

Hochschule für angewandte Wissenschaften Rosenheim

Fachbereich Informatik

## **Dokumentation**

**für das Projekt DAT:**

**Wetterstation auf Basis eines Arduino Mikrocontrollers mit Android  
Smartphoneanbindung**

**Version:** 0.8

16. Januar 2013

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Aufgabenstellung . . . . .	5
1.2	Hardware . . . . .	5
<b>2</b>	<b>Projektorganisation</b>	<b>6</b>
2.1	Team . . . . .	6
2.2	Kommunikation . . . . .	6
<b>3</b>	<b>Arbeitsmittel</b>	<b>7</b>
3.1	Versionsverwaltung . . . . .	7
3.2	Dokumentation . . . . .	7
3.3	Programmiersprachen . . . . .	7
3.3.1	Arduino . . . . .	7
3.3.2	Webseiten auf dem Arduino und dem externen Server . . . . .	7
3.3.3	Android . . . . .	7
3.3.4	Webserver . . . . .	7
3.4	Entwicklungsumgebung . . . . .	8
3.4.1	Arduino . . . . .	8
3.4.2	Eclipse . . . . .	8
3.5	Hardware . . . . .	8
3.5.1	Arduino Mikrocontroller . . . . .	8
3.5.2	Temperatur- und Luftdrucksensor BMP085 . . . . .	9
3.5.3	Luftfeuchtigkeitssensor HIH4030 . . . . .	9
<b>4</b>	<b>Realisierung</b>	<b>10</b>
4.1	Gesamtsystem . . . . .	10
4.2	Arduino Sensoren . . . . .	10
4.3	Arduino Webserver . . . . .	11
4.3.1	JSON Interface . . . . .	11
4.3.2	CSV Daten . . . . .	12
4.4	Android App . . . . .	12
<b>5</b>	<b>Probleme</b>	<b>14</b>
5.1	Arduino . . . . .	14
<b>6</b>	<b>Fazit</b>	<b>14</b>
<b>7</b>	<b>weiterer Ausbau</b>	<b>14</b>
7.1	Ausbau Arduino . . . . .	14
<b>8</b>	<b>Quellen</b>	<b>15</b>
<b>9</b>	<b>Glossar</b>	<b>17</b>
<b>A</b>	<b>Installationsanleitung WeatherWidgetApp Version 1.0</b>	<b>18</b>

<b>B</b>	<b>Installationsanleitung Arduino Programm Version 1.0</b>	<b>21</b>
<b>C</b>	<b>Installationsanleitung arduino.pl Version 1.0</b>	<b>25</b>
<b>D</b>	<b>Release Letter</b>	<b>26</b>
D.1	Inhalte . . . . .	26
D.2	Funktionalität Arduino Wetterstation . . . . .	26
D.3	Funktionalität Android Smartphone Widget . . . . .	27
D.4	Zusatzinformationen . . . . .	27
D.5	Ansprechpartner . . . . .	27

## Abbildungsverzeichnis

1	Arduino Uno . . . . .	5
2	Arduino Mega . . . . .	6
3	Temperatur- und Luftdrucksensor BMP085 . . . . .	9
4	Luftfeuchtigkeitssensor HIH-4030 . . . . .	10
5	Systemarchitektur . . . . .	10
6	Arduino Uno Sensor Verkabelung . . . . .	11
7	Arduino Mega Sensor Verkabelung . . . . .	12
8	JSON Daten . . . . .	13
9	CSV Daten . . . . .	13
10	Widget Screenshots . . . . .	16
11	Widget Screenshots . . . . .	19

# 1 Einleitung

## 1.1 Aufgabenstellung

Ziel des Projekts ist es eine Wetterstation auf Basis eines Arduino Mikrocontrollers zu erstellen. An dem Arduino sollen folgende Sensoren angeschlossen werden:

- Luftdrucksensor (siehe Kapitel 3.5.2)
- Temperatursensor (siehe Kapitel 3.5.2)
- Feuchtigkeitssensor (siehe Kapitel 3.5.3)

Der Arduino soll alle Sensoren in einem festgelegten Zeitintervall (z.B. alle 5 min.) auslesen, und entsprechend in einem Webinterface die aktuellen Werte anzeigen. Weiterhin soll eine History der gemessenen Wert auf einer angeschlossenen Micro-SD-Karte gespeichert werden. Die Messdaten sollen weiterhin bei Bedarf von dem Arduino über ein Ethernet-Shield an einen Webserver geschickt werden. Dieser wertet die Daten graphisch auf und zeigt sie als Zeitdiagramm in einer Webgui an. Ergänzend zu der Weboberfläche auf dem Webserver und dem Controller soll eine Smartphone App auf Android-Basis entstehen, mit welcher die Messdaten des Arduino graphisch angezeigt werden können.

## 1.2 Hardware

Als Mikrocontroller wurde, wie bereits in Punkt 1.1 erwähnt, ein Arduino verwendet. Die Sensor-Entwicklung wurde mit einem Arduino Uno (siehe Abbildung 1) begonnen.

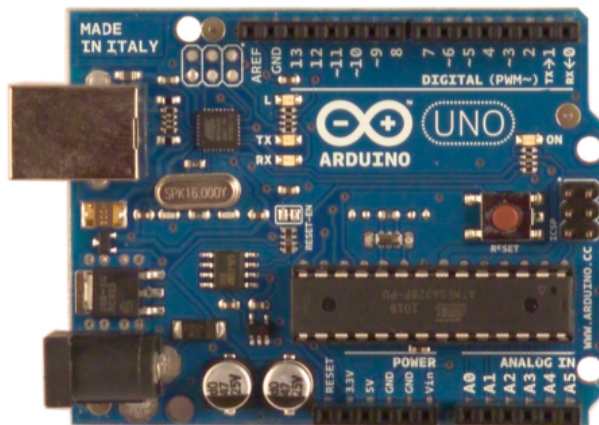


Abbildung 1:  
Arduino Uno

Während der Entwicklung stellte sich heraus, dass der Speicher des Arduino Uno für alle gestellten Anforderungen (SD-Karte, Webinterface, Webclient, usw.) nicht ausreichend ist. Aus diesem Grund gibt es eine eigene Version für den Uno ohne Unterstützung der SD-Karte. Um alle Features zu nutzen, ist der Arduino Mega (siehe Abbildung 2) nötig. Dieser besitzt unter anderem mehr Speicher, um grössere Programme aufnehmen zu können. Es muss hier für den Mega kein zusätzlicher Entwicklungsaufwand betrieben werden, da der Source-Code des Uno mit allen Sensoren einfach übernommen werden kann.

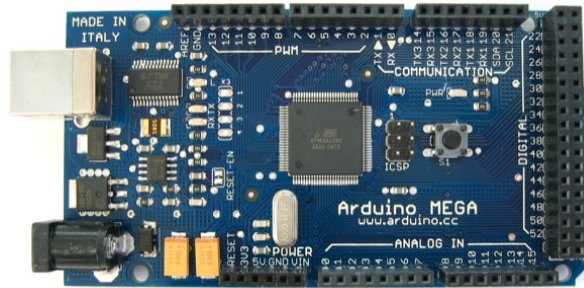


Abbildung 2:  
Arduino Mega

## 2 Projektorganisation

### 2.1 Team

Das Team bestand aus folgenden zwei Informatik Studenten im siebten Fachsemester der Hochschule Rosenheim.

Name	E-Mail
Alexander Zenger	weather@zengers.de
Maximilian Oberberger	arduinoWeather@oberbergers.de

### 2.2 Kommunikation

Die Kommunikation während der Projektphase erfolgte vorrangig über E-Mail und durch Tickets im Bug-Tracking System Trac<sup>1</sup>, welches auf einem privaten Server betrieben wurde.

---

<sup>1</sup>Projekthomepage: <http://trac.edgewall.org>

## 3 Arbeitsmittel

### 3.1 Versionsverwaltung

Um jederzeit Zugriff auf alle aktuellen projektspezifischen Dokumente und Quellcode zu haben und um Änderungen nachverfolgen zu können haben wir uns für das dezentrale Versionsverwaltungstool Git<sup>2</sup> entschieden. Das Repository wurde auf dem selben privaten Server wie Trac betrieben, und war in selbigem einsehbar.

### 3.2 Dokumentation

Für die Erstellung unserer Dokumente wird das Textsatzsystem LaTeX<sup>3</sup> verwendet. LaTeX ist ein Softwarepaket, welches die Benutzung des Textsatzprogramms TeX<sup>4</sup> mit Hilfe von Makros vereinfacht.

Die Erstellung der Dokumentation für den Quellcode unserer Android Applikation erfolgt mit Hilfe des Tools doxygen<sup>5</sup>. Doxygen erlaubt es, auf sehr einfache Weise, die Dokumentation aus den Kommentaren im Quellcode zu erstellen.

### 3.3 Programmiersprachen

Im folgenden Abschnitt werden die verwendeten Programmiersprachen kurz erläutert.

#### 3.3.1 Arduino

Für die Entwicklung der Arduino-Software wird die Arduinotypische Sprache C, bzw. C++ verwendet.

#### 3.3.2 Webseiten auf dem Arduino und dem externen Server

Die Darstellung der Webseiten des Webservers auf dem Mikrocontroller selbst, sowie dem externen Server wird mit HTML realisiert.

#### 3.3.3 Android

Die App wird Androidtypisch in Java 6 und XML implementiert.

#### 3.3.4 Webserver

Während der Entwicklung war ein Unix-Webserver (Debian 6) im Einsatz. Auf diesem wurde über einen Apache2-Webserver ein Perl-cgi-bin-Skript bereitgestellt.

---

<sup>2</sup>Erhältlich auf der Projekthomepage: <http://git-scm.com>

<sup>3</sup>Genauere Informationen findet man auf der Projekthomepage: <http://www.latex-project.org>

<sup>4</sup>siehe Wikipedia: <http://de.wikipedia.org/wiki/TeX>

<sup>5</sup>Dokumentation und Download findet man auf der Projekthomepage: <http://www.stack.nl/~dimitri/doxygen/>

## 3.4 Entwicklungsumgebung

### 3.4.1 Arduino

Es wird die Version 1.0.1 (Stand 09.07.2012 die aktuellste) der Arduino eigenen Entwicklungsumgebung für die Entwicklung verwendet. Die Entwicklung der Software kann Betriebssystem unabhängig erfolgen, da die Arduino Entwicklungsumgebung für jede Plattform verfügbar ist (OSX, Windows, Linux).

### 3.4.2 Eclipse

Um das Android-Smartphone Widget zu implementieren, wird die Eclipse Entwicklungsumgebung in Verbindung mit dem Android SDK von Google verwendet <sup>6</sup>. Es wird die Android Version 2.2 für die Entwicklung der App verwendet, da diese zum Zeitpunkt der Entwicklung die Unterstützung für die meisten Geräte hat.

Auch die Entwicklung des Widgets kann Betriebssystem unabhängig erfolgen, da Eclipse ebenfalls für jede Plattform verfügbar ist.

## 3.5 Hardware

### 3.5.1 Arduino Mikrocontroller

Der Anfangs verwendete Arduino Uno (siehe Abbildung 1) ist ein auf ATmega328 basierender Mikrocontroller. Er hat 14 digitale E/A Ports, 6 analoge Eingänge, einen 16 MHz Oszillator, einen USB-Anschluss, einen Stromanschluss, einen ICSP Header und einen Reset-Button.

#### **Zusammenfassung:**

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

Anschließend wurde der Arduino Mega (siehe Abbildung 2) verwendet. Er basiert auf einem ATmega1280. Er hat 54 digitale E/A Ports, 16 analoge Eingänge, einen 16 MHz Oszillation, einen USB-Anschluss, einen Stromanschluss, einen ICSP Header und einen Reset-Button.

---

<sup>6</sup>Installationsanleitung findet man auf der Google eigenen Entwickler Seite:  
<http://developer.android.com/tools/sdk/eclipse-adt.html>



**Zusammenfassung:**

Microcontroller	ATmega1280
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	128 KB of which 4 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

**3.5.2 Temperatur- und Luftdrucksensor BMP085**

Es wird der Sensor BMP085 der Firma Bosch (siehe Abbildung 3 mit Breakout Board) für die Temperatur- und Luftdruckmessungen verwendet. Dieser Sensor hat bei einer Messgenauigkeit von 0.3 hPa einen Luftdruck-Messbereich von 300 bis 1100 hPa. Er besitzt eine I2C-Bus Anbindung, sowie einen Versorgungsspannungsbereich von 1.8 bis 3.6 Volt.



Abbildung 3:  
BMP085

**3.5.3 Luftfeuchtigkeitssensor HIH4030**

Der verwendete Honeywell HIH-4030 Feuchtigkeitssensor (siehe Abbildung 4) gibt proportional zur Luftfeuchtigkeit eine analoge Spannung aus. Der Sensor besitzt einen Versorgungsspannungsbereich von 4 bis 5.8 Volt und hat eine typische Stromaufnahme von 200 $\mu$ A.



Abbildung 4:  
HIH-4030

## 4 Realisierung

Im folgenden wird auf die Realisierung unserer Wetterstation eingegangen.

### 4.1 Gesamtsystem

Das Gesamtsystem besteht aus dem Arduino-Mikrocontroller, den daran angeschlossenen Sensoren, einem optionalen Webserver sowie die implementierte Smartphone-App.

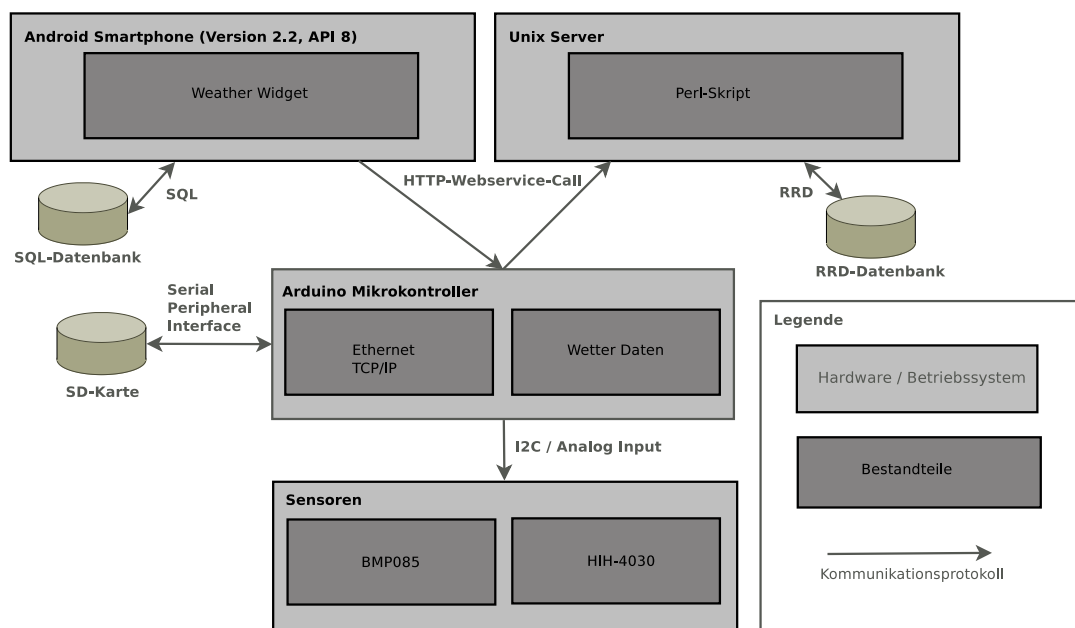


Abbildung 5:  
Systemarchitektur

### 4.2 Arduino Sensoren

Abbildung 6 zeigt wie die einzelnen Sensoren (in diesem Fall der Temperatur- und Luftdrucksensor bmp085, sowie der Luftfeuchtigkeitssensor hih4030) an den Arduino Uno angeschlossen werden.

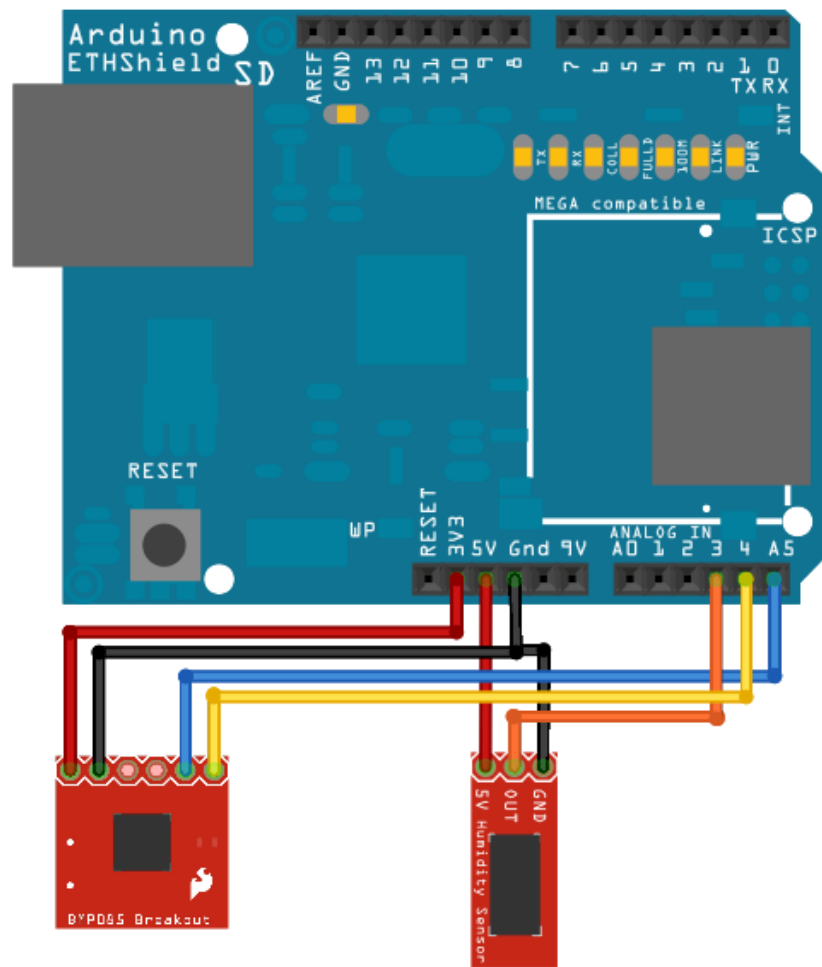


Abbildung 6:  
Arduino Uno Sensor Verkabelung

Der Anschluß der Sensoren an den Arduino Mega unterscheidet sich bezüglich des bmp085 im Vergleich zum Arduino Uno. Die Abbildung 7 zeigt dies.

### 4.3 Arduino Webserver

Der Webserver des Mikrocontrollers dient mehreren Zwecken. Zum einen können dort die Netzwerk-Einstellungen vorgenommen werden. Des weiteren können die verschiedenen Funktionen der Lösung gesteuert werden. Letztlich werden die aktuellen Sensordaten auf der Startseite angezeigt.

#### 4.3.1 JSON Interface

Der Webserver stellt die Sensordaten im JSON-Format bereit. Dies erfolgt über die URL `http://<ip>/json`. Diese können von jedem Gerät im Netzwerk mit einem HTTP-Request abgerufen werden, um die Daten entsprechend weiter zu verarbeiten. Dieses Interface nutzt

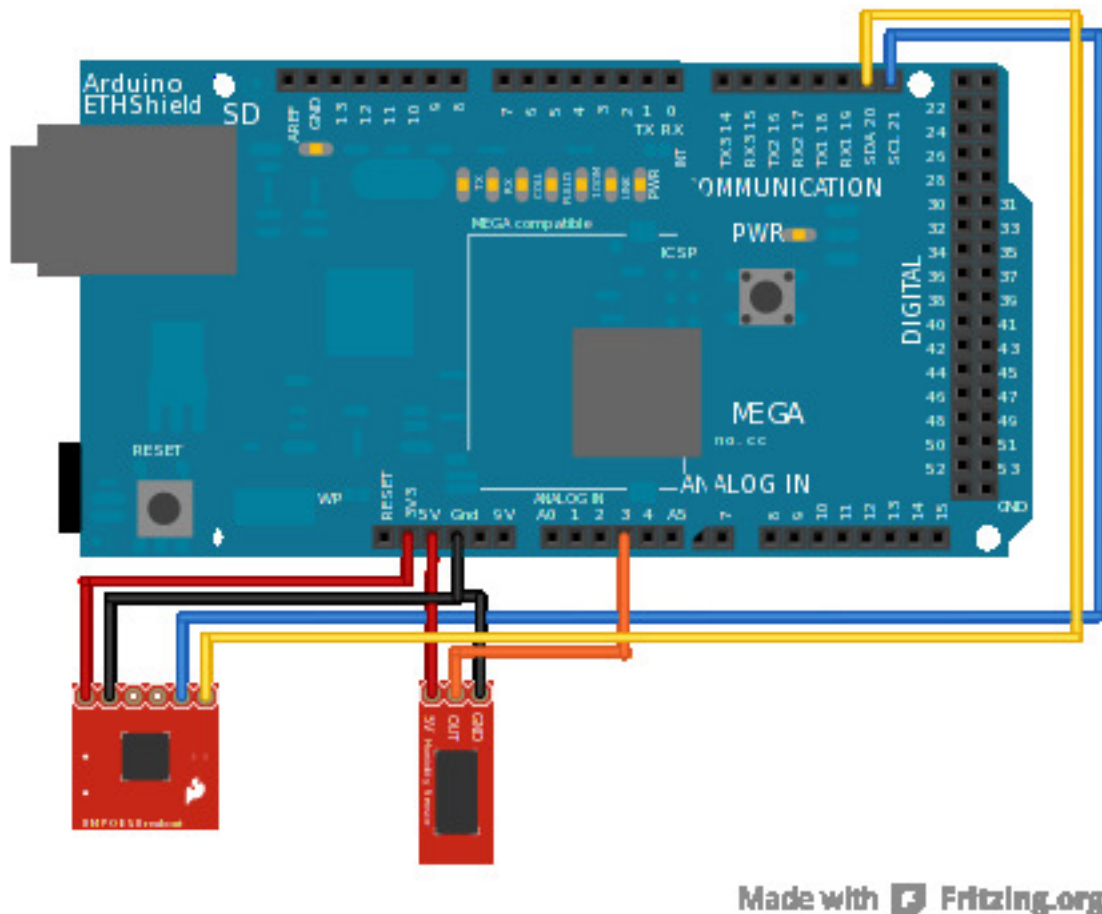


Abbildung 7:  
Arduino Mega Sensor Verkabelung

auch die im Projekt mitentwickelte Android App. Abbildung 8 zeigt Beispieldaten im JSON-Format.

#### 4.3.2 CSV Daten

Bei Benutzung des Arduino Mega ist es möglich, die Sensordaten auf eine Micro-SD-Karte zu schreiben. Diese wird dazu in dem entsprechenden Steckplatz auf dem EthernetShield geschoben. Der Controller schreibt, nach Aktivierung der Funktion in den Einstellungen, bei jedem Zyklus die aktuellen Sensordaten in die „data.csv“ auf der SD-Karte. Diese kann vom Webserver über die URL <http://<ip>/data.csv> heruntergeladen werden. Abbildung 9 zeigt dies.

### 4.4 Android App

Die Android App liefert ein Widget mit, welches ohne Eingriff des Nutzers alle halbe Stunde seine Daten aktualisiert. Über den „Sync“-Button kann der User aktiv eine Aktualisierung anstoßen (siehe Abbildung 10 (a), Seite 16).

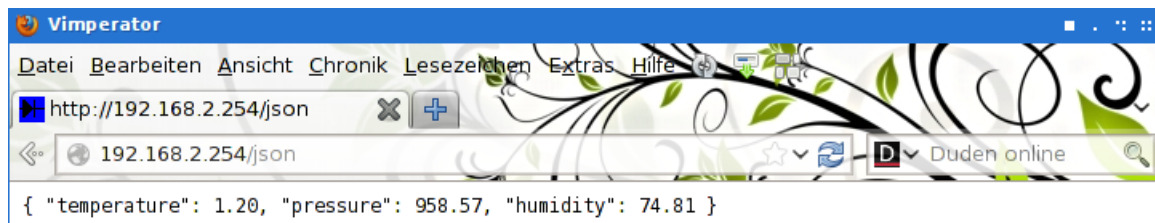


Abbildung 8: Sensordaten im JSON-Format

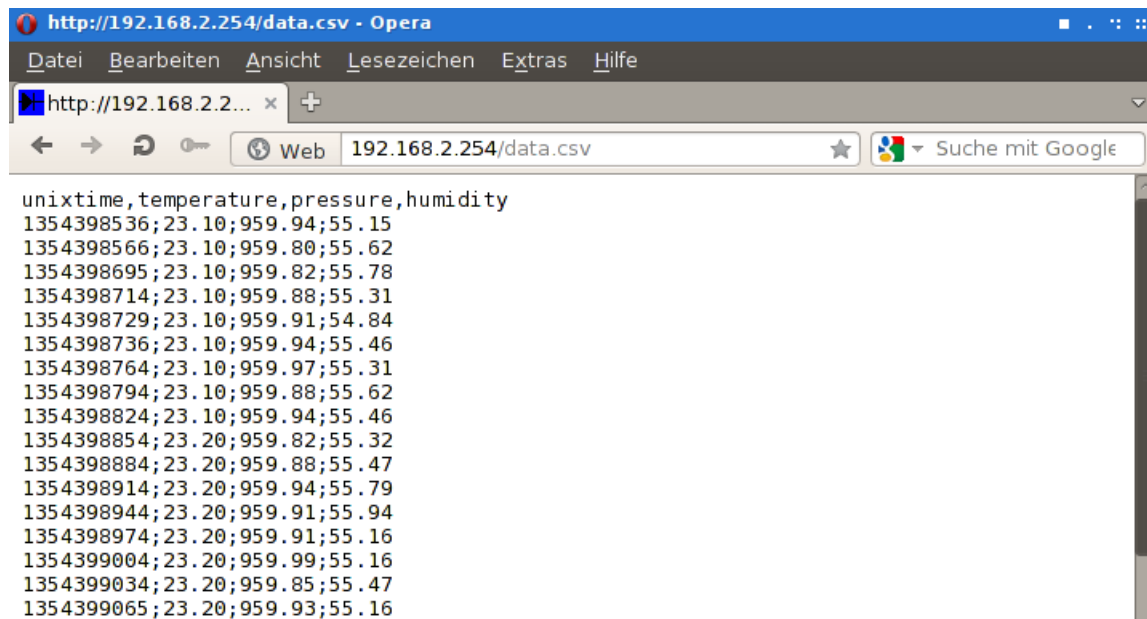


Abbildung 9: Sensordaten in CSV-Datei

Durch einen Klick auf das Widget selbst erreicht man die graphische Darstellung der Messdaten und kann weiter in das Einstellungsmenu navigieren (siehe Abbildung 10 (d)-(e), Seite 16). Über das Einstellungsmenu ist es dem Benutzer des Widgets möglich, einen anderen Server und/oder eine neue URL-Erweiterung einzugeben (siehe Abbildung 10 (b)-(c), Seite 16).

## 5 Probleme

### 5.1 Arduino

Während der Entwicklung kam es nach einiger Zeit zu Speicherproblemen auf dem Mikrocontroller. Dieser hat 32 KB Flashspeicher, welcher für den Bootloader und die Speicherung des Programms inklusive der Bibliotheken genutzt wird. Nachdem die grundlegenden Funktionen (Datenauslesen im Intervall, Webserver, Ausgabe auf der seriellen Konsole, Senden der Daten zum Server) implementiert waren, sollte es noch optional möglich sein die Daten auf einer SD-Karte zu speichern. Nach dem einbinden der entsprechenden Bibliothek, wurde jedoch die Kapazität des Flashspeicher überschritten. Versuche dem durch eine Reduzierung des Codes entgegen zu wirken schlugen fehl. Aus diesem Grund muss für dieses Feature der Arduino Mega verwendet werden.

## 6 Fazit

Insgesamt fällt unser Fazit des Projektes sehr positiv aus. Es konnten alle gesetzten Anforderungen / Ziele umgesetzt werden. Die aktuellen Sensordaten können direkt auf dem Controller über den dort laufenden Webserver abgerufen werden. Ebenso stellt dieser die Daten per JSON zur Verfügung. Des weiteren ist es möglich, die jeweils aktuellen Daten im 5 Minuten Intervall an einen Server per HTTP zu senden. Die Einstellungen dazu und andere können ebenso über das Webinterface angepasst werden. Wie am Anfang bereits vermutet, reicht der Arduino Uno nicht aus, um alle Features bereitzustellen. Deshalb ist für die Speicherung der Sensordaten der Arduino Mega notwendig. Mehr dazu unter [5.1](#), Seite [14](#). Außer diesem, sind keine größeren Probleme oder Schwierigkeiten aufgetreten, und das Projekt konnte erfolgreich abgeschlossen werden.

## 7 weiterer Ausbau

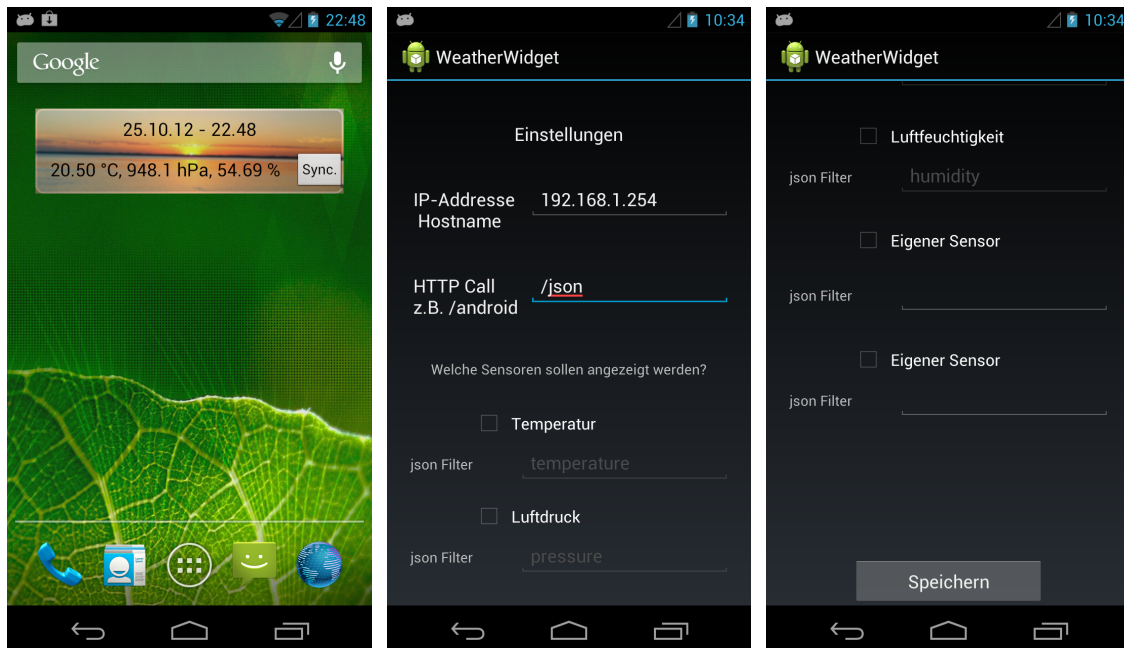
Während der Entwicklung sind einige Ideen aufgekommen, welche jedoch im Rahmen des Projektes nicht mehr umgesetzt werden konnten. Diese sollen hier in kurzer Form dokumentiert werden.

### 7.1 Ausbau Arduino

- Version mit SD-Karten Unterstützung für den Arduino Uno. Dafür auf andere Features verzichten.
- Ältere Sensor Daten auf der Hauptseite anzeigen. Z.B. im Array gespeichert.
- Bei Ziel-Server und NTP-Server statt IP, DNS-Namen eintragen.
- Aufbereitung der gesammelten Daten direkt auf dem Arduino. Z.B. mit Hilfe von Google Chart Tools.
- Einstellung der richtigen Zeitzone, mit Beachtung von Sommer- Winterzeit.

## 8 Quellen

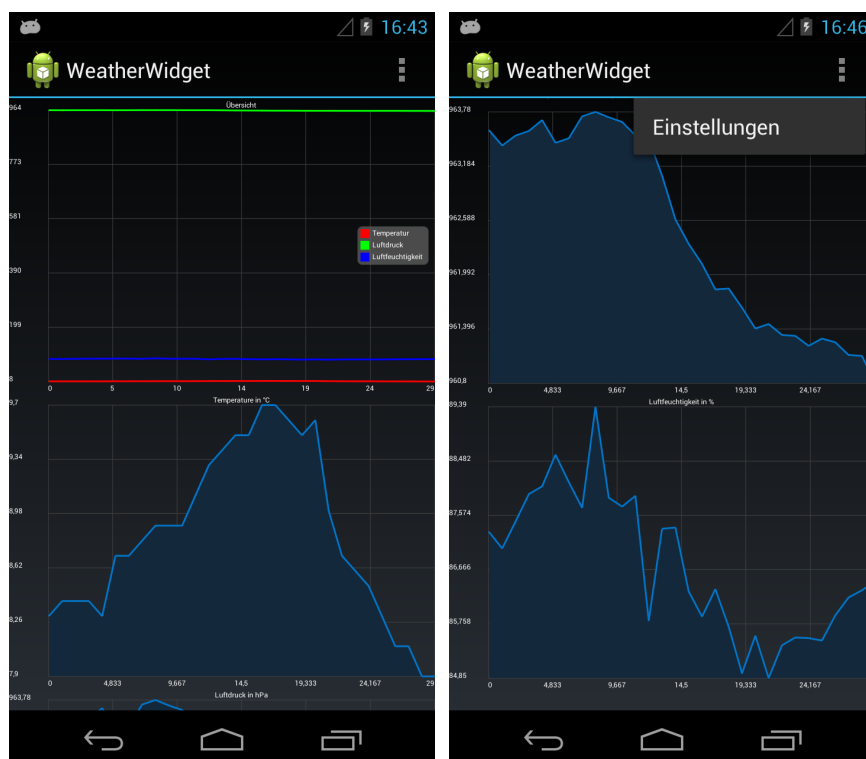
Beschreibung	URL
Arduino Hauptseite	<a href="http://arduino.cc/">http://arduino.cc/</a>
HIH-4030 Feuchtigkeitsensor	<a href="http://www.watterott.com/de/Feuchtigkeitssensor-HIH-4030-Breakout">http://www.watterott.com/de/Feuchtigkeitssensor-HIH-4030-Breakout</a>
BMP085 Druck- und Temperatursensor	<a href="http://www.watterott.com/de/Breakout-Board-mit-dem-BMP085-absoluten-Drucksensor">http://www.watterott.com/de/Breakout-Board-mit-dem-BMP085-absoluten-Drucksensor</a>
Android Hauptseite	<a href="http://www.android.com/">http://www.android.com/</a>
Arduino Mega	<a href="http://www.watterott.com/de/Arduino-Mega-2560">http://www.watterott.com/de/Arduino-Mega-2560</a>
Arduino Uno	<a href="http://www.watterott.com/de/Arduino-Uno-Rev-2-SMD">http://www.watterott.com/de/Arduino-Uno-Rev-2-SMD</a>
Arduni Ethernet-Shield	<a href="http://www.watterott.com/de/Arduino-Ethernet-Shield">http://www.watterott.com/de/Arduino-Ethernet-Shield</a>



(a) Widget Screenshot

(b) Einstellungen Teil 1 Screenshot

(c) Einstellungen Teil 2 Screenshot



(d) Graph Screenshot

(e) Graph Screenshot mit Einstellungen

Abbildung 10:  
Widget Screenshots



## 9 Glossar

Begriff	Erklärung
bmp085	Der bmp085 ist ein Temperatur- und Luftdrucksensor der Firma Bosch. Er wird im Zusammenhang mit der Wetterstation auf einem Breakout Board verwendet.
Arduino	Die Arduino-Plattform ist eine aus Soft- und Hardware bestehende Physical-Computing-Plattform. Beide Komponenten sind im Sinne von Open Source quelloffen. Die Hardware besteht aus einem einfachen I/O-Board mit einem Mikrocontroller und analogen und digitalen Ein- und Ausgängen.
hih4030	Der HIH-4030 ist ein Luftfeuchtigkeitssensor der im Zusammenhang mit der Wetterstation genau wie der bmp085 auf einem Breakout Board verwendet wird.
Breakout Board	Als Breakout Board oder auch Leiterplatte wird der Träger des Sensors bezeichnet. Es erleichtert die Entwicklung mit Sensoren, an denen die I/O-Pins sehr eng bei einander liegen. Durch das Breakout Board werden diese „handlicher“ zur Verfügung gestellt.
I/O	Mit Ein-/Ausgabe (abgekürzt E/A; engl. Input/Output; kurz I/O) bezeichnet man die Kommunikation / Interaktion eines Informationssystems mit seiner 'Außenwelt'.
Wetter-Sensor	Mit Wetter-Sensor ist ein Sensor zur Messung von klimatischen Verhältnissen (Wind, Luftfeuchtigkeit, Luftdruck, Temperatur, ...) gemeint
JSON	JavaScript Object Notation, ist ein weitverbreitetes lesbares Dateiformat im Textformat

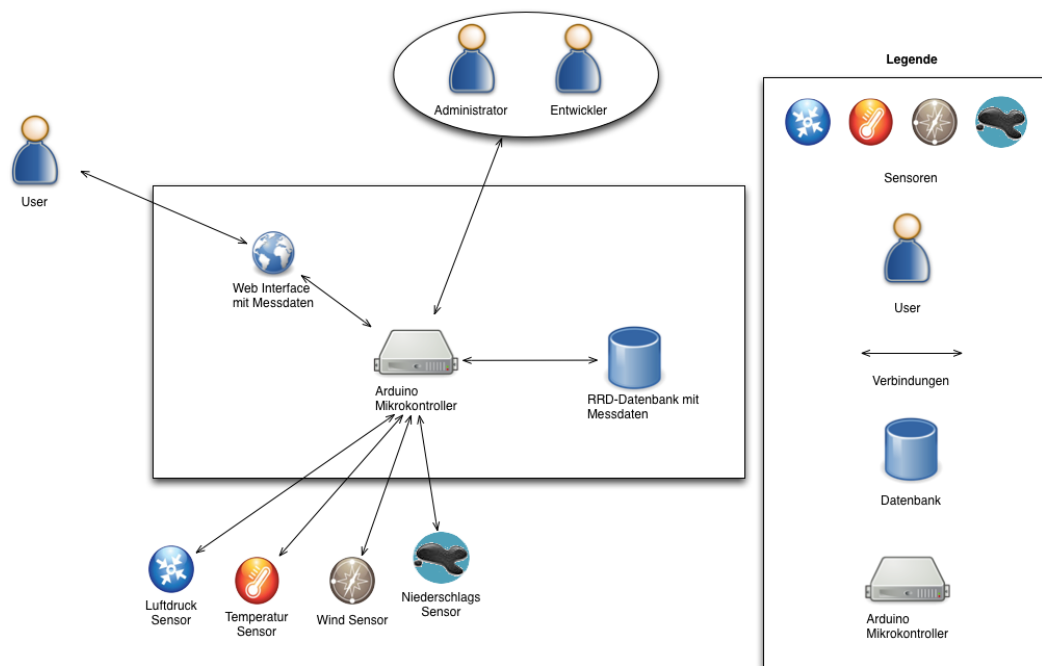
## A Installationsanleitung WeatherWidgetApp Version 1.0

Diese kurze Anleitung beschreibt die Installation der WeatherWidgetApp auf einem Android Telefon.

### 1. Voraussetzungen für die Installation/Benutzung der Anwendung:

- Android Betriebssystem Version 2.2 oder neuer
- Je nach Installationsart:
  - eine Internetverbindung auf dem Telefon
  - eine Dateimanager App auf dem Telefon
  - USB-Verbindung des Telefons zu einem Computer, sowie die Installieren-Android-Tools auf selbigem

### 2. Übersicht über das System



### 3. Installationsschritte

- über eine Dateimanager App:
  - (a) Kopieren der APK-Datei auf das Telefon
  - (b) Aufrufen der Datei über einen Dateimanager
- über eine USB-Verbindung zu einem Computer:
  - (a) Das Telefon mit einem Rechner per USB verbinden
  - (b) Am Telefon unter Einstellungen – > Apps – > Entwicklung, den Punkt „USB Debugging“ aktivieren
  - (c) Auf dem Rechner via „adb install <datei>“ die Installation starten

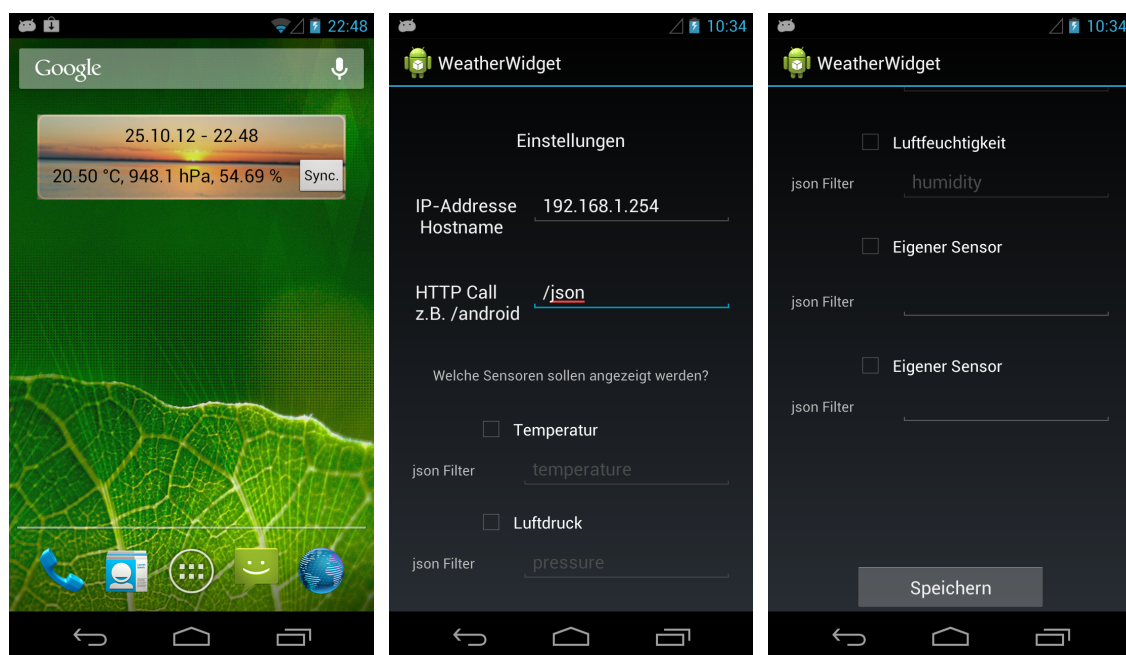
- über eine Internetverbindung auf dem Telefon:
  - (a) Wenn noch nicht getan, müssen Sie in den Einstellungen (Einstellungen – > Sicherheit) einen Haken bei „Unbekannte Herkunft“ setzen.
  - (b) Mit einem Browser auf dem Telefon die APK Datei direkt herunterladen - Die URL dazu finden Sie im Release Letter
  - (c) Der gewohnte Dialog zur Bestätigung der Android Anwendung erscheint auf dem Telefon mit dem Hinweis, welche Rechte die App benötigt.

#### 4. Starten der Anwendung

Die Anwendung wird, wie bei Android üblich, in der Widget Übersicht des Telefons durch ein Icon verlinkt. Dieses kann auf den Bildschirm gezogen werden (siehe Abbildung 11 (a)).

#### 5. Einstellungen des Widgets vornehmen

Im Einstellungsmenu (siehe Abbildung 11 (b) - (c)) des Widgets kann man folgende Einstellungen setzen:



(a) Widget Screenshot

(b) Einstellungen Teil 1 Screenshot  
hot

(c) Einstellungen Teil 2 Screenshot

Abbildung 11:  
Widget Screenshots

- IP-Adresse der Wetterstation (Arduino oder externer Server)
- HTTP-Call  
Für die Verwendung des Arduino als Server ist dies /json. Bei der Verwendung eines externen Servers ist es der HTTP-Pfad nach der IP-Adresse / dem Hostnamen zum perl cgi-script.

- Sensoren

Hier können die drei default Sensoren (Luftdruck, Luftfeuchtigkeit und Temperatur) und zwei zusätzliche Sensoren ausgewählt werden.

## B Installationsanleitung Arduino Programm Version 1.0

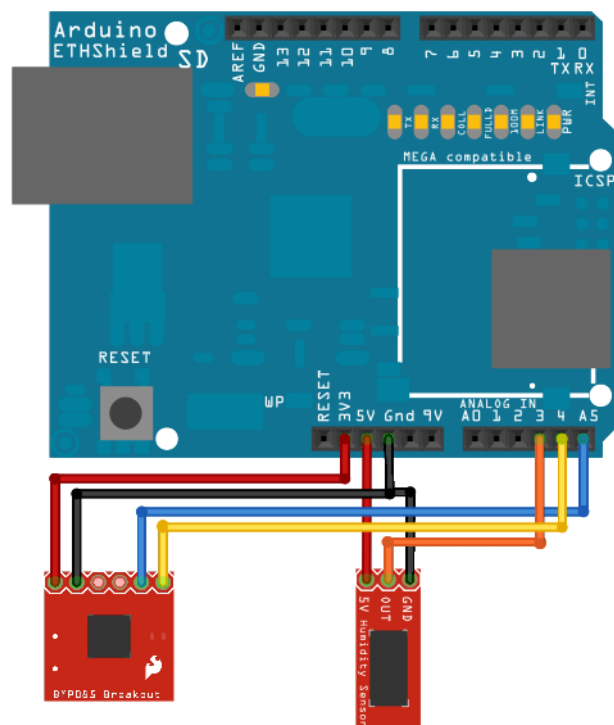
### 1. Voraussetzungen für die Installation/Benutzung der Anwendung

- Arduino Mega oder Arduino Uno (je nach Feature)
- Arduino Ethernet-Shield
- freier Ethernet-Anschluss um das Arduino Ethernet-Shield anzustecken
- Je nach gewünschtem Funktionsumfang:
  - Luftdruck- und Temperatursensor bmp085
  - Feuchtigkeitssensor hih4030
  - MicroSD-Card
- externer Webserver (optional)
  - Webserver um cgi-bin Skripte auszuführen (z.B. Apache2)
  - perl (getestet mit Version v5.10.1)
  - CPAN-Modul RRD::Simple
  - CPAN-Modul CGI
  - CPAN-Modul Log

### 2. Verkabelung des Systems

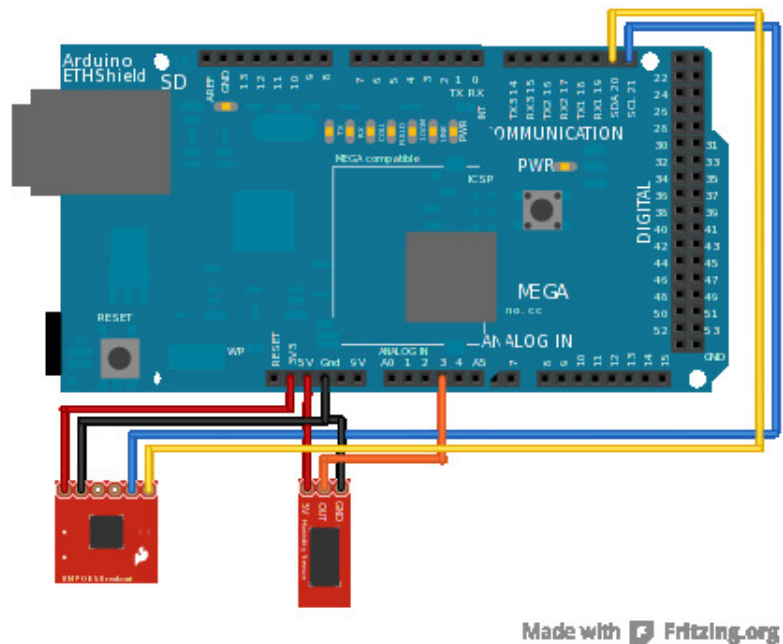
#### (a) Arduino Uno

Wenn die beiden Sensoren bmp085 und hih4030 verwendet werden, und der Arduino Uno benutzt wird, müssen diese wie folgt verkabelt werden:

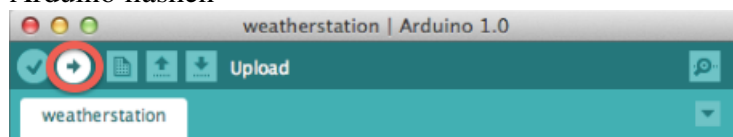


**(b) Arduino Mega**

Wenn die beiden Sensoren bmp085 und hih4030 verwendet werden, und der Arduino Mega benutzt wird, müssen diese wie folgt verkabelt werden:

**3. Installationsschritte** (flashen des Arduino Mikrocontrollers)

- (a) Entwicklungsumgebung des Arduino starten
- (b) Arduino Wetterstations Programm Version 1.0 öffnen
- (c) Arduino über USB am Computer anschließen
- (d) Arduino flashen

**4. Konfigurationsschritte** (Einstellungen am Arduino vornehmen)

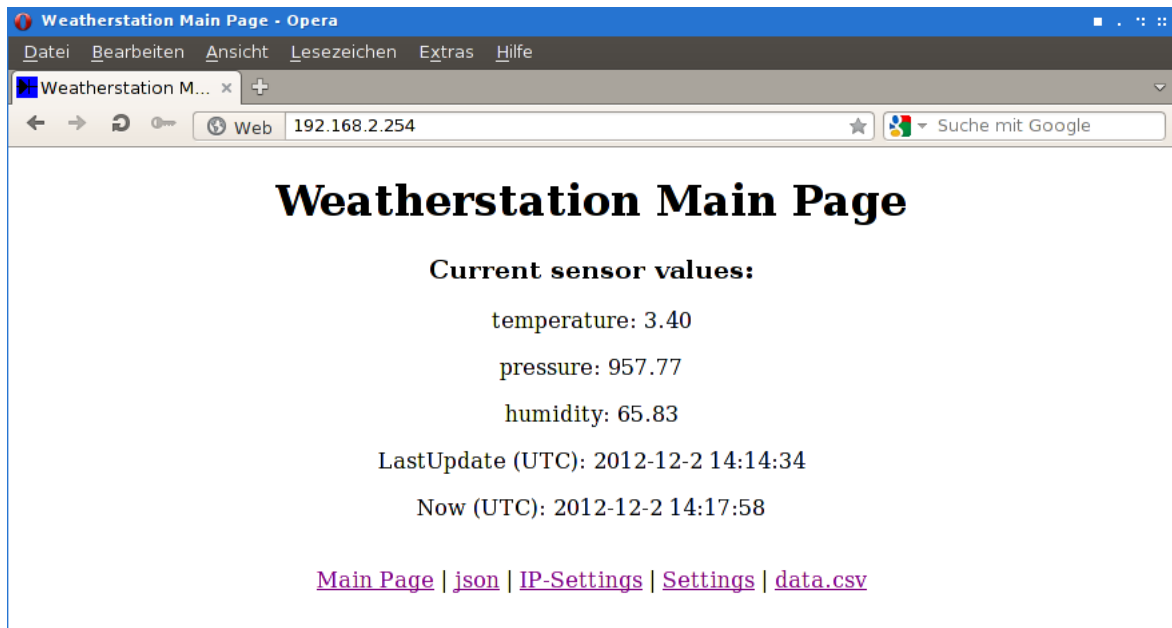
Nachdem der Arduino über das Ethernet-Shield am Netzwerk angeschlossen wurde, können Sie verschiedene Einstellungen am Arduino über eine Weboberfläche vornehmen.

Folgende Einstellungen können vorgenommen werden:

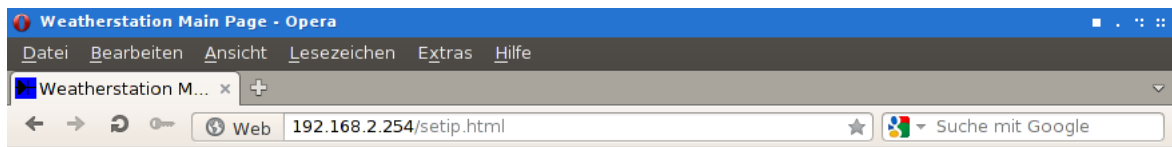
- Serielle Ausgabe des Arduino an-, bzw. ausschalten
- Die Kommunikation mit einem externen Server an-, bzw. ausschalten
- Die Eigenen IP-Adresse des Arduino einstellen.
- Gateway-Adresse für den Arduino einstellen.
- Adresse eines externen Servers eingeben.

- Der Pfad auf dem Server ist fest eingestellt auf /cgi-bin/arduino.pl

Um die Einstellungen vorzunehmen, muss der Arduino über das Ethernet-Shield mit dem Netzwerk verbunden sein. Nach dem ersten Start, ist der Arduino auf die IP 192.168.1.254 fest eingestellt. Um diese zu ändern, muss also temporär die IP des PCs in dieses Netz gelegt werden, falls ein anderes genutzt wird. Die Startseite des Controllers sieht wie folgt aus:



Die Einstellungen für das Netzwerk können über die URL `<ip>/setip.html`, oder durch den Link IP-Settings auf der Startseite, vorgenommen werden. Nachdem die Einstellungen geändert wurden, muss der Controller neu gestartet werden.



## Weatherstation SetIp Page

Please insert your IP values for the Arduino.  
These have to match your local network, or otherwise the Arduino controller is no longer reachable. If this happens, please re-flash the controller with this software. It will then use the default ip 192.168.1.254  
After changing them, you have to restart the controller

IP current: 192.168.2.254

IP Adresse: ...

GW current: 192.168.2.1

GW Adresse: ...

NTP current: 178.63.9.212

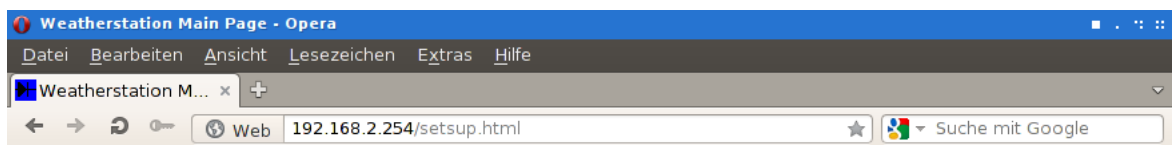
NTP Adresse: ...

Server current: 192.168.2.9

Server Adresse: ...

[Main Page](#) | [json](#) | [IP-Settings](#) | [Settings](#) | [data.csv](#)

Die verschiedenen Funktionen des Controllers können über die URL `<ip>/setup.html`, oder durch den Link Settings auf der Startseite, vorgenommen werden. Soll eine SD-Karte verwendet werden, ist es wichtig das diese vor der Aktivierung eingelegt wird. Nachdem die Einstellungen geändert wurden, muss der Controller neu gestartet werden.



## Weatherstation Settings

- ☒ enable sdcard
- ☐ enable server
- ☒ enable serial

[Main Page](#) | [json](#) | [IP-Settings](#) | [Settings](#) | [data.csv](#)



## C Installationsanleitung arduino.pl Version 1.0

Diese kurze Anleitung beschreibt die Installation der arduino.pl-Datei auf einem externen Unix-Server.

### 1. Voraussetzungen für die Installation:

- Server mit Anbindung an das Internet und fester IP-Adresse.
- Webserver (z.B. Apache2).
- Root-Rechte (Administrator-Rechte) auf dem Server.

### 2. Installationsschritte:

Die folgenden Installationsschritte beziehen sich auf einen Webserver mit debian6 als Betriebssystem.

- (a) arduino.pl in den cgi-bin-Ordner des Servers kopieren (unter debian6 ist das per default /usr/lib/cgi-bin/)  
z.B.:  
`$ cp arduino.pl /usr/lib/cgi-bin`
- (b) arduino.pl ausführbar machen und den Besitzer auf den Webserver-User (unter debian6 und apache2-Webserver ist das www-data)  
`$ chown a+x /usr/lib/cgi-bin/arduino.pl`

## D Release Letter

Willkommen zum Wetterstation Version 1.0 Release. Es handelt sich hier um ein Studenten-Projekt der FH-Rosenheim im Fach DAT. Dieses Projekt umfasst eine Wetterstation auf Basis eines Arduino Mikrocontrollers, eine Android Smartphoneanbindung sowie ein perl-cgi-bin-Skript um einen externen Server einzubinden.

### D.1 Inhalte

- Smartphone Applikation (weatherWidgetApp.apk) steht zum Download verfügbar.  
URL zum Herunterladen:  
<http://maxoberberger.net/weatherWidgetApp/weatherWidgetApp.apk>
- Arduino Mikrocontrollerprogramm ohne SD-Karten Unterstützung (weatherstationUno.ino) steht zum Download verfügbar.  
URL zum Herunterladen:  
<http://maxoberberger.net/weatherWidgetApp/weatherstationUno.tar.gz>
- Arduino Mikrocontrollerprogramm mit SD-Karten Unterstützung (weatherstationMega.ino) steht zum Download verfügbar.  
URL zum Herunterladen:  
<http://maxoberberger.net/weatherWidgetApp/weatherstationMega.tar.gz>
- Installationsanleitungen für die Smartphone Applikation und das Arduino Mikrocontrollerprogramm stehen zum Download verfügbar.  
URL zum Herunterladen (Android):  
<http://maxoberberger.net/weatherWidgetApp/InstallationsanleitungAndroid.pdf>  
  
URL zum Herunterladen (Arduino):  
<http://maxoberberger.net/weatherWidgetApp/InstallationsanleitungArduino.pdf>
- Perl cgi-bin Skript für die Verwendung der Arduino Wetterstation mit einem externen Server.  
URL zum Herunterladen:  
<http://maxoberberger.net/weatherWidgetApp/arduino.pl>
- Installationsanleitung für das Perl cgi-bin Skript.  
URL zum Herunterladen:  
<http://maxoberberger.net/weatherWidgetApp/InstallationsanleitungPerlCgi.pdf>

### D.2 Funktionalität Arduino Wetterstation

- Update der gemessenen Werte alle fünf Minuten

- Übertragung der gemessenen Werte zu einem externen Server via http-Post (optional).
- Bereitstellen der letzten gemessenen Werte über eine Weboberfläche
- Bereitstellung eines Einstellungsmenu um die Wetterstation den eigenen Wünschen anzupassen.
- Auf zwei Sensoren voreingestellt (bmp085, hih4030).

### **D.3 Funktionalität Android Smartphone Widget**

- Holt den zuletzt gemessenen Wert im 30 Minuten Rhythmus entweder von einem externen Server oder von der Wetterstation selbst
- graphische Anzeige einer Messhistory der letzten 30 Werte aller ausgewählten Sensoren
- Auswahl verschiedener Sensoren möglich.
- Anzeige der letzten aktuellen Werte im Widget direkt
- Um zwei weitere Sensoren erweiterbar

### **D.4 Zusatzinformationen**

- Internetverbindung zwingend notwendig für:
  - die Übertragung der Messdaten an einen externen Server (Wetterstation).
  - das Abholen der Messdaten von einem externen Server oder dem Mikrocontroller (Widget).

#### **Minimale Systemvoraussetzungen Android Widget**

- Android Betriebssystem Version 2.2 oder neuer
- Optimiert für Geräte mit 4" Displays

#### **Minimale Systemvoraussetzungen Arduino Wetterstation**

- Arduino Uno
- Ethernet Shield
- ein Sensor (idealerweise mit BreakoutBoard)

### **D.5 Ansprechpartner**

Alex Zenger <alexander.zenger@stud.fh-rosenheim.de>

Max Oberberger <maximilian.oberberger@stud.fh-rosenheim.de>