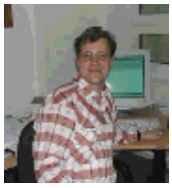
LinuxFocus article number 284
http://linuxfocus.org

Measuring air pressure with Linux

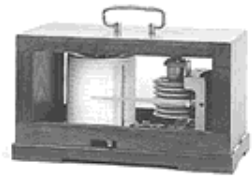


by Ralf Wieland
<rwieland(at)zalf.de>

About the author:

I use Linux (since 0.99pl12) to program environmental simulations, neural nets, and fuzzy systems. I'm also interested in electronics and hardware and I use Linux in these areas too.

Translated to English by:
Guido Socher
<guido(at)linuxfocus.org>



Abstract:

If you buy some sort of measurement module you always get a driver for one of the Windows variants. Linux users get usually nothing. It does not have to be like that, because it is technically often easier to write a hardware driver for Linux than for Windows. The argument of some manufactures that there are too few Linux customers is questionable as Linux users are usually more active and experimenting with new things. Anyway, often the solution is to write the driver yourself. I wanted to note down the steps to get the driver working. Maybe it will be useful also for other people. This will not be a kernel level driver but I will use the parallel port.

Introduction

For the development of a measurement system you have to clarify the following 3 questions first:

- What is the meaning of the measurand in physics?
- How do you get the data into the computer?
- What do you want to do with the data?

These questions may sound trivial but it is good to think about them at each step of the development.

The physics

Air pressure is interesting for several reasons. For sailors and mountain climbers it is an indication for a coming change in weather. But it is also fun for me to watch the air pressure at home. Many other people seem to do the same as one can see from all the little weather stations found in living rooms.

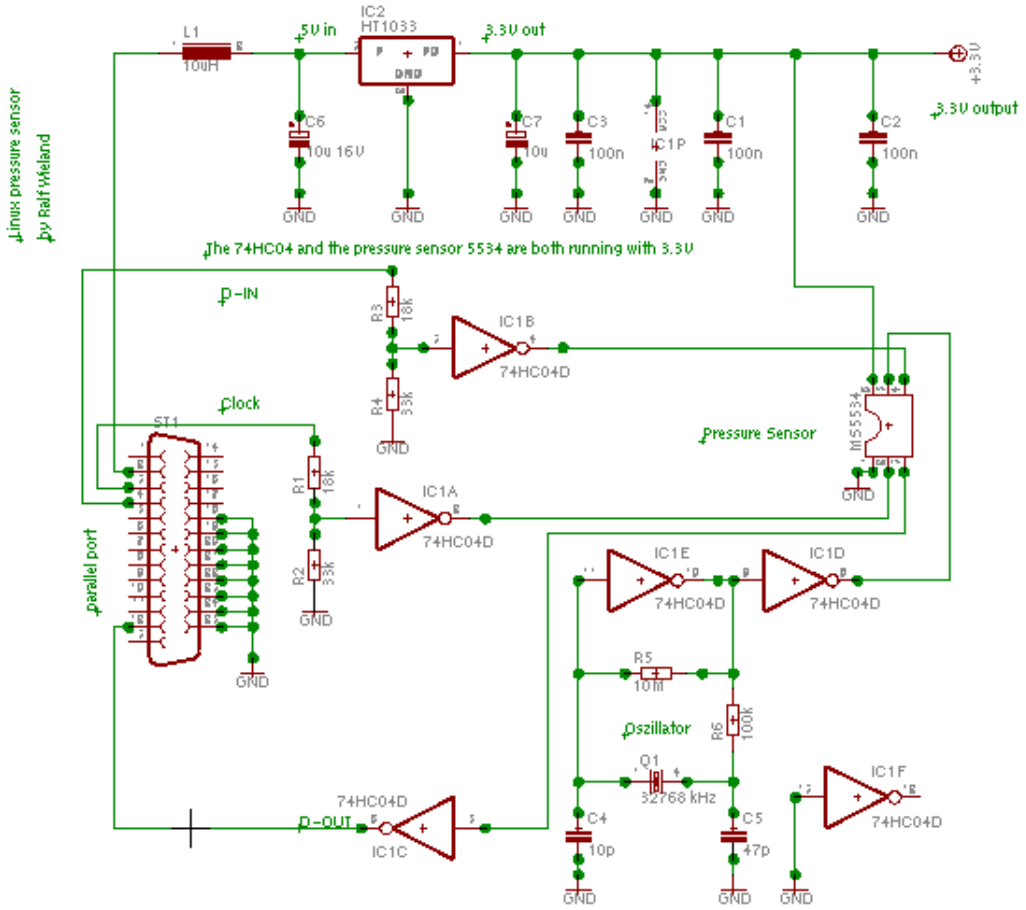
These things are however often more toys than real measurement equipment. If you want to do serious measurements of air pressure then you have to consider a few things. Air pressure can be measured accurately with a Barograph as shown in the title picture above. They are purely mechanical and cost a lot. To avoid the cost one can use an electronic semiconductor sensor. Motorola produces e.g a number of sensors for different purposes. Semiconductor sensors are always temperature and voltage dependent. You have to take this into account. Air pressure is relatively simple to measure since it is the same inside and outside the house. You don't have to build the sensor for outside use and protect it against rain. You can have the sensor directly next to your PC. This is also good for the temperature compensation since temperature does not change as much inside the house as outside. To get results with high accuracy you will however need an electronic temperature compensation. The proposed circuit is available as a kit from ELV, a European electronic distributor. ELV has given me the permission to publish the circuit. The temperature compensation is in this cases done via software. The sensor measures not only the pressure but also the temperature. The algorithm for the temperature compensation is documented in the datasheet of Intersema's air pressure sensor. The voltage stabilization is done with a 3.3V regulator.

The air pressure is not only dependent on the weather but also on the height above sea level. To compare the air pressure in different locations you need to normalize to the value at sea level. This is internationally done with the following formula:

p0=p/(1-6.5*h/288000)^5.255

This formula considers not only the dependence on the altitude but also the fact that the temperature decreases in higher locations. The air pressure as measured will now be compensated with the altitude over see level. You can calculate that the air pressure changes approximately by 1.2mbar/10m. The altitude over sea level is found in the program myclient as `#define HIGH_NN`. The remaining question is how accurate the measurement is. The accuracy depends directly on the sensor. You have to distinguish between the resolution and the accuracy of the results. The sensor has an accuracy of +-1.5mbar between 750..1100mbar at a temperature of 25°. The resolution is however much higher: 15Bit+-7Bit that is 1/2^15 = 3*10^-5 or 0.03mbar. In other words the last digits have no meaning. The program will therefore round the results to 1 digit behind the comma. Still we have to clarify how often one should measure. An interval of 5..10min should be enough for most purposes. To filter out random changes we will measure every 10..60s and then apply a low pass filter. The CPU load generated by this polling of data is negligible. I used a measurement interval of 10s and the values are stored every 2min. This produces 60*24/2=720 measurement values every day.

The interface between Sensor and PC



Click on the schematic for a bigger picture.

As the interface I used the parallel port. This makes sense since most modern printers will use the USB port. The advantage is that the parallel port is very easy to program and you don't need much additional hardware. Some inverter gates to adjust the voltage levels and to generate a clock signal.

If you don't want to buy the kit from ELV then you can also solder the parts onto a generic experimentation board.

A few comments on the circuit:
The numbers on the left are the pins on the parallel port. Pins 18..25 are ground and 12,3,4 are data lines. Pin 2 is used as power supply for the circuit. Resistors R1/R2 and R3/R4 are used to adapt the 5V of the parallel port to 3.3V. Both the 74HC04 and the sensor are running with 3.3V.

The program was written in C. It reads the data from the sensor and implements the digital low pass filter as well as the temperature compensation. The code should be easy to follow. You can download it at the end of this article. To retrieve the data from the sensor a number of pulses are sent to the sensor. The sensor interprets these pulses as commands and replies. The protocol is documented in the datasheet for the sensor.

Using and visualizing the measurement results

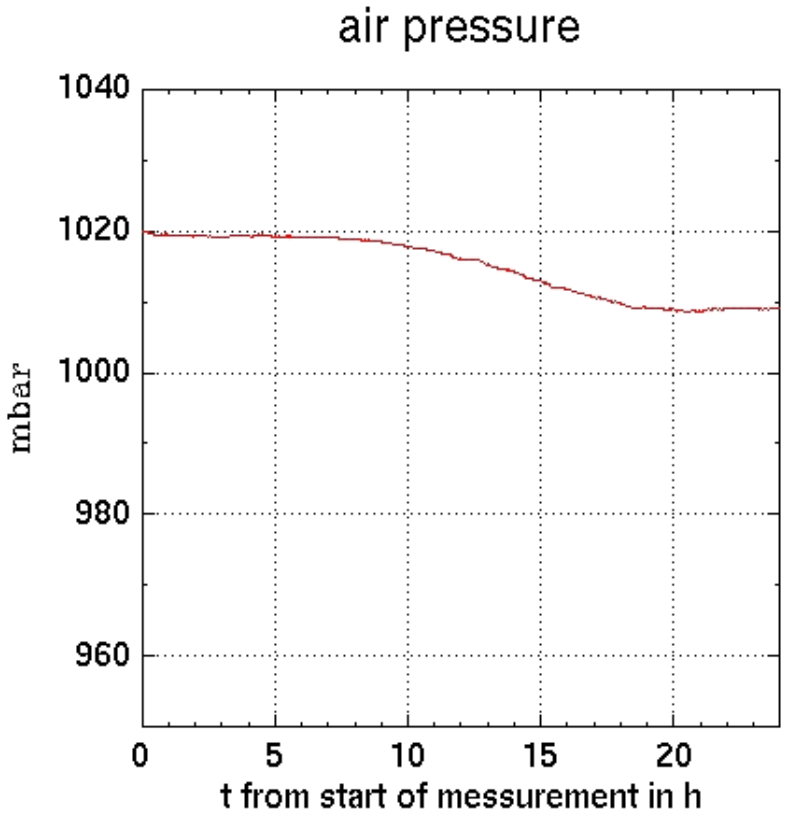
To measure air pressure with a normal PC over several days may not find may friends in you family since normal PCs are noisy and the PC would need to run all the time. To use a DIL/NetPC is an interesting alternative. These PCs use very little power and do not make any noise. Unfortunately they are not cheap and they can only be accessed via network. This led me to the idea of using the Linux-Server at work for this purpose. This server is anyhow running all the time. Those who don't have access to such a server can maybe use an old PC and deposit it somewhere in the cellar where it does not disturb. To retrieve the data I developed a server and client application. A limitation of this is currently that the server process must run with root rights to access the parallel port directly. An alternative would be to use a driver like Parapin to use the parallel port without root rights. This requires however normally the re-compilation of the kernel in order to add the Parapin driver. If you plan to make the data available over the Internet then you should definitely consider using a non privileged user. For my purposes this simple solution was enough.
The client application and the server process use TCP/IP sockets for communication. The server runs in 2 threads. One that gets the data from the sensor and one that handles the networking. These processes are implemented with the PThreads-Library. The client is very simple. It gets the data and corrects the height according to the above formula. The data is simply printed to stdout in 2 columns. The first one is the time in hours as a floating point number. It is counting the hours since the start of the measurement. The second column is the air pressure as a floating point number. It looks as follows:

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

This can be feed directly in to a plotting program like gnuplot or Plotutils. I have used Plotutils because of the better visualizing quality. To compare two different prints it is important to use the same scale in both plots. I prefer therefore absolute scaling on the axis:

```
./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
```

The client program "myclient" connects to the server "modell1" to get the data and provides it via the pipe to the Plotutils (the graph program). The scaling of the air pressure is between 950mbar..1040mbar. The time unit is 24 hours. If you change to option -T X to -T ps then a postscript files will be generated. This can then easily be printed on paper.



The diagram shows the end of a good weather period and the start of rain.

Installation

The installation is very easy. Check again the circuit and then connect it to the parallel port. The software can be unpacked with tar -zxvf druck-0.1.tar.gz. After that change to the source directory and modify the myclient.c for your height over sea level (HIGH_NN). Compile it by typing "make". Before you can use the module you should add a port number in /etc/services for the socket communication. Add the following line:

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

After that you can start the server process as root with ./druck LPT1 (or LPT2). If everything is OK then the server will print every 10 seconds the raw data of pressure and temperature together with the time. The client application can now be started anywhere in the same network (or on the same machine). The script druck.sh visualizes the data via the graph program from Plotutils. You have to edit the script for the correct server name. The entry in /etc/services has to be done on the machine where the server process is running and on the machine where the client runs.

Final remarks

It is annoying that manufactures often do not provide drivers for Linux. However this gave me the interesting opportunity to develop an own solution which I have presented here. Only the network client/server application made a satisfying solution since the data can now be read out from different machines. Still a few things are open. How can those values be provided via the Internet? If several people build up a weather station with maybe temperature, air pressure, humidity and wind speed (wind speed is not easy to measure) how can then the data be exchanged? Which data format should be used? Maybe somebody has an idea and this can be presented in a different article.

References

- Datasheet of the pressure sensor from Motorola: http://e-www.motorola.com/brdata/PDFDB/docs/MPX4100A.pdf, local copy: MPX4100A.pdf
- Source code of the software: download Seite für diesen Artikel
- Electronic distributor ELV: The kit from ELV (Articel-Nr.: 68-388-03)
- Datasheet of the pressure sensor from Intersema http://www.intersema.ch/pro/module/file/da5534.pdf, local copy: da5534.pdf
- Mini-PCs DIL
- Parallel port driver Parapin
- Linuxthreads PThreads
- Homepage Gnuplot
- Homepage Plotutils

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

2005-01-14, generated by lfparser_pdf version 2.51