

Pflichtenheft

Für das Projekt „Sting – Structure Monitoring“

beauftragt von der Systecs GmbH

bearbeitet von Enes Akkus, Marc Bauer,   
Boris Garkusha und Robin Müller   
von der Hochschule Esslingen



Inhalt

[1 Zielbestimmung 3](#_Toc532831108)

[1.1 Musskriterien 3](#_Toc532831109)

[1.2 Wunschkriterien 3](#_Toc532831110)

[1.3 Abgrenzungskriterien 3](#_Toc532831111)

[2 Produkteinsatz 4](#_Toc532831112)

[2.1 Anwendungsbereiche 4](#_Toc532831113)

[2.2 Zielgruppen 4](#_Toc532831114)

[2.3 Betriebsbedingungen 4](#_Toc532831115)

[3 Produktdaten 4](#_Toc532831116)

[4 Qualitätsanforderungen 5](#_Toc532831117)

[5 Technische Produktumgebung 5](#_Toc532831118)

[5.1 Software 5](#_Toc532831119)

[5.2 Hardware 5](#_Toc532831120)

# 1 Zielbestimmung

## Musskriterien

* Das Unternehmen Systecs Informationssysteme GmbH soll durch das Produkt in der Lage sein, Messdaten von Räumen mit einem Raspberry Pi zu erheben
* Man soll in der Lage sein, Messdaten von Räumen auf einer Website aufzurufen
* Die Messdaten sollen in einer Webapp angezeigt werden
* Es sollen Daten zur Temperatur und Luftfeuchtigkeit erhoben werden
* Es sollen Intervalle für die Messdaten eingestellt werden können, in denen die Messdaten bleiben sollen
* Eine Statusübersicht über die Räume wird verfügbar gemacht, in der erkennbar wird ob die Räume in den gegebenen Wertgrenzen liegen
* Neue Daten werden dauerhaft vom Pi generiert und übertragen in regelmäßigen Intervallen
* Die erhobenen Daten sollen in Azure verwaltet werden
* Die Webapp wird über einen Azure Service gehostet
* Azure soll die Daten an die Webapp weiterleiten
* Die App soll mit beliebig vielen Raspberry Pis betrieben werden können (im Rahmen der Limitationen von Azure und der verwendeten technischen Produktumgebung)
* Dokumentation der API des Raspberry Pi
* Der Source Code soll kommentiert werden, um eine leichtere Instandhaltung zu ermöglichen
* Die Webapp soll mit einem Mobile First Ansatz entwickelt werden
* Die Webapp soll ein responsive Design besitzen, um auf einer großen Anzahl von Geräten gut benutzbar zu sein
* Die Webapp soll eine englische Benutzeroberfläche besitzen

## Wunschkriterien

* Messdaten in einem Datenbankdienst von Azure speichern
* Erstellung von Graphen mit den historischen Daten
* Cloud-to-Device Methods, um von der Weboberfläche aus Aktionen auf dem Pi auslösen zu können
* Ausstattung des Pis mit mehr Komponenten zur Messung von weiteren Parametern und zur Indizierung des Status
* Messungen auf dem Pi können in sehr geringen Intervallen (<= 3s) ausgeführt werden
* Skalierbarkeit der Webapp und C# Anwendung
* Die Webapp soll auf möglichst vielen weit verbreiteten Browsern laufen

## Abgrenzungskriterien

* Es soll kein Support für die Anwendung übernommen werden
* Die C# Anwendung soll nur für den Raspberry Pi optimiert sein
* Für den Raspberry Pi soll keine graphische Oberfläche erstellt werden

# Produkteinsatz

Es soll ein Programm, das auf Windows 10 IoT Devices ausgeführt werden kann, programmiert werden. Dieses Programm soll Messdaten durch angeschlossene Sensoren aufnehmen und diese an die Azure Cloud senden. Das Programm soll in C# programmiert sein.

Es soll eine Webapp programmiert werden. Diese soll den Nutzer Raumdaten aus der Azure Cloud Diese soll Messdaten von der Azure Cloud lesen und ausgeben.

## 2.1 Anwendungsbereiche

Überwachung von gewerblichen und auch privaten Räumen, Etagen und Gebäuden.

## 2.2 Zielgruppen

Nutzer mit einem grundlegenden Verständnis von der eingesetzten Hardware, welche in der Lage sind ein Betriebssystem zu installieren.

## 2.3 Betriebsbedingungen

Raspberry Pi:

* in trockener Umgebung
* Temperaturgrenzen der Hardware werden eingehalten
* Zugang zum Internet
* Angeschlossene Sensoren
* Windows IoT Core Betriebssystem

Webapp:

* Internetfähiges Endgerät
* Web browser (Edge, Chrome, Opera, Firefox)

# 3 Produktdaten

Es werden folgende Daten für die Anwendung aufgezeichnet und persistent gespeichert:

* Gemessene Temperatur
* Gemessene Luftfeuchtigkeit
* Gemessener Luftdruck
* Höhe über dem Meeresspiegel
* Datum und Uhrzeit der Messungen

# 4 Qualitätsanforderungen

|  |  |  |  |
| --- | --- | --- | --- |
| Produktqualität | sehr gut | gut | normal |
| **Funktionalität** |  |  |  |
| Angemessenheit | **X** |  |  |
| Richtigkeit |  | **X** |  |
| Ordnungsmäßigkeit |  | **X** |  |
| **Zuverlässigkeit** |  |  |  |
| Fehlertoleranz |  | **X** |  |
| Wiederherstellbarkeit |  |  | **X** |
| **Benutzbarkeit** |  |  |  |
| Verständlichkeit | **X** |  |  |
| Erlernbarkeit | **X** |  |  |
| Bedienbarkeit | **X** |  |  |

# 5 Technische Produktumgebung

## 5.1 Software

Raspberry Pi:

* Windows IoT Core
* C#
* .Net Standard / UWP

Azure:

* Azure Tables
* Azure IoT Hub
* Azure App Services

Client:

* TypeScript
* Bootstrap 4
* Angular 7

Entwicklung:

* Visual Studio Enterprise
* Visual Studio Code
* Git
* GitHub

## 5.2 Hardware

* Raspberry Pi
* Sensoren für Raspberry Pi