

# Pflichtenheft

Projekt: Low power wakeup receiver

Version 0.1

Cédric Renda, Manuel Tischhauser

Name	Datum	Unterschrift
Prof. Dr. Heinz Mathis		
Selina Malacarne		
Cédric Renda		
Manuel Tischhauser		

## Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>3</b>
<b>2. Auftrag</b>	<b>3</b>
<b>3. Funktionale Anforderungen</b>	<b>3</b>
3.1. Überblick über die Systemfunktionen . . . . .	3
3.2. Actors . . . . .	4
3.3. Kurzbeschreibung der Use Cases . . . . .	4
3.4. <Use Case Name> . . . . .	4
<b>4. Sonstige Anforderungen</b>	<b>6</b>
<b>A. Referenzen</b>	<b>7</b>

## 1. Einleitung

Im Gebäudemanagement ist es üblich, Belegungspläne an den Eingängen der Räume anzubringen. Oftmals sind diese in Papierformat und müssen bei einer Änderung von Hand gewechselt werden. Mit dieser Methode werden kurzfristige Belegungen nicht aufgezeigt. Dies könnte man umgehen, wenn man mit Displays arbeitet, die über eine drahtlose Schnittstelle aktualisiert werden können. Dabei stellt sich allerdings das Problem, dass man entweder Kabel für die Netzeinspeisung verlegen muss, oder Batterien verwendet, die regelmässig ersetzt werden müssen. Im Idealfall entfällt die Speisung komplett.

## 2. Auftrag

Im Rahmen dieser Semesterarbeit sollen folgende Punkte abgearbeitet werden:

- Recherche bezüglich Schnittstelle und Energy Harvesting.
- Vor- und Nachteile bestehender Technologien abwägen und geeignete Hardware wählen.
- Erstellen eines lauffähigen Prototypen.

## 3. Funktionale Anforderungen

Die funktionalen Anforderungen sind die wichtigsten Teile zur Beschreibung des Systems. Für die Beschreibung der funktionalen Anforderungen bieten sich Use Cases an, nicht nur für Software, sondern auch für das ganze System. Jeder Use Case beschreibt dabei eine Funktion. Im folgenden Raster ist vorgesehen, dass bei jedem Use Case auch nicht funktionale Anforderungen wie Antwortzeiten, etc. direkt beim Use Case stehen, falls diese zu dieser Funktion gehören.

Bei der Use Case Definition ist wichtig, dass die Granularität nicht zu fein gewählt wird. Allfällige Ausnahmefälle einer Funktion sollen beispielsweise in der Beschreibung des entsprechenden Use Cases geschehen und nicht allenfalls in einem "Unter-Use Case".

Statt mittels Use Cases kann dieses Kapitel auch mit **Systemfunktion 1**, **Systemfunktion 2**, etc. gegliedert werden.

### 3.1. Überblick über die Systemfunktionen

Hier kommt ein Use Case Diagramm hinein, das alle Funktionen zeigt. Allenfalls genügt ein Link zum Abschnitt Systemübersicht.

### 3.2. Actors

Kurzbeschreibung der Actors.

### 3.3. Kurzbeschreibung der Use Cases

Jeder einzelne Use Case soll kurz beschrieben werden.

### 3.4. <Use Case Name>

Ist für jeden Use Case zu beschreiben. Die folgenden Unterkapitel sollen für jeden einzelnen Use Case vollständig vorhanden sein. Falls bei einem Abschnitt nichts zu schreiben ist, dann soll dies entsprechend vermerkt werden, z.B. falls ein Use Case keine Vorbedingungen braucht oder keine nicht-funktionalen Anforderungen vorhanden sind, kann bei diesem Abschnitt einfach das Wort "keine" stehen.

#### 3.4.1. Vorbedingungen

Zustand des Systems bevor der Use Case eintritt, z.B. kann hier stehen, dass ein System erfolgreich initialisiert sein muss, damit diese Funktion ausgeführt werden kann.

#### 3.4.2. Nachbedingungen

Zustand des Systems nachdem der Use Case durchlaufen ist, z.B. kann hier bei einem Kalibrations-Use Case stehen, dass das System kalibriert oder allenfalls in einem Fehlerzustand ist.

#### 3.4.3. Nicht-funktionale Anforderungen

Zusicherungen, die für Design und Realisierung wichtig sind, wie z.B. Antwortzeit, Häufigkeit, Priorität usw.

### 3.4.4. Hauptszenario

Beschreibung des Use Cases, ggf. gegliedert in Einzelpunkte. Beschrieben wird der Normalfall. Variationen werden mit Unterszenario-Nummer erwähnt (*[S-1]*, *[S-2]*, usw.) und separat als Unterszenarien beschrieben. Fehlerfälle werden mit Fehlerszenario-Nummern angegeben (*[E-1]*, *[E-2]*, usw.) und separat als Fehlerszenarien beschrieben.

Beispiele: Falls gewünscht können zusätzliche Informationen erfasst werden *[S-1]*.

Beim Einlesen der Daten können die Fehler *[E-1]*, *[E-2]* oder *[E-3]* auftreten.

### 3.4.5. Unterszenarien

*[S-1]* Zusatzinformationen erfassen  
*[S-2]* ...

#### 3.4.5.1. *[S-1]* Zusatzinformationen erfassen

#### 3.4.5.2. *[S-2]* ...

### 3.4.6. Fehlerszenarien

*[E-1]* Daten nicht verfügbar  
*[E-2]* Falsches Datenformat  
*[E-3]* ...

#### 3.4.6.1. *[E-1]* Daten nicht verfügbar

#### 3.4.6.2. *[E-1]* Falsches Datenformat

#### 3.4.6.3. *[E-2]* ...

### 3.4.7. Regeln

Gültigkeits- und Validierungsregeln, Berechnungsformeln usw.

### 3.4.8. Anmerkungen

### 3.4.9. Beispiele

## 4. Sonstige Anforderungen

In diesem Kapitel sollen alle bisher noch nicht spezifizierten Anforderungen, möglicherweise in Unterabschnitten, definiert werden, z.B.

- Leistungsanforderungen (Performance), die nicht direkt einer Systemfunktion zugeordnet werden konnten.
- Entwurfseinschränkungen (Design Constraints): Anforderungen, welche die Entwickler bei der Wahl des Designs einschränken.
- Qualitätsanforderungen wie Zuverlässigkeit, Verfügbarkeit, Sicherheit (Safety und Security, ist nicht dasselbe!)
- Wartbarkeit
- Portabilität
- Logging und Tracing
- Service und Support

## A. Referenzen

- [1] IEEE. *Recommended Practice for Software Requirements Specification*. ANSI/IEEE Std 830-1998, 1998.

Do not repeat items covered in other documents or in a global project definitions and acronyms document