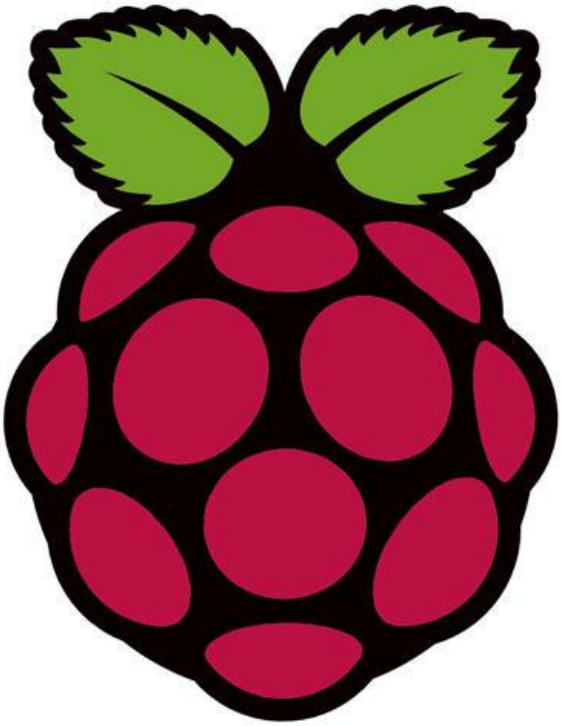

Wetterstation mit Raspberry Pi

Version: 1.0

Datum: 14.12.14

Projektteam: Andreas Hasler / David Daniel

Inhaltsverzeichnis

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 / Hardware Entwurf
0.6	16.11.2014	David Daniel	Software Entwurf (Steuerung) / Qualitätssicherung / Testing
0.7	16.11.2014	Andreas Hasler	Entwurf Webseite
0.8	23.11.2014	Andreas Hasler	Diverse Stellen überarbeitet
1.0	14.12.2014	Andreas Hasler	Finalisierung

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.

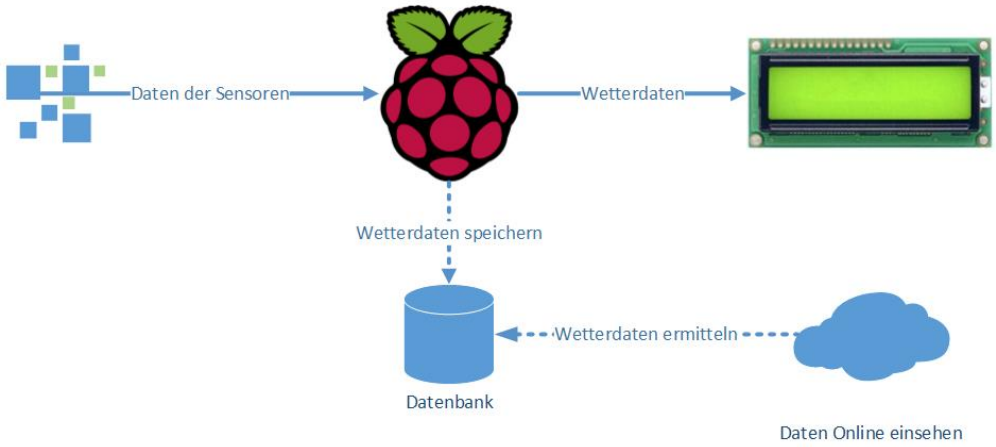

Abbildung 1: Projektidee (Skizze)

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:

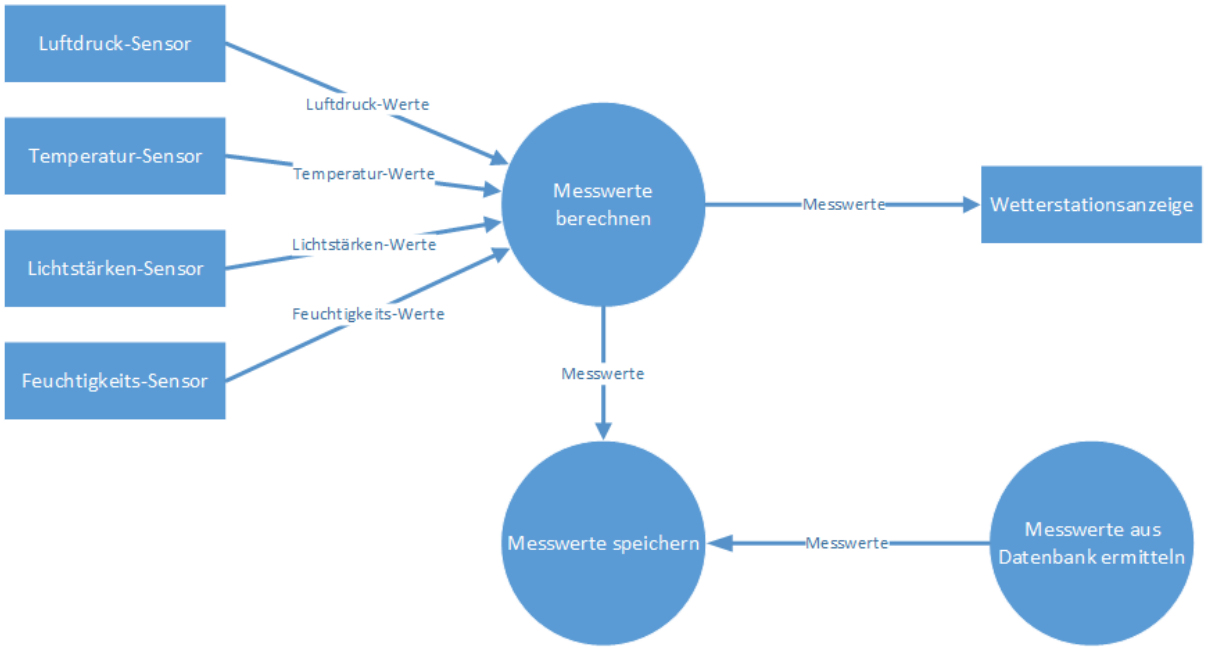

Abbildung 2: Kontextdiagramm des Projekts

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	**17.11.2014**
Schaltung / Hardware umsetzen	22.11.2014
Software implementieren (Ermittlung Messwerte, Weitergabe der Messwerte an den LCD-Bildschirm)	07.12.2014
Applikationstest und Abnahme	15.12.2014
Projektdokumentation finalisieren	**15.12.2014**
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

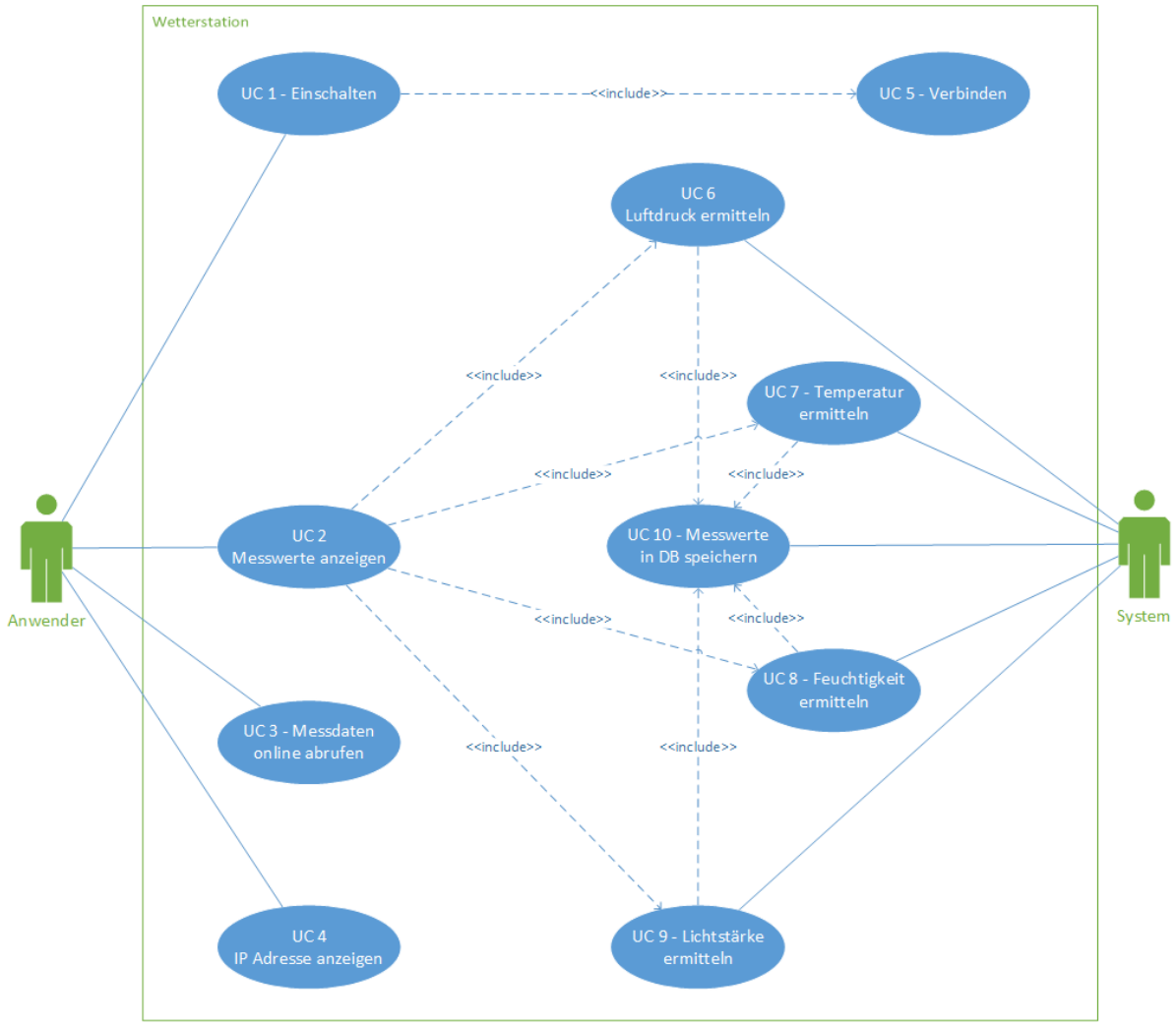

Abbildung 3: Use Cases

6.2 Beschreibungen

UC 1 - Einschalten	
Beschreibung	Die Wetterstation einschalten
Stakeholder	Anwender
Uses	*UC 5 - Verbinden*
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	*UC 6 - Luftdruck ermitteln, UC 7 - Temperatur ermitteln, UC 8 - Feuchtigkeit ermitteln, UC 9 - Lichtstärke ermitteln*
Vorbedingungen	• Wetterstation eingeschaltet • Messwerte durch die Sensoren ermittelt
Nachbedingungen	Messwerte werden auf dem LCD-Display angezeigt
Ablauf	1. 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 anzeigt) 2. 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	• Messdaten in der Datenbank vorhanden • IP-Adresse des Raspberry Pi bekannt
Nachbedingungen	Die Messwerte konnten Online ermittelt werden
Ablauf	1. 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	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	*System (Raspberry Pi)*
Vorbedingungen	• 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	1. Verbindung mit der Hardware zur Ermittlung der Messwerte aufbauen (IP-Verbindung). a. Bei einem Kommunikationsfehler soll dieser auf dem LCD Display ausgeben 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	*System (Raspberry Pi)*
Uses	*UC 10 - Messwerte in DB speichern*
Vorbedingungen	• 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	1. Prüfen ob der Sensor verfügbar ist a. Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 2. Ermitteln des aktuellen Messwertes 3. 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 7 - Temperatur ermitteln	
Beschreibung	Der aktuelle Temperatur wird von der Hardware mittels einem Sensor ermittelt
Stakeholder	*System (Raspberry Pi)*
Uses	*UC 10 - Messwerte in DB speichern*
Vorbedingungen	• 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	1. Prüfen ob der Sensor verfügbar ist a. Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 2. Ermitteln des aktuellen Messwertes 3. 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 8 - Feuchtigkeit ermitteln	
Beschreibung	Die aktuelle Feuchtigkeit wird von der Hardware mittels einem Sensor ermittelt
Stakeholder	*System (Raspberry Pi)*
Uses	*UC 10 - Messwerte in DB speichern*
Vorbedingungen	• 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	1. Prüfen ob der Sensor verfügbar ist a. Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 2. Ermitteln des aktuellen Messwertes 3. 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 9 - Lichtstärke ermitteln	
Beschreibung	Die aktuelle Lichtstärke wird von der Hardware mittels einem Sensor ermittelt
Stakeholder	*System (Raspberry Pi)*
Uses	*UC 10 - Messwerte in DB speichern*
Vorbedingungen	• 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	1. Prüfen ob der Sensor verfügbar ist a. Sensor nicht verfügbar: Fehlermeldung an LCD-Display ausgeben. Abbruch des Use Cases 2. Ermitteln des aktuellen Messwertes 3. 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	*System (Raspberry Pi)*
Vorbedingungen	• Messwerte wurden von den entsprechenden Sensoren ermittelt • Datenbank auf dem Raspberry Pi verfügbar
Nachbedingungen	Messwerte wurden in der Datenbank hinterlegt
Ablauf	Messwerte werden in der Datenbank abgespeichert (bei einem allfälligen Zugriffsfehler wird der Fehler nicht nach aussen populiert).

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

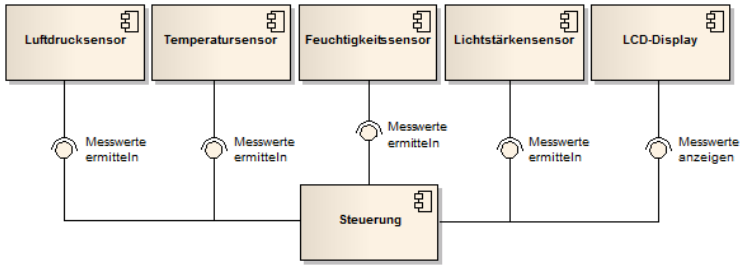

Abbildung 4: Direkte Anbindung der elektronischen Bauteile an das Raspberry Pi / die Steuerung

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

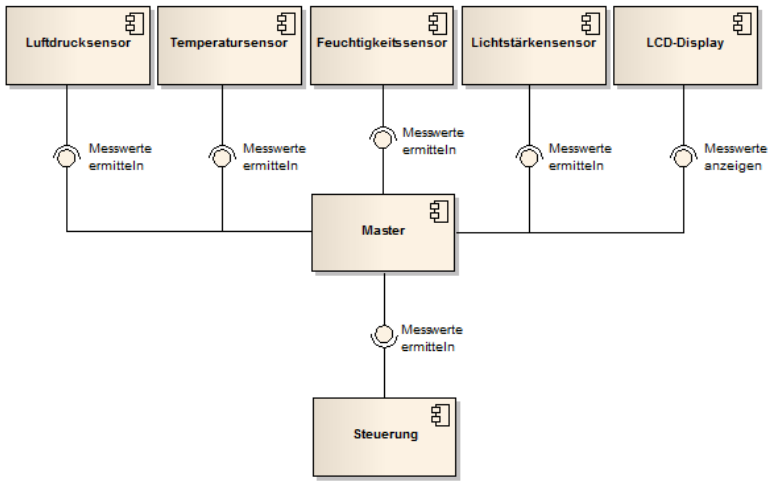

Abbildung 5: Indirekte Anbindung der elektronischen Bauteile an das Raspberry Pi / die Steuerung

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

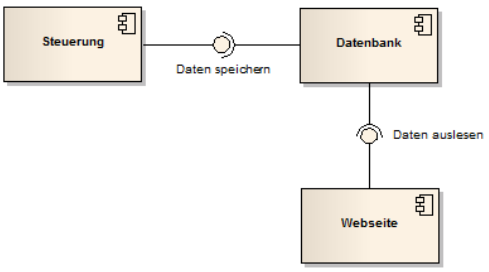

Abbildung 6: Webseite mit PHP mit Zugriff auf die Datenbank

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

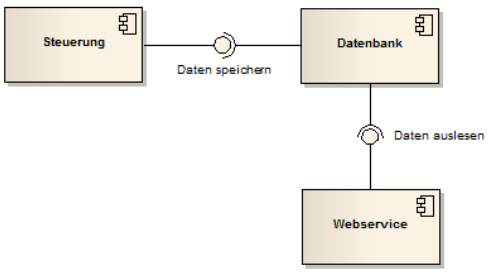

Abbildung 7: Webservice mit PHP mit Zugriff auf die Datenbank

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

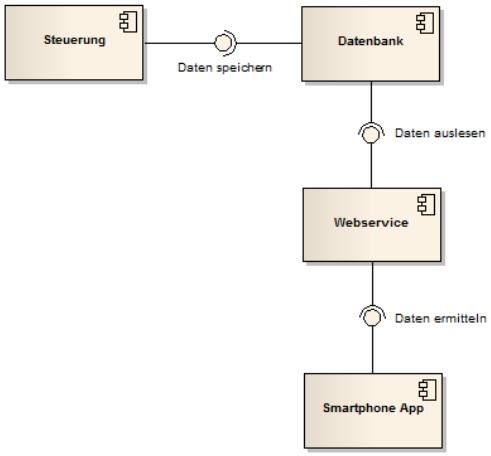

Abbildung 8: Webservice mit PHP mit Zugriff auf die Datenbank mit einer zusätzlichen Smartphone App (Windows Phone 8)

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, Asynchron, Synchronisation). Hingegen kann die Möglichkeit eines Ausfalls eines Master-Bausteins bei sachgemässem 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

Nachfolgend wird der Hardware-Entwurf der Wetterstation inkl. Ansteuerung des Raspberry Pi dargestellt. Folgende Hardware-Komponenten werden verwendet:

- 3 Sensoren (Temperatur und Druck, Feuchtigkeit, Lichtstärke)
- LCD-Anzeige mit 4 Zeilen
- Master (Zusammenführung der 3 Sensoren und der LCD-Anzeige) zur Verbindung mit dem Raspberry Pi
- Stromversorgung (Wandler) mit 6 bis 27 V DC Eingangsspannung und 5 V DC Ausgangsspannung für die Hardware-Bausteine und das Raspberry Pi
- Raspberry Pi

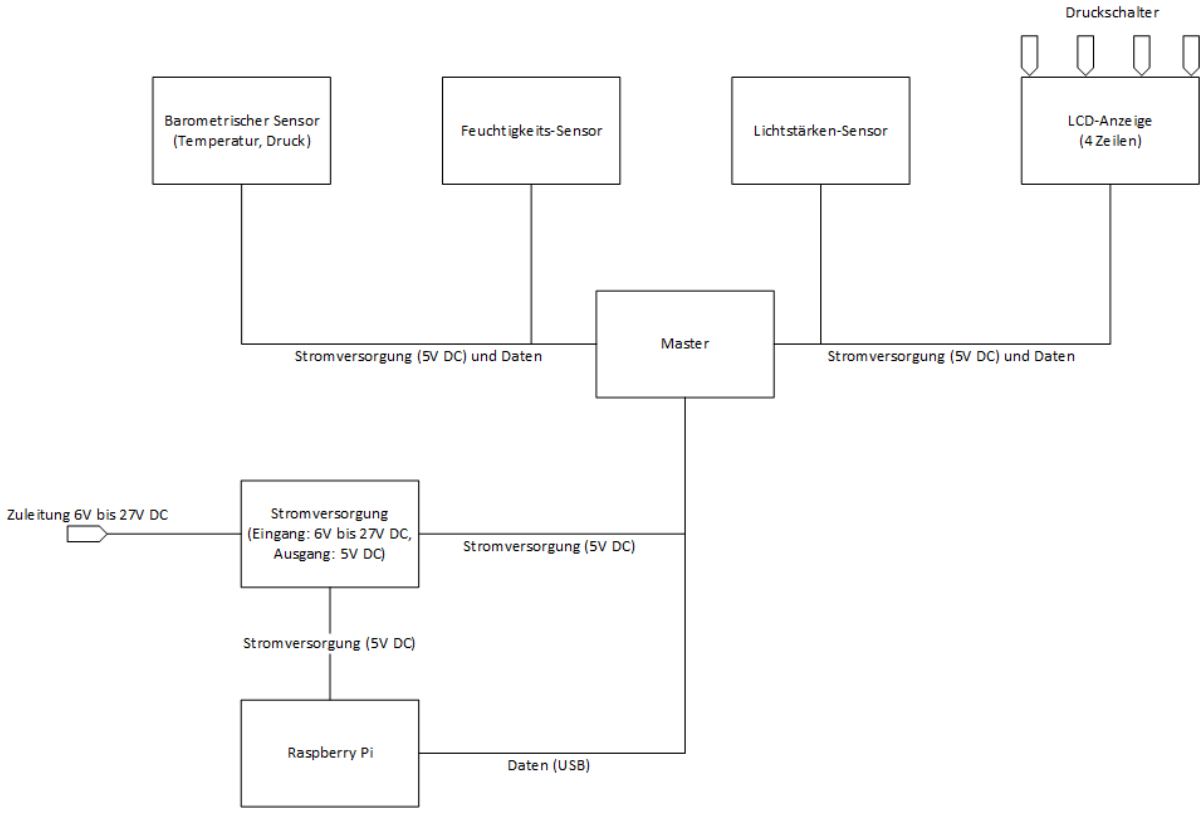

Abbildung 9: Entwurf der Schaltung der Sensoren, der LCD-Anzeige, sowie des Raspberry Pi

8.1.1 Technische Spezifikationen

Im Folgenden werden die einzelnen Sensoren, der Master, die LCD-Anzeige sowie die Stromversorgung mittels Schaltplan und einem Auszug aus dem Datenblatt detaillierter beschrieben.

Barometer-Sensor (Temperatur, Luftdruck)

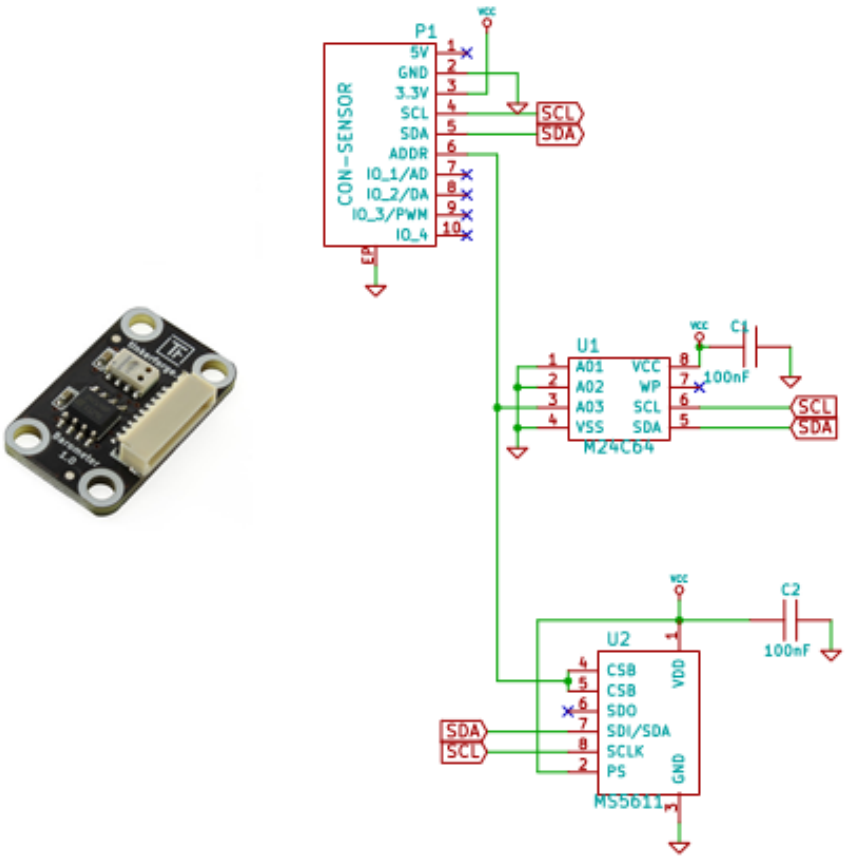

Abbildung 10: Barometer Bricklet / Schaltplan des Barometer Bricklet (Quelle: Tinkerforge)

Eigenschaft	Wert
Stromverbrauch	1mA
Luftdruck-Bereich	10 bis 1200 mbar (Genauigkeit +/- 1.5 mbar)
Temperatur	-40 °C bis 85 °C (in 0.01 °C Schritten)
Abmessung	25 x 15 x 5 mm
Gewicht	2g

Lichtstärken-Sensor

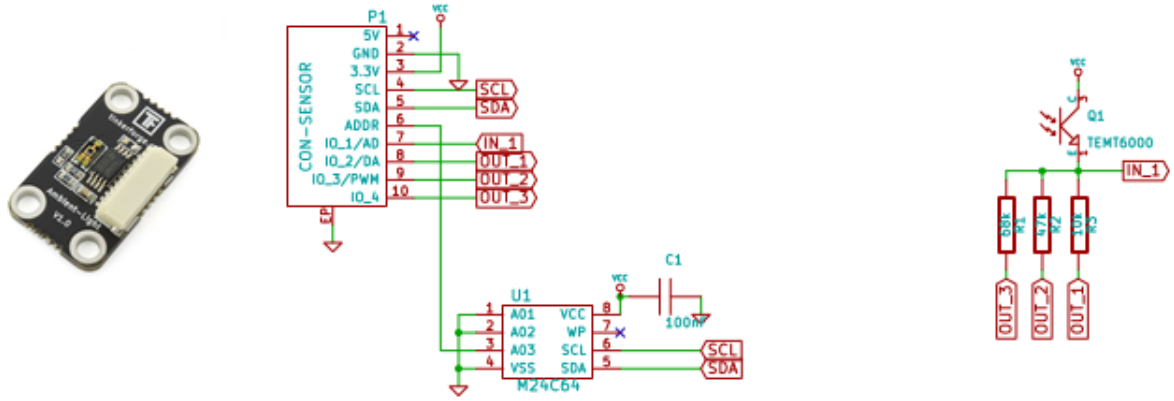

Abbildung 11: Lichtstärken Bricklet / Schaltplan des Lichtstärken Bricklet (Quelle: Tinkerforge)

Eigenschaft	Wert
Stromverbrauch	1mA
Beleuchtungsstärke	0 Lux bis 900 Lux (in 0.1 Lux Schritten, 12 Bit Auflösung)
Abmessung	25 x 15 x 5 mm
Gewicht	2g

Luftfeuchtigkeit-Sensor

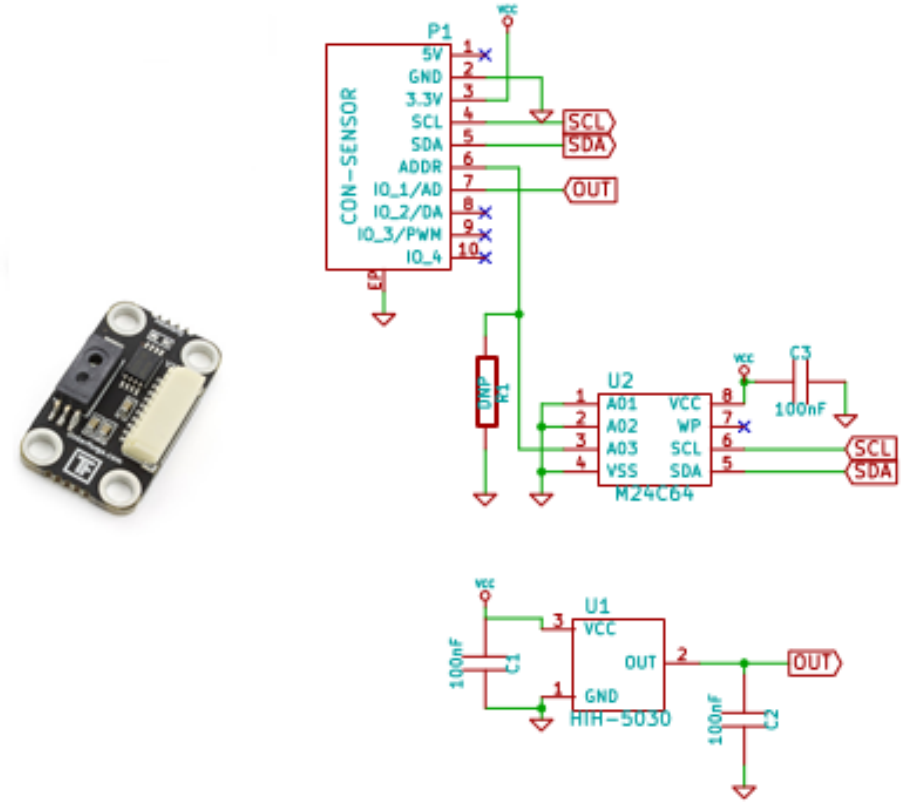

Abbildung 12: Luftfeuchtigkeit Bricklet / Schaltplan des Luftfeuchtigkeit Bricklet (Quelle: Tinkerforge)

Eigenschaft	Wert
Stromverbrauch	1mA
Beleuchtungsstärke	Relative Luftfeuchtigkeit (RH) 0% bis 100% in 0.1% RH Schritten (12 Bit Auflösung)
Abmessung	25 x 15 x 5 mm
Gewicht	2g

Master

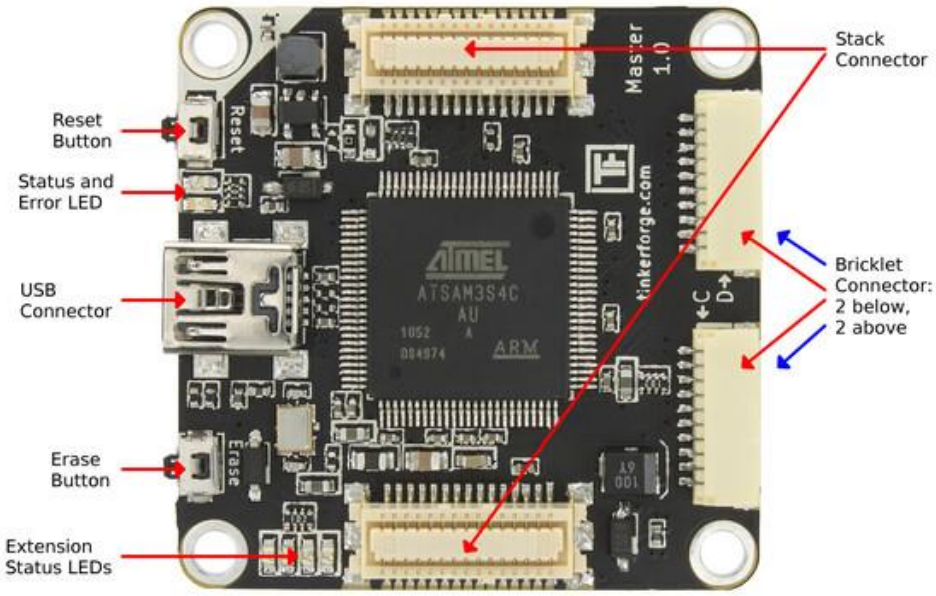

Abbildung 13: Master Brick (Anschlüsse für die Sensoren / Bricklets) (Quelle: Tinkerforge)

Der detaillierte Schaltplan des Master Bricks ist sehr umfangreich und kann auf https://github.com/Tinkerforge/master-brick/raw/master/hardware/master-schematic.pdf heruntergeladen werden.

Eigenschaft	Wert
Stromverbrauch	68mA
Bricklet Anschlüsse	4 (Weiterleitung der Daten an die USB-Schnittstelle)
Abmessung	40 x 40 x 16mm
Gewicht	12g

LCD-Anzeige

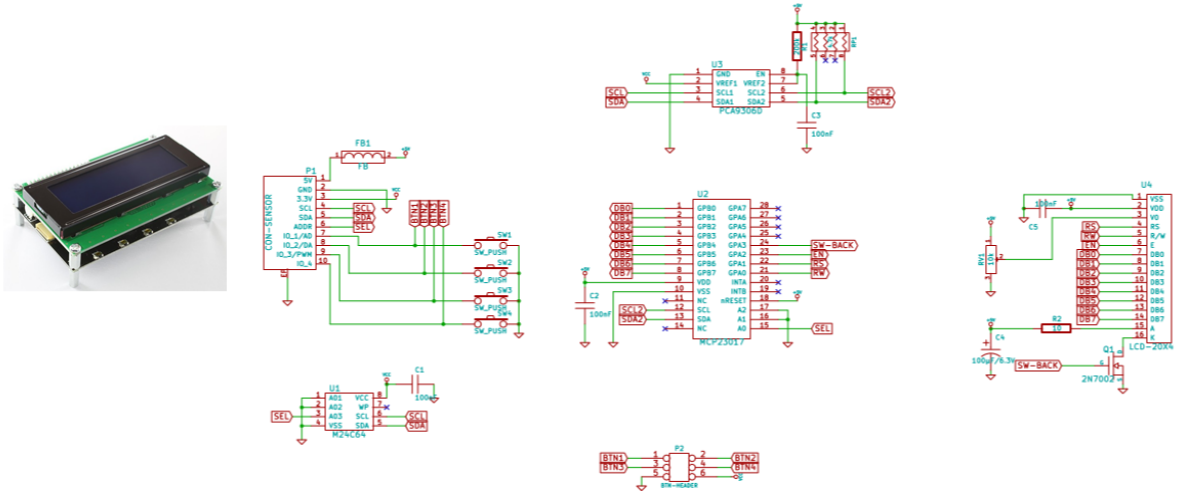

Abbildung 14: LCD-Anzeige / Schaltplan der LCD-Anzeige (Quelle: Tinkerforge)

Eigenschaft	Wert
Stromverbrauch	36mA
LCD	Alphanummerisch, 20 Zeichen pro Zeile, 4 Zeilen
Hintergrundbeleuchtung	Blau, per Software schaltbar
Kontrast	Einstellbar per Potentiometer
Abmessung	60 x 98 x 22mm
Gewicht	96g

Stromversorgung

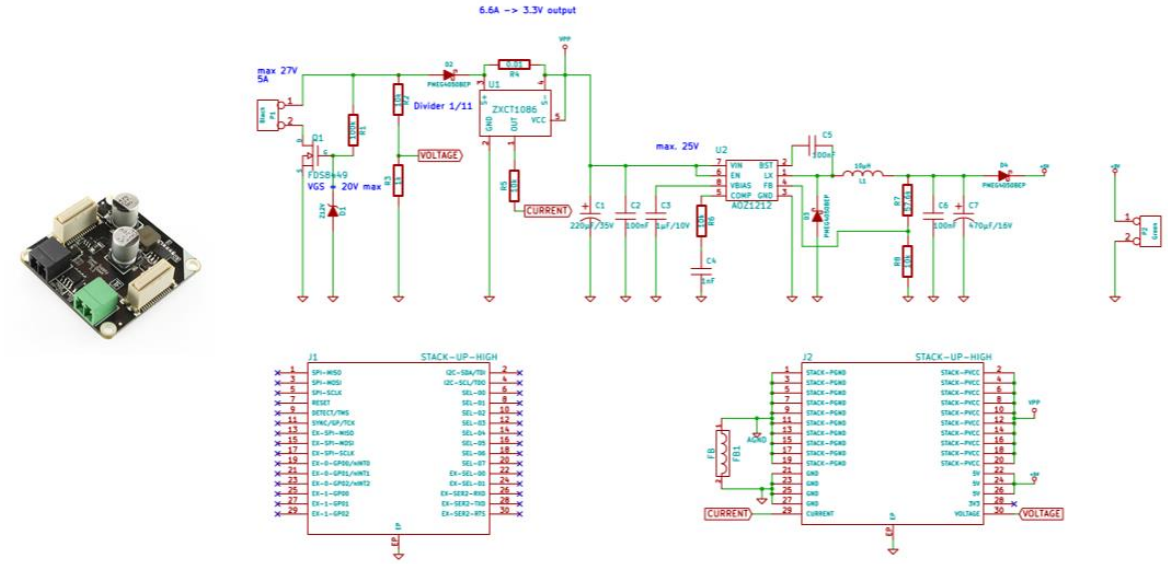

Abbildung 15: Stromversorgung / Schaltplan der Stromversorgung (Quelle: Tinkerforge)

Eigenschaft	Wert
Stromverbrauch	20-30mA (abhängig von der Eingangsspannung)
Min. Eingangsspannung	6V DC
Max. Eingangsspannung	27V DC
Max. Ausgangsstrom	5V Versorgung: 3A
Abmessung	40 x 40 x 16mm
Gewicht	14g

8.1.2 Hardware komplett

Die beiden nachfolgenden Bilder zeigen die komplett zusammengebaute Hardware inkl. dem Raspberry Pi. In der Abbildung 16 ist die LCD-Anzeige (noch ohne Software), sowie 2 der Sensoren und der Master sichtbar. Auf der Abbildung 17 insbesondere das Raspberry Pi sichtbar, welches mit dem Master für die Übertragung der Sensordaten verbunden ist.

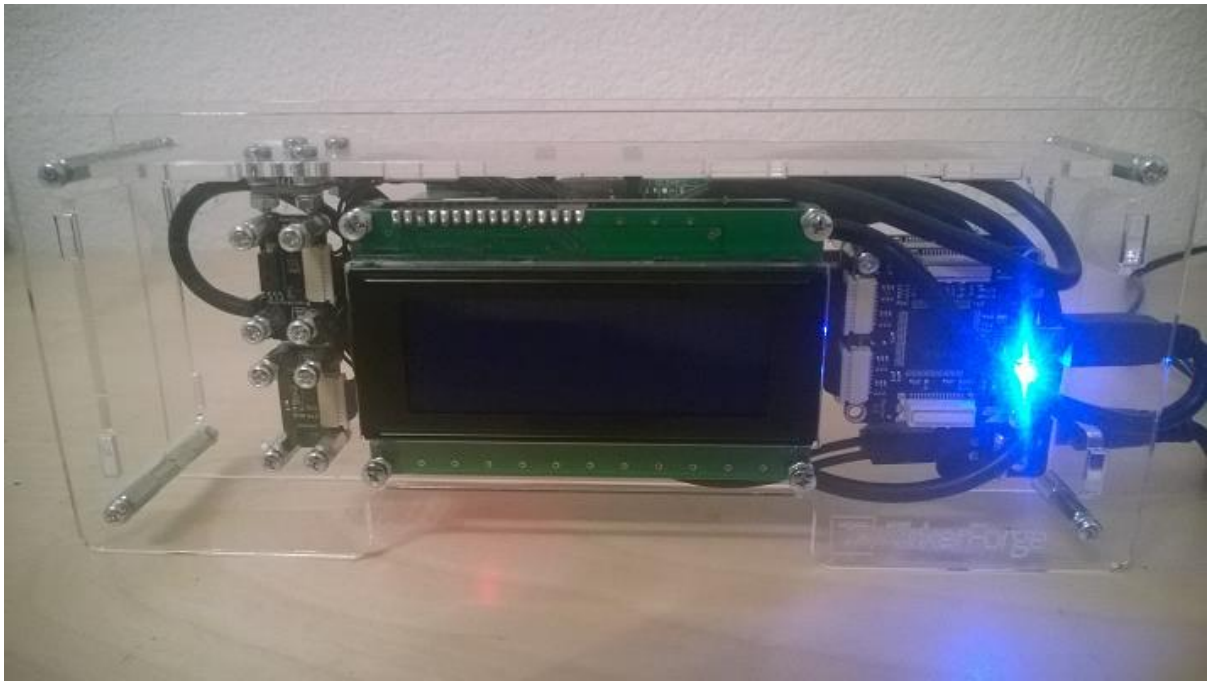

Abbildung 16: Ansicht Vorne der Wetterstation

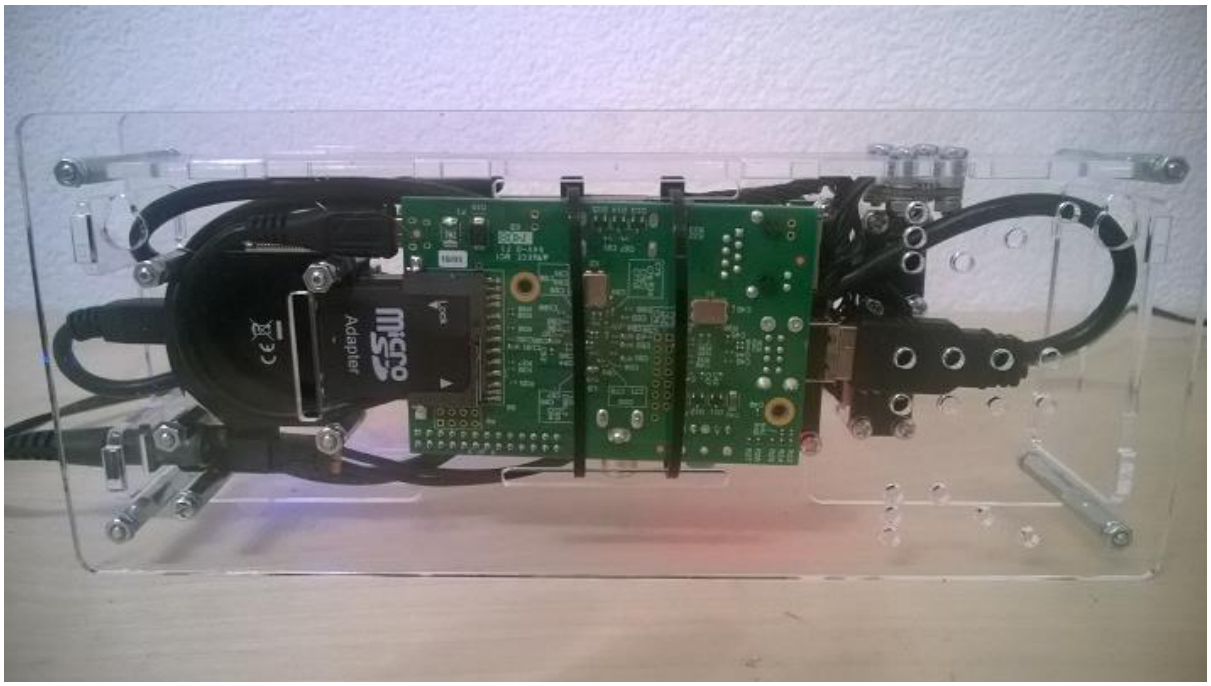

Abbildung 17: Ansicht Hinten der Wetterstation mit Raspberry Pi

8.2 Steuerung

Da die Anwendung nach dem Einschalten des Gerätes nicht mehr gestoppt wird, bis das Gerät ausgeschaltet wird, handelt es sich um einen Prozess, welcher im Hintergrund läuft. Dieser Prozess ermittelt die Messwerte von den Sensoren, speichert diese in der Datenbank und stellt die Werte auf dem LCD dar. Somit benötigt der Prozess während der gesamten Laufzeit Zugriff auf die genannten Komponenten.

8.2.1 Sensoren

Die Sensoren für den Luftdruck, die Luftfeuchtigkeit sowie für die Lichtstärke melden Änderungen per Event an die Steuerung, einzig die Temperatur muss mit einem Interval von 500ms abgefragt werden. Ein Sensor benötigt daher eine Beobachter-Schnittstelle und die Möglichkeit, die aktuellen Messwerte zu liefern.

8.2.2 LCD-Display

Das LCD-Display wird zur Darstellung von verschiedenen Werten verwendet, darin werden sowohl Messwerte mit unterschiedlichen Einheiten (Grad, Pascal oder hPa, Prozent etc.) sowie die IP Adresse und evtl. auch die Zeit dargestellt. Aufgrund des beschränkten Platzes auf dem Display wäre es schwierig, dem Anwender des Displays freizustellen, wie der anzuzeigende Inhalt auf dem Display dargestellt werden soll. Das Gerät muss lediglich die gewünschten Werte auf dem Display darstellen, es reicht daher, wenn die Abstraktion des Displays dies entsprechend ermöglicht. Somit muss das Display eine Möglichkeit bieten, einen Text wie die IP Adresse, eine Temperatur etc. oder alle aktuellen Messwerte zugleich anzuzeigen.

Durch die Notwendigkeit, dass das Display die Anzeige aktualisiert, was das Beobachten der Sensoren bedingt, bietet es sich an, einen Model-View Ansatz zu verfolgen. Die Sensoren bilden so das Model, während dem das Display als View auf Änderungen am Model lauscht.

8.2.3 Schalter am LCD-Display

Der oder die Schalter fungieren als Botschafter, welche die Anwendung benachrichtigen, sobald ein Schalter gedrückt wurde. Hierzu bietet sich daher ebenfalls ein klassisches Beobachter-Modell an.

8.2.4 Ablauf

Die Änderungen der Messwerte wird entweder per Event von den Sensoren gemeldet oder müssen gepolt werden (Temperatur, alle 500ms). Die Anwendung zeigt die Messwerte anschliessend auf dem LCD-Display dar und speichert die Messwerte innerhalb eines konfigurierbaren Intervalls in der Datenbank. Auf Anfrage über einen Schalter wechselt die Anwendung die Anzeige und stellt den gewünschten Wert dar. Das periodische Speichern der Werte beeinflusst jedoch nicht die Anzeige, für die Anzeige werden stets aktuelle Werte ermittelt. Sobald die Anzeige wechselt, wird der entsprechende Wert ermittelt und angezeigt. Anschliessend werden Änderungen des aktuell dar- gestellten Wertes registriert und bei jeder Änderung wird die Anzeige aktualisiert. Der Ablauf ist in der Abbildung 18 (Ermittlung und Darstellung der Messwerte) ersichtlich. In Abbildung 19 ist der Programmfluss der periodischen Speicherung der Messwerte visualisiert. Dieser Vorgang läuft unter einem separaten Thread kontinuierlich ab.

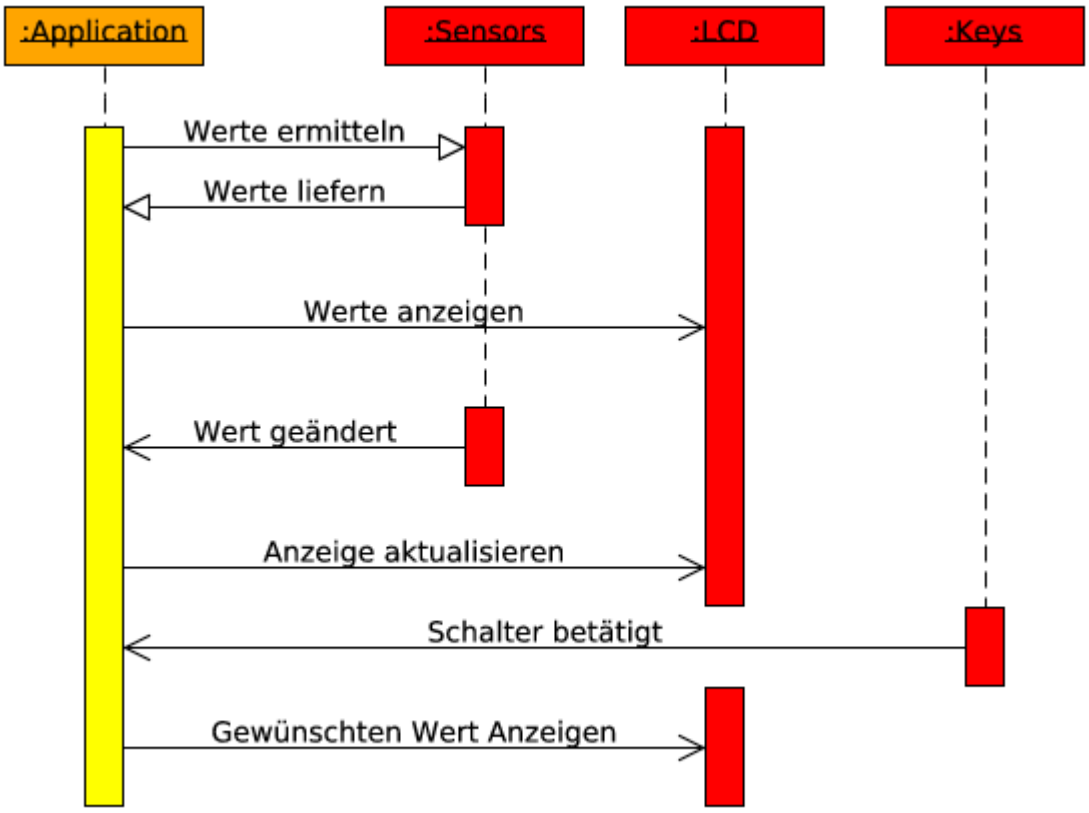

Abbildung 18: Die Anwendung reagiert auf Änderungen der Messwerte und auf Tastendruck

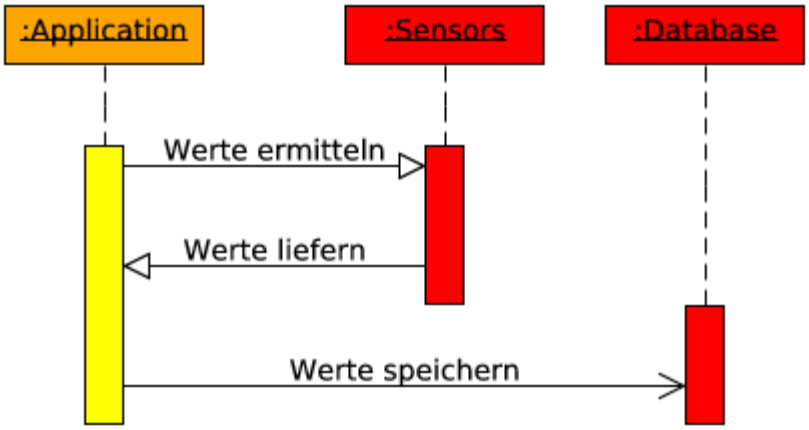

Abbildung 19: Die Messwerte werden periodisch mittels Polling gespeichert

8.2.5 Struktur

Statt einen monolithischen Ansatz zu verfolgen und dem LCD die Rolle der View zu zuordnen, sollte eher angestrebt werden, einen modularen Aufbau zu erzielen. Da andernfalls das Hinzufügen von weiteren Sensoren und die Implementierung der View schwierig sind. Eine einfache Indirektion löst dieses Problem: Statt dass die View auf alle Sensoren lauscht, existiert zu jedem Sensor eine kleine View, welche nur auf einen einzelnen Sensor lauscht und das LCD als Zeichenoberfläche erhält. Auf diese Weise können die einzelnen Sensoren leichter ausgetauscht werden und die Anwendung weist eine schwächere Kopplung auf. Zusammenfassend ergibt sich die Klassenstruktur wie sie in Abbildung 3 dargestellt ist. Der Grund, weswegen keine Schnittstelle Observer definiert wurde, liegt darin, dass mit C++ gearbeitet wird und statt einer Schnittstelle für den Observer lediglich ein Funktionsobjekt verwendet werden kann. Da der Funktor vorgängig mit Argumenten gebunden werden kann, muss dem Observer beim Aufruf kein spezifisches Argument mitgegeben werden. Generell interessiert den Beobachter lediglich, dass sich der Wert des beobachteten Sensors geändert hat. Eine Schnittstelle des Beobachters, welche beispielsweise den Sensor als Argument erhält, wäre nutzlos, da es sich bei dem Sensor, welcher der Observer erhalten würde, um eine Schnittstelle handeln würde, um den konkreten Sensor zu erhalten, wäre daher ein casting notwendig. Dies entfällt durch das vorgängige Binden des Funktors, beispielsweise kann eine Instanz einer `std::function<void()>` mittels einer Closure oder einer gebundenen Member Funktion erzeugt werden.

Aus Platzgründen wurden nicht alle Attribute und Methoden der Klasse Application notiert, generell handelt es sich um eine grobe Sicht auf die Architektur. Eine zentrale Rolle spielt der Konstruktor der Klasse, darin werden die Views beiden Sensoren als Beobachter

registriert und die Anzeige mit den aktuellen Werten versehen. Bei einem Wechsel der Anzeige können schliesslich die registrierten Beobachter deregistriert werden um die IP Adresse oder im Fehlerfall die Fehlermeldung anzuzeigen. Eine Alternative zu dem erwähnten Modell, in welchem die Views auf Änderungen am Modell lauschen, besteht darin, dass nicht die Views direkt auf Änderungen am Display lauschen sondern Application. Dies kann realisiert werden, indem wie angedeutet, eine Methode `onValueChanged` als Grundlage dient, einen Observer zu erstellen. Die Methode kann mit `this` und der zugehörigen View gebunden werden, um einen Funktor zu erzeugen, welcher als Beobachter eines Sensors dient. Ana- log dazu muss schliesslich noch dasselbe für den Thread `dbWriter` getan werden, indem statt bloss den Observer mit dem Sensor zu speichern, zusätzlich noch ein Funktor gespeichert wird, welcher den zugehörigen Wert in Values setzt. Schliesslich kann der Thread die Map durchwandern, jeden zugehörigen Funktor mit einem Value aufrufen, den Zeit Stempel setzen und die so assemblierten Werte mittels der Datenbank speichern. Im Falle eines Wechsels der Anzeige werden jeweils die Beobachter bei den Sensoren an- resp. abgemeldet.

In der Abbildung 20 auf der nächsten Seite wird das Klassendiagramm der Steuerung dargestellt.

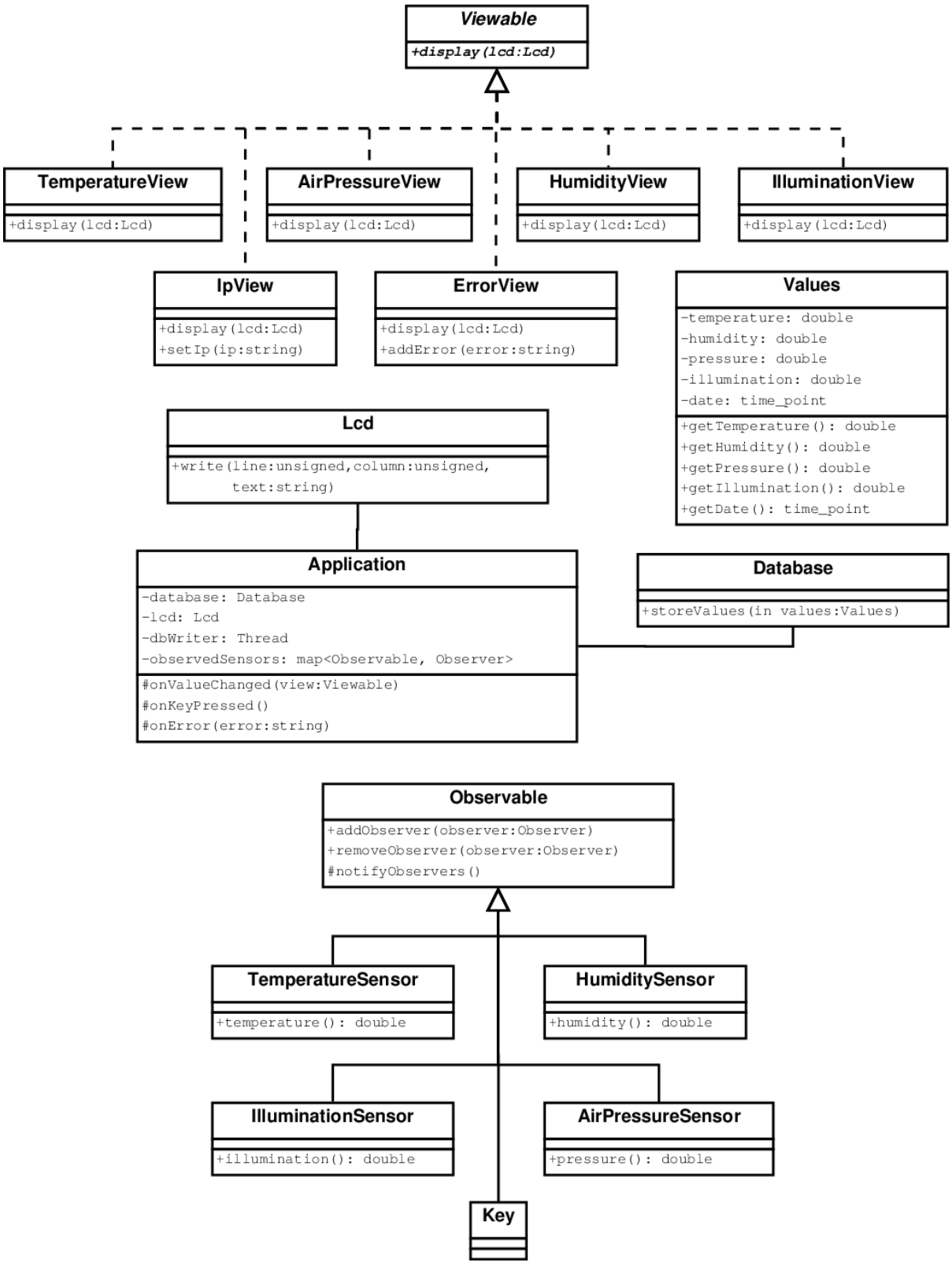

Abbildung 20: Die Messwerte werden periodisch mittels Polling gespeichert

8.3 Datenbank

Eine Sammlung von Messwerten wird in der Datenbank abgelegt, zu welcher folglich ein entsprechendes Schema folgende Felder umfasst (ohne Primärschlüssel):

Feldbezeichnung (Tabelle piw_values)	Datentyp
Id	Integer
Created_At (Unix Timestamp)	Integer
Temperature	Real
Humidity	Real
Pressure	Real
Illumination	Real

Das Auslesen der Messwerte aus der Datenbank wird durch eine andere Anwendung durchgeführt (Web-Anwendung), daher muss die Abstraktion, welche hier eingesetzt wird, dies vorerst nicht unterstützen. Die Daten werden in einer SQLite Datenbank abgelegt, auf diese Weise muss kein separater Datenbankserver betrieben werden und eine externe Anwendung (Web Anwendung) kann ebenfalls mittels einem standardisierten Verfahren (SQL) auf die Daten zugreifen.

8.4 Webseite

8.4.1 Benutzeroberfläche

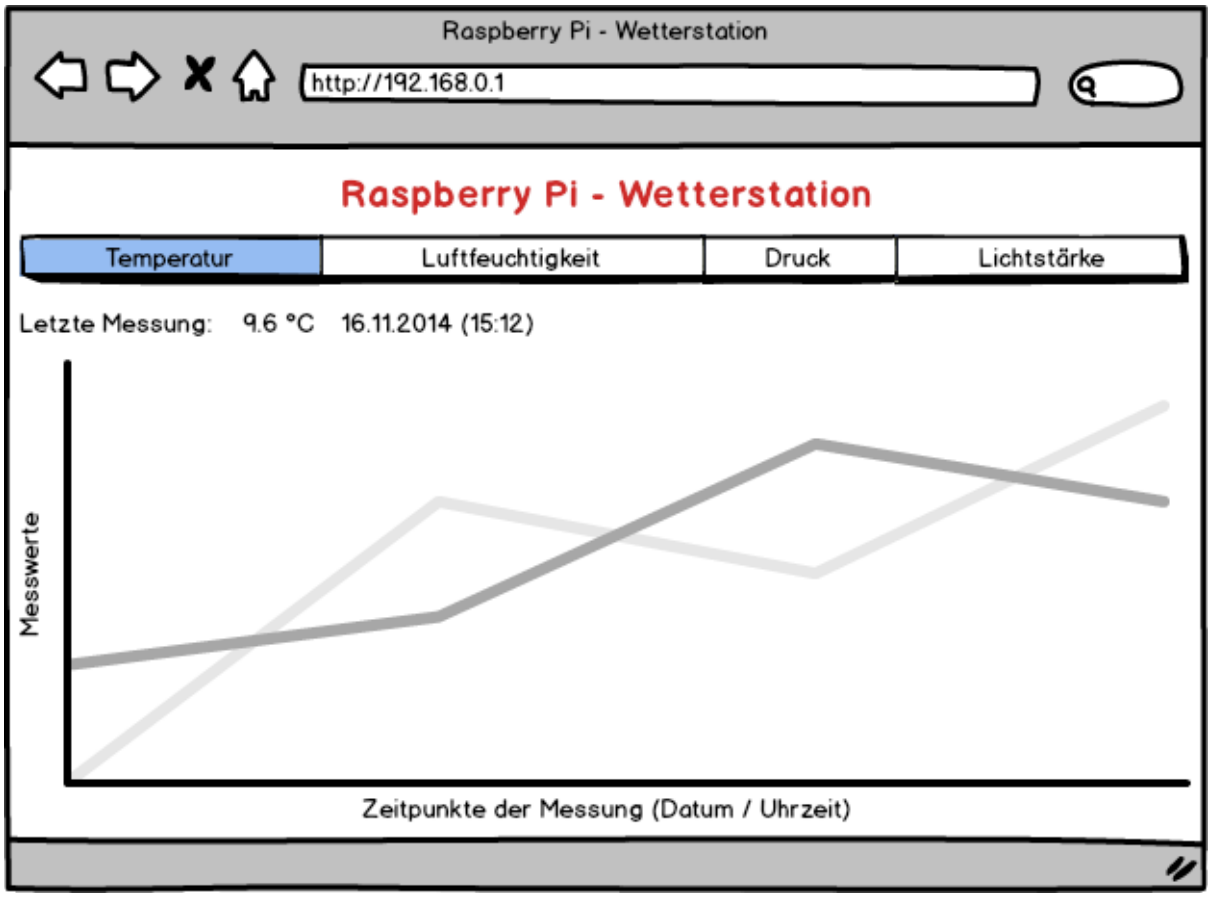

Abbildung 21: Benutzeroberfläche der Web-Anwendung

Auf der Benutzeroberfläche kann zwischen den unterschiedlichen Sensordaten navigiert werden. Auf der entsprechenden Seite wird der aktuelle (zuletzt gemessene) Datensatz dargestellt, sowie ein Verlauf der gemessenen Werte der vergangenen 7 Tage.

8.4.2 Software Entwurf

Im der nachfolgenden Abbildung wird das Design der Web-Anwendung schematisch dargestellt.

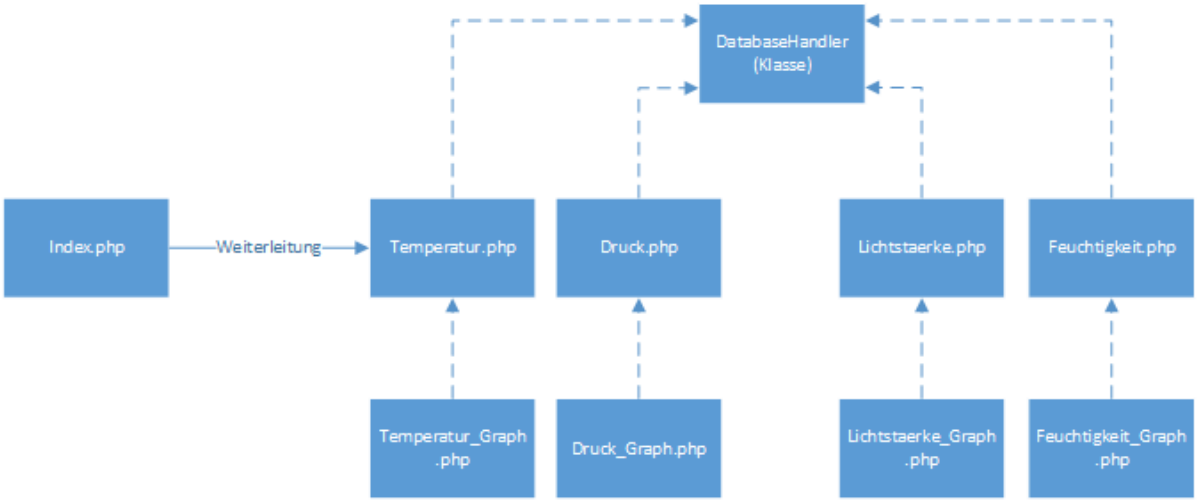

Abbildung 22: Benutzeroberfläche der Web-Anwendung

Es werden 9 PHP Seiten erstellt (Index, Temperatur, Druck, Lichtstärke und Feuchtigkeit) wobei Index nur zur Weiterleitung auf die Seite der Temperatur verwendet wird. Für den Zugriff auf die SQLite Datenbank (zur Abfrage der letzten Messwerte) wird eine separate Klasse erstellt (DatabaseHandler) welche durch alle Seiten mittels einem Include verwendet werden kann. Die *Graph* Dateien erstellen mittels einem Datenbankzugriff die Diagramme, welche auf den einzelnen Seiten als Bild dargestellt werden.

Folgende Komponenten werden für den Betrieb der Webseite benötigt und auf dem Raspberry Pi installiert:

- Apache Webserver mit PHP 5 Bibliothek
- JPGraph Bibliothek (Diagrammerstellung mit PHP)
- SQLite 3 / PHP-Erweiterung für SQLite 3

9 Qualitätssicherung

Um den notwendigen Qualität Standard zu ermitteln, werden die funktionalen Anforderungen herbeigezogen um daraus entsprechenden Test Szenarien abzuleiten.

- Luftdruck, Temperatur, Feuchtigkeit und Lichtstärke müssen korrekt ermittelt, dargestellt und gespeichert werden.
    - Mittels manuellen Tests und einem Referenz Gerät kann verifiziert wer- den, ob die Werte in einem gültigen Toleranzbereich liegen. Falls dies über das LCD verifiziert werden kann, ist auch sichergestellt, dass die Werte entsprechend ermittelt werden.
    - Mittels manueller Überprüfung kann verifiziert werden, ob der entsprechende Inhalt in der Datenbank abgelegt wurde. Beispielsweise mittels einer SQLite Shell.
- Sämtliche Informationen müssen über eine Web Applikation eingesehen wer- den können.
    - Auch dies kann mittels manueller Überprüfung auf dieselbe Weise verifiziert werden: Sämtliche online dargestellten Werte müssen im Vergleich mit den Werten eines Referenz Gerätes in einem gültigen Toleranzbereich liegen. Zudem kann überprüft werden, ob zuvor gespeicherte Daten abrufbar sind.

Zudem müssen weitere Risiken berücksichtigt werden, wobei externe Fehlerquellen wie ein Ausfall der Stromzufuhr ignoriert werden, solche Fehler können nicht adressiert werden. Ein Ausfall des LCD oder ein festsitzender Schalter können ebenfalls nicht behandelt werden. Es sollte jedoch sichergestellt werden, dass keine falschen Werte gespeichert werden, falls ein Sensor ausfällt, sollte dies von der Anwendung erkannt werden.

Zudem ist es möglich, dass irgendwann der Speicher ausgeht. Es wäre auch denk- bar, dass eine sehr grosse Datenbank Probleme verursacht. Es bleiben zusätzlich die folgenden möglichen Fehler, welche behandelt werden müssen:

- Ausfall von Sensoren
    - Im Falle von ausgefallenen Sensoren sollen die entsprechenden Werte nicht gespeichert werden. Auf dem LCD soll in diesem Fall eine Fehlermeldung angezeigt werden.
- Kein Speicherplatz mehr verfügbar

  - Falls das Speichern der Messwerte fehlschlägt, weil kein Speicherplatz mehr verfügbar ist, soll auf dem LCD ebenfalls eine Fehlermeldung angezeigt werden.
- Kein Netzwerk verfügbar
  - In diesem Fall soll bei einem Wechsel der Anzeige zur IP Adresse eine Fehlermeldung angezeigt werden.

10 Testing

10.1 Test Konzept

Folgende Ziele werden mit den Tests der Wetterstation verfolgt:

- Funktionen gemäss den in Kapitel 6 definierten Use Cases überprüfen
- Überprüfung der Umsetzung aller funktionaler Anforderungen
- Allgemeine Stabilität / Fehlertoleranz der Wetterstation

Nicht getestet werden in diesem Zusammenhang die nicht funktionalen Anforderungen (die nicht funktionalen Anforderungen werden mit der Abschluss Präsentation des Projekts als erfüllt angesehen). Der Test umfasst einen kombinierten System- und Abnahmetest welcher separat durch die 2 Projektmitglieder (Andreas Hasler, David Daniel) ausgeführt werden. Der Test hat den Charakter eines Blackbox-Tests (ohne genauere Analyse des Source-Codes der Steuerung oder der Web-Anwendung).

10.1.1 Testabbruch

Sollten während dem Test gravierende Mängel zum Vorschein kommen (Steuerung wird nicht gestartet oder stürzt ab, Web-Anwendung nicht erreichbar oder startet mit Exceptions) wird der Test abgebrochen und zur Implementierungsphase zurückgekehrt. Bei einer allfälligen Wiederaufnahme der Tests ist darauf zu achten, dass alle Tests nochmals durchlaufen werden müssen.

10.1.2 Hardware / Software

Für die Tests wird die Hardware der Wetterstation mit integriertem Raspberry Pi verwendet (die Web-Anwendung benötigt den Apache Webserver als Grundlage). Für die Web-Anwendung kann jeder gängige Browser (bspw. Internet Explorer, Firefox, Opera, Safari etc.) verwendet werden. Um die Messwerte der Wetterstation auf Plausibilität zu prüfen wird zusätzlich ein separates Gerät benötigt, welches ebenso diese Messwerte ermitteln kann.

10.1.3 Toleranzen der Messwerte

Gegenüber dem zum Test verwendeten externen Gerät zur Überprüfung der Messwerte gelten folgende Toleranzen:

- Feuchtigkeit: +/- 2 %
- Temperatur: 1 °C
- Lichtstärke: 10 lx
- Luftdruck: 5 hPa

10.2 Testfälle

TC 1 – Einschalten	
Beschreibung	Die Wetterstation wird eingeschaltet.
Bezieht sich auf…	• Use Case 1 (Einschalten) • Use Case 5 (Verbinden)
Vorbedingungen	Die Wetterstation ist ausgeschaltet.
Ablauf	Wetterstation an das 230V Netz anschliessen
Erw. Ergebnisse	Die Wetterstation beginnt nach dem Boot-Vorgang des Raspberry Pi mit der Aufzeichnung der Wetterdaten (sichtbar auf dem LCD-Display).

TC 2 – Messwerte anzeigen	
Beschreibung	Die Messwerte der Sensoren werden auf dem LCD-Display angezeigt (Plausibilitätskontrolle der Werte)
Bezieht sich auf…	• Use Case 2 (Messwerte anzeigen) • Use Case 6-9 (Luftdruck, Temperatur, Feuchtigkeit, Lichtstärke ermitteln)
Vorbedingungen	• TC 1 erfüllt (Wetterstation gestartet) • Möglichkeit der parallelen Messung der Wetterdaten • Die Sensoren sind funktions- und einsatzfähig
Ablauf	Schalter 1 an der Wetterstation betätigen (wurde das Raspberry Pi neu gestartet, wird dieser Schritt nicht benötigt)
Erw. Ergebnisse	Die Messwerte werden plausibel auf dem LCD-Display angezeigt (Kontrolle mittels der parallelen Messung der Wetterdaten).

TC 3 – IP-Adresse anzeigen	
Beschreibung	Die IP-Adresse wird auf dem Display des Raspberry Pi angezeigt
Bezieht sich auf…	Use Case 4 (IP-Adresse anzeigen)
Vorbedingungen	TC 1 erfüllt (Wetterstation gestartet)
Ablauf	Schalter 1 an der Wetterstation betätigen
Erw. Ergebnisse	• Die IP-Adresse des Raspberry Pi wird auf dem LCD-Display angezeigt. • Die dargestellte IP-Adresse ist mittels einem Ping von einem Computer im selben Netzwerk erreichbar.

TC 4 – Messwerte in Datenbank speichern	
Beschreibung	Die von der Steuerung ermittelten Messwerte werden in der Datenbank abgelegt
Bezieht sich auf…	Use Case 10 (Messwerte in DB speichern)
Vorbedingungen	TC 1 erfüllt (Wetterstation gestartet)
Ablauf	Mittels SQLite Shell wird auf dem Raspberry Pi geprüft, ob die Messwerte in die Datenbank übernommen wurden.
Erw. Ergebnisse	Die Messwerte wurden in der Datenbank gespeichert

TC 5 – Messdaten über Web-Anwendung abrufbar	
Beschreibung	Die von der Wetterstation ermittelten Messdaten können mittels der Web-Anwendung in einem Browser abgerufen werden
Bezieht sich auf…	Use Case 3 (Messdaten online abrufen)
Vorbedingungen	• TC 1 erfüllt (Wetterstation gestartet) • TC 3 erfüllt (IP-Adresse des Raspberry Pi bekannt) • TC 4 erfüllt (Daten werden in Datenbank gespeichert)
Ablauf	• Im Browser die IP-Adresse des Raspberry Pi eingeben • Die Messwerte in den einzelnen Registern abrufen
Erw. Ergebnisse	• Pro Bereich (Temperatur, Luftdruck, Luftfeuchtigkeit, Lichtstärke) müssen die Messwerte dargestellt werden • Die letzte Messung / die letzten 10 Messwerte in einem Diagramm mit dem zugehörigen Datum und der entsprechenden Zeit

TC 6 – Ausfall mind. eines Sensors	
Beschreibung	Während des Betriebs fällt mind. ein Sensor aus und kann keine Werte mehr an die Wetterstation liefern
Bezieht sich auf…	• Use Case 2 (Messwerte anzeigen) • Use Case 6-9 (Luftdruck, Temperatur, Feuchtigkeit, Lichtstärke ermitteln)
Vorbedingungen	TC 1 erfüllt (Wetterstation gestartet)
Ablauf	Verbindung zwischen Master und dem Sensor trennen (für alle Sensoren wiederholen)
Erw. Ergebnisse	Auf dem LCD-Display muss bei entsprechenden Wert ein Fehler angezeigt werden

TC 7 – Recovery mind. eines Sensors	
Beschreibung	Ist der Sensor wieder verfügbar, muss diesen Messwert von der Wetterstation auch wieder erfasst werden
Bezieht sich auf…	• Use Case 2 (Messwerte anzeigen) • Use Case 6-9 (Luftdruck, Temperatur, Feuchtigkeit, Lichtstärke ermitteln)
Vorbedingungen	• TC 1 erfüllt (Wetterstation gestartet) • TC 6 erfüllt (Verbindung zu mind. einem Sensor getrennt)
Ablauf	Verbindung zwischen Master und dem Sensor wieder herstellen
Erw. Ergebnisse	Der Messwert des entsprechenden Sensors muss wieder auf dem LCD-Display erscheinen

10.3 Issues

Während der Testphase wurde für die Issues auf aceproject.com ein Bugtracker eingerichtet, damit die Issues welche beim Testen auftreten übersichtlich definiert sind.

Issue	Beschreibung	Status
#1	Steuerung startet nicht wenn ein Sensor ausgesteckt ist	**OK**
#2	Keine Fehlermeldung bei ausgestecktem Sensor während dem Betrieb der Steuerung	**CLOSED (NOT POSSIBLE)**
#3	Temperatur nicht plausibel	**CLOSED (NOT POSSIBLE)**

10.4 Systemtest / Abnahmetest

Während dem Systemtest / Abnahmetest wurde das folgende Protokoll erstellt:

Testfall	Beschreibung	Resultat
TC 1	Einschalten	**OK**
TC 2	Messwerte anzeigen	**OK** **(Temperatur NOT OK)**
TC 3	IP-Adresse anzeigen	**OK**
TC 4	Messwerte in Datenbank speichern	**OK**
TC 5	Messdaten über Web-Anwendung abrufbar	**OK**
TC 6	Ausfall mind. eines Sensors	**NOT POSSIBLE**
TC 7	Recovery mind. eines Sensors	**OK**

Ergänzung zum TC 2: Alle Messwerte wurden korrekt ausgelesen und waren plausibel bis auf die Temperatur. Das Problem ist, dass der Barometer-Brick die Temperatur des Chips und nicht der Umgebung misst. Bei der Anschaffung der Bausteile der Wetterstation war dies auf den ersten Blick in der Spezifikation nicht klar ersichtlich. Eine mögliche Lösung ist einen speziellen Temperatur-Brick einzusetzen (verfügbar ebenfalls von Tinkerforge).

Ergänzung zum TC 6: Der Master-Brick liefert auch einen Wert des Sensors wenn dieser nicht mit verbunden ist. Aus diesem Grunde ist es nicht eindeutig möglich während dem Betrieb herauszufinden, wenn ein Sensor nicht mehr verbunden ist. Dies ist das Problem des Frameworks von Tinkerforge.

Entscheid über das Testresultat: Test bestanden bis auf die Temperatur

Die Tests wurden durchgeführt von: Andreas Hasler / David Daniel (14.12.2014)

11 Fazit

Grundsätzlich sind wir als Team mit dem Resultat der Projektarbeit zufrieden. Der zu Beginn des Projektes scheinbare Vorteil auf eine, bis auf das zusammenbauen der Komponenten, beinahe fertige Hardware von Tinkerforge aufzubauen hat sich im Laufe des Projektes als Herausforderung herausgestellt. So konnten wir beispielsweise auf Grund des Frameworks von Tinkerforge die Ermittlung eines ausgesteckten (defekten) Sensors nicht umsetzen. Zudem hat uns bei der Anschaffung die Spezifikation des Barometer-Bricklets in die Irre geführt, so dass die Temperaturanzeige nicht komplett zufriedenstellend ist. Dies könnte gelöst werden, in dem ein separater Temperatur-Brick eingesetzt wird und so die Hardware und die Steuerung noch ergänzt werden müssen. Aus Zeitgründen war dies im Umfang des Projekts nicht mehr möglich (dabei ist auch die Lieferzeit einzuberechnen).