BFO DPS310 barometer

Rudolf Widmer-Schnidrig

Black Forest Observatory, University of Stuttgart
Heubach 206, D-77709 Wolfach, Germany

Abstract

We describe an autonomously recording barometer which can resolve ambient atmospheric pressure variations for signal periods longer than 30 seconds, operates on a single Lithium D-cell battery for 6 months and maintains absolute time by synchronizing with the DCF77 radio clock signal available over much Europe. Its weight is 180 g and fits into a 500 ml twist top jar. The barometer should be well suited for research in meteorology and geophysics.

Keywords: Barometer, MEMS pressure sensor, atmospheric pressure, acoustic gravity waves, microbaroms.

1. Motivation – Design Goals

At the Black Forest Observatory (BFO) we want to capture the temporal and spatial variation of the atmospheric surface pressure as these variations constitute the dominant source of background noise in long-period seismic observations recorded by the seismometers, gravimeters, strain- and tiltmeters deep in the BFO mine. The idea is that by actually measuring the pressure fluctuations we know the forcing of the solid Earth and may be able to predict its elastic response and subsequently the atmosphere induced signal component in the seismic data. The period band of interest lies roughly between one hour and one minute. To capture the spatial pressure variation the deployment of a large number of continuously recording barometers is envisioned. To make this a feasible endeavor these barometers must on the one hand resolve the pressure fluctuations in the target frequency band and, on the other hand, must be affordable. With the availability of modern MEMS pressure sensors, both criteria can be met leading to the design and construction of the barometer presented in this document.

2. Trade-offs in the design

Considering that the environment around the Black Forest Observatory (BFO) is sparsely populated and covered with a dense forest we reject the option of powering the barometers by solar panels or from the utility power grid. Instead we chose a **battery powered design**.

Concerning the handling of the data we opted for decentralized off-line data storage and thereby do without real-time data. The reason to do so is both because of the lack of cell phone reception near BFO and because of power considerations. Wireless data transmission is a power hungry endeavor. Additionally it should be noted that at this early stage of the project we do not intend to implement a real-time correction of our seismic data. For a feasibility study, as we have it in mind, real-time data feeds are not needed. Thus every barometer shall be equipped with a **microSD mass storage** device. One big disadvantage of this choice is the loss of real-time state-of-health information. Hence it may happen that when visiting the barometers to collect the acquired data we find that the microSD card is empty.

For the selection of a time source we were faced with a choice between DCF77 [1] and GPS. While GPS offers a more precises time signal it also requires a clear view of the sky. In contrast **DCF77 radio clock** receivers can even operate below the ground: in fact they synchronize in the BFO mine at 100 m depth. Considering that we want to hide the barometers and not attract the attention of curious hikers, dogs or wild animals we opted for DCF77. Achievable time quality with DCF77 is better than 10 ms which is sufficient for our application, considering the roughly 10 m/s propagation speed of weather fronts and acoustic gravity waves.

3. The Infineon DPS310 MEMS pressure sensor

We have tested several MEMS based pressure sensors from Infineon and Bosch and found that the DPS310 has the lowest self-noise in our target frequency band. Figure 1 shows the result of a huddle test of three pressure sensors operated in the same room at BFO. They are the DPS310 MEMS sensor from Infineon, the DigiQuarz barometer 6016B from Parosci and the IFS3000 infrasound sensor from Hyperion. Between DC and 30 mHz (33 s period) the DPS310 MEMS sensor compares well against the more than 100 (!) times more expensive classical barometers. The excellent comparison with the Parosci at low frequency is also a testimony to the excellent temperature compensation implemented in the MEMS sensor. At high frequency ($f > 30$ mHz) both the Parosci and the DPS310 are limited by their self-noise while the PSD of the infrasound sensor keeps

Email address: `widmer@gis.uni-stuttgart.de` (Rudolf Widmer-Schnidrig)

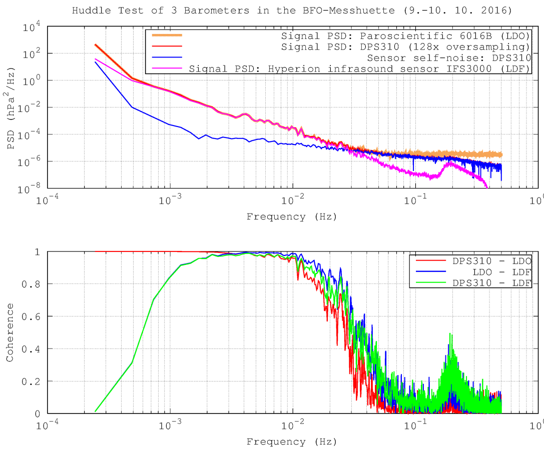

Figure 1: Huddle test of three pressure sensors in the BFO magnetics hut. The three sensors are the DigiQuarz 6016B barometer from Paroscientific, the IFS3000 infrasound sensor from Hyperion and the MEMS barometer DPS310 from Infineon. The upper panel shows the PSD of the three signals together with the sensor self-noise of the DPS310. The latter has been estimated with a three-channel correlation analysis [5]. The bottom panel shows the pairwise coherences. The coherence of sensor pairs involving the infrasound sensor falls off at low frequencies (f<2 mHz) as these are outside the pass-band of the infrasound sensor. Above 30 mHz both barometers are limited by sensor self-noise. While the infrasound sensor can resolve the microbarom peak at 0.2 Hz this signal only shows up in the barometers as a slightly increased coherency.

decreasing. Clearly, on quiet days at BFO neither the Parosci nor the DPS310 can resolve the background pressure fluctuations above 30 mHz. We will however show an example of a microbarom at 7 s period recorded with the DPS310 in the Black Forest during winter when microbaroms were strongly excited in the North Sea (see fig. 7). Considering that our target period band stops at 1 minute the DPS310 seems a perfect match for our requirements.

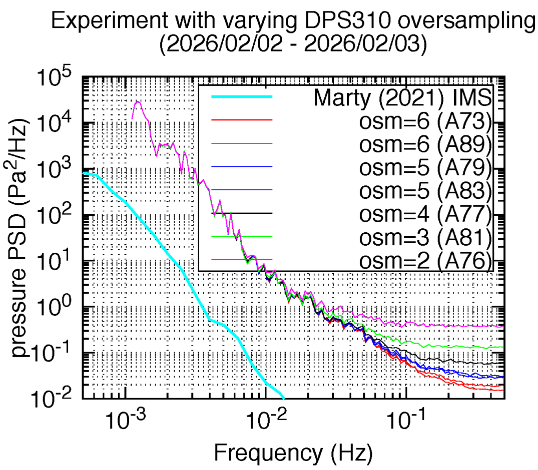

Figure 2: Huddle test of barometers with different amounts of oversamplig.

4. Oversampling in the DPS310

Internally the DPS310 can conduct repeated pressure measurements at a very high pace. To increase the pressure resolution the manufacturer offers oversampling by factors of 2^n where n is an integer number between 1 and 7. The price for doing so leads to increased power consumption and hence shorter autonomy time. Figure 2 illustrates the benefit of oversampling. The salient point is that only the high frequency signal components benefit from oversampling. This observation can be used to find the most suitable choice of operating parameters. If not stated otherwise we have used an oversampling of $n = 5$ for the figures in the writeup.

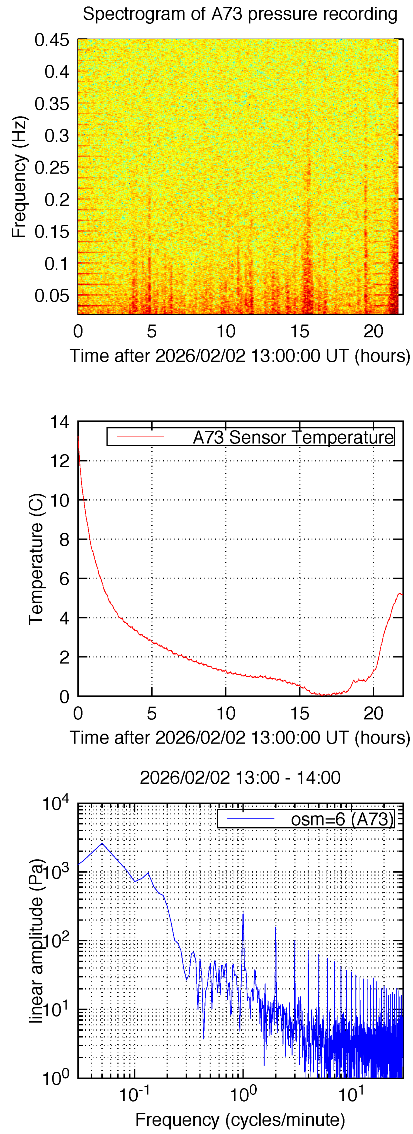

Figure 3: Effect of a rapid temperature variation on pressure recording. The top panel shows a spectrogram of a 22 hour long pressure recording. The middle panel shows the temperature variation in that same time interval. The bottom panel shows an amplitude spectrum of the first hour of data with a frequency axis in units of cycles per minute.

5. Temperature compensation in the DPS310

In some sense every barometer is also a thermometer. That is barometers react not only to pressure changes but also to changes in the temperature. For this reason the DPS310 contains also a thermometer to measure its temperature. Both pressure and temperature measurements support oversampling. For the same oversampling temperature and pressure measurements take equally long. Every raw pressure reading must be compensated for the temperature. Coefficients needed to compute thermally compensated and calibrated pressure value are stored inside the MEMS sensor. The firmware running on the MCU always uses the most recent temperature value to compute the temperature corrected pressure. The question then arises: how often should the temperature be measured? Every second? Every minute? We initially opted to take a temperature reading every minute. Is this good enough also at times when the sensor temperature changes rapidly (at sunrise)? A data example from an outdoor deployment without any thermal protection of the barometer is shown in figure 3. At the start of the recording the temperature rapidly decreases while it increases again towards the end. Simultaneously with these rapid variations harmonics are visible in the spectrogram. The bottom panel shows an amplitude spectrum of the first hour of data. The peaks occur exactly at integer multiples of 1 cycle/minute. Considering that the temperature measurements are taken once per minute we interpret the harmonics in the pressure spectrogram as a consequence of a suboptimal temperature correction of the pressure readings: towards the end of every minute the used temperature value is no longer valid since the ambient temperature has already changed. Is this an acceptable artifact in the data? Since our target frequency band is below 1 min^{-1} we could tolerate these peaks.

A more subtle temperature effect comes from the heating of the barometer due to activity of the SD drive as illustrated by the heat pulses in figure 4. Their occurrence at half hour intervals is a clear indication that they are caused by the writing to the SD drive. We rule out the possibility of electrical cross-talk because the SD drive is only powered up for less than 30 s while the shown temperature pulses last at least 3 minutes. So we likely see ohmic heating of the barometer board due to SD drive activity. Without a temperature compensation of the raw pressure readings these pulses would also show up in the pressure data. The sharp onset of the temperature pulses necessitates rapid sampling of the temperature or else these pulses will also show up in the pressure data. If our interpretation of the cause of these pulses is correct, then increasing the separation between SD drive and DPS310 chip on the PCB should reduce this effect. Unfortunately the two are close to each other with the current incarnation (see fig. 5). We have inspected the longest continuous pressure record that we have collected so far (21 days in length) for spectral lines at the frequency of the SD drive activity and could not find any spurious lines. So this effect may be too small to show up in the temperature corrected pressure data.

Ideally we would want to measure the temperature before each pressure reading and get rid of the above two issue entirely. This is how the Parosci DigiQuarz operates. For a battery powered device this may reduce the autonomy time too much.

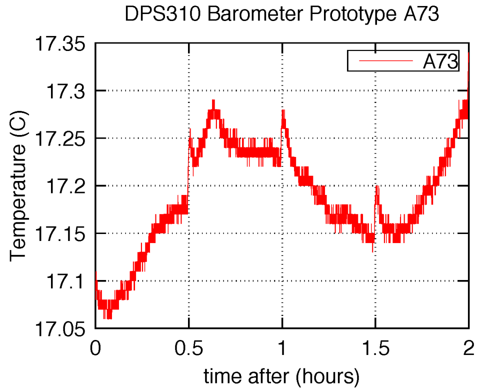

Figure 4: Temperature pulses sensed by the DPS310 barometer.

Increasing the frequency of the temperature measurements to once every 30 seconds will remove the peak at 16.66 mHz completely and the lowest harmonic will only appear at 33.3 mHz. Combining this with an effort to thermally shield the barometers and hence smooth out any temperature variation should further reduce these artifacts. We note that these artifacts are not strictly band-limited: at signal periods of the temperature variations we expect also a distortion of the pressure amplitudes.

We leave the final choice in this trade-off between autonomy time and pressure resolution to the user by introducing a "MODE" parameter with which a selection between six different parameter triplets can be made. One triplet consists of the pressure sample interval, the temperature sample interval and the DPS310 oversampling (see table 1).

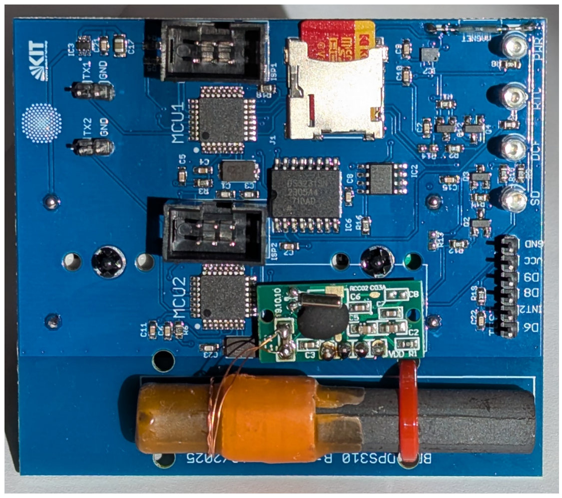

Figure 5: Top view of the barometer board. The DPS310 chip is IC7 to the right of the microSD drive.

6. Functional description of the barometer board

At the core of the board (figures 5 and 6) are two microcontrollers (MCU) ATMega328P: one dedicated to the decoding of the DCF77 radio signal and the other for the remaining tasks:

Figure 6: Bottom view of the barometer board with the D-Cell battery.

- every second: take pressure and/or temperature readings from the DPS310 pressure sensor,
- every second: get current time from the DS3231 RTC clock,
- every second: write time stamped pressure readings to the FRAM memory chip,
- every hour: dump the FRAM contents to the microSD memory card,
- every hour: monitor supply voltage. Only operate the microSD drive if supply voltage is high enough,
- once per day: wake up the DCF-MCU for a renewed synchronization of the RTC with DCF time.

To monitor the activity of the board four LEDs and TX pins are available. The LEDs are labeled ”PWR”, ”PPS”, ”DCF” and ”SD”. The PWR LED indicates that the board is powered. The PPS LED shows the 1PPS signal generated by the DS3231 RTC clock: it is 500 ms on and 500 ms off. The PPS signal is used to wake up MCU1 once per second for the pressure reading. While MCU2 is awake and trying to synchronize with the 77.5 kHz DCF signal, the DCF LED shows the DCF code. This code consists of short and long pulses of 100 and 200 ms duration, respectively. They start at the beginning of each second. Once synchronization is achieved and the RTC clock is reset to the current DCF time MCU2 powers itself down and the DCF LED stops blinking. The fourth LED labeled SD shows the writing activity of the microSD drive. To save power all LEDs are only operating if a magnetic Reed switch is activated: hence one has to place a permanent (Neodymium) magnet within a few centimeters of the Reed switch before the LEDs light up. The Reed switch is in the little glass tube next to the PWR LED.

Detailed information about the activity of the board can be obtained by connecting to the UART serial output of the two MCUs via test pins TX1 or TX2. The **serial line parameters are set to 115200 baud and 8N1**. Using a FT232RL FTDI Serial to USB adapter module a PC can be connected to the board to listed to the two MCUs. On TX2 the DCF77 synchronization can be monitored. On TX1 the time stamped pressure and temperature readings are displayed. Both MCUs are also quite verbose during their boot process. On a Linux PC assuming that the FTDI module attached to the USB port is linked to the device /dev/ttyUSB0 and assuming that the user is a member of the group ”dialout” the following command should bring the output of either MCU to the screen

```
screen /dev/ttyUSB0 115200
```

The board does not have a power switch. When the battery is inserted the board starts by first synchronizing with DCF77 and once that is achieved the recording of pressure starts.

Reboot of the board To reboot the board, either quickly remove the battery or alternatively short the test pins VCC and GND.

The test pins labeled D6, INT2, D8 and D9 are digital lines connecting the two MCUs. INT2 is the pin with which MCU1 can wake up MCU2, which happens either only at midnight or at 8 hour intervals. (The interval is hard coded in the firmware).

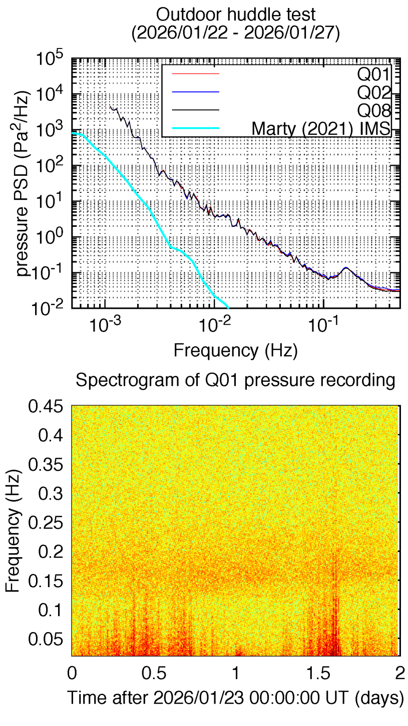

Figure 7: In winter atmospheric conditions in central Europe are favorable for the observation of microbaroms. The figure shows the result of a huddle test with three barometers: the microbarom peak is clearly present in the power spectral densities. It shows up as a diffuse horizontal band centered around 0.15 Hz in the spectrogram.

7. Operating instructions

Power: A Lithium-Thionyl-Chlorid (Li-SOCI2) D-Cell 3.6V SAFT 33600 is recommended. The average current drawn is 3.4 mA. Once per hour the SD drive is activated resulting in peak currents of 50 mA for approx. 10 seconds. These values should help with the selection of alternative batteries. The SAFT 33600 has an advertised capacity of 17000 mAh which translates to an autonomy time of 17000/3.4 = 5000 hrs = 208 days (This assumes an oversampling of $n = 5$).

Mass storage: Data are written in hourly files to the microSD card. One such file has 70kB in size which translates to 50MB/-month and 613MB/year. Compared with the above mentioned autonomy time it becomes clear that 1 GB or 2 GB microSD card is more than enough for one deployment. The firmware is linked with the Arduino SdFat library and supports FAT16, FAT32, and exFAT file systems on standard SD, SDHC, and SDXC cards. I have experimented with many brands of SD cards from different sources and found that writing to an microSD card from an 8MHz microcontroller is not without pit falls. In the end I found one brand of cards that never let me down: ”INTENSO 2 GB”. The KODAK 32 GB painted in yellow/red, not yellow/black, from AliExpress was also reliably performing. I tested only three 4 GB SanDisk cards and they worked. None of the cheap brands sold on the AliExpress platform worked for me. I suspect that the larger and more modern SD cards simply draw more power than the barometer board can provide. In fact, to reach stable performance I had to make sure that the SDcard is only accessed while the MCU2 dedicated for DCF77 decoding is asleep.

Yet another pit fall concerns the microSD formatting: 2GB uSD cards can be formatted both with FAT16 and FAT32. However the barometer firmware fails if it encounters FAT32 on a 2GB microSD card while FAT32 on a 4GB or larger microSD card is fine.

preparing the microSD card: the barometer expects to find one directory on SD card: DPS310. All hourly files are written to that directory. The barometer will fail to record data if it cannot find that directory.

Barometer configuration via files on the microSD card: A three letter station code can be provided in a text file ”STATION.TXT” located in the root directory of the microSD card. Everything beyond the first three characters of that file is ignored. This allows us to rename the barometer after every redeployment to a new location. If no name is provided on the microSD card then the station is named ”XXX”. This is not a big issue because all barometers have also a unique 8 digit serial number (permanently stored in the FRAM) which is also contained in the header of each data file. However, without an individual station code the hourly file names from different barometers are no longer unique. So beware!

A limited choice of operating modes of the barometer can be selected via the MODE parameter provided in the file ”MODE.TXT”. If no such file is present in the root directory of the microSD card then the default mode 2 will be selected. Allowed combinations of parameter values are given in table 1.

Data file structure - Header section: The hourly files are plain ASCII files with a 9 line header (see below) followed by the data with one data point per record. The file names are composed of a time stamp followed by the station name like this: **26B20G30.A89**. Here 26B20G30 is made up like YYMDDHmm where YY is the year 2026, M is the month with ”A” for January, ”B” for February etc. 20 is the day of the month: thus 2026/02/20. H stands for the hour with ”A” for ”00”, ”B” for ”01” and ”X” for ”23”. Finally ”mm” stands for the minute. After the dot is the three letter station code ”Q08”. Note that the file name is generated using the time when the FRAM is being written: so it matches the time of the last sample in the file, not the first.

mode	osm	Δt(p) (s)	Δt(T) (s)	Δt(FRAM) (min)	I_{AVG} (mA)	τ (days)
1	5	1	60	60	2.93	241
2	5	1	30	60	2.93	241
3	5	2	2	60	2.92	242
4	6	2	2	60	3.30	214
5	4	1	1	30	2.92	242
6	5	1	1	30	3.30	214

Table 1: The six operating modes of the barometer: OSM is the oversampling in the DSP310, Δt(p) the sample interval for the pressure, Δt(T) the sample interval for the temperature and Δt(FRAM) is the time between writing to the microSD card. The average current, I_{AVG}, consumed by the barometer board and the estimated autonomy time, τ, assuming a 17Ah battery are given in the last two columns. For modes 3 and 4 pressure and temperature are taken on the even and odd seconds, respectively. The modest influence of the oversampling on the total power consumption is due to the fact that the board consumes about 2.4 mA in standby mode.

```
File name: DPS310/26B20G30.A89
Station name: A89
Barometer SN: 26010089
Firmware compiled on: Feb 19 2026 10:45:29
Supply voltage: 3414 mV
MODE parameter (1...7): 5
DPS310 oversampling (1...7): 4
1771567681 latest synchronization with DCF: 2026-2-20 6:8:1 UT
1771567681  RTC offset: -28 ms
1771567250 960350
1771567250 1436
1771567251 960348
1771567251 1437
```

Figure 8: Header lines of a typical data file followed by the first four data samples and their UNIX time stamps. With the operating mode set to MODE=5 the oversampling is OSM=4 and every second both a pressure and temperature reading is taken (see table 1).

Data file structure - Data section:

The data section contains the records with the time stamped pressure and temperature values. Every data record consists of one line that begins with a UNIX time stamp. These UNIX time stamps count the number of seconds since January 1, 1970. On a Linux system the below command converts the UNIX time 1770112835 to a human readable date: 2026/02/03 10:00:35

```
date -u --date="@1770112835" +'%Y/%m/%d %H:%M:%S'
```

The time stamps are followed by either a pressure and temperature values and are given as integer numbers: the pressure in tenths of Pascals, the temperature in one hundredths of a degree Celsius. The magnitude of the numbers permits to unambiguously identify if its a pressure or a temperature value. Pressure values, p, lie strictly in the range $800000 < p < 1100000$ while

Figure 9: Naked and field ready version of the barometer. A slightly modified 500 ml twist-top PCB container from Packari is used. To let the ambient pressure fluctuations enter the housing unattenuated six 0.5 mm diameter holes have been drilled into the container near the threading for the lid. Hence the holes are hidden from view in this picture.

temperatures samples, T, lie in the range $-4000 < T < 8000$.

8. RTC clock drift

Between synchronizations with DCF77 the RTC chip DS3231 keeps track of time and date. The drift of the RTC time away from DCF77 is estimated with every synchronization to DCF77. This is documented as RTC offset in the file header. A value of 999999 indicates that no synchronization occurred in the time span covered by the current data file. The RTC offset is also written into the data section in units of micro seconds. This way it cannot be mistaken for a pressure or temperature value.

9. Supply voltage

The microcontroller has an internal reference voltage which can be used to measure the voltage supplied by the battery. If the voltage drops below 3.0 Volts the firmware will no longer attempt to access the microSD drive. This is a safety measure in order to not corrupt the SD file system.

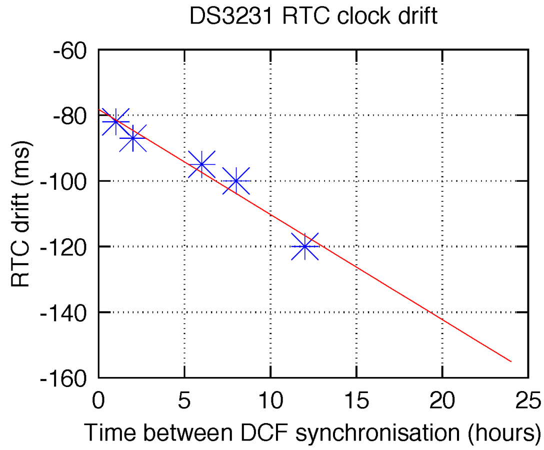

Figure 10: Drift of DS3231 RTC clock relative to DCF77 estimated with five separate barometers in an indoor huddle test. The RTC drift is expected to depend on temperature and on temperature variations. In the shown example the drift is -3.2 ms/hour or 1ppm which is well within the specifications of the DS3231. In view of this clock drift we chose to synchronize the RTC at midnight every 24 hours.

10. Deployment

The housing of the barometer needs special care: it shall protect the board from humidity while at the same time it shall not attenuate the ambient pressure fluctuations. We have chosen (figure 9) a layered combination of water tight plastic bags and a rigid PCB food container with small holes drilled into it. To close the plastic bag we use plastic locking clamps. Thus the barometer board is placed inside a sealed plastic bag which in turn is placed into the food container. For additional protection we place the food container in yet another bag. This latter bag may be a dog poop bag to minimize interest from curious eyes. To keep the PCB board dry we include a tea bag with some desiccant.

11. Acknowledgements

Early help for this project came from Karl-Heinz Jäckel (GFZ Potsdam) back in 2016 when the whole project still ran on a bread board with two Arduino Pro Mini. Udo Klein for publishing the noise resilient DCF77 decoder library [4]. Petr Foltyn from SolidusTech [2] suggested many essential power saving modifications and never lost the patience to explain electronics to a seismologists. Thank you, Petr! Luca Spinner patiently taught a FORTRAN dinosaur how to program in C. Infineon for publishing an Arduino compatible library to access their DPS310 [3]. Last but not least I thank my colleagues at BFO for technical support and insightful feedback: Peter Duffner, Thomas Forbriger and Walter Zürn.

References

[1] Physikalisch Technische Bundesanstalt (PTB) Braunschweig. DCF77 reach. `https://www.ptb.de/cms/en/ptb/fachabteilungen/abt4/fb-44/`

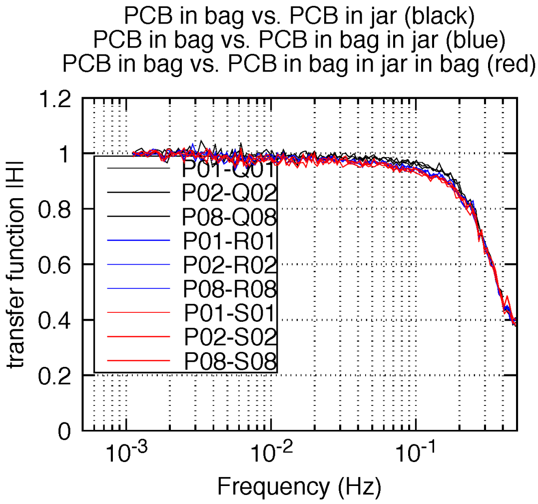

Figure 11: We have experimentally verified that the many layers that make up the barometer housing do not attenuate or degrade the signal quality in any way. The figure shows the result of a huddle test with barometers that have one, two and three layers: barometer in a plastic bag, barometer in a plastic bag in a PET jar, barometer in a plastic bag in a PET jar in a dog poop bag. A slight signal attenuation is notable above 100 mHz.

`ag-442/dissemination-of-legal-time/dcf77/reach-of-dcf77.html`, 2026.

[2] Petr Foltyn. Solidustech. `https://www.solidustech.com`, 2026.

[3] Infineon. DPS310 Arduino library. `https://github.com/Infineon/DPS310-Pressure-Sensor`, 2026.

[4] Udo Klein. The noise resilient DCF77 decoder library. `https://github.com/udoklein/dcf77`, 2026.

[5] Reinoud Sleeman, Arie van Wettum, and Jeannot Trampert. Three-channel correlation analysis: A new technique to measure instrumental noise of digitizers and seismic sensors. *Bulletin of the Seismological Society of America*, 96(1):258–271, 02 2006.