

Facharbeit

aus dem Fach

Physik

Thema:

Konzeption und Bau einer elektronischen Wetterstation

Verfasser: Sebastian Schuon
Leistungskurs: Physik
Kursleiter: Herr Miller
Abgabetermin: 02.02.04

Erzielte Note: in Worten:
Erzielte Punkte: in Worten:

Mündliche Prüfung zur Facharbeit:

Prüfungstag:
Erzielte Note: in Worten:
Erzielte Punkte: in Worten:

Abgabe im Sekretariat am:

.....

(Unterschrift des Kursleiters)

Inhaltsverzeichnis

1	Einleitung.....	3
2	One-Wire-Netzwerke.....	4
	2.1 Einführung.....	4
	2.2 Anwendung.....	5
3	Die Sensoren.....	6
	3.1 Feuchtesensor.....	6
	3.2 Windgeschwindigkeitssensor.....	7
	3.3 Windrichtungssensor.....	8
	3.4 Temperatursensor.....	9
	3.5 Helligkeitssensor.....	9
	3.6 Luftdrucksensor.....	10
	3.7 Niederschlagssensor.....	11
	3.8 weitere Sensortypen.....	12
	3.8.1 Grundwasser-Pegel-Sensor.....	12
	3.8.2 Dachflächenfenster-Sensor.....	12
	3.8.3 Feuermelder-Sensor.....	12
	3.8.4 Sichtweitensensor.....	13
4	Software.....	14
	4.1 Der Daemon.....	14
	4.2 Die Datenbank.....	15
	4.3 Die Benutzeroberfläche.....	15
	4.4 SMS-Service.....	16
5	Schlussgedanken und Danksagung.....	17
6	Literaturverzeichnis.....	18
7	Anhang.....	19
	7.1 Ausgewählte Messungen.....	19
	7.2 Die Datenbankstruktur.....	20
	7.2.1 Die Tabelle für die Sensoren.....	20
	7.2.2 Die Tabelle für die Messwerte.....	21
	7.2.3 Die Tabelle für Logs.....	21
	7.2.4 Die Tabelle für Trigger.....	22
	7.3 Die Triggersyntax.....	22
	7.4 Dateiliste der Softwarekomponenten.....	23
	7.4.1 Daemon.....	23
	7.4.2 Benutzeroberfläche.....	24
	7.5 Die Schaltpläne.....	25

1 Einleitung



Eine Wetterstation - eine an sich schon seit mehr als hundert Jahren bekannte Installation, aber trotzdem noch ein spannendes Themengebiet, wie ich finde.

Für mich stand bei diesem Projekt eher der technische Aspekt im Vordergrund als etwas wirklich Neues zu entdecken: ich suchte eine Möglichkeit die „1-Wire-Technologie“ zu erproben und die in unserem Haus vorhandenen Sensoren zu vernetzen. Deshalb sind vor allem die Messwerte der Wetterstation für mich wichtig, um sie mit den Daten der anderen Sensoren zu kombinieren. Beispielsweise machen die Sensoren an

den Dachflächenfenstern deutlich mehr Sinn, wenn man sie mit dem Regensensor kombiniert. Dann kann eine Warnmeldung ausgegeben werden, falls es regnet und die Fenster geöffnet sind.

Die Wetterstation lässt sich aber auch in der Schule verwenden. Dort hat die Wetterstation allerdings den Zweck, Umweltdaten u.a. für den Erdkunde-Unterricht (z.B. Klimadiagramm) oder für den Einsatz im Fach „Regenerative Energien“ (hier zur Auswertung der Energiebilanzen) zu erfassen.

Meine Ausführungen gliedern sich in drei große Abschnitte: Anfangs möchte ich die zu Grunde liegende Vernetzungstechnik „1-Wire“ erläutern. Dann werden die aktuell integrierten Sensoren beschrieben und ein Ausblick auf weitere, mögliche Sensoren gegeben. Im dritten Teil werden die Software-Komponenten erläutert, welche für Datenerfassung, Auswertung und Darstellung verantwortlich sind. Zusätzlich befindet sich im Anhang weitere technische Dokumentation.

2 One-Wire-Netzwerke

2.1 Einführung^{[2],[11]}

Die 1-Wire-Technologie wurde von der Firma *Dallas Semiconductor / Maxim (Da/Semi/Maxim)* entwickelt. Es handelt sich hierbei um eine Netzwerktechnik, welche nur eine „Twisted-Pair“-Leitung zur Kommunikation benötigt. Das System basiert auf einer Master/Slave-Architektur und besteht grundsätzlich aus drei Bestandteilen: 1) dem Busmaster 2) der Verdrahtung 3) den 1-Wire-Geräten (d.h. Slaves).

Das Netzwerk nutzt eine der beiden Leitungen zur Stromversorgung und Kommunikation, die andere als Masse. Erstere wird durch einen Pull-Up-Widerstand auf 5V (dies entspricht dem High-Pegel) gezogen. Um nun eine logische Eins zu schreiben, wird der Bus für weniger als 15µs auf den Low-Pegel gezogen (\approx Masse), für eine logische Null wird der Bus für mindestens 60µs auf den Low-Pegel gezogen. Zur Kommunikation führt der Master einen Reset des Busses durch. Dies bedeutet, er zieht den Bus für mehr als 480µs auf Low. Danach wartet er auf Antworten von den Slaves. Wenn er nun mit einem speziellen Slave kommunizieren will, ruft er dessen Adresse auf und startet einen „Handshake“. Falls dieser erfolgreich verläuft, sendet der Master nun die baustein-spezifischen Befehle und führt den nötigen Datenaustausch aus.

Damit es nicht zu Verwechslungen auf dem Bus kommt, trägt jeder Baustein eine eindeutige 64-Bit-Seriennummer. Diese wurde bereits vom Werk her mit einem Laser eingebrannt.

Eine besondere Spezialität des 1-Wire-Netzwerkes ist die "parasitäre" Stromversorgung. Damit ist es möglich, den Slave-Baustein über den Bus mit zu versorgen. Dazu sorgen eine Diode und ein Kondensator dafür, dass die Stromversorgung überbrückt werden kann, falls der Bus gerade für Kommunikation verwendet wird. Hier wird der Kondensator geladen, wenn gerade keine Kommunikation stattfindet und der Bus sich auf High befindet. Wenn der Bus jetzt auf Low fällt (d.h. Daten übertragen werden), sorgt die Diode dafür, dass sich der Kondensator nicht in Richtung des Busses entlädt, sondern nur in Richtung des Slaves entladen kann.

2.2 Anwendung

Da das Kommunikationsprotokoll des 1-Wire-Netzes relativ komplex ist, wird hier ein Leitungstreiber-Baustein eingesetzt. *DalSemi/Maxim* bietet dazu verschiedene Bausteine an, die jeweils auf der Nicht-1-Wire-Seite unterschiedliche Schnittstellen bedienen.

Für Test-Zwecke habe ich einen vorhandenen DS2490^[9] genutzt, welcher eine USB-1.1-Schnittstelle bietet. Zum Entwickeln bot diese Schnittstelle einige Vorteile (bereits vorhandene Stromversorgung durch den USB-Bus, Hot-Plug-Fähigkeit), für den späteren Dauereinsatz kommt allerdings der DS2480B^[8] zum Zuge, da die hier unterstützte serielle Schnittstelle stabiler und Ressourcen schonender ist. So kann als Messrechner auch ein älterer PC benutzt werden. Allerdings muss hier noch eine Konvertierung des Pegelsignals am UART auf Computer-Niveau (5V auf 12V) erfolgen, dies lässt sich allerdings einfach mit einem MAX232-Derivat^[10] erledigen.

DalSemi/Maxim bietet eine Reihe von Messbausteinen an, welche bereits ein 1-Wire-Interface integriert haben und weitere Aufgaben erledigen können. Die von mir im Rahmen dieses Projektes verwendeten Typen, sollen kurz vorgestellt werden:

DS2450^[7]

Ein Vierfach-Analog/Digital-Wandler mit 16 Bit Auflösung. Messungen können nur zwischen Masse und einer Eingangsspannung durchgeführt werden. Der Messbereich liegt zwischen 0 V und maximal 5,12 V.

DS2423^[5]

Ein 4-kBit Speicher mit zwei integrierten Countern, welche extern getriggert werden können.

DS1820-Familie^[3]

Eine Familie von Temperatur-Sensoren, welche eine maximale Abweichung von 0,5° bis 2° bieten. Die Auflösung beträgt typischer Weise 0,1°C. Alle Sensoren sind für parasitäre Stromversorgung vorbereitet. Bei einigen Modellen ist diese schon fest verdrahtet.

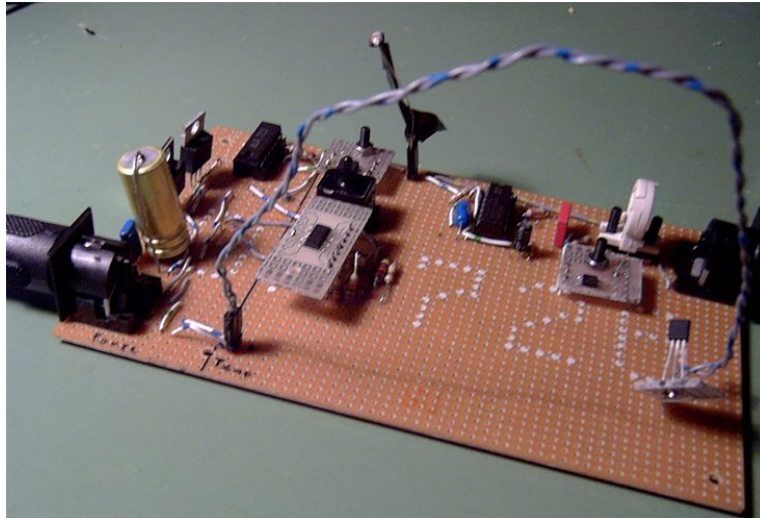
DS2438^[6]

Eigentlich ein Chip zur Überwachung von Akkus, aber er besitzt einen A/D-Wandler, mit welchem masse-frei gemessen werden kann.

DS2408^[4]

Ein Baustein mit einem kleinen eingebauten Speicher und mit acht I/O-Ports

3 Die Sensoren



Die Platine mit den aufgebauten Sensoren

3.1 Feuchtesensor

Mittels eines Kondensators kann die relative Luftfeuchte der Luft bestimmt werden. Da sich je nach Feuchte die Dielektrizitätskonstante ändert, kann man nun eine Beziehung zwischen Kapazität und der Feuchte mit Hilfe der Dielektrizitätskonstante herstellen.

In der Praxis wird ein Feuchtesensor („Philips H1“^[16]) von *Philips* verwendet. Dieser stellt im Prinzip einen Plattenkondensator dar. Als Dielektrikum wird eine Folie, welche auf beiden Seiten eine Goldbeschichtung trägt, verwendet. Hiermit kann der Sensor eine Temperaturkompensation leisten, die sonst extern nötig geworden wäre.

Um allerdings die Kapazität des Kondensators in eine digitalisierbare Größe zu verwandeln ist ein gewisser technischer Aufwand nötig. Die entsprechende Schaltung wurde der Dokumentation eines anderen Feuchtesensors^[15] entnommen. Hier behilft man sich mit zwei Timer-Bausteinen vom Typ 555^[19]. Dies ist ein Baustein, welche einen Takt generieren kann, dessen Frequenz von einem RC-Glied abhängig ist. In der Schaltung wird ein IC vom Typ 556^[19] verwendet, da hier in einem IC zwei 555-Timer integriert sind.

Ein erster Timer (Timer1) generiert einen Arbeitstakt mit fester Frequenz (d.h. mit festen Werten für das RC-Glied). Ein zweiter Timer (Timer2) wird durch das Taktsignal von Timer1 getriggert, Timer2 befindet sich damit in dem Betriebs Monoflop. Das RC-Glied von Timer2 besteht aus einem Widerstand mit bekanntem Wert und dem Feuchtesensor als Kondensator. Die Änderung der Kapazität des Feuchtesensors

bewirkt damit eine Modulation des Tastverhältnisses. Dieses Signal wird nun noch über einen D/A-Wandler (im Prinzip ein RC-Glied) geleitet, um eine zur Kapazität proportionale Spannung für den A/D-Wandler zu erhalten.

Ein einfacher Funktionstest kann mittels Anhauchen erfolgen. Eine schon etwas genauere Kalibrierung ergibt sich durch die Referenzwerte, welche im Datenblatt angegeben sind. Dazu misst man die Spannung des High-Zustandes und muss die Werte für Kapazität und Widerstand des RC-Gliedes von Timer2 wissen. Daraus lässt sich mittels der in den Application Notes des 555 aufgeführten Formeln die Kapazität des Sensors errechnen.

Noch genauer lässt sich die Schaltung mittels eines Feuchtemessgerätes kalibrieren.

3.2 Windgeschwindigkeitssensor



Um die Windgeschwindigkeit zu erfassen, benutze ich ein Kreuzschalenanemometer. Dieses besteht idealerweise aus vier Halbschalen, welche an den Enden eines Kreuzes angebracht sind. Am Mittelpunkt des Kreuzes ist eine Achse angebracht, mit welcher der Sensor mittels einem möglichst reibungsfreien Kugellager befestigt

werden kann. Der Vorteil der Anordnung ist, dass der Sensor unabhängig von der Anströmrichtung des Windes ist. Das Funktionsprinzip kann auch auf drei armige Anemometer übertragen werden, allerdings ist das Anlaufverhalten dann nicht mehr so gut.

Der bei meiner Installation verwendete Sensor wurde aus einer bestehenden Installation übernommen. Er entstammt dem Kosmos-Baukasten "Kachelmanns Wetterstation". Die Zahl der Umdrehungen wird mittels eines Reed-Kontaktes gemessen. Dies ist zwar eine relativ gute, aber nicht perfekte Methode, da durch den an der Spule vorbei bewegten Magneten auf diesen (hemmende) Kräfte ausgeübt werden. Ideal wäre eine Lichtschranke, die mittels einer an der Achse angebrachten Scheibe mit Schlitz geöffnet wird. Ich habe auf diese Methode verzichtet, da die oben erwähnte Ungenauigkeit durch die Reedkontakte vernachlässigbar klein ist .

Um die Impluse vom Anemometer auszuwerten, verwendet man einen Frequenzzähler. Für die Implementierung eines Frequenzzählers gibt es zwei mögliche Varianten: eine einfachere und eine kompliziertere. Ich habe mich für die einfachere

Variante entschieden, da meiner Meinung nach die oben erwähnte Ungenauigkeit der Messwerte vertretbar ist. Der Vollständigkeit halber soll aber auch die andere Variante erläutert werden.

Die einfache Variante nutzt einen Zähler, in diesem Fall ein DS2423^[6]. Dieser wird geöffnet und sein Zustand ausgelesen. Nun lässt man diesen Zähler für eine gewissen Zeit offen (Torzeit). Danach wird der Counter wieder geschlossen und sein Zustand ausgelesen. Aus der Differenz geteilt durch die Torzeit, erhält man die Frequenz. Zu Ungenauigkeit kommt es, wenn man das Tor kurz vor einem neu eintreffenden Impuls schließt. Die Folge ist, dass die errechnete Frequenz zu niedrig ist. Diese Ungenauigkeit nimmt mit zunehmender Torzeit bzw. höherer Frequenz ab.

Der kompliziertere Ansatz benötigt zusätzlich einen Taktgenerator. Ein eingehender Impuls öffnet den Counter, an welchem der Taktgenerator anliegt. Dadurch zählt der Zähler nun solange hoch, bis durch einen weiteren Impuls der Counter wieder geschlossen wird. Da die Anzahl der Impulse des Taktgenerators bekannt sind, kann man mit Hilfe des Zählerstandes die Frequenz berechnen. Es ist zu beachten das die verwendete Taktfrequenz deutlich höher ist als die Frequenz, welche man messen möchte, sonst entstehen ähnliche Probleme wie bei der ersten Variante.

Ein Funktionstest des Sensors kann mit einem Föhn oder einem PC-Lüfter erfolgen. Die nötigen Angaben für die Umrechnung von Frequenz in Windgeschwindigkeit wurden mittels eines elektrischen Fahrradtachometers ermittelt, welcher dem Kosmos-Baukasten beilag, dem auch das Anemometer entnommen wurde. In diesem Baukasten wird der Fahrradtachometer zur Anzeige der Windgeschwindigkeit benutzt. Eine Überprüfung dieser Werte erfolgte mit einem Auto. Dazu wird der Sensor bei windstiller Umgebung aus dem Fenster des Autos gehalten. Am Tachometer kann man nun die Geschwindigkeit ablesen. Idealerweise nutzt man zur Geschwindigkeitsbestimmung einen GPS-Empfänger, da dieser genauer ist als der Tacho. Der Autotacho ist bauartbedingt etwas ungenau, da nur die Umdrehungszahl pro Sekunde mit einem festen Wert für den Umfang des Reifens multipliziert wird. Allerdings erfährt ein Reifen während seines Einsatzes naturgemäß Abnutzung und sein Umfang wird geringer.

3.3 Windrichtungssensor

Das Prinzip eines Windrichtungssensors ist relativ einfach: eine Windfahne dreht sich in den Wind und somit bestimmt man die Windrichtung.

Um die Richtung der Achse der Windfahne in elektrische Signal umzuwandeln, wird ein Endlos-Potentiometer verwendet. Somit reicht es, den Widerstand des Potentiometers zu messen, denn dieser ist direkt-proportional zur Windrichtung. Um die Temperaturunabhängigkeit zu verbessern, wird als zweiter Widerstand der Gesamt-widerstand des Potentiometers verwendet. Zur Messung des Widerstands dient die 5V-Spannungsquelle und ein Kanal des A/D-Wandlers.

Eine Kalibrierung kann sehr einfach mit einem Geodreieck und einem Kompass erfolgen.

3.4 Temperatursensor

Zur Messung der Temperatur wird der Einfachheit halber der fertige Baustein DS1820 ^[3] verwendet. Dieser kann als BlackBox betrachtet werden, zumal er alle relevanten Daten direkt auf dem 1-Wire-Bus ausgibt.

Die Funktionsweise lässt sich leicht mit einem handelsüblichen Thermometer überprüfen. Eine Feinkalibrierung kann später in der Software erfolgen.

3.5 Helligkeitssensor

Zur Bestimmung der Helligkeit wird ein Fotowiderstand verwendet. Ein solcher Fotowiderstand ist ein Halbleiterbauelement, welches seinen Widerstandswert beleuchtungsabhängig ändert. Intern bestehen diese Bauelemente aus einem Plättchen, auf das z.B. Cadmium-Sulfid aufgedampft wird, welches lichtempfindlich ist und je nach Beleuchtung unterschiedlich leitfähig ist. Um die Helligkeit zu bestimmen, muss man nur der Widerstand des Sensor messen.

Ich habe einen Sensor der Firma *Heimann* verwendet, Typ A9060^[12]. Hier ist die lichtempfindliche Schicht bereits in einem Gehäuse verpackt, allerdings habe ich ihn noch zum Schutz vor Wind und Wetter unter ein Marmeladeglas gestellt.

Die Umrechnung der abfallenden Spannung bzw. des Widerstand des Sensors in Lux lässt sich mit Hilfe der im Datenblatt angegebenen Kennlinie bewerkstelligen. Eine Kalibrierung kann mit einem Lux-Meter erfolgen. Allerdings ist dies relativ schwierig, da die Kennlinie nur scheinbar linear ist, da die Skalen jeweils logarithmisch sind.

3.6 Luftdrucksensor

Um den Luftdruck zu messen, bedient man sich am besten eines fertigen Sensors, da die technischen Anforderungen sehr hoch sind, um einen ausreichend präzisen Sensor herzustellen.

Bei den von mir verwendeten Sensoren handelt es sich um SDX15A2^[18] von Sensortronics, einer Firma in Puchheim. Dies ist ein fertig kalibrierter Sensor, der elektrisch gesehen eine Widerstandsbrücke darstellt. Intern besteht der Sensor aus einem Plättchen, auf welches eine leitende Substanz aufgebracht ist. Je nach Druck biegt sich das Plättchen und der Widerstand ändert sich.

Leider lässt sich der Sensor nicht direkt an den A/D-Wandler anschließen. Dafür sind zwei Sachverhalte verantwortlich: zum einen kann der 16-Bit A/D-Wandler nicht massfrei messen. Dies würde dazu führen, dass man nur eine Seite der Brücke messen kann und die werksseitige Kalibrierung damit nutzlos geworden wäre. Zum anderen beträgt der Maximalausschlag des Sensors 90mV bei 12V Betriebsspannung und für die bei uns anzutreffenden Luftdruckwerte bewegen sich in einem 10%-breiten Intervall des Messbereichs des Sensors (Messbereich des Sensors: 0-1034 hPa, erwartete Messwerte liegen im Bereich von 950-1050 hPa -> Messintervall ca. 9mV). Die obere Grenze des Messbereichs des Sensors liegt in diesem Fall sogar unter dem erwarteten maximalen Luftdruck, aber man muss wissen, dass der Sensor durchaus höheren Luftdruck zu vermessen vermag. Es besteht erst Gefahr von Schäden am Sensor ab ca. 2000 hPa. Um nun besser messen zu können und auch der ersten Tatsache Rechnung zu tragen, kommt ein Vorverstärker zum Einsatz. Hier habe ich die von Sensortronics vorgeschlagene Schaltung^[17] verwendet, allerdings einen anderen Operationsverstärker verwendet (MAX4169^[14]) als vorgeschlagen, da dieser leichter zu beschaffen war.

Grob lässt sich der Sensor kalibrieren, wenn man die Spannung am Sensor misst und gleichzeitig die Spannung am Vorverstärker. Nun kann man aus den im Datenblatt angegebenen Werten den herrschenden Luftdruck berechnen. Um allerdings eine genauere Kalibrierung zu erreichen, kann man die im Internet angebotenen Daten des *Deutschen Wetterdienstes* nutzen (z.B. bei *wetter.com*). Daraus lässt sich innerhalb der Software noch einen Korrekturfaktor bestimmen.

Man muss beachten, dass die gemessenen Werte den momentanen Druck p in der momentanen Höhe h repräsentiert. Falls man nun seine Messwerte mit den Werten aus der Wetterkarte vergleichen will, muss man noch eine Umrechnung erfolgen, da

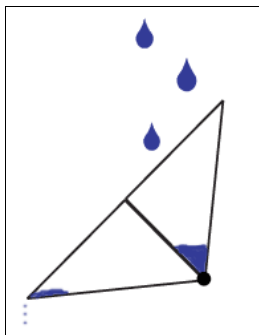
diese Werte auf „Normal Null“ (NN) und 15°C bezogen gemessen werden. Dafür verwendet man die barometrische Höhenformel, wie sie z.B. in der Brockhaus Enzyklopädie gefunden werden kann:

$$h_1 - h_0 \approx R \cdot T \cdot \ln\left(\frac{p}{p_0}\right) \quad \text{oder} \quad p_0 \approx p \cdot e^{\frac{h_1}{R \cdot T}} \quad \text{mit } h_0 \approx 0 \text{ m}$$

R ist dabei eine aus der allg. Gaskonstanten abgeleitete Größe mit $R = 29,27 \text{ m/K}$. Für die Temperatur T in Kelvin nimmt man den Mittelwert zwischen der gemessenen Temperatur und dem Bezugswert von 15°C.

3.7 Niederschlagssensor

Die meisten Niederschlagssensoren beruhen auf dem Prinzip einer Schale, in welche Regenwasser tropft und die entleert wird, sobald sie voll ist. Dadurch wird ein Impuls ausgelöst, welcher gezählt wird. Damit kann man die Regenmenge bestimmen, da das Volumen der Schale bekannt ist.



Mein Sensor ist eine Wippe mit je einer Schale auf einer Seite^[1]. Dies hat den Vorteil, dass wenn die eine Seite gefüllt ist, und sich damit zur Seite neigt um sich zu entleeren, die andere Seite automatisch angehoben wird und der Regen nun in diese tropfen kann.

Die Umkippvorgänge werden mit Hilfe eines Reed-Kontaktes erfasst und von einem Zähler des Typs DS2423^[5] registriert.

Dieser Zähler kann nun von der Software über das 1-Wire-Netz ausgelesen werden.

Am besten testet man die Apparatur erst im Waschbecken, um die Auswirkungen verschieden starker Regengüsse zu ermitteln. Bei dieser Gelegenheit sollte man auch gleich das Fassungsvermögen der Schalen bestimmen.

Ein Problem ergibt sich eigentlich bei allen Niederschlagssensoren: Wenn die Temperatur unter 0°C sinkt, beginnt das Wasser in den Schalen zu gefrieren und damit die Apparatur unbrauchbar zu machen. Entweder man schaltet die Erfassung dann per Software ab, welche ja auch die Temperatur erfasst und damit entscheiden kann, ab wann keine zuverlässigen Messungen mehr möglich sind. Alternativ könnte man auch die Schalen beheizen, was sich aber als relativ kompliziert erweist, da man Kabel für die Heizschleifen zu den Schalen legen müsste, welche wiederum die Beweglichkeit der Schalen behindern und somit evtl. zu einer ungenaueren Messung beitragen bzw. ein Umkippen komplett verhindern.

3.8 weitere Sensortypen

Diese Sensoren befinden sich bei mir gerade in der Entwicklung, deshalb will ich sie hier nur grob umreißen.

3.8.1 Grundwasser-Pegel-Sensor

Nach den unzähligen Überschwemmungen in Deutschland habe ich mich gefragt, wie man wohl eine Art Frühwarn-System bauen könnte. Da unser Garten über einen Brunnenschacht verfügt, habe ich beschlossen, hier den Pegel des Grundwasser zu messen. Wenn dieser zu hoch steigt, sollte man sich überlegen, ob man Pumpen für den Keller kauft.

Die Messeinrichtung besteht aus einem Differenzialdrucksensor, an dessen einen Eingang nichts angeschlossen wird, also der momentane Luftdruck herrscht. Am anderen Eingang schließt man einen Schlauch an, welcher in den Brunnen hinab hängt. Nun muss man eine Wassersäule in diesem Schlauch ansaugen, so dass sich nur eine kleine Luftblase zwischen Wassersäule und Differenzialdrucksensor bildet. Dies ist nötig, da der Sensor nicht für Flüssigkeiten gedacht ist. Je nach Pegelstand, drückt das Wasser auf die Luftblase oder dehnt sie aus. Mittels des Sensors kann man nun den Druck in der Blase messen und damit dann die Pegelhöhe berechnen.

3.8.2 Dachflächenfenster-Sensor

Unsere Dachfenster haben wir mit Sensoren ausgestattet, mit deren Hilfe bestimmt werden kann, ob ein Fenster geschlossen oder geöffnet ist. Als Sensoren werden Reed-Kontakte verwendet. Deren Signal wird mit Hilfe des achtfach I/O-Bausteins DS2408^[4] digitalisiert und ins 1-Wire-Netz eingebunden.

Die Software prüft sobald sie am Niederschlagssensor Regen registriert, ob alle Fenster geschlossen sind. Falls nicht, wird der Benutzer mittels einer SMS informiert (siehe dazu 4.4)

3.8.3 Feuermelder-Sensor

Die in unserem Haus installierten Feuermelder werden mit einem Anschluss an das 1-Wire-Netz nachgerüstet. Die Feuermelder funktionieren nach dem optischen Prinzip, d.h. wenn in einer Lichtschranke der Strahl durch Rauchpartikel teilweise abgelenkt wird, löst der Feuermelder aus. An der Lichtschranke wird das Alarmsignal abgegriffen und in digitalisierter Form in das 1-Wire-Netz eingespeist. Zusätzlich kann der Feuermelder durch das 1-Wire-Netz auf Alarm geschaltet werden. Dies ist nützlich,

dass, falls einer der Feuermelder Alarm auslöst, alle Feuermelder ein akustisches Signal abgeben. Im Fall eines Alarms wird der Benutzer mittels einer SMS informiert. Es wäre auch vorstellbar, direkt eine Meldung an die Feuerwehr weiter zu leiten. Dies ist allerdings kritisch auf Grund der Möglichkeit von Fehlalarmen.

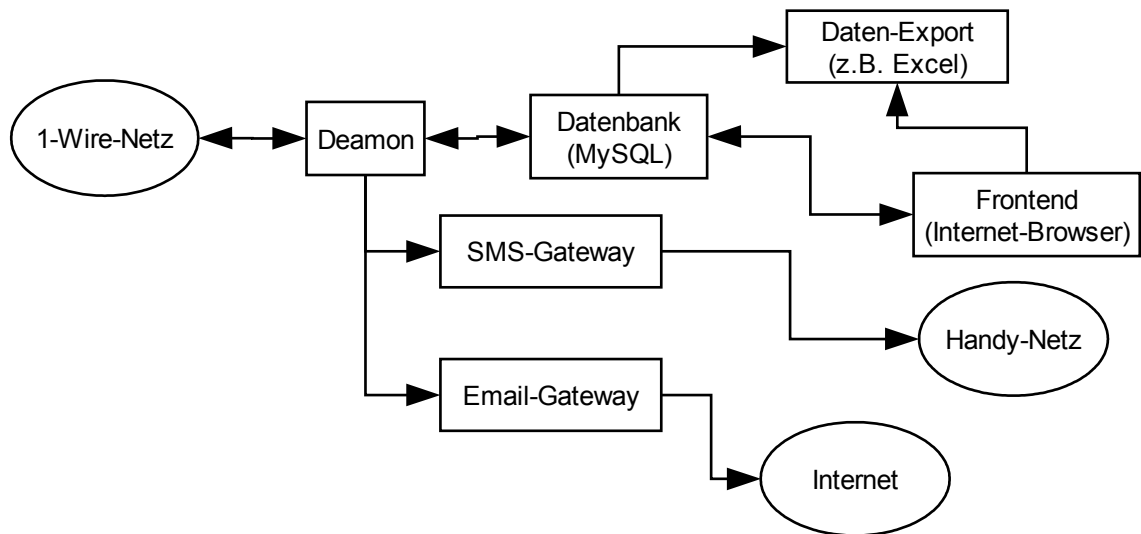
3.8.4 Sichtweitensensor

Inspiziert von den oft neben der Autobahn anzutreffenden Sichtweitensensoren habe ich eine ähnliche Apparatur konstruiert, da mein Wohnort in einer oft nebligen Gegend liegt.

Dazu wird von einer Infrarot-Sendediode ein Lichtstrahl ausgesandt. In ca. 2m Entfernung ist eine IR-Empfangsdiode positioniert, an der eine Spannung proportional zur Empfangsstärke des Signals gemessen werden kann. Diese wird mit einem A/D-Wandler vom Typ DS2450 digitalisiert.

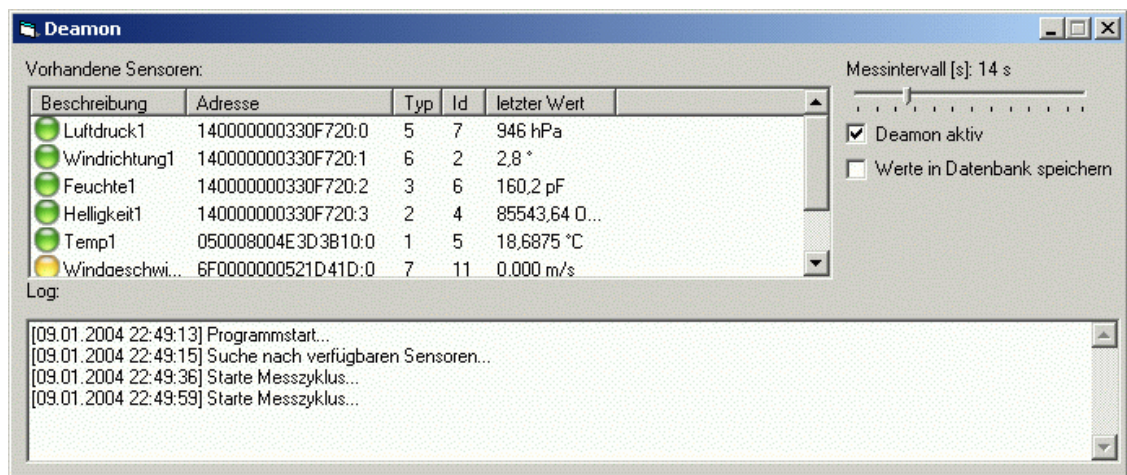
Als Messwert erhält man dann nur eine prozentuale Sichtverschlechterung gegenüber dem Idealzustand. Eine Kalibrierung kann nur in sofern erfolgen, dass man den Idealzustand bei gutem Wetter misst und diesen als Maximalwert (100%) setzt. Nun verdeckt man den Empfänger komplett und erhält so einen Minimalwert (0%).

4 Software



Aufbau und Zusammenspiel der Software-Komponenten

4.1 Der Deamon



Der Begriff „Daemon“ stammt aus der Unix-Welt und repräsentiert dort ein Programm, welches im Hintergrund läuft und eine bestimmte Aufgabe erfüllt. Genau so arbeitet das Datenerfassungsprogramm. Es lässt sich auswählen, welche Sensoren abgefragt werden sollen und ob deren Daten in die Datenbank übernommen werden sollen. Auch lässt sich das Messintervall einstellen.

In einem Textfenster protokolliert das Programm seine Tätigkeit und gibt gegebenenfalls Warnungen und Fehlermeldungen aus. Alle Meldungen werden zusätzlich in der Datenbank protokolliert, damit sie über die Benutzeroberfläche (siehe 4.3) gelesen werden können. Desweiteren wird bei jedem Messzyklus überprüft, ob ein Trigger (siehe dazu 7.3) wahr geworden ist und evtl. die angegebene Reaktion ausgeführt.

Der Daemon ist in Visual Basic implementiert, da diese Sprache eine schnelle Entwicklung der Software gestattet. Allerdings ist der Deamon damit an ein Win32-kompatibles Betriebssystem gebunden.

Zur Kommunikation mit dem 1-Wire-Netz wird das von *Da/Semi/Maxim* zur Verfügung gestellte Software Developer Kit verwendet, insbesondere hier das OWAPI.

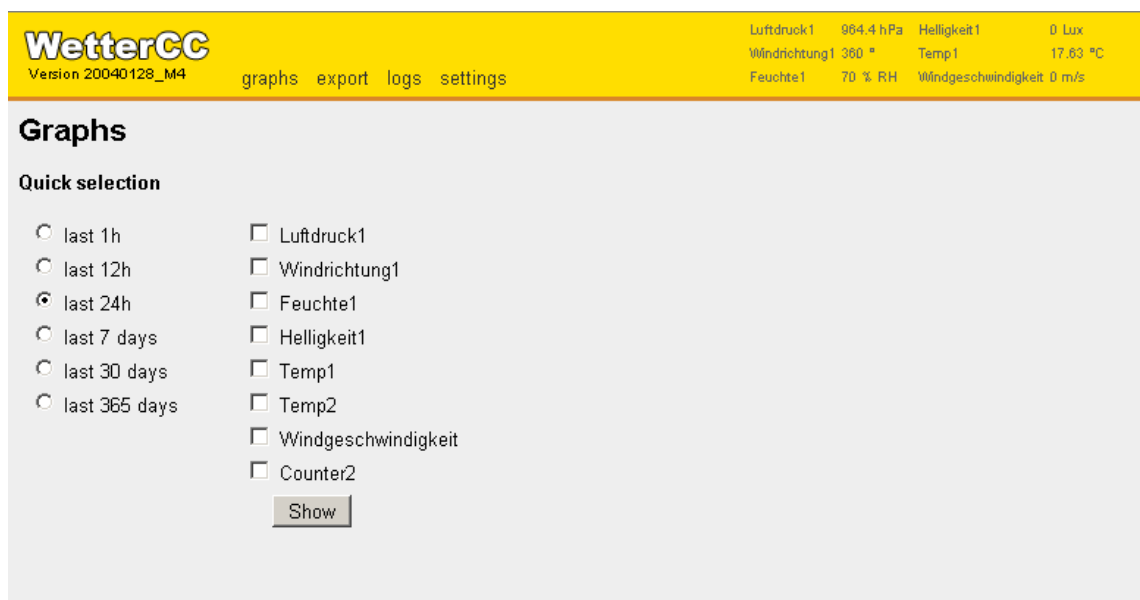
4.2 Die Datenbank

Zur Speicherung der Daten wird eine SQL¹-kompatible Datenbank verwendet. Meine Wahl ist auf die MySQL-Datenbank gefallen. Sie ist unter einer OpenSource-Lizenz verfügbar, hat sich bei mir in früheren Projekten gut bewährt und es existiert eine gute Anbindungsmöglichkeiten an verschiedenste Programmiersprachen. Des weiteren kann die Datenbank mittels einer TCP-Verbindung einfach über ein Netzwerk angesprochen werden.

Ein weiterer Vorteil einer SQL-Datenbank ist, dass später auch andere Entwickler auf die darin gespeicherten Daten einfach und standardisiert zugreifen können.

Der Aufbau der Datenbank ist im Anhang unter 7.2 näher erläutert.

4.3 Die Benutzeroberfläche



The screenshot shows the WetterCC web interface. At the top, there is a yellow header with the logo 'WetterCC' and 'Version 20040128_M4'. To the right of the header, there are several data points: Luftdruck1: 964.4 hPa, Helligkeit1: 0 Lux, Windrichtung1: 360 °, Temp1: 17.63 °C, Feuchte1: 70 % RH, and Windgeschwindigkeit: 0 m/s. Below the header, there are navigation links: 'graphs', 'export', 'logs', and 'settings'. The main content area is titled 'Graphs' and contains a 'Quick selection' section. This section has two columns of radio buttons and checkboxes. The first column contains time period options: 'last 1h', 'last 12h', 'last 24h' (selected), 'last 7 days', 'last 30 days', and 'last 365 days'. The second column contains data type options: 'Luftdruck1', 'Windrichtung1', 'Feuchte1', 'Helligkeit1', 'Temp1', 'Temp2', 'Windgeschwindigkeit', and 'Counter2'. A 'Show' button is located at the bottom of the selection area.

Die Benutzeroberfläche soll mehrere Aufgaben erfüllen:

- Anzeige der aktuellen Messwerte
- Ausgabe von Statusmeldungen
- Visualisierung der Messwerte
- Einfacher Datenexport

1 SQL: standard query language, Standard zum Speichern und Abfragen von Daten

- Aufstellen und Ändern von Regeln für die Benachrichtigung von Benutzern (Trigger)

Für die Implementierung habe ich mich für die Script-/Programmiersprache PHP entschieden, weil

- Sprache und Entwicklungsumgebung unter einer GPL²-ähnlichen Lizenz verfügbar sind
- der Interpreter für sehr viele Betriebssysteme verfügbar ist (Win32, diverse Unixe)
- sehr viel zusätzliche Software verfügbar ist (z.B. Module zur Erstellung von Diagrammen)
- eine Ausgabe als HTML ermöglicht dem Betrachter eine Unabhängigkeit von Betriebssystem des Messrechners, sowie eine Unabhängigkeit vom Ort des Messrechners

PHP setzt auf einem Webserver (in meiner Konfiguration Apache) auf, um eine dynamisch erstellte Website auszugeben. Der Benutzer kann nun nach Eingabe von von *http://mein_messrechner/* die Website mit allen oben geforderten Funktionen erreichen.

4.4 SMS-Service

Eine andere Form der Kommunikation mit dem Nutzer sind Textmeldungen für das Handy (SMS). Dazu benutzt der Deamon ein SMS-Gateway um den Benutzer bei etwaigen wichtigen Ereignissen zu informieren (z.B. Feueralarm) oder auch einen morgendlichen Wetterbericht zu versenden.

Das SMS-Gateway ist eine Software-Komponente, die über ein Datenkabel mit einem angeschlossenen Handy über die serielle Schnittstelle kommuniziert und eine SMS absenden kann.

Um an das verwendete Kommunikationsprotokoll für das Handy zu gelangen, habe ich auf die mitgelieferte Siemens Data Suite das Verfahren des *reverse engineering* angewandt. Durch gezieltes Belauschen der seriellen Schnittstelle konnte ich eine SMS-Sendevorgang mitschneiden. Mit Hilfe einer im Internet verfügbaren Anleitung¹³ konnte ich schließlich den Vorgang selbst nachprogrammieren. Allerdings ist dieser nicht trivial, denn es muss u.a. eine Zeichensatz-Konvertierung (8bit nach 7bit) durchgeführt werden.

² GPL: general public licence, eine Lizenz die dem Benutzer sehr viele Freiheiten gewährt

5 Schlussgedanken und Danksagung

Letztendlich kann ich sagen, dass mir dieses Projekt viel Spass bereitet hat und ich mein Wissen deutlich erweitern konnte. Auch wenn die Fehlersuche zeitweilig recht anstrengend und mühsam war.

Insgesamt habe ich gelernt, dass die einzelnen Teilprojekte besser durchgeplant werden müssen und meine, meist naiven Ansätze zum Scheitern verurteilt waren und diese entweder verbessert oder durch komplexere Ansätze ersetzt werden mussten. Hier haben sich vor allem die vielen Tücken der modernen Technik ausgewirkt, die nicht immer auf den ersten Blick zu erkennen waren. Woher soll z.B. der unbedarfte Schüler auch wissen, dass der Quarz ganz nah am Chip verlötet werden muss, und nicht „schön“ an einer weiter entfernten Stelle auf der Platine angebracht werden darf. Oft habe ich Erfahrungen dieser Art gemacht, aber aus diesen Fehlern lernt man vermutlich am besten.

Bedanken möchte ich mich vor allem bei meinem Vater, der mich immer unterstützt und gefördert hat. Mein Dank gilt auch meinem Physik-Lehrer, welcher es mir gestattete, selbst ein Thema für die Facharbeit zu wählen und mich nach Kräften unterstützt hat.

Vielen Dank auch an Prof. Dr. Ing. Eberhard Schuon, Dipl. Ing. Jörg Schuon, Dipl. Ing. Frieder Schuon und Marianne Schuon, welche mir mit wertvollen Tipps zur Seite standen.

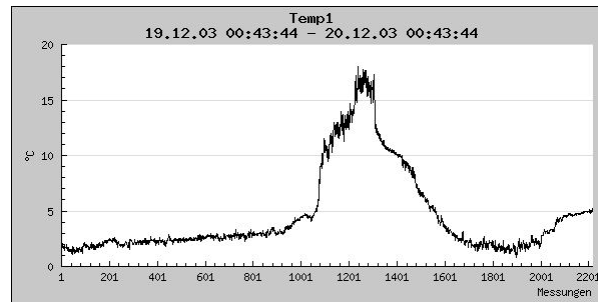
Zuletzt möchte ich mich auch bei Dallas Semiconductor / Maxim für ihre Versorgung mit kostenlosen Bauteilen und bei Sensortecnics für die zur Verfügung gestellten Drucksensoren bedanken.

6 Literaturverzeichnis

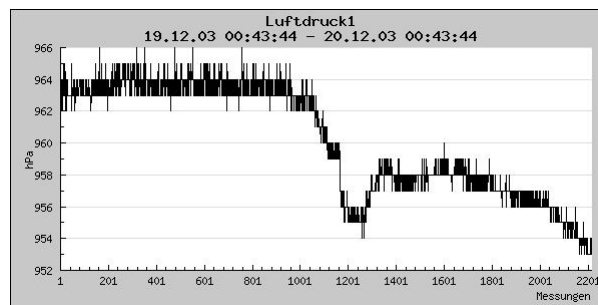
- [1] Awtrey, D., A 1-Wire Rain Gauge, in: Sensors Magazin Vol. 16 No. 12, Hrsg. Advanstar, USA, Dezember 1999, Reprint for Dallas Semiconductor
- [2] Dallas Semiconductor / Maxim, App Note 148: Guidelines for Reliable 1-Wire Networks, o.O., o.J.
- [3] Dallas Semiconductor / Maxim, Datasheet: DS1820, o.O., o.J.
- [4] Dallas Semiconductor / Maxim, Datasheet: DS2408, o.O., o.J.
- [5] Dallas Semiconductor / Maxim, Datasheet: DS2423, o.O., o.J.
- [6] Dallas Semiconductor / Maxim, Datasheet: DS2438, o.O., o.J.
- [7] Dallas Semiconductor / Maxim, Datasheet: DS2450, o.O., o.J.
- [8] Dallas Semiconductor / Maxim, Datasheet: DS2480B, o.O., o.J.
- [9] Dallas Semiconductor / Maxim, Datasheet: DS2490, o.O., o.J.
- [10] Dallas Semiconductor / Maxim, Datasheet: MAX232, o.O., o.J.
- [11] Dallas Semiconductor / Maxim, Engineering Journal Vol. 4, o.O., o.J., S. 3 – 8
- [12] Heimann, Datenblatt A9060, o.O., o.J.
- [13] Hüttisch, N., Kurznachrichten auf die perverse Art, http://www.nobbi.com/sms_pdu.htm vom 1.2.2004
- [14] Maxim, Datasheet: MAX4169, o.O., Oktober 1997
- [15] Panametrics, Datasheet: Mini Cap 2, o.O., o.J.
- [16] Philips Components, Product specification: Humidity sensor 2322 691 90001, o.O., 12. November 1996
- [17] Sensortechncs, App Note: SSAN-38, Puchheim, April 1999, S. 1F
- [18] Sensortechncs, Datasheet: SDX...A Series, Puchheim, März 2002
- [19] STMicroelectronics, Datasheet: NE556, o.O., Juli 1998

7 Anhang

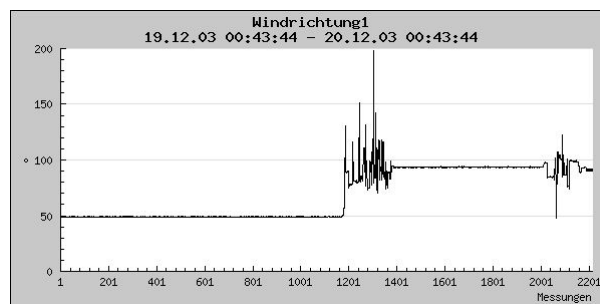
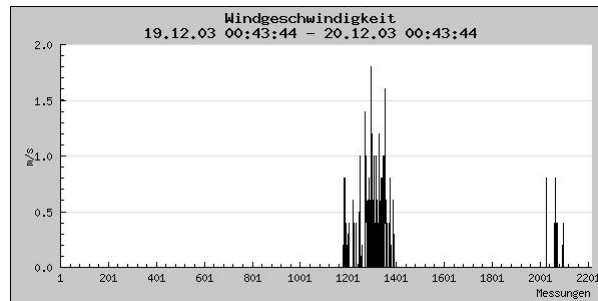
7.1 Ausgewählte Messungen



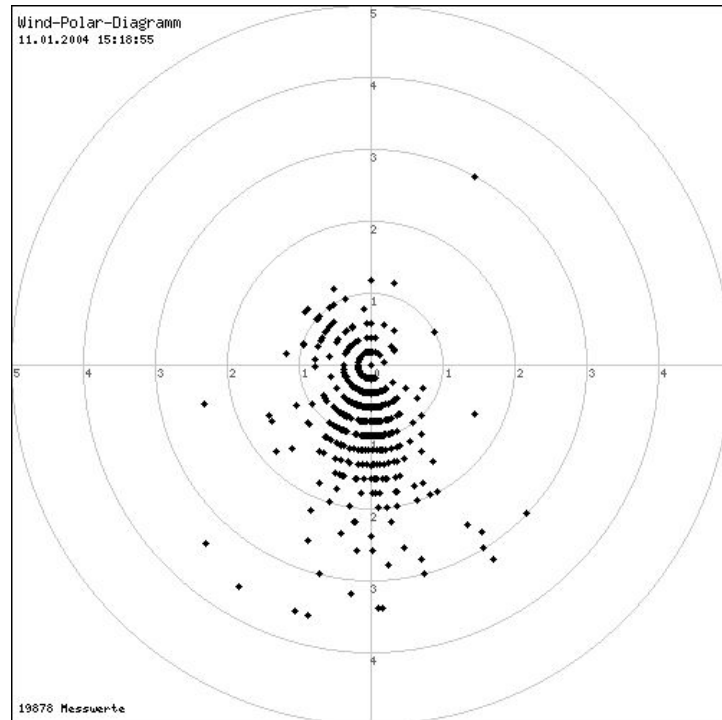
Temperaturverlauf gemessen über einen Tag



Druckverlauf gemessen über einen Tag



Windgeschwindigkeit und Windrichtung im Verlauf eines Tages.
Deutlich ist zu erkennen, dass sich die Windrichtung nur bei Wind ändert.



Dieses Diagramm zeigt die Verteilung des Windes. Man kann hier gut die Hauptwindrichtung erkennen.

7.2 Die Datenbankstruktur

Für die Wetterstation sind vier Tabellen in der Datenbank vorgesehen: eine für die Verwaltung der Sensoren, eine für die Speicherung der Messwerte, eine für Log-Meldung sowie eine für Trigger.

7.2.1 Die Tabelle für die Sensoren

Die Sensoren werden mit Hilfe der Tabelle *t_sensoren* verwaltet. Diese enthält folgende Felder:

Feldname	Typ	Beschreibung
a_sensor_id	Integer	Eindeutige Nummer für einen Sensor
a_1wire_id	Varchar	Die 1-Wire-Adresse des Bausteins, an dem der Sensor angeschlossen ist
a_1wire_kanal	Integer	Der Kanal am 1Wire-Baustein. Die Zählung beginnt bei 0.
a_bezeichnung	Varchar	Ein Name für den Sensor
a_genauigkeit	Double	Gibt die Genauigkeit des Sensors in Prozent an
a_sensor_typ	Integer	Gibt den Typ des Sensors an. Momentan sind folgende Werte möglich: 0: unbekannt 1: Temperatur 2: Helligkeit 3: rel. Luftfeuchte 4: Spannung 5: Luftdruck 6: Windrichtung 7: Windgeschwindigkeit

Die Abfrage um diese Tabelle zu erstellen lautet damit:

```
CREATE TABLE t_sensoren (  
    a_sensor_id int(11) auto_increment,  
    a_lwire_id varchar(100),  
    a_lwire_kanal int(11),  
    a_bezeichnung varchar(100),  
    a_genauigkeit double,  
    a_sensor_typ int(11),  
    PRIMARY KEY (a_sensor_id)  
) TYPE=MyISAM;
```

7.2.2 Die Tabelle für die Messwerte

Der Name der Tabelle lautet *t_messwerte* und sie enthält folgende Felder:

Feldname	Typ	Beschreibung
a_messung_id	Integer	Eine fortlaufende Nummer mit der eine Messung eindeutig identifiziert werden kann
a_sensor_id	Integer	Verwendeten Sensor an, vgl. <i>t_sensoren</i> .
a_messung_value	Double	Gemessener Wert
a_messung_einheit	Varchar	Einheit der gemessenen Größe
a_messung_zeit	Integer	Zeitpunkt der Messung, als UNIX-Timestamp ³

Die Abfrage um diese Tabelle zu erstellen lautet damit:

```
CREATE TABLE t_messwerte (  
    a_messung_id int(11) auto_increment,  
    a_sensor_id int(11),  
    a_messung_value double,  
    a_messung_einheit varchar(100),  
    a_messung_zeit int(11),  
    PRIMARY KEY (a_messung_id)  
) TYPE=MyISAM ROW_FORMAT=COMPRESSED;
```

7.2.3 Die Tabelle für Logs

Meldungen des Logs werden in der Tabelle *t_logs* gespeichert. Sie ist folgendermaßen aufgebaut:

Feldname	Typ	Beschreibung
a_log_id	Integer	Eine fortlaufende Nummer welche eine Meldung eindeutig kennzeichnet.
a_priority	Integer	Die Prorität der Meldung (1: niedrig, 2: mittel, 3: hoch)
a_category	Integer	Kategory (1: daemon, 2: trigger)
a_status	Integer	Status der Meldung (0: ungelesen, 1: gelesen)
a_message	Varchar	Die eigentliche Meldung
a_time	Integer	Zeitpunkt des Erscheinens der Meldung, als UNIX-Timestamp

Die Abfrage um diese Tabelle zu erstellen lautet damit:

```
CREATE TABLE t_logs (  
    a_log_id tinyint(4) auto_increment,  
    a_priority tinyint(4),  
    a_category tinyint(4),  
    a_status int(11),  
    a_message varchar(255),
```

³ UNIX-Timestamp: Sekunden seit dem 1.1.1970 in GMT

```
        a_time int(11),
        PRIMARY KEY (a_log_id)
) TYPE=MyISAM;
```

7.2.4 Die Tabelle für Trigger

Die Trigger werden in der Tabelle *t_triggers* gespeichert. Diese enthält folgende Felder:

<i>Feldname</i>	<i>Typ</i>	<i>Beschreibung</i>
a_trigger_id	Integer	Eine fortlaufende Nummer mit der ein Trigger eindeutig identifiziert werden kann
a_active	Integer	Gibt an, ob ein Trigger aktiv ist (1: ja, 0: nein)
a_name	Varchar	Name des Triggers
a_rule	Text	Regel für den Trigger, siehe 7.3
a_action	Integer	Was passiert, wenn der Trigger wahr wird? (1: Email, 2: SMS)
a_action_destination	Varchar	Ziel der Meldung via Email / SMS
a_run_once	Integer	Gibt an, ob der Trigger nur einmal wahr werden kann (1: ja, 0: nein)

Die Abfrage um diese Tabelle zu erstellen lautet damit:

```
CREATE TABLE t_triggers (
    a_trigger_id int(11) auto_increment,
    a_active int(11),
    a_name varchar(255),
    a_rule tinytext,
    a_action int(11),
    a_action_destination varchar(255),
    a_run_once tinyint(4),
    PRIMARY KEY (a_trigger_id)
) TYPE=MyISAM;
```

7.3 Die Triggersyntax

Eine Triggerregel besteht aus einem oder mehreren Statements. Diese Statements können untereinander boolisch verknüpft werden. Dazu werden die Ausdrücke AND und OR verwendet. Ein Statement hat jeweils die Form:

Sensor(*id*) Vergleichsoperator Wert

Für *id* muss eine gültige Sensor-ID eingesetzt werden. Mögliche Werte für *Vergleichsoperator* sind >, <, >=, <=, =. Für *Wert* sind beliebige Zahlenwerte möglich.

Eine gültige Regel sieht z.B. dann folgendermaßen aus:

```
Sensor(1) >= 20 AND Sensor(4) < -11,5
```

7.4 Dateiliste der Softwarekomponenten

Im Folgenden werden alle wichtigen Dateien für die Softwarekomponenten aufgeführt. Die anderen Dateien, welche sich in den Verzeichnissen befinden, sind meist von der IDE oder dem Compiler erzeugte Hilfsdateien sowie Bilder oder Icons.

7.4.1 Deamon

Verzeichnis *Wetterstation_Deamon*:

Verzeichnis *Icons*:

Farbige Icons für den Zustand der Sensoren

Verzeichnis *MyVbQL – Source*:

Komponenten für den Zugriff auf die MySQL-Datenbank. Diese Software stammt von <http://www.icarz.com/mysql/>

Verzeichnis *SendMail – Source*:

Komponenten für den Email-Versand via SMTP. Diese Software stammt von *freevbcode.com / Dean Dusenbery*

frmAdd.frm

Formular zum Hinzufügen von Sensoren

frmMain.frm

Oberfläche für den Deamon sowie die meisten Funktionen des Deamons.

frmStartup.frm

Dieses Formular wird angezeigt, wenn der Deamon startet.

Wetterstation_Deamon.vbp

Die Projektdatei.

Verzeichnis *shared*:

md1Wire.bas

Funktionen für die 1-Wire-Kommunikation

mdEmail.bas

Funktionen für den Versand von Emails

mdmktm.bas

Funktion für Unix-Timestamps

mdSMS.bas

Funktionen zum Versand von SMS

mdTimeZone.bas

Funktionen zum Zeitzone-Ermittlung

mdTriggers.bas

Funktionen zur Auswertung von Triggern

7.4.2 Benutzeroberfläche

Verzeichnis *WebInterface*:

Verzeichnis *include*

Verzeichnis *gfx*:

Hier befindet sich die Funktion zum Erstellen des Wind-Polar-Diagramme

Verzeichnis *jpgraph*:

Hier befindet sich die Bibliothek für das Erstellen der Diagramme. Diese stammt von <http://www.aditus.nu/jpgraph/>

Verzeichnis *ui*:

export.inc.php

Export-Funktionalität

graphs.inc.php

Auswahl und Anzeige der Diagramme

logs.inc.php

Ausgabe der (Warn-)Meldungen

settings.inc.php

Erstellen und Bearbeiten von Triggern

ui_start.inc.php

Ausgabe des oberen Teils der Vorlage für die Webseite

ui_end.inc.php

Ausgabe des unteren Teils der Vorlage für die Webseite

connect.inc.php

Diese Datei stellt die Verbindung zur Datenbank her.

diagramm.inc.php

Mit dieser Datei wird die Diagramm-Bibliothek angesteuert.

status.inc.php

Von dieser Datei wird der Status der Sensoren ausgegeben.

time.inc.php

Diese Datei stellt Funktionen für die Behandlung von Datum-Strings zur Verfügung

sql_prevention.inc.php

Diese Datei schützt vor SQL-Injection-Attacken

Verzeichnis *ui*:

index.html

Das Frameset für die Benutzeroberfläche

cur_values.htm

Vorlage für die aktuellen Werte der Sensoren

main.htm

Vorlage für das Frame *main*.

navigation.htm

Das Menu links oben

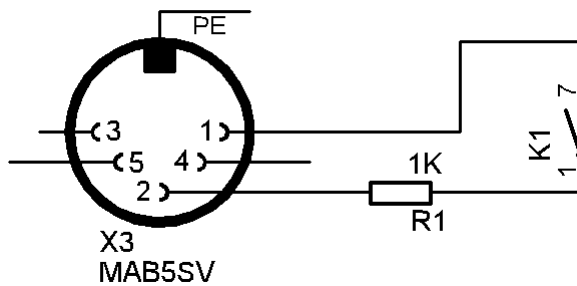
index.php

Hauptseite, bindet dann aber Unterseiten ein.

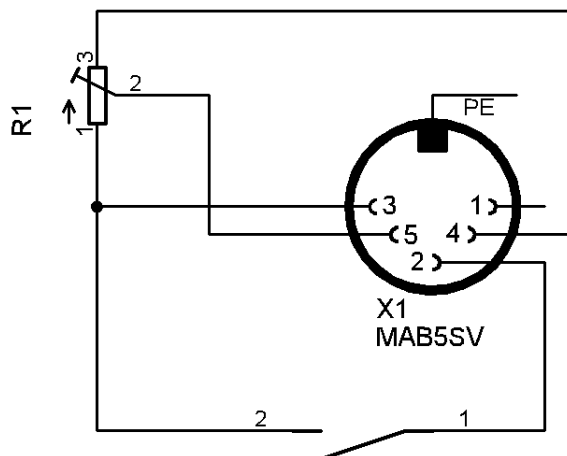
settings.inc.php

Hier werden Einstellungen wie z.B. die Datenbankverbindung angegeben

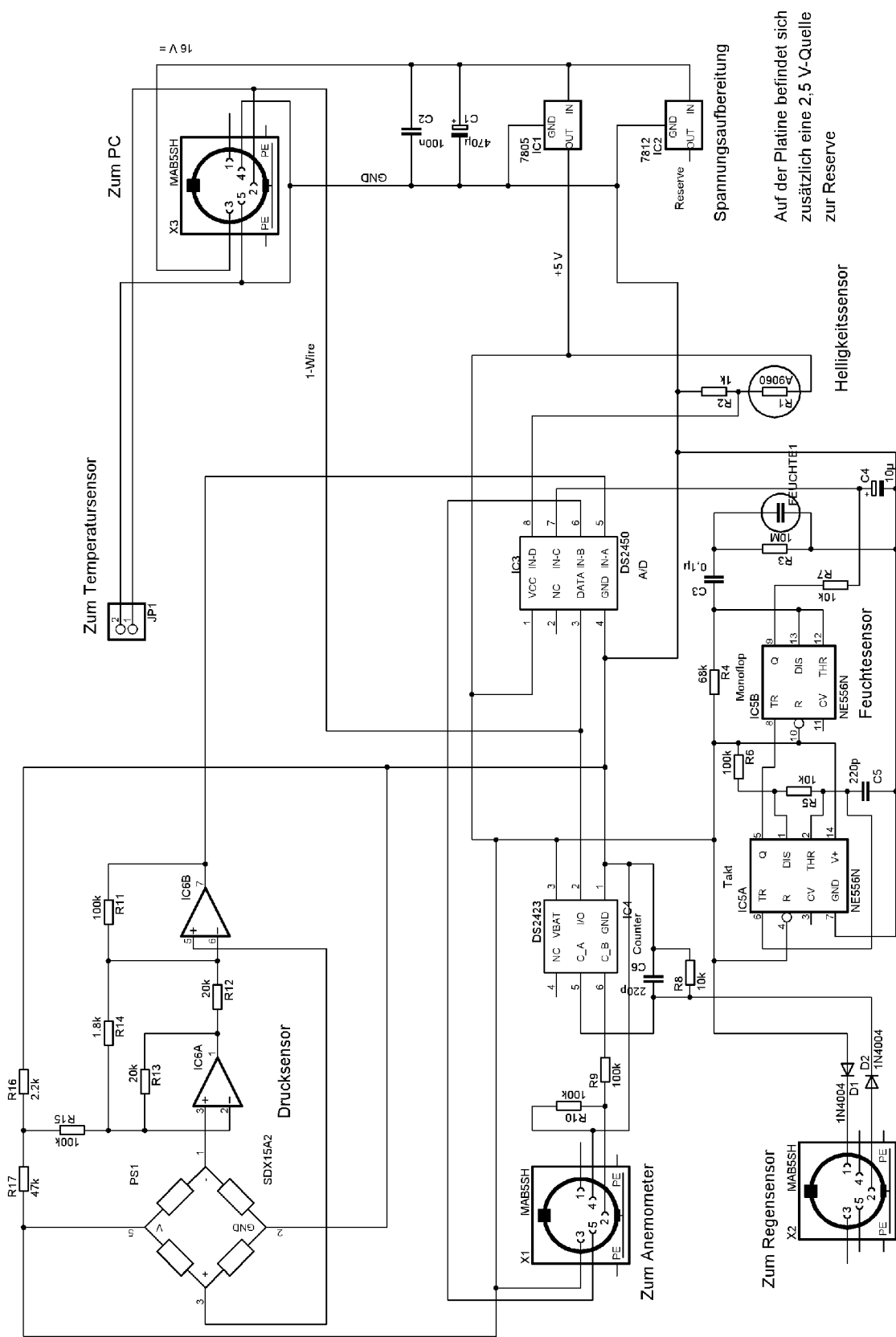
7.5 Die Schaltpläne



Schaltplan des Regensensors mit Verbinder zur Wetterstation



Schaltplan des Windgeschwindigkeits- und Windrichtungssensor.



Auf der Platine befindet sich zusätzlich eine 2,5 V-Quelle zur Reserve

Spannungsaufbereitung

Ich erkläre hiermit, dass ich die Facharbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benützt habe.

Puchheim....., den 02.02.2004.....

Ort

Datum

.....
(Unterschrift des Schülers)