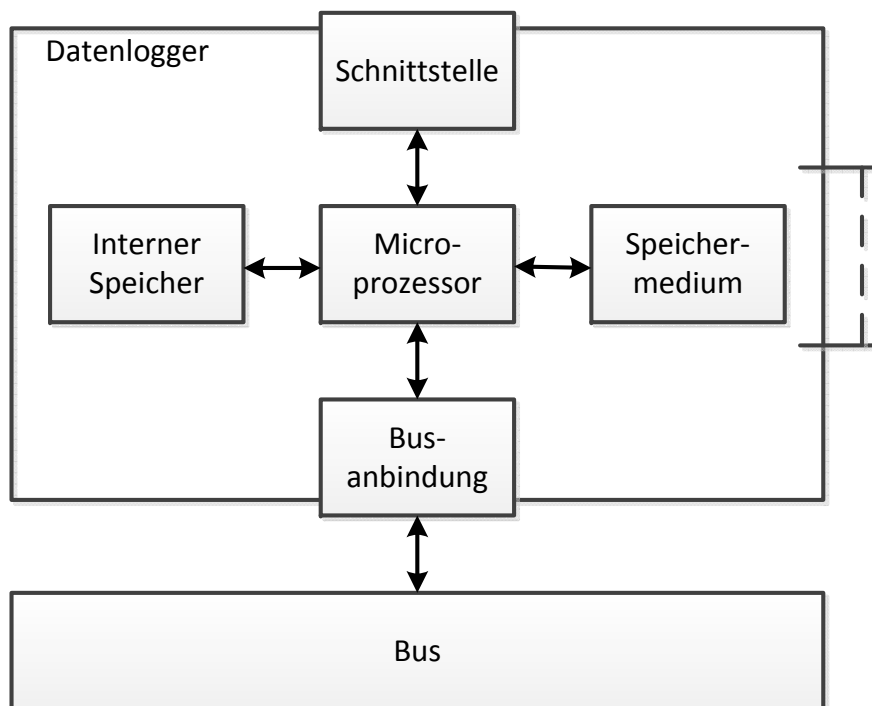


Abbildung 7.1.: Hardwarekonzept des Datenloggers.

#### 7.1.2. 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 Mikroprozessor verarbeitet,

im internen Speicher zwischengespeichert und über das Bussystem an den Datenlogger übertragen. Abbildung 7.2 zeigt das Hardware-Konzept der Sensoreinheit.

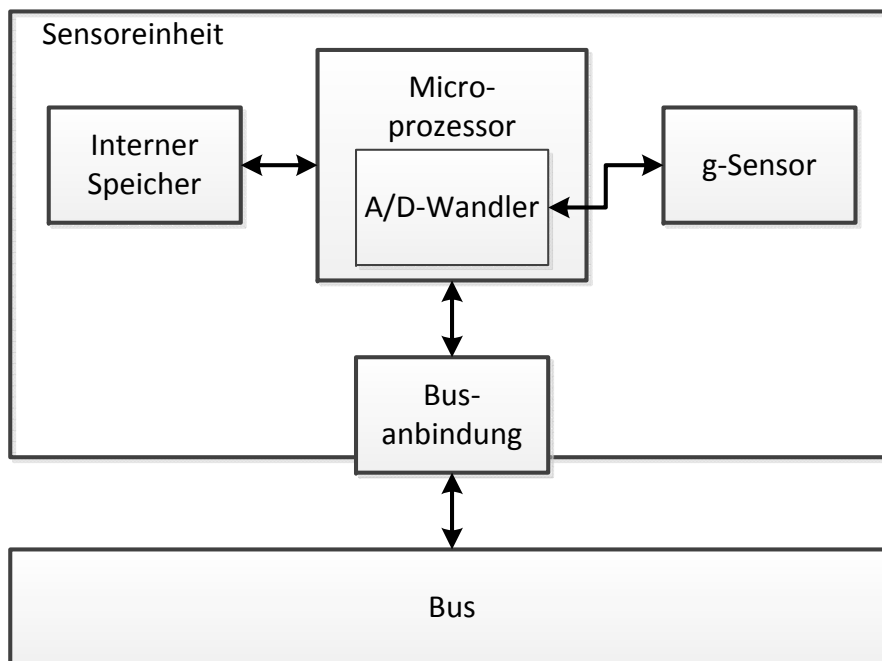


Abbildung 7.2.: Hardwarekonzept der Sensoreinheit.

### 7.1.3. 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.

## 7.2. Komponentenauswahl

### 7.2.1. Mikroprozessor

Bei der Auswahl des Mikroprozessors 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).
- 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 Digitaler 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 7.3 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 8.4 wird näher darauf eingegangen.

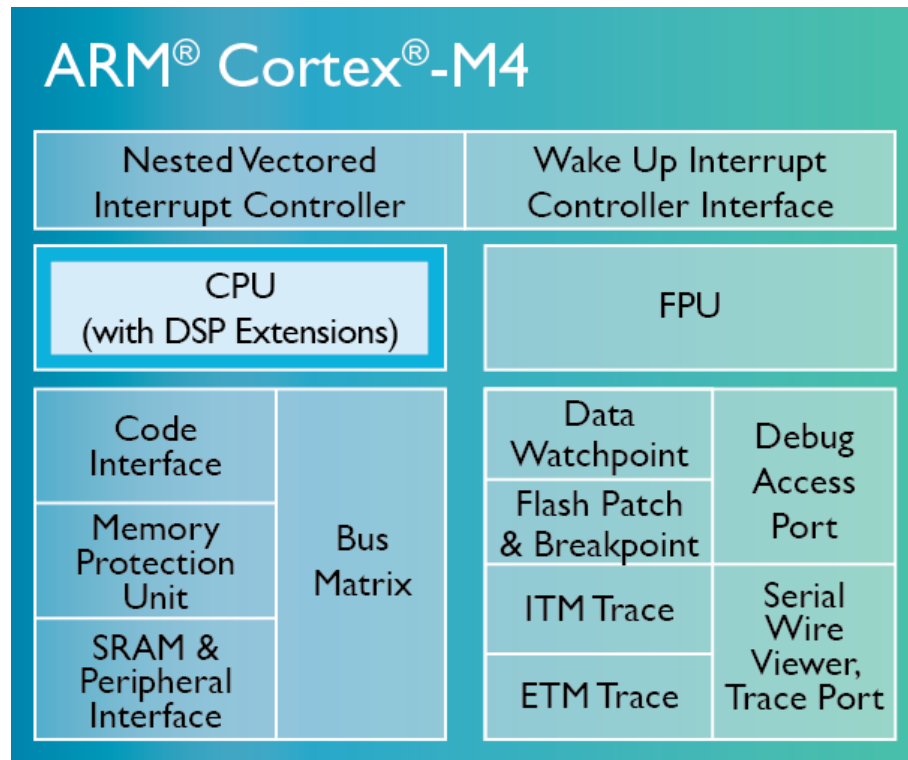
**DSP** Der *ARM Cortex-M4* verfügt über DSP-Funktionen, eine sog. single-cycle multiply-accumulate unit (MAC). In einer single-cycle MAC können Multiplikationen in einem einzigen Prozessorzyklus ausgeführt werden. Normalerweise benötigt ein Prozessor für eine Multiplikation bis zu mehreren zehn Zyklen, 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 [1].

**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.

**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.

Abbildung 7.3.: Chipdiagramm der *ARM Cortex-M4* Architektur [1].

**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 7.1 dargestellt.

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

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 7.1.: Fähigkeiten des NXP LPC4088 Prozessors [2].

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

Tabelle 7.2.: Zusätzliche Fähigkeiten des NXP LPC4088 QuickStart Boards von *Embedded Artists* [3].

### 7.2.2. Bus-System

Anhand folgender Kriterien wurde ein Bussystem ausgewählt:

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

In Tabelle 7.3 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 (Daisychain). Jedes Paket müsste vom Microcontroller weitergeleitet werden, wenn es für einen anderen Empfänger bestimmt ist. Dies führte 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 Daisychain mit PoE, neben dem Datenkabel wäre noch ein Kabel für die Stromversorgung notwendig.

**Vergleich CAN-Bus und RS485** CAN und RS: Stecker nicht definiert => wasserdichte Stecker einfach zu finden.

**Entscheidung** CAN-Bus erfüllt alle Kriterien und erlaubt es, den Busmaster am Ende des Bus zu platzieren. Dies ist ein Vorteil gegenüber RS485, wo der Master in der Mitte platziert werden sollte. CAN-Bus bietet bereits Kollisionserkennung 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 ein klarer Vorteil von RS485 darstellt. Insgesamt überwiegen die Vorteile von CAN-Bus klar.

### 7.2.3. 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.

	Bitrate	Distanz	Clients	Besonderheiten
<b>CAN</b>	1 MBit/s 125 kBit/s	40 m 500 m	> 20	<ul style="list-style-type: none"> <li>+ Collision Detection (CD) umgehen mit Polling durch Master.</li> <li>+ Bei synchronem CAN wird CD durch ID gelöst.</li> <li>+ CAN Controller sendet Interrupt Request bei erhaltener Nachricht.</li> </ul>
<b>SPI</b>	..100 MBit/s	< 1 m	slave select	<ul style="list-style-type: none"> <li>- Pro Client eine Slave Select Leitung</li> <li>- Daisy Chain <math>\Rightarrow</math> alle MC beschäftigt.</li> <li>- Bei Ausfall eines MC ganzer Bus unterbrochen.</li> </ul>
<b>RS485</b>	35 MBit/s 100 kBit/s	10 m 1200 m	>32	<ul style="list-style-type: none"> <li>- Master am besten in der Mitte des Bus <math>\Rightarrow</math> ungünstig.</li> <li>- Braucht 2..4 Drähte (bei Full Duplex)</li> <li>- braucht pull-up und pull-down Widerstände <math>\Rightarrow</math> mehr Leistungsaufnahme.</li> </ul>
<b>Ethernet</b>	100 MBit/s	100 m	> 20	<ul style="list-style-type: none"> <li>+ Stromversorgung bei Power over Ethernet (PoE) integriert.</li> <li>- kein Bus sondern allenfalls Daisychain.</li> <li>- bei Daisychain kein PoE möglich.</li> </ul>
<b>Feldbus</b>				ist eine Familie von Bussen, z.B. CAN-Bus
<b>I2C</b>	0.4..5 Mbit/s	wenige Meter	< 20	nur für kurze Distanzen, Bitrate nimmt rasch ab.

Tabelle 7.3.: Entscheidungsmatrix für die Auswahl des Bussystems.



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

Tabelle 7.4.: Entscheidungsmatrix zur Auswahl des Speichermediums.

**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..90 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. Begnügt man sich mit weniger Details, reichen 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 wasser-dichten Zugang zum Medium voraus. Da heute Speichermedien mit kleinem Platzbedarf erhältlich sind, könnte ein solcher Zugang recht einfach mit einem Schraubverschluss realisiert werden.

**Vergleich** In Tabelle 7.4 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.

**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.

#### 7.2.4. Sensor

-Referenzen in  
7.4

swahl beschrei-

### 7.2.5. Schnittstelle

Schnittstellenaus-  
beschreiben

## 7.3. Komponenten

Texten

### 7.3.1. Cortex M4 Mikroprozessor

Cortex M4 besch

Flash Speicher

Flash Speicher b  
ben

SDRAM

SDRAM beschre

### 7.3.2. Beschleunigungs-Sensor

Sensor beschreib

### 7.3.3. CAN Bus

CAN Bus beschr

CAN Transceiver

CAN Transceiver  
schreiben

### 7.3.4. SD Karte

SD-Karte/MCI b  
ben

### 7.3.5. UART Schnittstelle

UART beschreib

### 7.3.6. Gehäuse

Gehäuse beschre

## 7.4. Datenlogger

ger übersicht

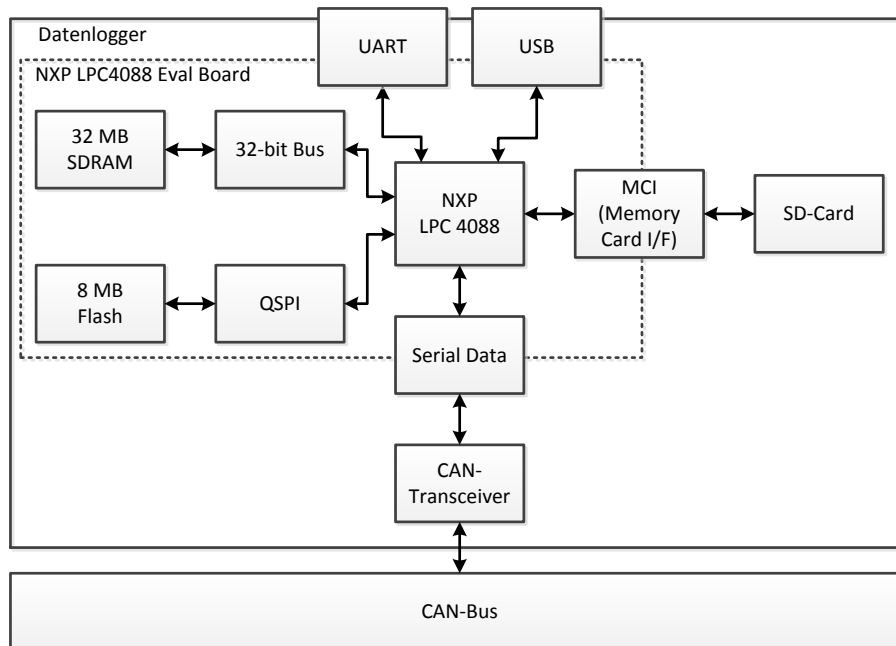


Abbildung 7.4.: Schematischer Hardware-Aufbau des Datenloggers.

## 7.5. Sensoreinheit

inheit übersicht

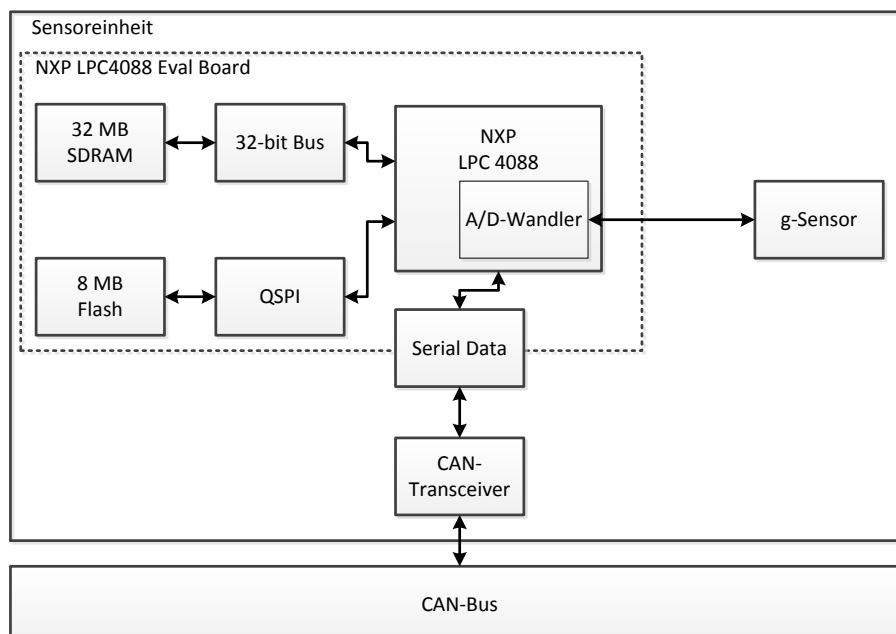


Abbildung 7.5.: Schematischer Hardware-Aufbau der Sensoreinheit.

## 8. Software-Konzept

### 8.1. Software-Stack

#### 8.1.1. Überblick

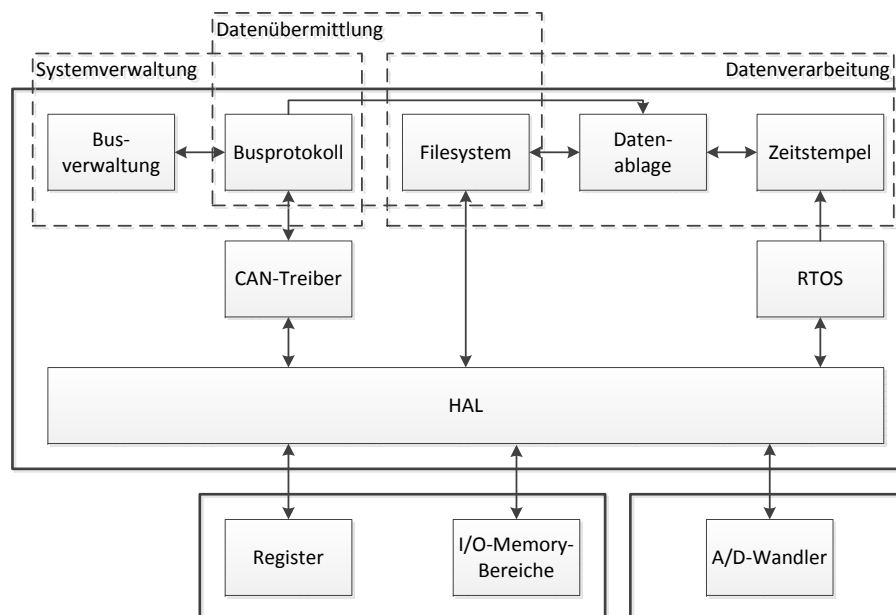


Abbildung 8.1.: Softwarestack des Datenloggers.

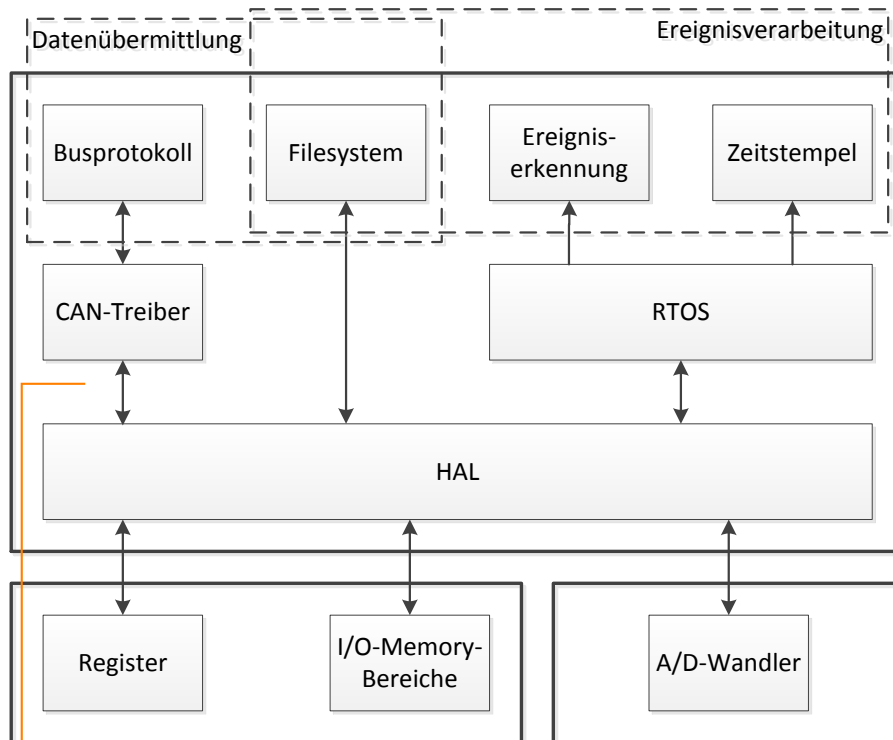


Abbildung 8.2.: Softwarestack der Sensoreinheit.

### 8.1.2. 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 Beschleunigungssensor 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 mit bis zu 400 kHz betrieben werden.

### 8.1.3. 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 einigen 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 [4]. 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 [6,

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.

### Hilbert-Transformation als FIR-Filter

Die Hilbert-Transformation kann mittels eines FIR-Filters angenähert werden. Ein Allpass mit gerader Filter-Ordnung und geeignet gewählten Koeffizienten (Abbildung 8.3) liefert eine gute Näherung. Je nach gewählter Ordnung des Filters ist der Rechenaufwand aber in ähnlicher Größenordnung wie mit DFT und IDFT.

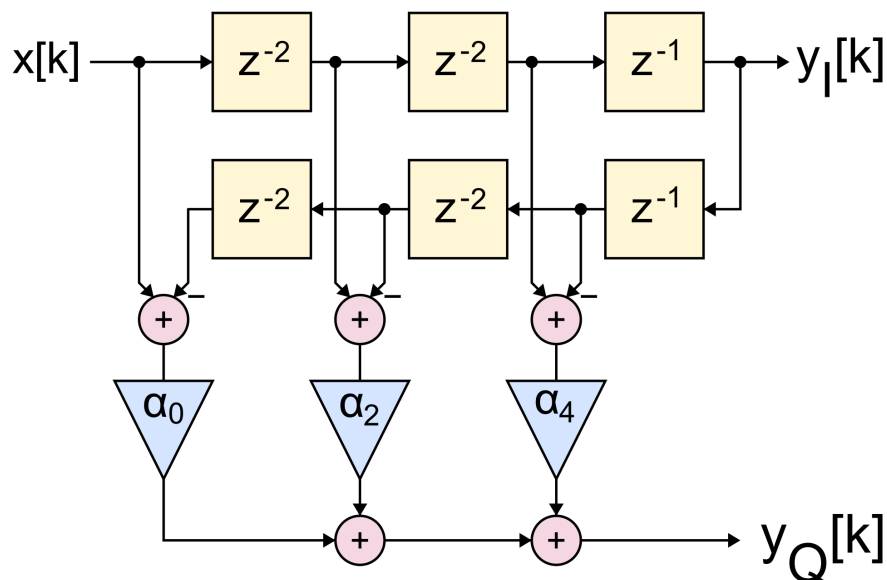


Abbildung 8.3.: Hilbert-Transformation als FIR-Filter [5].

### 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 8.4 zeigt alle möglichen Zustände der FSM und welche Ereignisse einen Übergang in einen anderen Zustand auslösen.

Ereignis-FSM an  
auf neue Stati

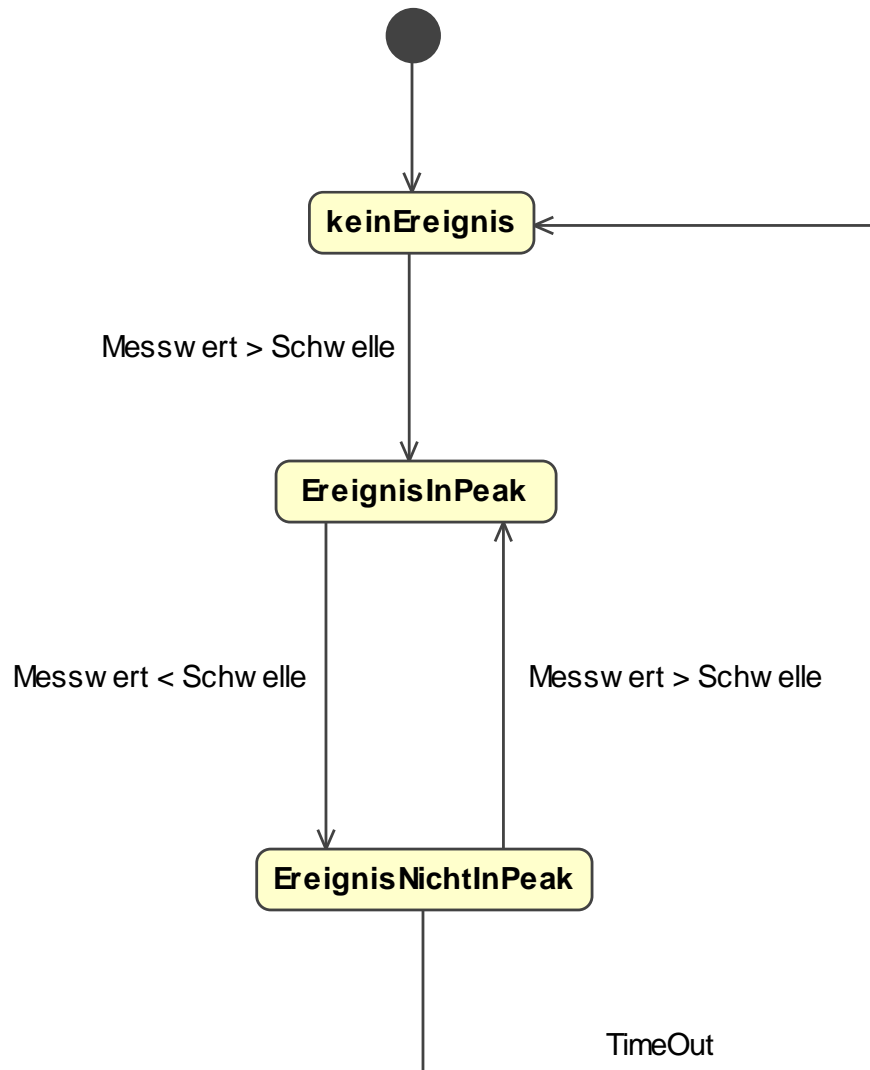


Abbildung 8.4.: Zustandsmaschine der Ereigniserkennung.

**Konfiguration der Zustandsmaschine** Über Parameter wird definiert, welche Signalform als Ereignis erkannt werden soll. In Abbildung 8.5 sind die Parameter dargestellt.

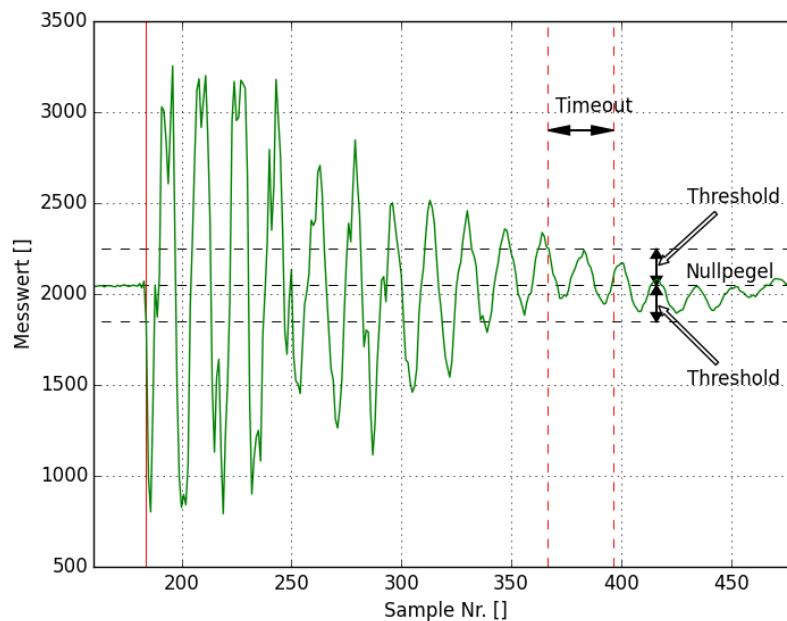


Abbildung 8.5.: Parameter der Ereigniserkennung.

**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 und damit nicht in jedem Fall gleich gross. Deshalb muss dieser Wert angepasst werden können.

**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.

**Timeout** Da ein Ereignis nicht nur aus 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.

### Ablauf der Zustandsmaschine

Im Folgenden wird der Ablauf in der Zustandsmaschine genauer erklärt. Im Zustandsdiagramm in Abbildung 8.4 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 'EreignisInPeak\_positiv'. In diesem Zustand verbleibt die FSM, bis ein anders klassierter 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 (Queue) geschrieben werden, von wo sie von der FSM abgeholt werden. So lange die Queue nicht überfüllt wird, gehen keine Messwerte verloren. Die Verarbeitung der Messwerte in der FSM erfolgt im NXP LPC4088 Prozessor schnell genug, um theoretisch mit einer Abtastrate bis 200  $kHz$  messen zu können.

ins Busprotokoll  
den Verlust von  
ten!

enhänge A/D-  
g und Ereigni-  
ng und Über-  
beschreiben, da-  
ommunikations-  
m, wo synchron,  
chron.

dene Betriebs-  
t Grafiken be-  
n

ungen, in wel-  
odus wie lange  
n werden kann,  
lange ein Sensor  
vorhandenen  
die Resultate  
speichern kann.  
s ein System  
n, das automa-  
schen verschie-  
odi hin- und  
ten kann. (Ist  
s heikel). Wie  
soren können in  
Modus gleich-  
n System betrie-  
den, bei welcher  
ate ist Schluss  
andbreite.

#### 8.1.4. Timestamp

#### 8.1.5. Verwaltung der Messstation

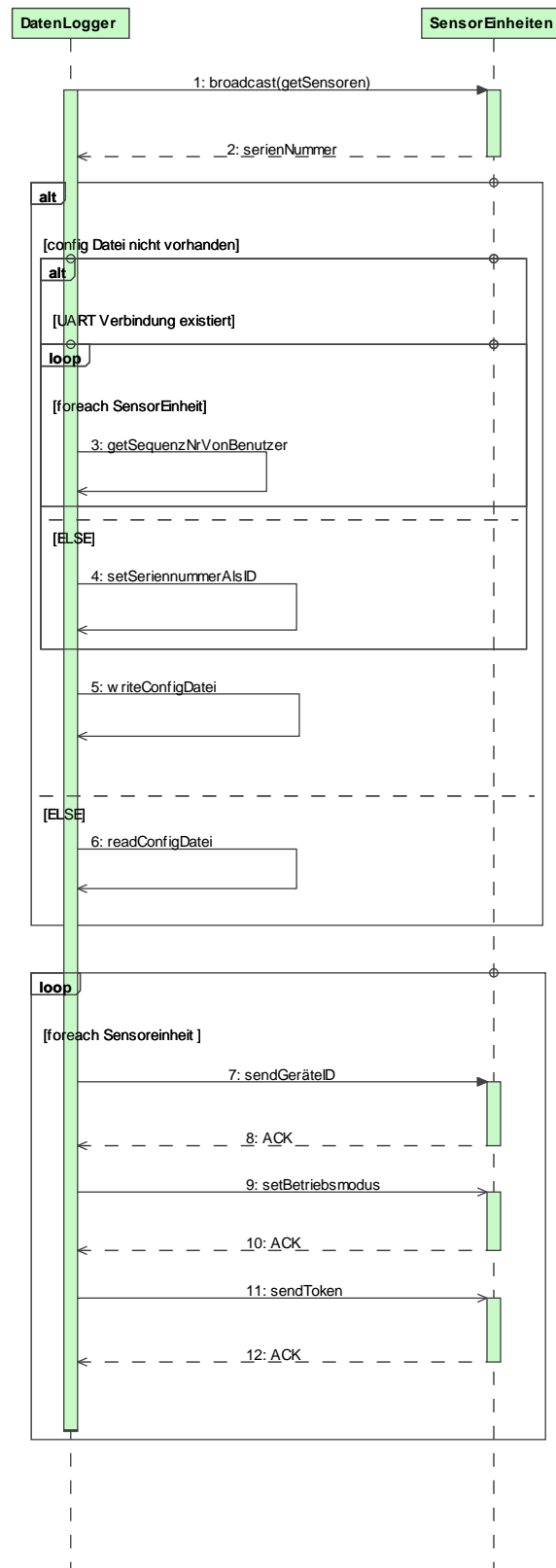


Abbildung 8.6.: Sequenzdiagramm des Startupvorgangs der Messstation.

### 8.1.6. Busprotokoll

### 8.1.7. Filesystem

### 8.1.8. UART-Kommandozeile

## 8.2. Funktionalität

## 8.3. Konfiguration

## 8.4. Parallele Prozesse

## 9. Resultate

läuft das zeug?

wie viele sensore  
leiden?

wie schnell kann

wie lange kann m  
zeichnen?

was geht, was ni

was haben wir n  
reicht?

wo haben wir me  
gefordert erreich

## 10. Diskussion

ir erfüllt?

schwierigkei-

ind wir stolz

nte man jetzt  
och machen?

och geplant?

## **11. Verzeichnisse**

# Literaturverzeichnis

- [1] ARM Ltd. Cortex-M4 Processor. <http://www.arm.com/products/processors/cortex-m/cortex-m4-processor.php>, Stand: 7.12.2014.
- [2] NXP Semiconductors. Lpc4088fbd208. [http://www.nxp.com/products/microcontrollers/cortex\\_m4/lpc4000/LPC4088F](http://www.nxp.com/products/microcontrollers/cortex_m4/lpc4000/LPC4088F), Stand: 7.12.2014.
- [3] NXP Semiconductors. Lpc4088fbd208. <http://www.nxp.com/demoboard/OM13063.html>, Stand: 7.12.2014.
- [4] Autor unbekannt. Wikipedia: Hilbert-Transformation, 11 2014. <https://de.wikipedia.org/wiki/Hilbert-Transformation>, Stand: 6.12.2014.
- [5] '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](https://commons.wikimedia.org/wiki/File:FIR_Hilbert_Transform_Filter.svg) License: <https://creativecommons.org/licenses/by-sa/3.0/deed.en>.
- [6] Sigisbert Wyrsh. Vorlesung Digitale Signalverarbeitung 1. Kap. 3, DFT und FFT, S. 48.

# Abbildungsverzeichnis

3.1. Eine Messstation mit einem Datenlogger, der mehrere Sensoreinheiten im Bach steuert.	9
7.1. Hardwarekonzept des Datenloggers.	16
7.2. Hardwarekonzept der Sensoreinheit.	17
7.3. Chipdiagramm der <i>ARM Cortex-M4</i> Architektur [1].	19
7.4. Schematischer Hardware-Aufbau des Datenloggers.	24
7.5. Schematischer Hardware-Aufbau der Sensoreinheit.	25
8.1. Softwarestack des Datenloggers.	26
8.2. Softwarestack der Sensoreinheit.	27
8.3. Hilbert-Transformation als FIR-Filter [5].	28
8.4. Zustandsmaschine der Ereigniserkennung.	29
8.5. Parameter der Ereigniserkennung.	30
8.6. Sequenzdiagramm des Startupvorgangs der Messstation.	32
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

7.1. Fähigkeiten des NXP LPC4088 Prozessors [2]. . . . .	20
7.2. Zusätzliche Fähigkeiten des NXP LPC4088 QuickStart Boards von <i>Embedded Artists</i> [3].	20
7.3. Entscheidungsmatrix für die Auswahl des Bussystems. . . . .	21
7.4. Entscheidungsmatrix zur Auswahl des Speichermediums. . . . .	22

# 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. 24, *siehe* Quantisierung & Bit-Breite

**Abtastrate** Definiert, in welchen zeitlichen Abständen ein Messwert erfasst werden soll. Üblicherweise wird dieser Wert in  $Hz$  oder  $s^{-1}$  angegeben. 13, 24, 28

**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. 10

**Blockgrösse** Anzahl Werte, die in einer DFT oder IDFT verrechnet wird. 24

**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. 2, 7, 9, 11–13, 16–19, 36, *siehe* Signal

**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. 10, *siehe* PC & Embedded System

**Datenlogger** Ein Gerät zur Sammlung und Speicherung von Messdaten von mehreren Sensoreinheiten. 2, 7–13, 16, 17, 19–21, 23, 28, 35, *siehe* Sensoreinheit

**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 Alleskönner 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 & Mikrokontroller

**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. 7–13, 19, 24, 26

**Ereigniserkennung** Auswertung von Messdaten um definierte Signalformen (Ereignisse) zu erkennen. 24, 25, *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. 25

**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. 25, 35

**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. 6, 14

**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. 16, 17

**Hilbert-Transformation** Mathematische Umrechnung einer Wertfolge, um die umhüllende Kurve zu erhalten. 24, 25, 35

**Kommandozeile** Eine Eingabeaufforderung auf dem Bildschirm, wo der Benutzer über eine Tastatur Befehle eingeben kann, die vom Computer interpretiert und ausgeführt werden. 10, *siehe* Computer

**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. *siehe* Pin

**Nullpegel** Signalpegel, wenn keine Beschleunigung gemessen wird. 27

**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. 11, 27

**Pin** Ein Anschluss an einem Chip oder einer Leiterplatte. Über Pins werden elektronische Bauteile miteinander verbunden und Peripheriegeräte wie z.B. Sensoren angeschlossen. 24, *siehe* Sensor

*siehe* Bit-Breite

**Sensor** Ein Messgerät für physikalische Größen wie Temperatur, Feuchtigkeit, Luftdruck oder Beschleunigung. 4, 5, 7, 8, 10–13, 17–19, 24, 28, 30

**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. 2, 9–14, 16–18, 22, 24, 35, *siehe* , Sensor & Datenlogger

**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. 24, *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

**Speichermedium** Ein Stück Hardware, auf dem Daten gespeichert werden können. 16

**Threshold** Englisch für Schwellenwert. 11, 24, 27, 28

**Timeout** Englisch für Zeitüberschreitung. 11, 27

**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. 2, 3, 11–13, 28

# Akronyme

**CD** Compact Disc VII, XIII

**DFT** diskrete Fourier-Transformation 24, 25

**FSM** Finite State Machine 25, 27, 28

**ID** Identifikationsnummer 10, 12

**IDFT** inverse diskrete Fourier-Transformation 24, 25

**PC** Personal Computer 12, 38

**WSL** Eidg. Forschungsanstalt für Wald, Schnee und Landschaft 6, 24

## Listings

# **A. 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](mailto:bruno.fritschi@wsl.ch) (Projektidee)

Dieter Rickenmann, [dieter.rickenmann@wsl.ch](mailto:dieter.rickenmann@wsl.ch) (Betreuung)

**Gruppengrösse:** 2,

**Reserviert für:** Tobias Welti (weltitob), Tobias Keller (kelleto1)

### **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  
Besprechungsprotokolle  
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

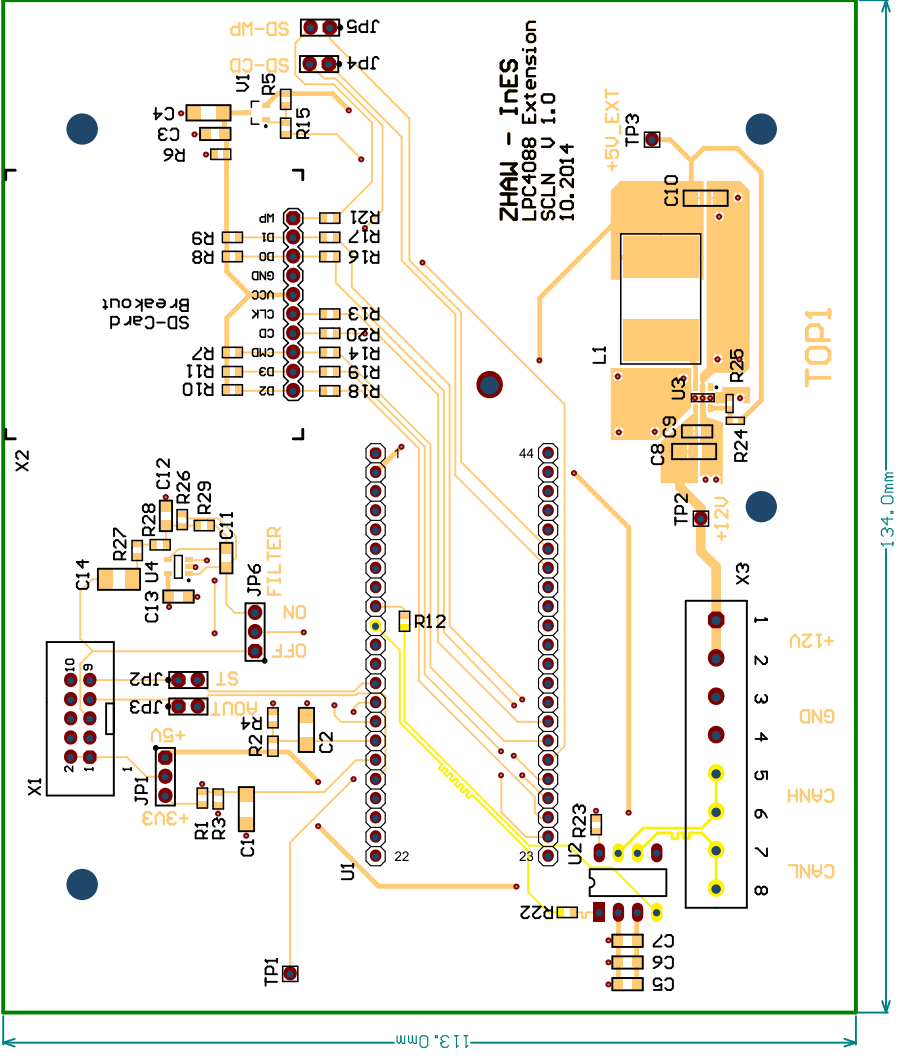
## A

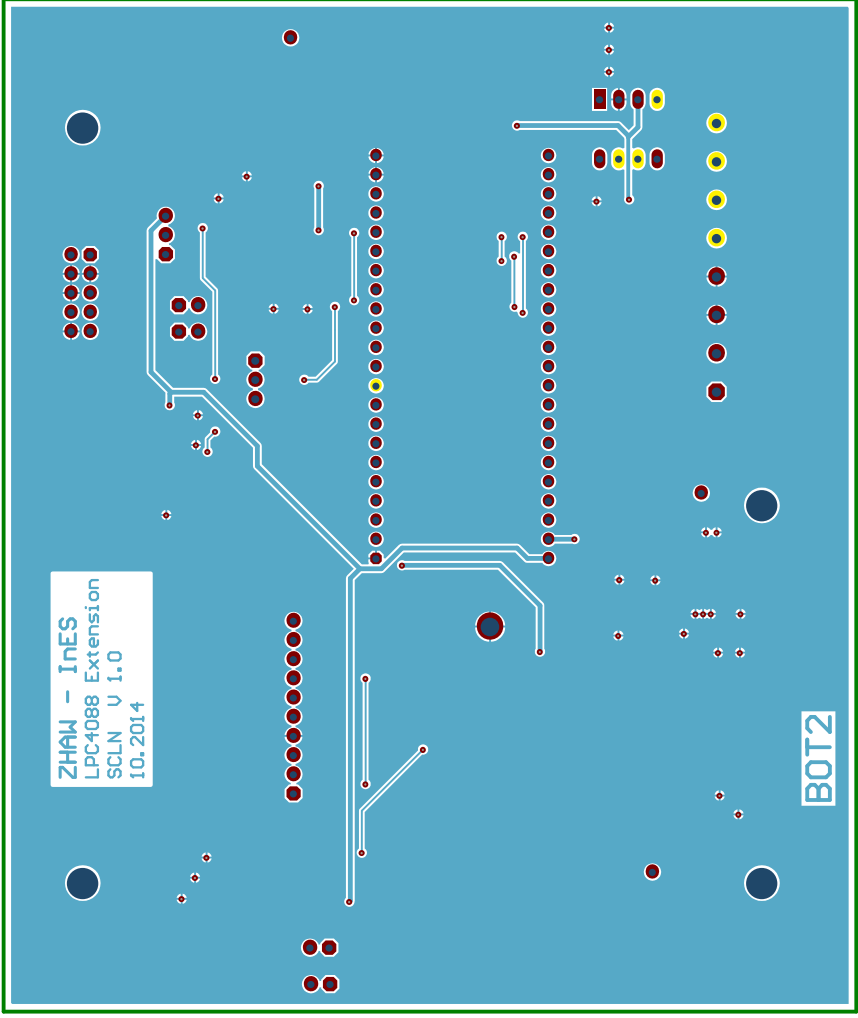


100



	D
--	---



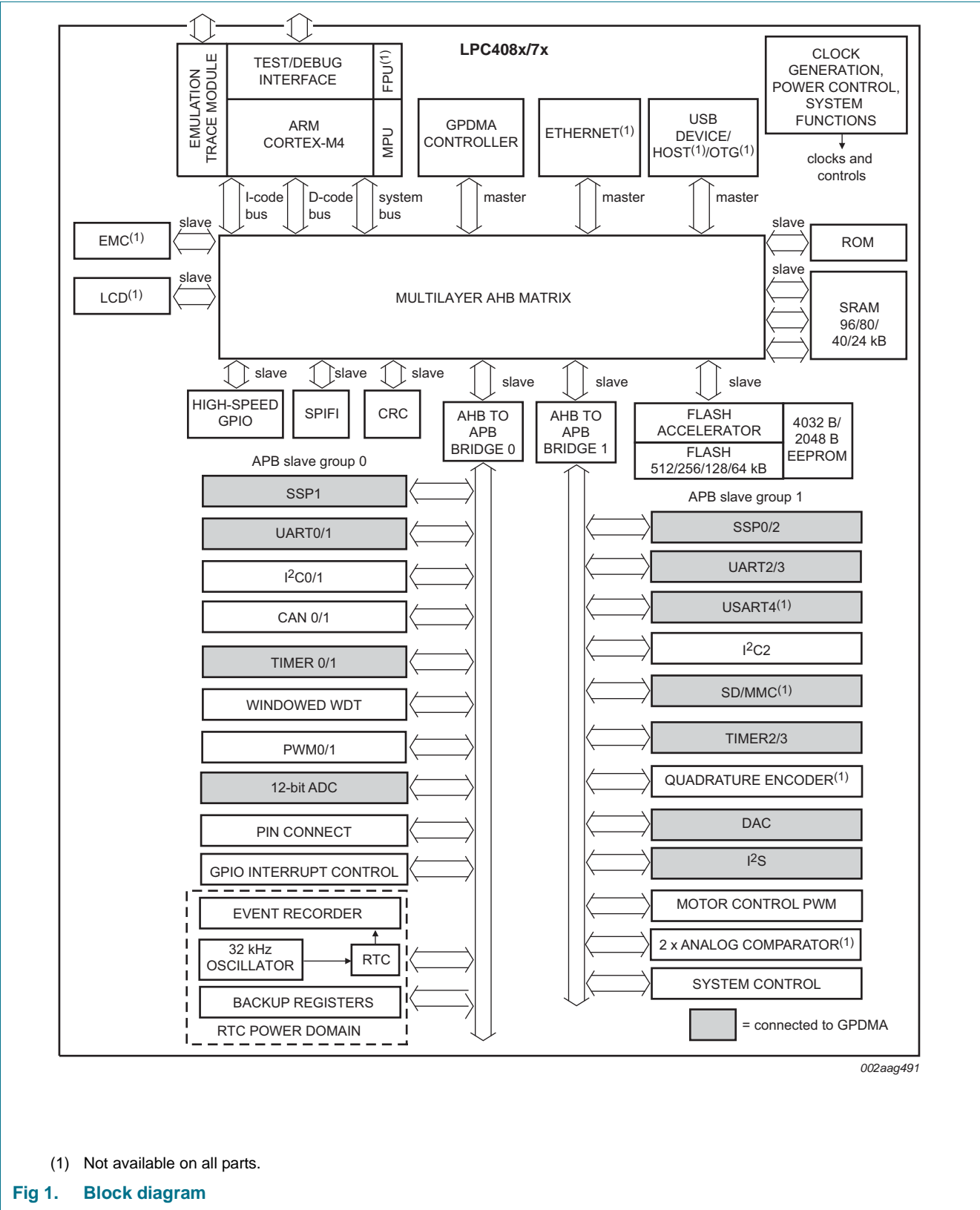


## **A.5. Datenblätter**

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



### A.5.2. Embedded Artists NXP LPC4088 QuickStart Board

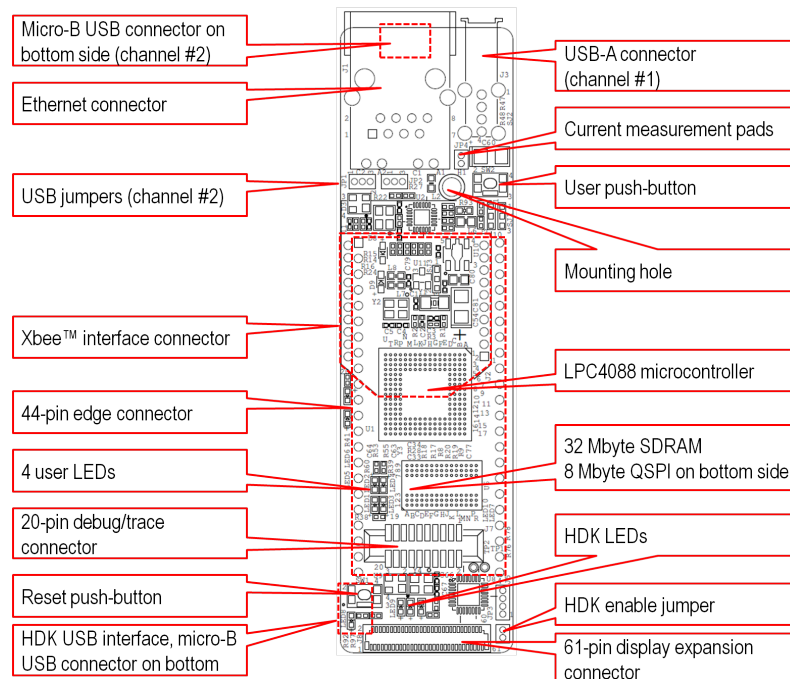


Abbildung A.1.: Hauptkomponenten des NXP LPC4088 QuickStart Boards von Embedded Artists.

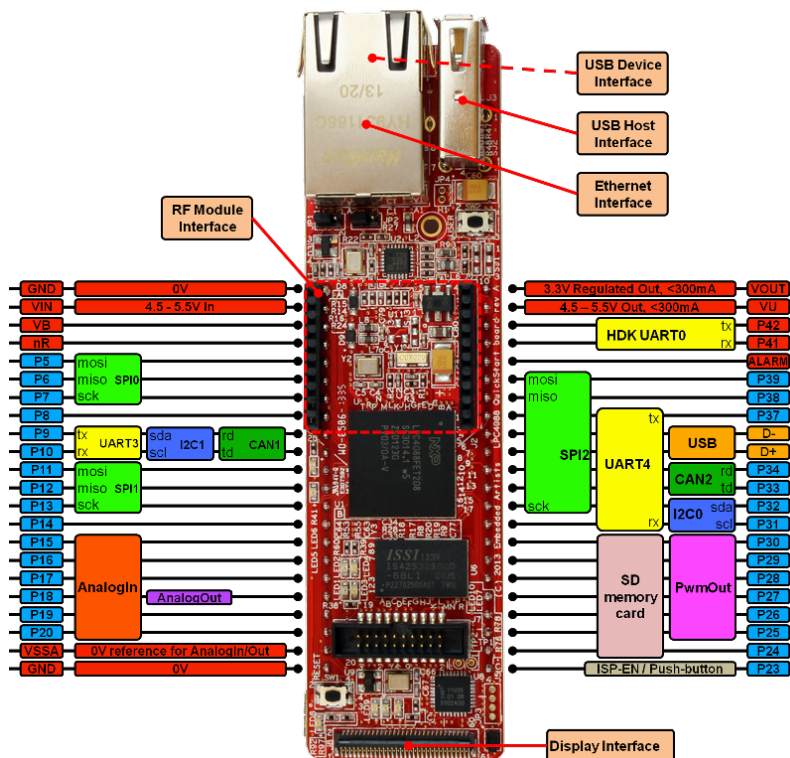


Abbildung A.2.: Pins des NXP LPC4088 QuickStart Boards von Embedded Artists.

(https://mbed.org/)

[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 [\(https://developer.mbed.org/users/embeddedartists/\)](https://developer.mbed.org/users/embeddedartists/) **EmbeddedArtists** [AB\(https://developer.mbed.org/users/embeddedartists/\)](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: [\(https://developer.mbed.org/users/embeddedartists/\)](https://developer.mbed.org/users/embeddedartists/) **EmbeddedArtists** [AB\(https://developer.mbed.org/users/embeddedartists/\)](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.

Table 1:											
Parameter	Conditions	ADXL001-70			ADXL001-250			ADXL001-500			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
SENSOR											
Nonlinearity	Includes package alignment		0.2	2		0.2	2		0.2	2	%
Cross-Axis Sensitivity			2			2			2		%
Resonant Frequency			22			22			22		kHz
Quality Factor			2.5			2.5			2.5		
SENSITIVITY											
Full-Scale Range	$I_{OUT} \leq \pm 100\text{ }\mu\text{A}$	−70		+70	−250		+250	−500		+500	g
Sensitivity	100 Hz		16.0			4.4			2.2		mV/g
OFFSET											
Zero-g Output	Ratiometric	1.35	1.65	1.95	1.35	1.65	1.95	1.35	1.65	1.95	V
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		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	To ground		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		30	50		30	50		30	50		k $\Omega$
OUTPUT AMPLIFIER											
Output Swing	$I_{OUT} = \pm 100\text{ }\mu\text{A}$	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 u  
tenblätter der ve  
ten Messgeräte u  
der Komponente  
Berechnungen, M  
te, Simulationsre  
Grafische Darste  
Fotos

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