



**School of  
Engineering**

InES Institute of  
Embedded Systems

## **Konzept**

# Messstation zur Registrierung von Geschiebe- Bewegungen im Fluss

---

**Autoren**

Tobias Keller  
Tobias Welti

---

**Datum**

15.09.2014

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>4</b>
1.1. Überblick . . . . .	4
1.2. Datenlogger . . . . .	4
1.2.1. Messdaten sammeln . . . . .	5
1.2.2. Kontrolle über das Bussystem . . . . .	5
1.2.3. Steuerung des Betriebs . . . . .	5
1.2.4. Schnittstelle nach Aussen . . . . .	5
1.3. Sensoreinheit . . . . .	5
1.3.1. Messdatenerfassung . . . . .	5
1.3.2. Ereigniserkennung . . . . .	5
1.3.3. Datenübertragung . . . . .	6
1.4. Bussystem . . . . .	6
<b>2. Hardware-Konzept</b>	<b>7</b>
2.1. Hardware-Architektur . . . . .	7
2.1.1. Datenlogger . . . . .	7
2.1.2. Sensoreinheit . . . . .	7
2.1.3. Bussystem . . . . .	7
2.2. Komponentenauswahl . . . . .	7
2.2.1. Mikroprozessor . . . . .	7
2.2.2. Bus-System . . . . .	8
2.2.3. Speichermedium . . . . .	9
2.2.4. Sensor . . . . .	11
2.2.5. Schnittstelle . . . . .	11
2.3. Komponenten . . . . .	11
2.3.1. Cortex M4 Mikroprozessor . . . . .	11
2.3.2. Beschleunigungs-Sensor . . . . .	11
2.3.3. CAN Bus . . . . .	11
2.3.4. SD Karte . . . . .	11
2.3.5. UART Schnittstelle . . . . .	11
2.4. Datenlogger . . . . .	11
2.5. Sensoreinheit . . . . .	12
<b>3. Software-Konzept</b>	<b>13</b>
3.1. Software-Stack . . . . .	13
3.1.1. Überblick . . . . .	13
3.1.2. Messdatenerfassung . . . . .	13
3.1.3. Ereigniserkennung . . . . .	13
3.1.4. Timestamp . . . . .	14
3.1.5. Busprotokoll . . . . .	14
3.1.6. Filesystem . . . . .	14
3.1.7. UART-Kommandozeile . . . . .	14
3.2. Funktionalität . . . . .	14
3.3. Konfiguration . . . . .	14
<b>4. Verzeichnisse</b>	<b>15</b>
Literaturverzeichnis . . . . .	16
(Abbildungsverzeichnis) . . . . .	17

(Tabellenverzeichnis) . . . . .	18
4.1. (Glossar) . . . . .	19
(Abkürzungsverzeichnis) . . . . .	19
(Listingverzeichnis) . . . . .	20
<b>A. Anhang</b>	<b>I</b>
A.1. Projektmanagement . . . . .	I
A.2. Weiteres . . . . .	I

## Liste der noch zu erledigenden Punkte

Tabelle verschiedener uCs zum Vergleich . . . . .	7
Literatur-Referenzen in Tabelle 2.2 . . . . .	9
Abbildung: Abb. der Software (in UML) . . . . .	13
Abbildung: Abbildung der State Machine der Ereigniserkennung . . . . .	13
Ereigniserkennung beschreiben . . . . .	13
Busprotokoll austüfteln. Darstellung siehe HW-Konzept Rioxo . . . . .	14

# 1. Einleitung

## 1.1. 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 Vibrationen, werten die Signale aus und erkennen Ereignisse, die einer vordefinierten Signalform entsprechen. Alle Einheiten sind über ein Bussystem verbunden, um miteinander kommunizieren zu können. Der prinzipielle Aufbau ist in Abbildung 1.1 ersichtlich. Die Stromversorgung der Anlage wird am Datenlogger angeschlossen. Parallel zum Kabel des Datenbusses wird die Stromversorgung der Sensoreinheiten geführt.

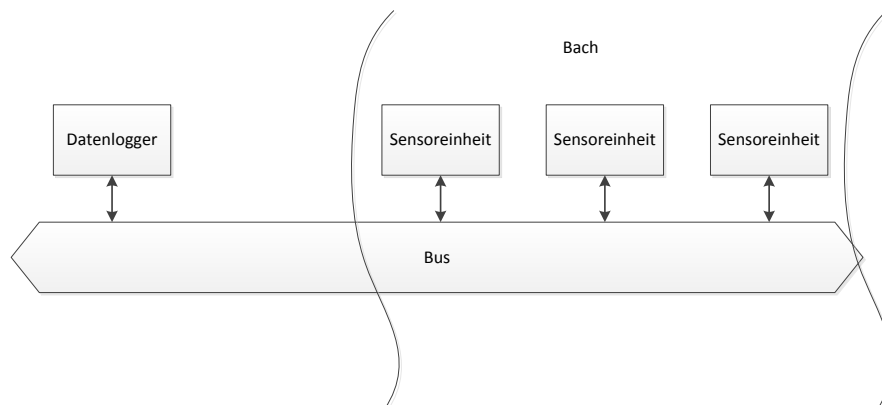


Abbildung 1.1.: Eine Messstation mit einem Datenlogger, der mehrere Sensoreinheiten im Bach steuert.

Diese drei Einheiten werden im Folgenden genauer definiert.

## 1.2. 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.

### 1.2.1. 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.

### 1.2.2. 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. 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.

### 1.2.3. Steuerung des Betriebs

Die Messstation hat verschiedene Betriebsmodi, die über den Datenlogger angewählt werden können. Der Datenlogger steuert die einzelnen Sensoreinheiten entsprechend an.

### 1.2.4. 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.

## 1.3. Sensoreinheit

Die Aufgaben der Sensoreinheit umfassen:

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

### 1.3.1. 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.

### 1.3.2. Ereigniserkennung

Im Mikroprozessor werden die Messdaten fortlaufend analysiert. Überschreitet das gemessene Signal einen gewissen Pegel (Threshold), markiert dies den Beginn eines Ereignisses. Das Ereignis ist beendet, wenn der Signalpegel für eine gewisse Zeit 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 aller Peaks übertragen werden.

### **1.3.3. 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.

## **1.4. 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.

## 2. Hardware-Konzept

### 2.1. 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.

#### 2.1.1. Datenlogger

Der Datenlogger sammelt die Daten der Sensoreinheiten über das Bussystem ein und speichert sie ab. Dafür benötigt er das Bussystem, 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.

#### 2.1.2. Sensoreinheit

Die Sensoreinheit benötigt einen Beschleunigungssensor, um die Einschläge von Geschosse zu messen. Über einen Analog-Digital-Wandler (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.

#### 2.1.3. 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.

## 2.2. Komponentenauswahl

### 2.2.1. Mikroprozessor

Bei der Auswahl des Mikroprozessors werden folgende Kriterien berücksichtigt:

- Rechenleistung genügend für allfällige zusätzliche Anforderungen.
- Analog-Digital-Wandler mit genügender Abtastrate und Auflösung.
- Digitaler Signal Prozessor integriert für die Verarbeitung der Messdaten.
- Ein-/Ausgänge für das Bussystem.
- Ein-/Ausgänge für den externen Speicher.
- möglichst geringer Stromverbrauch.



	Bitrate	Distanz	Clients	Besonderheiten
<b>CAN</b>	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.
<b>SPI</b>	..100 MBit/s	< 1 m	slave select	- Pro Client eine Slave Select Leitung - Daisy Chain $\Rightarrow$ alle MC beschäftigt. - Bei Ausfall eines MC ganzer Bus unterbrochen.
<b>RS485</b>	35 MBit/s 100 kBit/s	10 m 1200 m	>32	- Master am besten in der Mitte des Bus $\Rightarrow$ ungünstig. - Braucht 2..4 Drähte (bei Full Duplex) - braucht pull-up und pull-down Widerstände $\Rightarrow$ mehr Leistungsaufnahme.
<b>Ethernet</b>	100 MBit/s	100 m	> 20	+ Stromversorgung bei Power over Ethernet (PoE) integriert. - kein Bus sondern allenfalls Daisychain. - bei Daisychain kein PoE möglich.
<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 2.1.: Entscheidungsmatrix für die Auswahl des Bussystems.

### 2.2.2. 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 2.1 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.

	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 2.2.: Entscheidungsmatrix zur Auswahl des Speichermediums.

**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. Für CAN-Bus sind Bus-Treiber (Transceiver) erhältlich, die mit hohen Spannungen umgehen können, was das Bussystem robuster gegenüber Umwelteinflüssen macht.

### 2.2.3. 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..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 wasserdichten 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 2.2 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.

#### 2.2.4. Sensor

#### 2.2.5. Schnittstelle

### 2.3. Komponenten

#### 2.3.1. Cortex M4 Mikroprozessor

Flash Speicher

SDRAM

#### 2.3.2. Beschleunigungs-Sensor

#### 2.3.3. CAN Bus

CAN Transceiver

#### 2.3.4. SD Karte

#### 2.3.5. UART Schnittstelle

### 2.4. Datenlogger

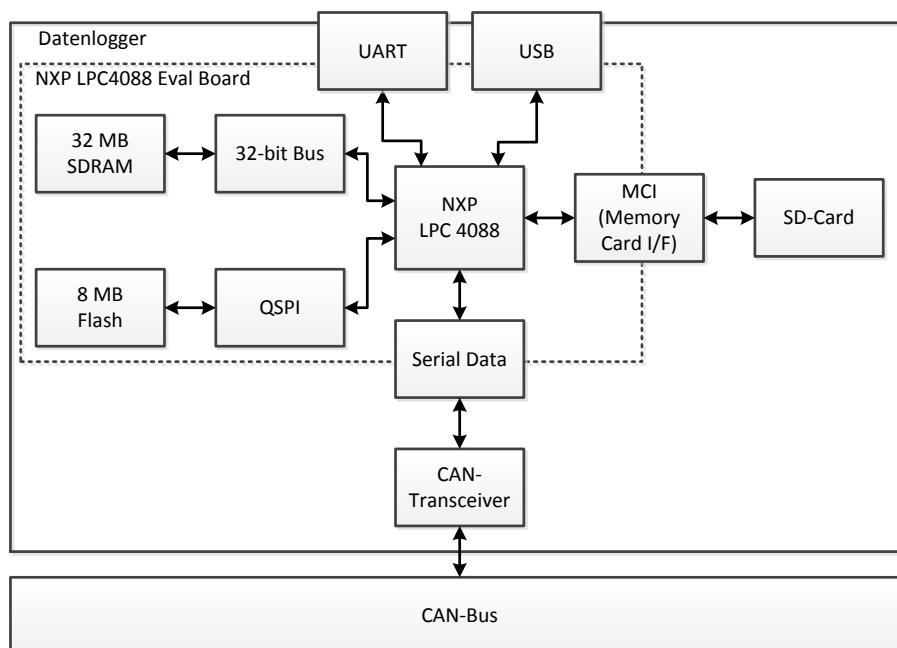


Abbildung 2.1.: Schematischer Hardware-Aufbau des Datenloggers.

## 2.5. Sensoreinheit

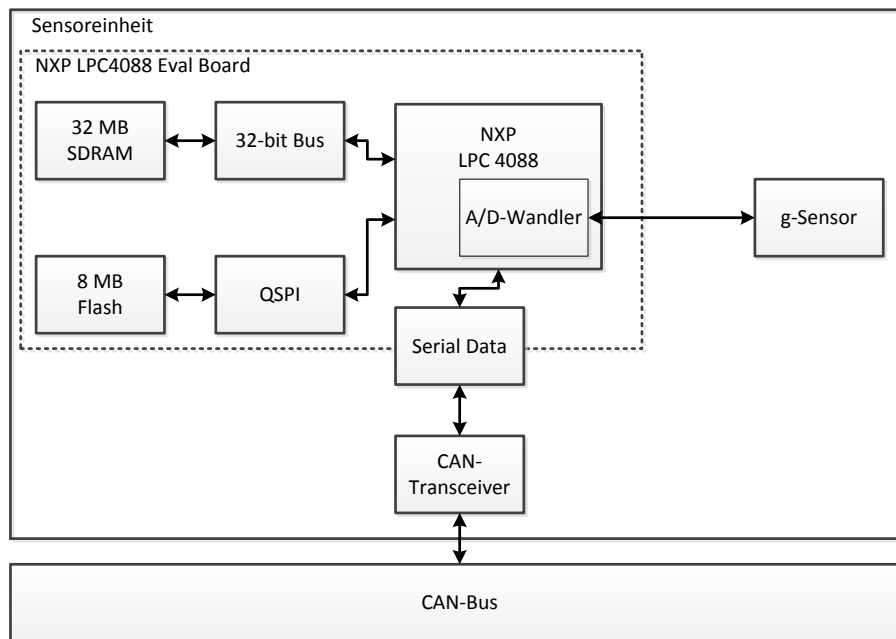
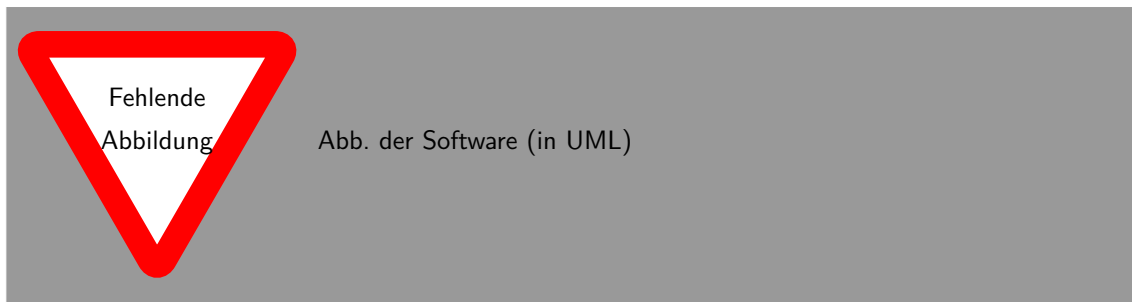


Abbildung 2.2.: Schematischer Hardware-Aufbau der Sensoreinheit.

## 3. Software-Konzept

### 3.1. Software-Stack

#### 3.1.1. Überblick

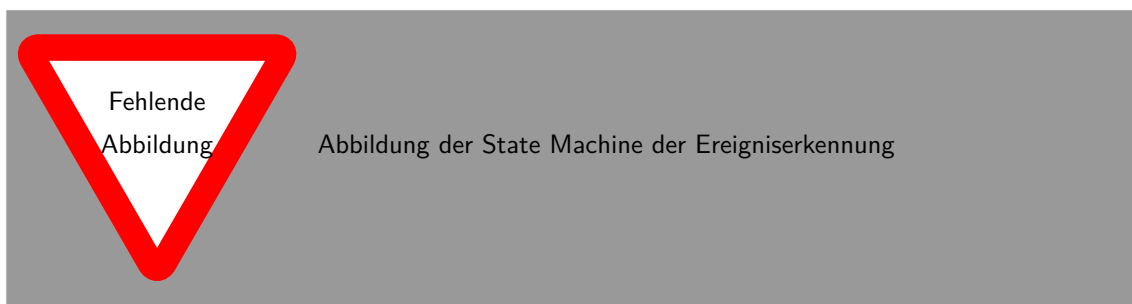


#### 3.1.2. Messdatenerfassung

Der NXP LPC4088 verfügt über einen 12-bit A/D-Wandler, der über einen Multiplexer auf acht Pins messen kann. Auf dem Quickstart Board stehen 6 Pins für A/D-Wandlung zur Verfügung. Für die geplante Anwendung reicht ein A/D-Eingang, da der g-Sensor die Beschleunigung nur auf einer Achse misst. Der A/D-Wandler des NXP LPC4088 wird mit einer Samplingrate von 10 kHz betrieben. Falls höhere Samplingraten nötig sind, kann der A/D-Wandler mit bis zu 400 kHz betrieben werden.

#### 3.1.3. Ereigniserkennung

Vom 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. Da die Hilbert-Transformation grossen Rechenaufwand benötigt, lösen wir die Ereigniserkennung einfacher.



Ereigniserkennung  
schreiben

**3.1.4. Timestamp****3.1.5. Busprotokoll**

okoll austüfteln.  
ing siehe HW-  
Rioxo

**3.1.6. Filesystem****3.1.7. UART-Kommandozeile****3.2. Funktionalität****3.3. Konfiguration**

## **4. Verzeichnisse**



# Literaturverzeichnis

- [1] B. Klaus and P. Horn, Robot Vision. Cambridge, MA: MIT Press, 1986.
- [2] L. Stein, »Random patterns,» in Computers and You, J. S. Brake, Ed. New York: Wiley, 1994, pp. 55-70.

# Abbildungsverzeichnis

1.1. Eine Messstation mit einem Datenlogger, der mehrere Sensoreinheiten im Bach steuert.	4
2.1. Schematischer Hardware-Aufbau des Datenloggers. . . . .	11
2.2. Schematischer Hardware-Aufbau der Sensoreinheit. . . . .	12

# Tabellenverzeichnis

2.1. Entscheidungsmatrix für die Auswahl des Bussystems. . . . .	8
2.2. Entscheidungsmatrix zur Auswahl des Speichermediums. . . . .	9

## 4.1. (Glossar)

In diesem Abschnitt werden Abkürzungen und Begriffe kurz erklärt.

Abk	Abkürzung
XY	Ix Ypsilon
YZ	Ypsilon Zet

## Listings

# A. Anhang

## A.1. Projektmanagement

- Offizielle Aufgabenstellung, Projektauftrag
- (Zeitplan)
- (Besprechungsprotokolle oder Journals)

## A.2. Weiteres

- CD mit dem vollständigen Bericht als pdf-File inklusive Film- und Fotomaterial
- (Schaltpläne und Ablaufschemata)
- (Spezifikationen u. Datenblätter der verwendeten Messgeräte und/oder Komponenten)
- (Berechnungen, Messwerte, Simulationsresultate)
- (Stoffdaten)
- (Fehlerrechnungen mit Messunsicherheiten)
- (Grafische Darstellungen, Fotos)
- (Datenträger mit weiteren Daten (z.B. Software-Komponenten) inkl. Verzeichnis der auf diesem Datenträger abgelegten Dateien)
- (Softwarecode)