

Wetterstation mit Raspberry Pi

Version: 0.5

Datum: 26.10.14

Projektteam: Andreas Hasler / David Daniel

Inhaltsverzeichnis

1	Informationen zum Dokument	4
1.1	Zweck des Dokuments.....	4
1.2	Versionskontrolle	4
1.3	Referenzierte Dokumente	4
2	Projektdefinition	5
3	Anforderungen	5
3.1	Funktionale Anforderungen.....	5
3.2	Nicht funktionale Anforderungen.....	6
4	Kontextdiagramm.....	6
5	Terminplan.....	7
6	Use Cases	8
6.1	Diagramm	8
6.2	Beschreibungen.....	9
7	Grobentwurf	14
7.1	Hardware	14
7.1.1	Direkte Anbindung	14
7.1.2	Indirekte Anbindung.....	14
7.2	Steuerung / Online-Schnittstelle.....	15
7.2.1	Webseite.....	15
7.2.2	Webservice	16
7.2.3	Smartphone-App mit Webservice-Zugriff	16
7.3	Lösungsfindung	16
7.3.1	Hardware	16
7.3.2	Software	17
8	Detailentwurf.....	18
8.1	Hardware / Schaltung	18
8.2	Steuerung	18
8.3	Webseite.....	18

9	Testkonzept	19
---	-------------------	----

1 Informationen zum Dokument

1.1 Zweck des Dokuments

Dieses Dokument beinhaltet die Projektdokumentation zum Projekt *Wetterstation*, welches im Zuge des 9. Semesters im Fach Embedded Systems und Hardware Hacking an der FFHS umgesetzt wurde.

1.2 Versionskontrolle

Ausgabe	Datum	Autor	Bemerkungen
0.1	27.09.2014	Andreas Hasler	Initialversion
0.2	28.09.2014	Andreas Hasler	Anpassungen Anforderungen und Terminplan
0.3	29.09.2014	Andreas Hasler	Anpassungen Anforderungen
0.4	10.10.2014	Andreas Hasler	Use-Cases hinzugefügt
0.5	25.10.2014	Andreas Hasler	Grobentwurf / Lösungsfindung

1.3 Referenzierte Dokumente

Dokument / Bemerkungen
Präsenz Block 2 (27.09.2014) mit der Aufgabenstellung auf Seite 11

2 Projektdefinition

Mit dem Raspberry Pi soll eine Wetterstation erstellt werden, welche Wetterdaten (Luftdruck, Temperatur, Feuchtigkeit und Lichtstärke) ermittelt und auf einem Display alternierend darstellt. Zusätzlich sollen die Wetterdaten auf dem Raspberry Pi in einer Datenbank persistent abgespeichert werden, so dass die aktuellsten Daten Online eingesehen werden können.

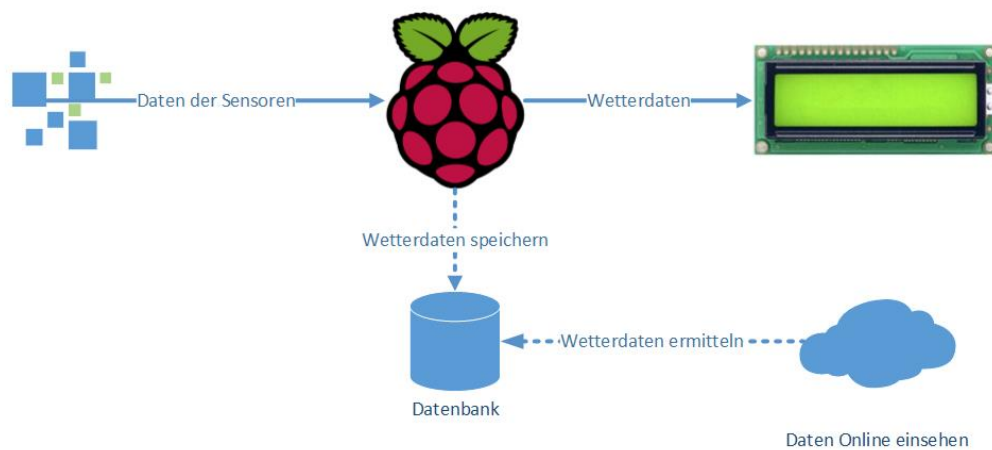


Abbildung: Projektidee (Skizze)

3 Anforderungen

Nachfolgend werden die funktionalen sowie die nicht funktionalen Anforderungen an das System beschrieben. Bei den funktionalen Anforderungen handelt es sich ausschliesslich um Muss-Anforderungen.

3.1 Funktionale Anforderungen

- Die Wetterdaten (Luftdruck, Temperatur, Feuchtigkeit und Lichtstärke) sind mittels Sensoren zu ermitteln.
- Die ermittelten Wetterdaten sind persistent in einer Datenbank abzuspeichern.
- Des Weiteren sind die ermittelten Wetterdaten auf einem Display auszugeben.
- Die aktuellsten Wetterdaten müssen Online eingesehen werden können.

3.2 Nicht funktionale Anforderungen

- Die Projekt muss am 14.12.2014 (inkl. Dokumentation) abgeschlossen sein
- Das Projekt muss mittels Präsentation am 20.12.2014 anlässlich der 5. Präsenz vorgestellt werden.
- Die Signal- und Datenverarbeitung hat auf dem Raspberry Pi zu erfolgen.

4 Kontextdiagramm

Nachfolgend wird das Kontextdiagramm des Projekts (inkl. den Kann-Zielen) darstellt:

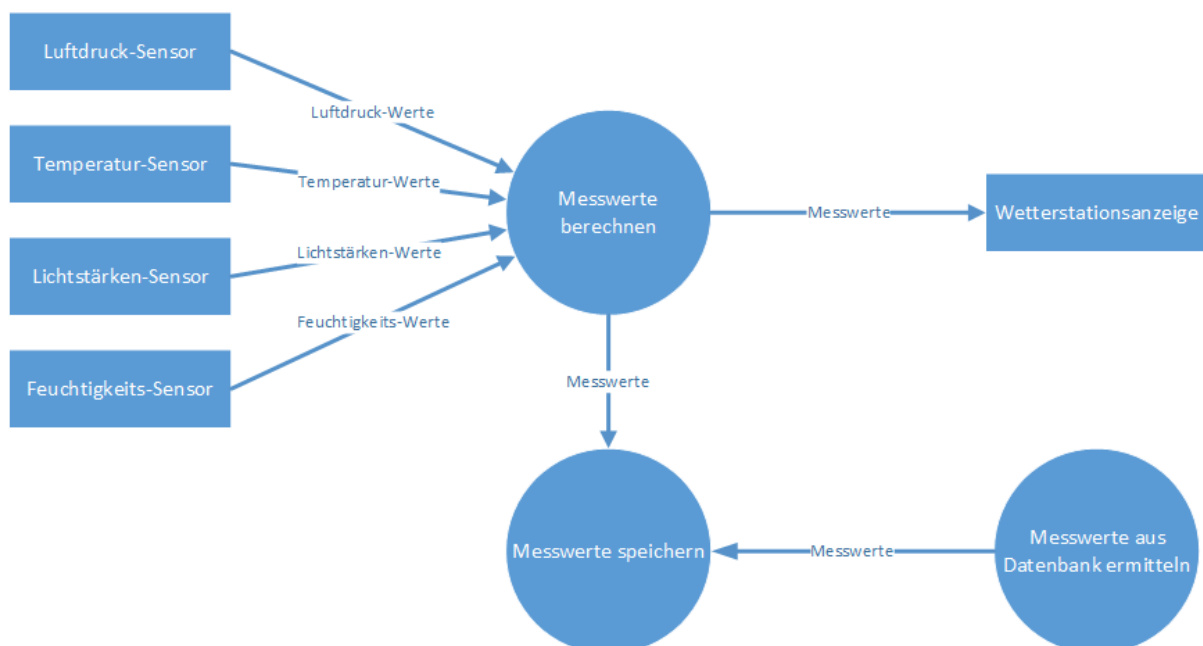


Abbildung: Kontextdiagramm des Projekts

Die Werte der einzelnen Sensoren werden ermittelt und in einer zentral berechnet / umgerechnet. Anschliessend werden die Daten an die Anzeige weitergeleitet. Zudem werden die Messwerte nach der Berechnung persistent abgespeichert, damit die Daten Online abgefragt werden können.

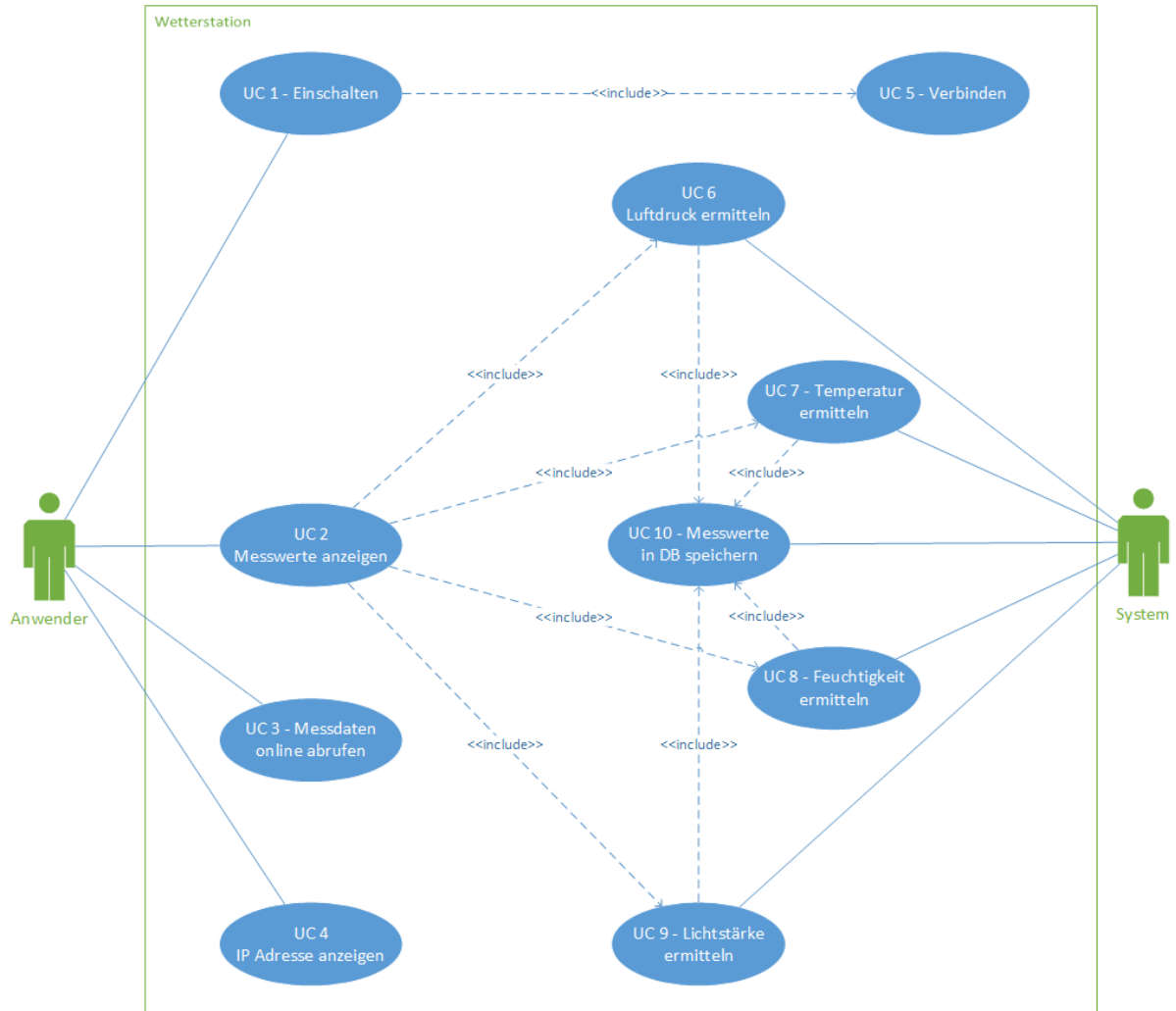
5 Terminplan

Bezeichnung	Termin
Projektskizze erstellt	05.10.2014
Anforderungen / Kontextdiagramm / Terminplan	12.10.2014
Use-Cases erstellen / verifizieren	19.10.2014
Lösungsentwürfe erstellen (Grobentwurf) / Lösungsfindung	26.10.2014
Schaltungsentwurf / Softwareentwurf / Testkonzept	16.11.2014
Schaltung / Hardware umsetzen	22.11.2014
Software implementieren (Ermittlung Messwerte, Weitergabe der Messwerte an den LCD-Bildschirm)	07.12.2014
Applikationstest und Abnahme	14.12.2014
Projektdokumentation finalisieren	14.12.2014
Präsentation anlässlich Präsenz 5	20.12.2014

Die Meilensteine (Abgaben in moodle) sind Fett markiert und sind zwingend einzuhalten.

6 Use Cases

6.1 Diagramm



6.2 Beschreibungen

UC 1 - Einschalten	
Beschreibung	Die Wetterstation einschalten
Stakeholder	Anwender
Uses	<i>UC 5 - Verbinden</i>
Vorbedingungen	Die Wetterstation ist noch nicht eingeschaltet
Nachbedingungen	Die Wetterstation ist eingeschaltet
Ablauf	Verbindung Wetterstation / 230V Steckdose mit Netzteil

UC 2 - Messwerte anzeigen	
Beschreibung	Die unterschiedlichen Messwerte sollen auf dem LCD Display der Wetterstation angezeigt werden
Stakeholder	Anwender
Uses	<i>UC 6 - Luftdruck ermitteln, UC 7 - Temperatur ermitteln, UC 8 - Feuchtigkeit ermitteln, UC 9 - Lichtstärke ermitteln</i>
Vorbedingungen	<ul style="list-style-type: none"> Wetterstation eingeschaltet Messwerte durch die Sensoren ermittelt
Nachbedingungen	Messwerte werden auf dem LCD-Display angezeigt
Ablauf	<ol style="list-style-type: none"> Anwender betätigt den Schalter 1 am LCD-Display (gilt nur wenn der Schalter 2, 3 oder 4 zuvor betätigt wurde, ansonsten werden die Messwerte standardmässig angezeigt) Messwerte werden auf dem LCD-Display angezeigt (Pro LCD-Reihe ein Messwert)

UC 3 – Messdaten online abrufen	
Beschreibung	Die Messdaten dem Anwender Online zur Verfügung stellen
Stakeholder	Anwender
Vorbedingungen	<ul style="list-style-type: none"> Messdaten in der Datenbank vorhanden IP-Adresse des Raspberry Pi bekannt
Nachbedingungen	Die Messwerte konnten Online ermittelt werden
Ablauf	<ol style="list-style-type: none"> Anwender verbindet sich mittels der bekannten IP-Adresse mit der Schnittstelle auf dem Raspberry Pi

	2. Messwerte werden aus der Schnittstelle auf Grund der Angabe des Datumbereichs (resp. des aktuellen Wertes) ausgelesen
--	--

UC 4 - IP Adresse anzeigen

Beschreibung	Die IP-Adresse des Raspberry Pi wird auf dem LCD-Display angezeigt (für Fernwartung oder Zugriffe auf die Online-Schnittstelle)
Stakeholder	Anwender
Vorbedingungen	Wetterstation eingeschaltet
Nachbedingungen	IP-Adresse des Raspberry Pi wird auf dem LCD-Display dargestellt
Ablauf	<ol style="list-style-type: none"> 1. Anwender betätigt den Schalter 2 am LCD- 2. IP-Adresse wird auf dem LCD-Display dargestellt

UC 5 - Verbinden

Beschreibung	Das Raspberry Pi verbindet sich beim Systemstart mit der Hardware zur Ermittlung der Messwerte.
Stakeholder	<i>System (Raspberry Pi)</i>
Vorbedingungen	<ul style="list-style-type: none"> • Wetterstation eingeschaltet • Hardware zur Ermittlung der Messwerte an das Raspberry Pi angeschlossen und bereit
Nachbedingungen	Das System ist mit der Hardware zur Ermittlung der Messwerte verbunden
Ablauf	<ol style="list-style-type: none"> 1. Verbindung mit der Hardware zur Ermittlung der Messwerte aufbauen (IP-Verbindung). <ol style="list-style-type: none"> a. Bei einem Kommunikationsfehler soll dieser auf dem LCD Display ausgegeben werden. b. Kann die Verbindung hergestellt werden, kann mit der Ermittlung der Messwerte begonnen werden.

UC 6 - Luftdruck ermitteln

Beschreibung	Der aktuelle Luftdruck wird von der Hardware mittels einem Sensor ermittelt
Stakeholder	<i>System (Raspberry Pi)</i>
Uses	<i>UC 10 - Messwerte in DB speichern</i>
Vorbedingungen	<ul style="list-style-type: none"> • Wetterstation eingeschaltet

	<ul style="list-style-type: none"> Verbindung zwischen dem Raspberry Pi und der Hardware hergestellt
Nachbedingungen	Messwert wird auf LCD-Display dargestellt oder aber es wird eine entsprechende Fehlermeldung beim Messwert angezeigt.
Ablauf	<ol style="list-style-type: none"> Prüfen ob der Sensor verfügbar ist <ol style="list-style-type: none"> Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases Ermitteln des aktuellen Messwertes Validieren des ermittelten Messwertes <ol style="list-style-type: none"> Messwert nicht plausibel: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases Ausgabe des Messwertes auf dem LCD-Display (UC 2) Speicherung des Messwertes in der DB (UC 9)

UC 7 - Temperatur ermitteln

Beschreibung	Der aktuelle Temperatur wird von der Hardware mittels einem Sensor ermittelt
Stakeholder	<i>System (Raspberry Pi)</i>
Uses	<i>UC 10 - Messwerte in DB speichern</i>
Vorbedingungen	<ul style="list-style-type: none"> Wetterstation eingeschaltet Verbindung zwischen dem Raspberry Pi und der Hardware hergestellt
Nachbedingungen	Messwert wird auf LCD-Display dargestellt oder aber es wird eine entsprechende Fehlermeldung beim Messwert angezeigt.
Ablauf	<ol style="list-style-type: none"> Prüfen ob der Sensor verfügbar ist <ol style="list-style-type: none"> Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases Ermitteln des aktuellen Messwertes Validieren des ermittelten Messwertes <ol style="list-style-type: none"> Messwert nicht plausibel: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases Ausgabe des Messwertes auf dem LCD-Display (UC 2) Speicherung des Messwertes in der DB (UC 9)

UC 8 - Feuchtigkeit ermitteln

Beschreibung	Die aktuelle Feuchtigkeit wird von der Hardware mittels einem Sensor ermittelt
Stakeholder	<i>System (Raspberry Pi)</i>
Uses	<i>UC 10 - Messwerte in DB speichern</i>
Vorbedingungen	<ul style="list-style-type: none"> Wetterstation eingeschaltet Verbindung zwischen dem Raspberry Pi und der Hardware hergestellt
Nachbedingungen	Messwert wird auf LCD-Display dargestellt oder aber es wird eine entsprechende Fehlermeldung beim Messwert angezeigt.
Ablauf	<ol style="list-style-type: none"> Prüfen ob der Sensor verfügbar ist <ol style="list-style-type: none"> Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases Ermitteln des aktuellen Messwertes Validieren des ermittelten Messwertes <ol style="list-style-type: none"> Messwert nicht plausibel: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases Ausgabe des Messwertes auf dem LCD-Display (UC 2) Speicherung des Messwertes in der DB (UC 9)

UC 9 - Lichtstärke ermitteln

Beschreibung	Die aktuelle Lichtstärke wird von der Hardware mittels einem Sensor ermittelt
Stakeholder	<i>System (Raspberry Pi)</i>
Uses	<i>UC 10 - Messwerte in DB speichern</i>
Vorbedingungen	<ul style="list-style-type: none"> Wetterstation eingeschaltet Verbindung zwischen dem Raspberry Pi und der Hardware hergestellt
Nachbedingungen	Messwert wird auf LCD-Display dargestellt oder aber es wird eine entsprechende Fehlermeldung beim Messwert angezeigt.
Ablauf	<ol style="list-style-type: none"> Prüfen ob der Sensor verfügbar ist <ol style="list-style-type: none"> Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases Ermitteln des aktuellen Messwertes Validieren des ermittelten Messwertes

	a. Messwert nicht plausibel: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 4. Ausgabe des Messwertes auf dem LCD-Display (UC 2) 5. Speicherung des Messwertes in der DB (UC 9)
--	---

UC 10 - Messwerte in DB speichern

Beschreibung	Die ermittelten Messwerte in die Datenbank speichern
Stakeholder	<i>System (Raspberry Pi)</i>
Vorbedingungen	<ul style="list-style-type: none"> Messwerte wurden von den entsprechenden Sensoren ermittelt Datenbank auf dem Raspberry Pi verfügbar
Nachbedingungen	Messwerte wurden in der Datenbank hinterlegt
Ablauf	Messwerte werden in den Datenbank abgespeichert (bei einem allfälligen Zugriffsfehler wird der Fehler nicht nach aussen populierte).

7 Grobentwurf

7.1 Hardware

Bei der Umsetzung der Hardware-Schaltung gibt es 2 mögliche Varianten, wie die elektronischen Bauteile (Sensoren und LCD-Display) mit dem Raspberry Pi (und damit mit der Steuerung) verbunden werden können:

- Direkte Anbindung
- Indirekte Anbindung über einen Master-Baustein

7.1.1 Direkte Anbindung

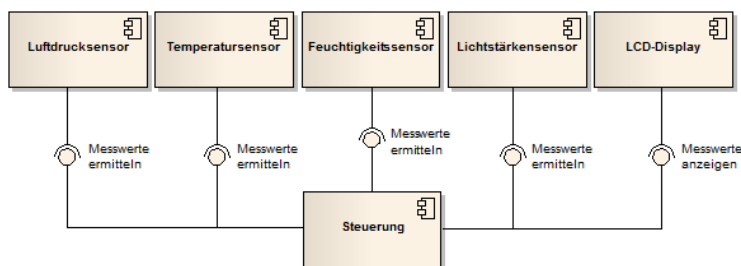


Abbildung: Direkte Anbindung der elektronischen Bauteile an das Raspberry Pi / die Steuerung

Bei der direkten Anbindung werden alle Sensoren sowie der LCD-Display direkt mit dem Raspberry Pi verbunden. Jedes elektronische Bauteil benötigt aus diesem Grunde eine separate Daten- sowie Stromzuleitung.

7.1.2 Indirekte Anbindung

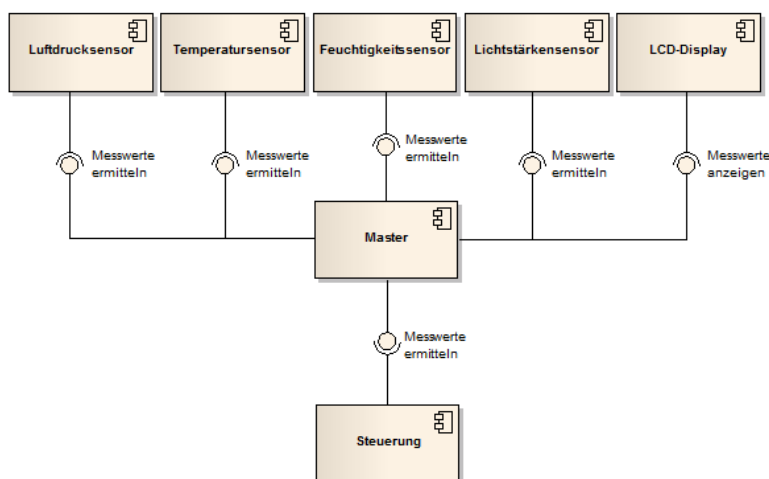


Abbildung: Indirekte Anbindung der elektronischen Bauteile an das Raspberry Pi / die Steuerung

Bei der indirekten Anbindung werden die Sensoren wie das LCD-Display an einen Master angeschlossen, welcher wiederum mit dem Raspberry Pi verbunden ist. Die

Stromversorgung erhalten die Bauteile durch den Master. Die Daten werden vom/zum Raspberry Pi über den Master gesandt.

7.2 Steuerung / Online-Schnittstelle

Die Steuerung ermittelt die Messwerte von den Sensoren und gibt diese innerhalb eines bestimmten Intervalls an das LCD-Display sowie an die Datenbank weiter. Die Steuerung wird nach dem Startvorgang des Raspberry Pi automatisch gestartet (kein manueller Eingriff nötig), so dass die Messwerte umgehend ermittelt und gespeichert werden. Damit die Daten von anderen Personen eingesehen werden können, wird eine Online-Schnittstelle definiert, welche die Daten gegen aussen zur Verfügung stellt.

Für die Umsetzung der Steuerung haben wir uns für C++ als Sprache entschieden. Von den auf Raspberry Pi verfügbaren Sprachen ist bei C++ die grösste Erfahrung vorhanden. Bei der Datenbank zur persistenten Speicherung der Messwerte haben wir uns für SQLite entschieden.

Die Online-Schnittstelle kann auf 3 unterschiedliche Arten umgesetzt werden:

- Webseite (PHP)
- RESTful Webservice (PHP)
- Smartphone-App (Windows Phone 8) im Zusammenspiel mit einem RESTful Webservice (PHP)

7.2.1 Webseite

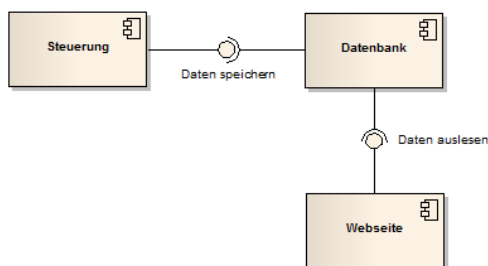


Abbildung: Webseite mit PHP mit Zugriff auf die Datenbank

Die Webseite greift auf die Messdaten, welche von der Steuerung in der Datenbank gespeichert wurden, zu. Die Daten werden anschliessend über ein mit PHP entwickeltes Web-UI zur Verfügung gestellt.

7.2.2 Webservice

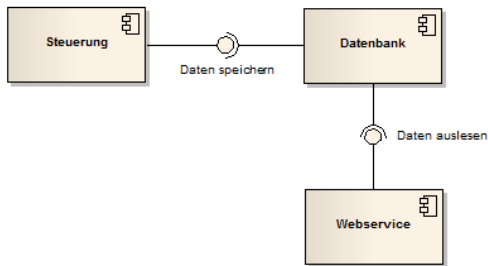


Abbildung: Webservice mit PHP mit Zugriff auf die Datenbank

Der Webservice soll als RESTful Service die Daten über eine Schnittstelle öffentlich zur Verfügung stellen. Die Umsetzung ist mit PHP geplant.

7.2.3 Smartphone-App mit Webservice-Zugriff

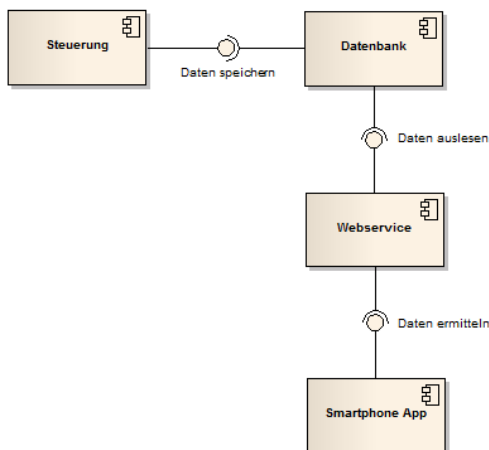


Abbildung: Webservice mit PHP mit Zugriff auf die Datenbank mit einer zusätzlichen Smartphone App (Windows Phone 8)

Bei dieser Variante soll ein RESTful Webservice mit PHP erstellt werden, welcher die Messdaten öffentlich zur Verfügung stellt. Zusätzlich zur Variante nur mit einem Webservice soll zusätzlich eine Smartphone-App (Windows Phone 8) erstellt werden, welche die Daten entsprechend konsumiert.

7.3 Lösungsfindung

7.3.1 Hardware

Beide Varianten lassen sich insbesondere durch einen Unterschied voneinander unterscheiden: Bei der direkten Anbindung führt jeder Sensor sowie der LCD-Bildschirm die Verbindung direkt auf das Raspberry Pi und damit auf die Steuerung. Bei der indirekten Anbindung werden die Verbindungen der elektronischen Bauteile zuerst auf einem Master-Baustein zusammengeführt und erst anschliessend auf die Steuerung gebracht.

Dies führt automatisch zu je einem Vor- wie auch Nachteil der beiden Varianten. So funktioniert bei einem Ausfall eines elektronischen Bauteils bei der direkten Anbindung die Wetterstation immer noch, wenn auch nur eingeschränkt. Allerdings ist die Schnittstelle zwischen den Bauteilen und der Steuerung komplexer, da für jedes Bauteil eine eigene Ansteuerung erstellt werden muss. Bei der indirekten Ansteuerung führt andererseits ein Ausfall des Master-Bausteins dazu, dass die Wetterstation nicht mehr funktioniert, da alle Verbindungen über diesen geführt werden. Hingegen ist die Anbindung an die einzelnen Bauteile einfacher, da nur eine Verbindung unterhalten werden muss.

Erfahrungsgemäss führen mehrere Schnittstellen gegenüber nur einer Schnittstelle in einer Steuerung eher zu mehr Problemen (Threads, Asynchronität, Synchronisation). Hingegen kann die Möglichkeit eines Ausfalls eines Master-Bausteins bei sachgemäsem Einsatz als sehr gering angenommen werden. **Aus diesem Grunde entscheiden wir uns für die indirekte Ansteuerung der elektronischen Bauteile.**

7.3.2 Software

Die Variante mit dem reinen Webservice scheidet aus, da diese Variante nur Benutzern mit Programmierkenntnissen einen Mehrwert bringt (und diese eine eigene Anwendung entwickeln müssten). Die Verbindung zwischen Webservice und Smartphone wäre zwar reizvoll (insbesondere weil diese Variante auf 2 unterschiedlichen Technologien aufsetzen würde), allerdings müsste selbst bei hybriden Apps pro Smartphone-Technologie (Android, iPhone, Windows Phone, Blackberry) eine App erstellt werden, zudem wären die Desktop-Benutzer ausgeschlossen. Aus diesem Grunde **bietet sich die Lösung mit der Webseite mit PHP an**, da diese sowohl von mobilen Benutzern (mit unterschiedlichen Technologien) als auch von Benutzern mit stationären Computern genutzt werden kann.

8 Detailentwurf

8.1 Hardware / Schaltung

8.2 Steuerung

8.3 Webseite

9 Testkonzept