



by Ralf Wieland
<rwieland(at)zalf.de>

Luftdruckmessung unter Linux



About the author:

Ich beschäftige mich mit Umweltsimulation, neuronalen Netzen und Fuzzy-Systemen, indem ich sie programmiere. Letzteres vollzieht sich immer unter Linux (seit 0.99pl12). Weiterhin bin ich an Elektronik und Hardware interessiert und versuche das mit Linux zu verbinden.

Abstract:

Kauft man heute ein Messmodul, so bekommt man immer einen irgendwie funktionierenden Treiber für eine Windows-Variante mitgeliefert. Der Linux-Nutzer geht in der Regel leer aus. Das muss nicht sein, da die Ankopplung von Hardware unter Linux einfacher ist, als unter Windows. Auch das Argument vieler Hersteller, dass es zu wenig Linux-Nutzer gibt, ist fragwürdig, da gerade Linux-Nutzer als besonders experimentierfreudig gelten. Wie dem auch sei, man muss selbst Hand anlegen. Die dazu notwendigen Schritte wollte ich einmal notieren, vielleicht sind sie auch anderen von Nutzen.

Translated to English by:
name of translator <??>

Einleitung

Für die Entwicklung eines jeden Messsystems sind folgende drei Fragen zu klären:

- Was bedeutet die Messgröße physikalisch?
- Wie bekommt man die Größe in den Computer?
- Was will man mit der Messgröße machen?

Obwohl die Fragen trivial klingen, sollte man sich vor einer Eigenentwicklung jeden Schritt genau überlegen.

Physikalisches Messprinzip

Der Luftdruck ist eine interessante Messgröße. Für Segler oder Bergsteiger gibt er Hinweise auf bevorstehende Wetteränderungen. Aber auch mir, als Großstadtbewohner macht die Beobachtung des Luftdrucks Spass. Dass das auch anderen so geht, sieht man in den vielen Wetterstationen, die die Wohnzimmer schmücken. Oft sind diese Geräte aber mehr Spielzeuge als ernsthafte Messgeräte. Will man ernsthaft den Luftdruck messen, so sind eine Reihe von Randbedingungen zu beachten.

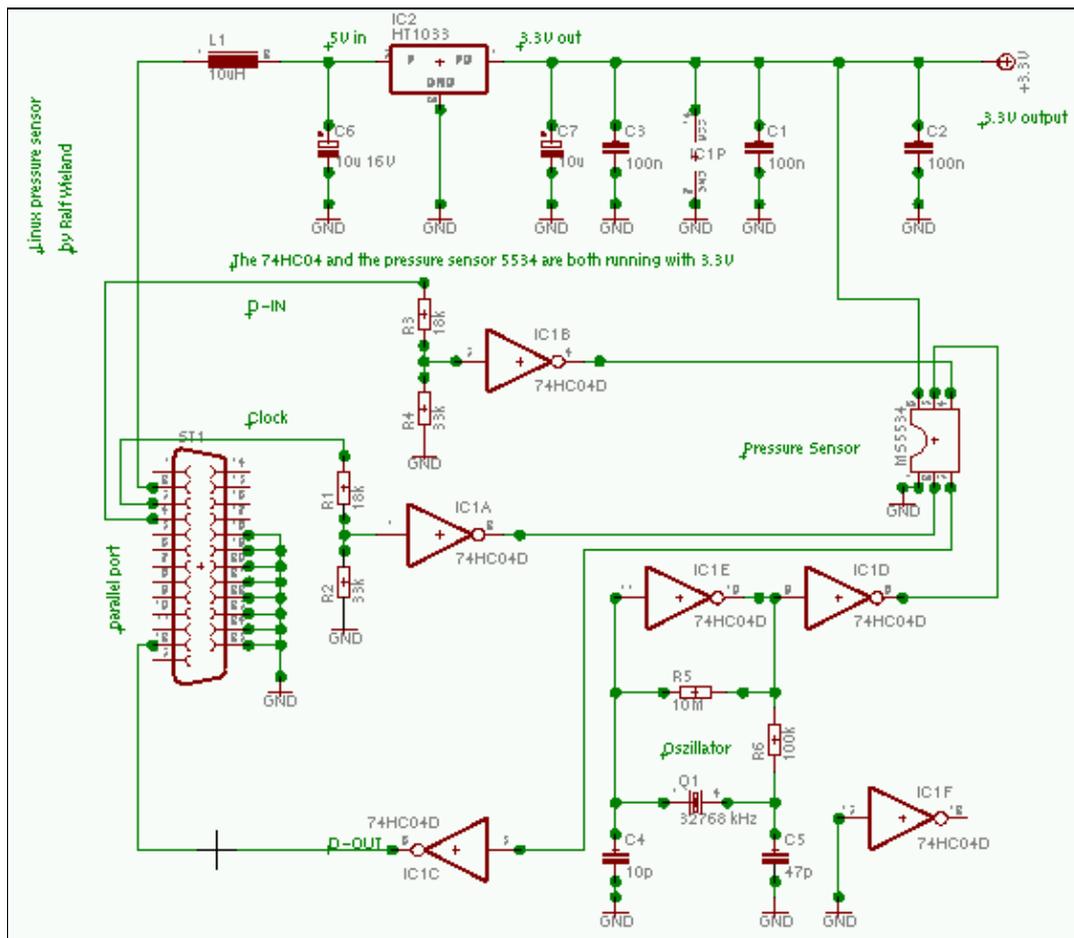
Der Luftdruck kann mit dem oben gezeigten Barographen recht genau ermittelt werden. So ein Gerät auf rein mechanischer Basis sieht nicht nur schön aus, es kostet auch recht viel. Um diese hohen Kosten zu umgehen, misst man heute den Luftdruck mit Halbleitersensoren. So fertigt z.B. Motorola eine ganze Serie von Drucksensoren für unterschiedliche Einsatzgebiete. Halbleitersensoren sind aber immer temperatur- und spannungsabhängig. Um diese Charakteristik muss man sich kümmern. Der Luftdruck ist vergleichsweise einfach zu ermitteln, da er im ganzen Haus als konstant angesehen werden kann. Damit braucht der Sensor nicht im Freien angebracht, sondern kann direkt am Rechner platziert werden. Das kommt einer Temperaturkompensation sehr zu Gute, da ja davon ausgegangen werden kann, dass die Temperatur am Rechner sich nicht sehr ändert. Trotzdem ist bei hoher Auflösung eine Temperaturkompensation unumgänglich. Bei dem vorgeschlagenen Sensor, der von ELV, einem renommierten europäischen Elektronikversender als Bausatz angeboten wird (und der auch freundlicherweise die Genehmigung zum Nachdruck der Schaltung gab), wird die Temperaturkompensation in der Software durchgeführt. Dazu misst der Sensor nicht nur den Druck, sondern auch die Temperatur. Der Algorithmus zur Temperaturkompensation kann dem Datenblatt von Intersema, dem Hersteller des Sensors, entnommen werden. Die Spannungskompensation übernimmt ein integrierter Spannungsregler. Damit ist die Frage der Sensorwahl und des Aufstellungsortes geklärt.

Der Luftdruck ist nicht nur vom Wettergeschehen abhängig, sondern auch von der Höhe über dem Meeresspiegel (NN). Will man den Luftdruck vergleichen, so ist er auf NN zu beziehen. Das kann mit der internationalen Höhenformel:

$$p_0 = p / (1 - 6.5 * h / 288000)^{5.255}$$

geschehen. Sie berücksichtigt neben der Höhenabhängigkeit des Luftdrucks auch die Abnahme der Temperatur mit der Höhe und erweitert damit die bekannte barometrische Höhenformel. Der gemessene Luftdruck p wird mit der Höhe h über NN korrigiert. Grob geschätzt kann man mit Druckverminderung von 1.2mbar/10m rechnen. Die Höhe über NN ist im Programm myclient als #define HIGH_NN einzugeben. Die Druckdifferenz wird dort nach der internationalen Höhenformel ermittelt. Mit Berücksichtigung der Höhe ist man noch nicht fertig. Es bleibt noch die Frage, wie genau will man messen und wie oft muss man messen. Die Genauigkeit hängt direkt vom verwendeten Sensor ab. Dabei ist zwischen Genauigkeit, also der Verlässlichkeit des Messwertes und der Auflösung, also wieviele Stellen angezeigt werden, zu unterscheiden. Der verwendete Wandler hat eine Genauigkeit von ± 1.5 mbar über einen Bereich von 750..1100mbar bei 25°. Die Auflösung beträgt aber 15Bit \pm 7Bit, d.h. $1/2^{15} = 3 * 10^{-5}$ oder 0.03mbar. Vergleicht man die Auflösung mit der Genauigkeit, so wird die Unsinnigkeit einer derart hohen Anzeigegenauigkeit offensichtlich. Im Programm wird daher auf eine Stelle hinter dem Komma gerundet. (Und das auch nur, damit die Anzeige nicht unsinnig springt.) Als Letztes ist noch zu klären, wie oft gemessen werden sollte. Der Luftdruck ändert sich nicht gerade sprunghaft. Erfasst man den Luftdruck alle 5..10min, so sollte das für alle in der Praxis auftretenden Fälle ausreichend sein. Um ein zufälliges Schwanken der Messwerte auszugleichen, mit anderen Worten um Rauschen auszufiltern, wird der Messwert alle 10..60s abgefragt und einer digitalen Tiefpassfilterung unterzogen werden. Das ist zwar nicht unbedingt nötig, schadet aber auch nicht. Die Belastung des Rechners ist dabei noch vernachlässigbar. Im konkreten Beispiel führt das letztlich dazu, dass alle 10s gemessen, aber nur alle 2min die Messwerte gespeichert werden. Damit fallen täglich $60 * 24 / 2 = 720$ Messwerte an, die dann einer Anzeige zuzuführen sind. Für eine Darstellung der Druckentwicklung stehen somit 720 Werte zur Verfügung, was eine sehr detailreiche Darstellung ermöglicht.

Interface Sensor/PC



Für eine größere Ansicht auf den Schaltplan klicken.

Als Interface wurde die parallele Schnittstelle des PC gewählt. Das ergibt einigen Sinn, nachdem die neueren Drucker den USB-Port nutzen. Außerdem ist die parallele Schnittstelle recht einfach zu programmieren, so dass die Aufgabe ohne zusätzliche Hardware lösbar war. Außer den Gattern, die zur Pegelanpassung und zur Erzeugung des von dem Sensor benötigten Taktes notwendig waren, wurde nur noch die Spannungsstabilisierung integriert. Falls man nicht auf den Bausatz zurückgreifen möchte, lassen sich die wenigen Bauteile auch bequem auf eine Universalleiterplatte löten. Die Entwicklung einer speziellen Leiterplatte dürfte nur in den seltensten Fällen lohnen.

Kurz zur Schaltung:

Die Zahlen am linken Rand bezeichnen die Anschlüsse an der parallelen Schnittstelle. Die Anschlüsse 18..25 bilden die Masseleitungen, die Anschlüsse 12,3 und 4 sind Dataleitungen, also Outputs auf den Sensor. Der Anschluss 12 ist der sogenannte Paper-Error, der einen Fehler im Drucker signalisiert und als Input in den Rechner fungiert. Über die Widerstände R1/R2 bzw. R3/R4 wird die Spannung von 5V auf einen Wert unter 3.3V reduziert. Sie dienen somit zur Pegelanpassung. Erwähnenswert ist noch der Taktgenerator für den Sensor, der schaltungstechnisch aber keine Besonderheiten aufweist. Mit der Stromversorgung von 5V auf stabile 3.3V ist die kleine Schaltung auch schon komplett.

Es wurde ein Programm entworfen und in C implementiert, das den Sensor abfragt, die Temperaturkompensation durchführt, die Werte einer digitalen Filterung unterzieht und die Werte letztlich speichert. Das Ganze ist recht übersichtlich und kann leicht nachvollzogen werden. Zur Abfrage des Sensors wird eine Folge von Impulsen an den Sensor gesendet. Der Sensor interpretiert diese Folge als Befehl und antwortet seinerseits mit einem als Impulsfolge kodierten Messwert. Dieses Handshake ist im Datenblatt sehr gut dokumentiert und kann anhand des Programmtextes leicht nachvollzogen werden.

Nutzung und Visualisierung der Messwerte

Eine Aufnahme und Speicherung der Messwerte ist nur ein Teil der Aufgabe bei der Entwicklung eines Messsystems. Die sehr wichtige Frage nach der Visualisierung ist zu klären. Es stellte sich nämlich heraus, dass eine Visualisierung von Druckentwicklungen während eines Tages bzw. einer Woche nicht unbedingt auf Gegenliebe von Familienmitgliedern stoßen. Der Rechner macht Lärm und verbraucht Energie. Um diesem Problem aus dem Wege zu gehen, kann man einen DIL/NetPC einsetzen. So ein Teil ist völlig lautlos und verbraucht auch wenig Energie. Leider sind diese Dinger nicht gerade billig. Außerdem haben DIL/NetPC keine eigene Auswerteeinheit, sondern müssen via Netzwerk abgefragt werden. Die Kopplung via Netzwerk schien interessant. Letzteres brachte mich auf die Idee, einen Linux-Server auf Arbeit für die Druckmessung zu missbrauchen. Wir setzen Linux-Server für vielfältige Aufgaben ein, da macht eine gelegentliche Abfrage des Sensors nicht viel aus. Wer nicht über so einen Luxus verfügt, kann vielleicht einen ausgedienten PC irgendwo im Keller platzieren, wo er kaum auffällt. (Strom verbraucht er aber trotzdem ;-)) Wie dem auch sei, eine Abfrage via Netzwerk musste her. Die Idee bestand darin, einen Server zur Datenerfassung und Speicherung aufzubauen, der dann bei Bedarf vom Client abgefragt werden kann. Damit ist eine multiple Nutzung im Netzwerk gesichert. Auch meine Kollegen kommen nun in den Genuss der Luftdruckmessung. Eine Einschränkung sei hier allerdings gemacht, der Server läuft in der jetzigen Form mit Rootrechten. Das ist erforderlich, da die Schnittstelle direkt angesprochen wird. Ein Treiber, wie Parapin, kann genutzt werden, um die Schnittstelle auch als nichtprivilegiertes Nutzer anzusprechen. Das setzt aber in den meisten Fällen die Rekompilation des Kernels voraus und ist nicht jedermanns Sache. Bei einer Verbindung zum Internet sollte aus Sicherheitsgründen diese Mühe auf sich genommen werden. Für meine Anwendung reichte die einfache Variante aus.

Es wurden also zwei Programme entwickelt. Einen Server, der die Daten erfasst und speichert und einen Client, der die Daten vom Server anfordert. Beide sind mittels einer Socket-Schnittstelle via Ethernet verbunden. Für den Server ergab sich dabei noch, dass hier zwei unabhängige Prozesse laufen. Einer, der die Daten vom Sensor abfragt und einer, der die Netzwerkverbindung abwickelt. Beide schließen sich gegenseitig aus, d.h. wenn die Schnittstelle abgefragt wird, muss der Netzwerkprozess warten und umgekehrt, wenn der Netzwerkprozess gerade die Werte übermittelt, darf der Abfrageprozess keinen neuen Wert bereitstellen. Diese Prozesse wurden leichtgewichtig mittels Threads realisiert. Dazu kam die PThreads-Library zum Einsatz, die aber in jeder Linux-Distribution enthalten sein sollte.

Der Client wurde bewusst einfach gehalten. Hier werden im wesentlichen nur die Daten empfangen und die Höhenkorrektur durchgeführt. Ausgegeben werden die Daten via stdout. Es werden ausgegeben:

die Zeit ab Messbeginn in Stunden als float
die Messwerte als float
gefolgt von einem nl.

Eine Ausgabe sieht dann beispielsweise wie folgt aus:

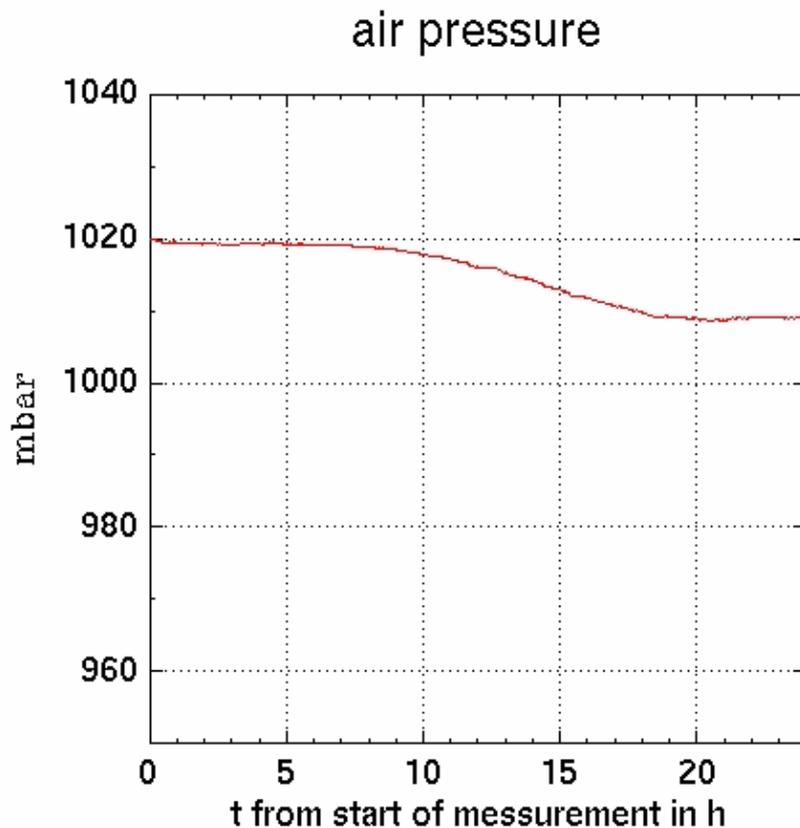
```
0.000000 1008.2
0.033333 1008.1
0.066667 1008.2
0.100000 1008.0
0.133333 1008.1
...
```

Damit können Visualisierungsprogramme wie gnuplot oder die Plotutils einfach genutzt werden. Ich habe wegen der guten Darstellungsqualität die Plotutils gewählt. Auch wenn die Darstellung als Diagramm mit den erwähnten Hilfsmitteln ziemlich leicht zu realisieren ist, sollte man sich doch ein paar Gedanken bezüglich vergleichbarer Darstellungen machen. Was nützen schöne formatfüllende Diagramme, die aber jeden Tag mit einer anderen Skalierung aufwarten? Man möchte schließlich einen Tag mit den vorhergehenden vergleichen und sich nicht immer mit anderen Skalierungen herumärgern. Deshalb bevorzuge ich eine absolute Skalierung.:

```
./myclient modell1 | graph -T X -C -g 3 -L "air pressure" \
```

```
-X "t from start of measurement in h" -Y "mbar" --x-limits 0 24\  
--y-limits 950 1040
```

Der Client "myclient" ruft vom Rechner "modell1" die Daten ab und gibt sie via Pipe den Plotutils zur Darstellung. Die Skalierung des Luftdrucks liegt zwischen 950mbar..1040mbar. Als Zeiteinheit werden 24 Stunden verwendet. Tauscht man in den Befehlszeilen die Option `-T X` in `-T ps`, so wird eine Postscriptbefehlsfolge erzeugt, die sich wunderbar drucken oder in eigene Dokumente einbinden lässt. Das nur als Hinweis, für jemanden der die Plotutils noch nicht ausgiebig nutzt.



Das Diagramm zeigt das Ende einer Schönwetterperiode und Einsetzen von Niederschlag.

Installation

Die Installation ist ziemlich einfach. Nachdem das Modul aufgebaut wurde, und auf Fehler (Brücken, kalte Lötstellen etc.) kontrolliert wurde, kann es an die parallele Schnittstelle angeschlossen werden. Die Software wird wie üblich mittels `tar -zxvf druck-0.1.tgz` entpackt und danach wechselt man ins Verzeichnis `druck-0.1` und kann nach Eintragen der Höhe über NN in die Datei `myclient.c` durch ein einfaches `make` die Übersetzung starten. Alles sollte problemlos laufen. Bevor das Modul aber genutzt werden kann, muss in der `/etc/services` der Port für die Socketschnittstelle eingetragen werden. Ich verwendete hierzu:

```
socktest      7123/tcp      # Air pressure sensor
```

den bei mir freien Port 7123 als socktest. Ist das geschehen, kann unter Root das Programm ./druck LPT1 (oder LPT2) gestartet werden. Ging alles klar, so meldet sich der Server und gibt alle 10 Sekunden die rohen Werte für Druck, Temperatur und die aktuelle Zeit aus. Damit ist gleichzeitig eine Kontrolle des Sensors gegeben. Der Client kann nun auf dem gleichem oder sinnvollerweise auf einem anderen Rechner gestartet werden. Dazu ist ebenfalls auf dem Client-Rechner der Port socktest in die /etc/services einzutragen. Um die Visualisierung etwas zu vereinfachen, wurde ein Shellskript druck.sh entwickelt. Dort ist der Servername durch den aktuellen Servernamen zu ersetzen und alles sollte laufen.

Schlussbemerkung

Aus dem Ärger, dass es fast nie einen Treiber für Linux bei dem Anschluss von Sensoren gibt, sollten die Schritte zur Entwicklung einer Schnittstelle einmal nachvollziehbar gestaltet werden. Es war interessant, dass bei einer scheinbar so einfachen Sache, wie dem Anschluss eines Luftdruckmoduls, doch eine Reihe von Dingen zu beachten sind, bis alles vernünftig läuft. Erst durch Einbeziehung der Netzwerkschnittstelle konnte eine befriedigende Lösung erzielt werden. Trotzdem bleiben noch eine Reihe von Dingen ungeklärt. Wenn auch die Nutzung der parallelen Schnittstelle mittels Parapin die leidigen Rootrechte verhindern kann, so ist eine weitergehende Nutzung dieser Werte via Internet noch völlig offen. Gesetzt den Fall, es wollen sich mehrere Leute Wetterstationen aufbauen, indem sie vielleicht noch Temperatur- und Luftfeuchtefühler integrieren (eine Windmessung ist recht kompliziert und durch den Laien nicht ohne weiteres durchführbar). Dann bleibt die Frage, in welcher Form sie ihre Daten austauschen. Vielleicht hat jemand eine Idee und kann diese Frage in einem gesonderten Beitrag erörtern?

Referenzen

- Datenblatt Drucksensor von Motorola:
<http://e-www.motorola.com/brdata/PDFDB/docs/MPX4100A.pdf>, lokale Kopie: [MPX4100A.pdf](#)
- Quellcode der Software: [download Seite für diesen Artikel](#)
- Elektronikversand ELV [Bausatz\(Artikel-Nr.: 68-388-03\)](#)
- Datenblatt Drucksensor von Intersema <http://www.intersema.ch/pro/module/file/da5534.pdf>, lokale Kopie: [da5534.pdf](#)
- Mini-PCs [DIL](#)
- Parallelporttreiber [Parapin](#)
- Linuxthreads [PThreads](#)
- Homepage [Gnuplot](#)
- Homepage [Plotutils](#)

<p>Webpages maintained by the LinuxFocus Editor team © Ralf Wieland "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: de --> -- : Ralf Wieland <rwieland(at)zalf.de> de --> en: name of translator <???</p>
---	---