

# Bau eines barometrischen Höhenmessers

The image is a composite graphic illustrating the construction of a barometric altimeter. It features a central photograph of a custom-built circuit board with various components like a microcontroller, capacitors, and a pressure sensor. Overlaid on this are several technical elements:

- Schematic Diagram:** A detailed circuit diagram showing the electrical connections between components. It includes a pressure sensor (P1, 20k), resistors (R3, R4, R5, R6, R7, R8, R9), an operational amplifier (IC1, LM358), and a microcontroller (IC2, ATmega16). The sensor is connected to a voltage divider network.
- Graph:** A line graph with a vertical axis representing altitude in meters, ranging from 100 to 550. The graph shows two data series: a solid red line and a dashed green line, both showing a positive correlation between altitude and sensor output.
- Code Snippets:** Several fragments of C code are scattered around the board and graph, including:

```
uart_init();  
sei();  
leds_set_busy();  
sleep_ms(200);  
leds_unset_busy();  
uart_puts_P(PSTR(">> Press any key to start calibration\n"));  
while(cal_active == 0 && i <= 20) {  
  if(uart_getc() != -1) {  
    cal_active = 1;  
  }  
}
```

Autor: Sebastian Schumb

Betreuender Lehrer: Herr Heinemeyer

## **Inhaltsverzeichnis**

|  |    |
|--|----|
| 1. Einleitung.....                       | 3  |
| 2. Theoretische Überlegungen.....        | 4  |
| 2.1. Luftdruck.....                      | 4  |
| 2.2. Die Internationale Höhenformel..... | 6  |
| 3. Praktische Umsetzung.....             | 11 |
| 3.1. Der Drucksensor.....                | 15 |
| 3.2. Der Messverstärker.....             | 15 |
| 3.3. Das Programm.....                   | 17 |
| 4. Test.....                             | 19 |
| 5. Fazit.....                            | 21 |
| 6. Quellenverzeichnis.....               | 22 |
| 6.1. Luftdruck und Höhenformel.....      | 22 |
| 6.2. Elektronik.....                     | 22 |
| 6.3. Programmierung.....                 | 23 |
| 7. Anhang.....                           | 24 |
| 7.1. Quelltextauszüge.....               | 24 |
| 7.2. Bilder.....                         | 28 |
| 7.3. Platinenlayouts.....                | 32 |

# 1. Einleitung

Niels Bohr soll während einer Physikprüfung einmal die Frage gestellt worden sein, wie man mit einem Barometer die Höhe eines Gebäudes bestimmen könnte.

Nach kurzer Überlegung antwortete er, man solle das Barometer an einer Schnur vom Dach des Gebäudes herunter lassen. Die Länge der Schnur und die Höhe des Barometers ergäben die Höhe des Gebäudes.

Irritiert von dieser unorthodoxen Antwort, beschlossen die Prüfer Bohr durchfallen zu lassen. Allerdings protestierte dieser unter Berufung darauf, dass seine Antwort im Grunde korrekt war und erhielt schließlich eine zweite Chance eine Antwort zu geben, die zumindest die Kenntnis von physikalischen Zusammenhängen nachweist.

Bohr überlegte kurz und nannte schließlich mehrere Methoden :

Als erstes schlug er vor das Barometer vom Dach fallen zu lassen und über die Fallzeit die Höhe zu berechnen.

Sein zweiter Vorschlag war aus den Längen der Schatten des Gebäudes und des Barometers, sowie dessen Höhe, die Höhe des Gebäudes zu errechnen.

Als dritte Möglichkeit schlug er vor das Barometer als Pendel zu benutzen und den Unterschied der gravitativen Wiederherstellungskraft zwischen dem Boden und dem Dach des Gebäudes zu bestimmen. Über diese Differenz kann anschließend die Höhe berechnet werden.

Er schlug auch vor das Barometer als Zollstock zu benutzen und die Höhe durch Abmessen zu berechnen.

An fünfter Stelle führte er die schlichte und orthodoxe Lösung an, die Höhe über den Druckunterschied zu bestimmen.

Gleich danach fügte er aber hinzu, dass man die gewünschte Information auch über den Hausmeister beziehen könnte, wenn man ihm im Gegenzug das Barometer überlässt.

Diese Anekdote hat sich leider so niemals zugetragen und gehört ins Reich der modernen Mythen.

Trotzdem soll sie als Einleitung zu dieser Arbeit dienen, in der es im Prinzip genau um die genannte Fragestellung geht.

Von den von Bohr vorgeschlagenen Methoden, lassen sich nur ein paar praktisch anwenden, wenn man dann noch den Verschleiß an Barometern gering halten und möglichst genaue Ergebnisse erzielen will, bleibt eigentlich nur noch der fünfte Vorschlag übrig :

Das Errechnen der Höhe über den Luftdruck.

Zielsetzung dieser Arbeit soll es sein, den Prototyp eines elektronischen Höhenmessers zu entwickeln, der den Luftdruck über einen geeigneten Sensor messen, die Höhe über dem Meeresspiegel errechnen und diesen Wert anschließend speichern kann.

Dazu werde ich zuerst die nötigen physikalischen Überlegungen zum Phänomen Luftdruck vorstellen, anschließend das Konzept und die Umsetzung meines Höhenmessers erklären und dessen Test, sowie die Ergebnisse erläutern.

## 2.Theoretische Überlegungen

### 2.1.Luftdruck

Die physikalische Größe Druck beschreibt im Prinzip eine Kraft auf eine Fläche.

Ein Pascal ist zum Beispiel der Druck, den eine Kraft von einem Newton auf die Fläche von einem Quadratmeter ausübt. Der Luftdruck auf eine Fläche, den wir in der Atmosphäre wahrnehmen können, resultiert aus der Gewichtskraft der Luftsäule über der Fläche.

Das Phänomen lässt sich mit dem Schweredruck im Wasser vergleichen. Allerdings ist Wasser nicht kompressibel, das heißt die Dichte einer Wassersäule, mit einer einheitlichen Temperatur, ist überall gleich. Daraus folgt das die Masse des Wassers über die Länge der Säule gleich verteilt ist. Deshalb ist die Gewichtskraft und damit auch der Druck auf die Grundfläche der Säule direkt proportional zur Höhe der Säule. Bei der doppelten Höhe hat die Säule die doppelte Masse und es herrscht der doppelte Druck auf die Grundfläche.

Luft ,im Gegensatz dazu, ist kompressibel. Setzt man Luft unter Druck kann man ihre Dichte bis zu einem gewissen Grad erhöhen.Das führt dazu, dass in einer Luftsäule die Masse nicht gleich verteilt ist. Je weiter man in der Luftsäule nach unten geht, desto höher wird der Druck und damit auch die Dichte der Luft an dieser Stelle. Deshalb ist der Druckverlauf in einer Luftsäule nicht linear sondern viel mehr exponentiell.

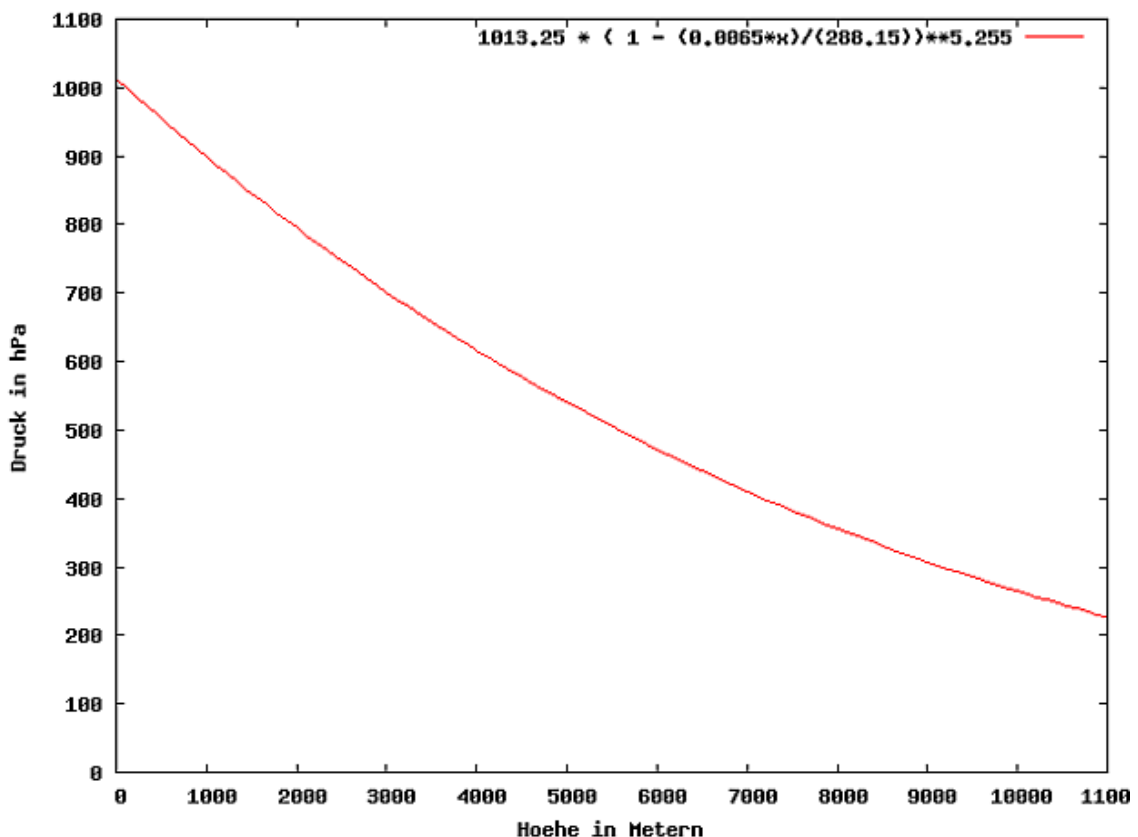


Abbildung 1: Druckverlauf bis 11km Höhe

Des weiteren kommt hinzu, dass die Temperatur in der Atmosphäre mit zunehmendem Abstand zur Erde abnimmt, was die Dichte noch einmal zusätzlich beeinflusst.

Auch durch Winde und Sonneneinstrahlung können Hoch und Tiefdruckgebiete entstehen, wo durch auf der selben Höhe zwei unterschiedliche Drücke herrschen können.

Im ersten Fall reißt ein sogenannter Jetstream, also ein Höhenwind, beim Beschleunigen Luftmassen aus seiner Umgebung mit, so dass ein sogenanntes dynamisches Tief entsteht, oder der Jetstream wird durch die Landschaftsform, z.B. ein Gebirge abgebremst, wobei Luftmassen nach unten und zur Seite gedrückt werden. Es entsteht ein sogenanntes dynamisches Hoch.

Im Fall von Sonneneinstrahlung erhitzt sich der Boden und die drüber liegenden Luftmassen, welche aufsteigen. Man spricht von einem thermischen Tief. Kommen die warmen Luftmassen über kälteres Gebiet kühlen sie aus und sinken ab. Der Luftdruck am Boden erhöht sich. In diesem Fall spricht man von einem thermischen Hoch.

Diese Meteorologischen Effekte können zwar eine Höhenmessung über den Luftdruck verfälschen, allerdings verfügen Hoch und Tiefdruckgebiete meist über eine Ausdehnung im mehrstelligen Kilometerbereich und bewegen sich relativ langsam. Es ist also unwahrscheinlich bei einer Höhenmessung von einem Hoch in ein Tiefdruckgebiet zu geraten oder umgekehrt. Allerdings kann dieser Fall auch nicht unbedingt ausgeschlossen werden. Deshalb finden reine barometrische Höhenmesser heute meistens dort Anwendung wo keine großen Strecken während der Messung zurückgelegt werden. Typische Beispiele wären das Bergsteigen, Wandern, oder der Modellflug. Aber auch in kleineren Flugzeugen, zum Beispiel beim Segelflug, werden noch einfache barometrische Höhenmesser eingesetzt, weil dort die Abweichungen akzeptabel sind.

Es gibt allerdings auch zwei Strategien die Abweichungen zu korrigieren.

Man kann entweder den Höhenmesser an einem Punkt bekannter Höhe eichen. Das geschieht, indem die Parameter in der internationalen Höhenformel, auf der die meisten Höhenmesser aufbauen, den zur Zeit herrschenden Bedingungen anpasst werden.

Falls das nicht möglich ist, weil man zum Beispiel eine benötigte Größe aus technischen Gründen nicht messen kann, kann man mit dem ungeeichten Höhenmesser die Höhendifferenz zu diesem Punkt messen und mit der Höhe des Punktes, die richtigen Werte berechnen.

## 2.2. Die Internationale Höhenformel

Um aus dem Luftdruck die Höhe über dem Meeresspiegel berechnen zu können, wird eine Funktion benötigt, die dem Druck die Höhe zuordnet oder umgekehrt.

Dafür müssen mehrere Einflüsse berücksichtigt werden.

Zum Beispiel die schon angesprochene Verteilung der Masse in der Luftsäule und die Temperaturabnahme mit zunehmender Höhe, die die Dichte beeinflussen.

Um diese Anforderungen zu erfüllen wurde die internationale Höhenformel aufgestellt.

Der Ansatz zur Herleitung dieser Formel betrachtet zunächst ein kubistisches Luftpaket mit der Grundfläche  $A$  und der möglichst kleinen Höhe  $\Delta h$  in einer Luftsäule.

Es wird angenommen, dass nur in vertikaler Richtung Kräfte auf das Luftpaket wirken.

Auf die Unterseite des Luftpakets wirken die Gewichtskraft und der Luftdruck der unter dem Paket herrscht. Die Gewichtskraft  $F_G$  kann man über das Volumen  $V$ , die Dichte  $\rho$  und die Erdbeschleunigung errechnen.

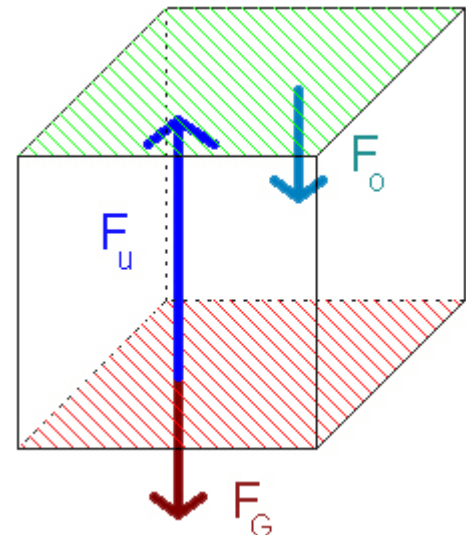


Abbildung 2: Kräfte auf das Luftpaket

$$F_G = V \cdot \rho \cdot g = \Delta h \cdot A \cdot \rho \cdot g$$

Die Kraft  $F_u$  die aus dem Luftdruck  $P_u$  auf die Unterseite resultiert, lässt sich über den Druck und die Fläche der Unterseite, der Grundfläche  $A$  also, ausdrücken.

$$F_u = A \cdot P_u$$

Auf die Oberseite des Luftpakets wirkt der Schweredruck  $P_o$ , der über dem Paket liegenden Luftsäule. Die daraus resultierende Kraft  $F_o$  kann man analog ausdrücken.

$$F_o = A \cdot P_o$$

Jetzt geht man davon aus, dass sich das Luftpaket nicht bewegt, das heißt, dass die resultierende Kraft Null ist, also der Druck auf die Unterseite die Gewichtskraft und den Druck von oben ausgleicht.

$$\begin{aligned}F_u &= F_G + F_o \\A \cdot P_u &= A \cdot \Delta h \cdot \rho \cdot g + A \cdot P_o \\A \cdot P_u &= A \cdot (\Delta h \cdot \rho \cdot g + P_o)\end{aligned}$$

Jetzt kann die Grundfläche, die nicht Null sein sollte, eliminiert und  $P_o$  über  $P_u$  ausgedrückt werden.

$$\begin{aligned}P_u &= \Delta h \cdot \rho \cdot d + P_o \\P_u &= \Delta h \cdot \rho \cdot g + P_u + \Delta P \\0 &= \Delta h \cdot \rho \cdot g + \Delta P \\ \Delta P &= -\Delta h \cdot \rho \cdot g\end{aligned}$$

$$\frac{\Delta P}{\Delta h} = -\rho \cdot g$$

Diese Gleichung beschreibt den Zusammenhang zwischen Dichte, Erdbeschleunigung, der Höhendifferenz und der Druckdifferenz. Allerdings ist wie eingangs schon erwähnt die Dichte vom Druck abhängig. Das lässt sich über das ideale Gasgesetz nach Boyle-Mariotte ausdrücken.

$$\rho = \frac{P \cdot M}{R \cdot T}$$

Rho steht dabei wie oben für die Dichte,  $P$  für den Druck,  $M$  für die molare Masse der Gase in der Atmosphäre,  $R$  für die universelle Gaskonstante und  $T$  für die Temperatur in Kelvin.

Nach dem Einsetzen ergibt sich :

$$\begin{aligned}\frac{\Delta P}{\Delta h} &= -\frac{P \cdot M}{R \cdot T} \cdot g \\ \frac{\Delta p}{P} &= -\frac{M}{R \cdot T} \cdot \Delta h \cdot g\end{aligned}$$

Wählt man jetzt  $\Delta h$  minimal klein, kann man die Verhältnisse in diesem Teil der Luftsäule näherungsweise bestimmen. Will man nun über größere Höhendifferenzen, zum Beispiel von  $h_0$  bis  $h_1$ , die Druckdifferenz bestimmen, muss man viele dieser kleinen Abschnitte übereinander legen. Mathematisch kann man das ausdrücken, in dem man die rechte und linke Seite der Gleichung für jeden Abschnitt löst und aufaddiert, oder über beide Seiten integriert, was im Prinzip das selbe ist.

$$\int_{P(h_0)}^{P(h_1)} \frac{dP}{P} = \int_{h_0}^{h_1} \frac{-M}{R \cdot T} \cdot g \cdot dh$$
$$\int_{P(h_0)}^{P(h_1)} \frac{dP}{P} = -\frac{M}{R \cdot T} \cdot g \cdot \int_{h_0}^{h_1} dh$$

Die beiden Integrale lassen sich unter Anwendung von Integrationsgesetzen lösen.

$$\ln\left(\frac{p(h_1)}{p(h_2)}\right) = -\frac{M}{R \cdot T} \cdot g \cdot (h_1 - h_0)$$
$$\ln\left(\frac{p(h_1)}{p(h_2)}\right) = -\frac{M}{R \cdot T} \cdot g \cdot \Delta h_g$$

Jetzt hat man eine Abhängigkeit zwischen dem Druck auf der Höhe  $h_0$  und der Höhe  $h_1$ . Zu beachten ist noch, dass  $\Delta h_g$  in diesem Fall nicht, wie  $\Delta h$  anfangs die Höhe eines Abschnitts, sondern die Differenz der beiden betrachteten Höhen darstellt. In eine andere Form gebracht, erhält man die Funktion  $p(h_i)$  die den Druck auf der Höhe  $h_i$  mit den Konstanten  $M$ ,  $R$ ,  $T$  und dem Druck auf der Höhe  $h_0$  darstellt.

$$\frac{p(h_1)}{p(h_0)} = e^{\frac{-M}{R \cdot T} \cdot g \cdot \Delta h_g}$$
$$p(h_1) = p(h_0) e^{\frac{-M}{R \cdot T} \cdot g \cdot \Delta h_g}$$

Diese Funktion findet sich als barometrische Höhenformel in einführender Literatur und im Schulunterricht. In dieser einfachen Form wären jetzt schon Messungen einer Höhendifferenz möglich, allerdings nur über kurze Distanzen. Die Funktion bildet zwar die exponentielle Abnahme des Drucks mit zunehmender Höhe näherungsweise nach, berücksichtigt aber nicht, dass die Temperatur mit zunehmender Höhe abnimmt. Sie nimmt eine isotherme Atmosphäre an. Der Abkühlungseffekt ist über kurze Distanzen zwar vernachlässigbar, aber bereits ab 1000m oder 2000m können erste Abweichungen auftreten.

Der einfachste Ansatz um die Temperatur mit einzubeziehen sieht eine lineare Abnahme der Temperatur mit zunehmender Höhe vor. Für den Bereich der Troposphäre, also bis 11km Höhe ist dies meist eine relativ gute Näherung.



$$T(h) = T(h_0) - a \cdot (h - h_0)$$

Der Parameter  $a$  in dieser Funktion stellt den sogenannten Temperaturgradienten dar. Er gibt als Proportionalitätsfaktor die Temperaturänderung pro Höhenänderung an.  $T(h)$  wird als  $T$  in die einfache Version der Höhenformel eingesetzt.

$$\int_{P(h_0)}^{P(h_1)} \frac{dP}{P} = \int_{h_0}^{h_1} -\frac{M}{R \cdot (T(h_0) - a \cdot (h - h_0))} \cdot g \cdot dh$$

$$\int_{P(h_0)}^{P(h_1)} \frac{dP}{P} = -\frac{M}{R} \cdot g \cdot \int_{h_0}^{h_1} \frac{dh}{T(h_0) - a \cdot (h - h_0)}$$

Da  $T$  nun nicht mehr konstant ist, kann es nicht mehr vor das Integral gezogen werden.

Mit  $\int \frac{1}{b-ax} dx = -\frac{1}{a} \cdot \ln(b-ax)$  lässt sich das Integral auflösen.

$$\ln\left(\frac{P(h_1)}{P(h_2)}\right) = \frac{M}{R} \cdot g \cdot \frac{1}{a} \cdot \ln\left(\frac{T(h_0) - a \cdot (h_1 - h_0)}{T(h_0)}\right)$$

Nun kann man wieder durch Umstellen, die barometrische Höhenformel erhalten.

$$P(h_1) = P(h_0) \cdot e^{\frac{M}{R} \cdot g \cdot \frac{1}{a} \ln\left(1 - \frac{a \cdot \Delta h}{T(H_0)}\right)}$$

Das sich der Logarithmus und die e-Funktion gegenseitig aufheben kann man die Funktion noch vereinfachen.

$$P(h_1) = P(h_0) \cdot \left(1 - \frac{a \cdot \Delta h}{T(h_0)}\right)^{\frac{M}{R \cdot a} \cdot g}$$

Nimmt man für  $h_0$  Meereshöhe an, und setzt die restlichen Parameter auf die Werte der Standardatmosphäre :  $P(h_0) = 1013,25 \text{ hPa}$ ;  $a = 0,0065$ ;  $T(h_0) = 288 \text{ K}$ ;  $M = 0,02896 \text{ kg mol}^{-1}$ ;  $R = 8,314 \text{ J K}^{-1} \text{ mol}^{-1}$  erhält man die Höhenformel, die unter anderem in der Fliegerei genutzt wird.

$$P(h) = 1013,25 \text{ hPa} \cdot \left( 1 - \frac{0,0065 \frac{\text{K}}{\text{m}} \cdot h}{288,15 \text{ K}} \right)^{\frac{0,02896 \text{ kg}}{8,314 \frac{1}{\text{K mol}} \cdot 0,0065 \frac{\text{K}}{\text{m}}}}$$

Diese Funktion, die der Höhe einen Druck zuordnet kann man jetzt einfach umstellen um zu einem gemessenen Luftdruck die Höhe zuordnen zu können.

$$h(P) = \left[ - \left( \frac{P}{1013,25 \text{ hPa}} \right)^{\frac{8,314 \frac{1}{\text{K mol}} \cdot 0,0065 \frac{\text{K}}{\text{m}}}{0,02896 \text{ kg}}} + 1 \right] \cdot \frac{288,15 \text{ K}}{0,0065 \frac{\text{K}}{\text{m}}}$$

Oder vereinfacht :

$$h(P) = \left[ - \left( \frac{P}{1013,25} \right)^{\frac{1}{5,255}} + 1 \right] \cdot \frac{288,15}{0,0065} \text{ m}$$

Mit dieser Formel lässt sich jetzt zu einem gemessenen Druck eine Höhe über dem Meeresspiegel zu ordnen, ohne das weitere Größen gemessen werden müssen. Sie gilt in der gesamten Troposphäre, also in der Schicht der Atmosphäre in der sich das Wettergeschehen abspielt. Die Troposphäre erstreckt sich von Meeresniveau bis 11km Höhe.

### 3. Praktische Umsetzung

Zu Beginn der praktischen Umsetzung des Höhenmessers stand die Suche nach einem geeigneten Drucksensor.

Nach einer kurzen Internetrecherche wurde der MPX4115A von Freescale bzw. Motorola gefunden. Dieser Sensor stammt aus der Automatisierungstechnik, deckt einen mehr als ausreichend großen Messbereich ab, ist einfach anzusteuern und gleichzeitig vergleichsweise billig verfügbar.

Zur Auswertung der Druckdaten sollte ein Mikroprozessor benutzt werden.

Dabei fiel die Wahl auf den ATmega32 von Atmel. Dieser Prozessor verfügt über einen integrierten 10bit breiten AD-Wandler, mit dem das Analoge Signal des Sensor zur weiteren Verarbeitung digitalisiert werden kann und genug RAM für komplexere Berechnungen. Außerdem beziehen sich die meisten Programmbeispiele die AVR bereitstellt auf diesen Chip, was ihn relativ benutzerfreundlich macht.

An den Prozessor sind zunächst 2 Leds angeschlossen, eine gelbe und eine rote, die den Programmstatus wiedergeben. Gelb steht für beschäftigt, sie leuchtet zum Beispiel wenn gerade ein Wert gemessen wird. Rot steht für Fehler sie leuchtet immer, wenn der Prozessor auf einen Fehler trifft, wegen dem das Programm abgebrochen werden muss.

Zur Energieversorgung wurde ein Lithiumpolymerakumulator eingeplant. Diese

Akkumulatoren liefern relativ lang eine hohe Spannung, die erst kurz bevor der Akkumulator komplett entladen ist, einbricht. Sie sind damit ideal um eine konstante Spannungsversorgung der Schaltung sicher zu stellen. Ein Akkumulator mit 2 Zellen hat eine Nennspannung von 7,3-7,4V. Der Mikroprozessor und der Sensor benötigen aber nur 5V. Deshalb wurde ein 7805-Längsregler benutzt, der stabile 5V bereitstellt. Eine grüne Led direkt am Ausgang des Reglers zeigt an, dass dieser korrekt funktioniert.

Nach ersten groben Überschlagsrechnungen war schnell klar, dass bei einer direkten Verbindung von

Sensor und Prozessor die Genauigkeit unter dem großen Wertebereichs des Sensor leiden würde. Als Konsequenz daraus wurde beschlossen nur einen kleinen Teil des Wertebereichs des Sensors zu nutzen und diesen über einen Messverstärker so zu verstärken, dass er die Bandbreite des AD-Wandlers voll ausnutzt.

Nun galt es nur noch eine Lösung zur Speicherung der Messwerte zu finden. Der ATmega32 verfügt intern über einen sogenannten Eeprom-Speicher, dieser wäre zwar ausreichend groß, lässt aber nur eine beschränkte Anzahl an Schreibzyklen pro Speicherblock zu.

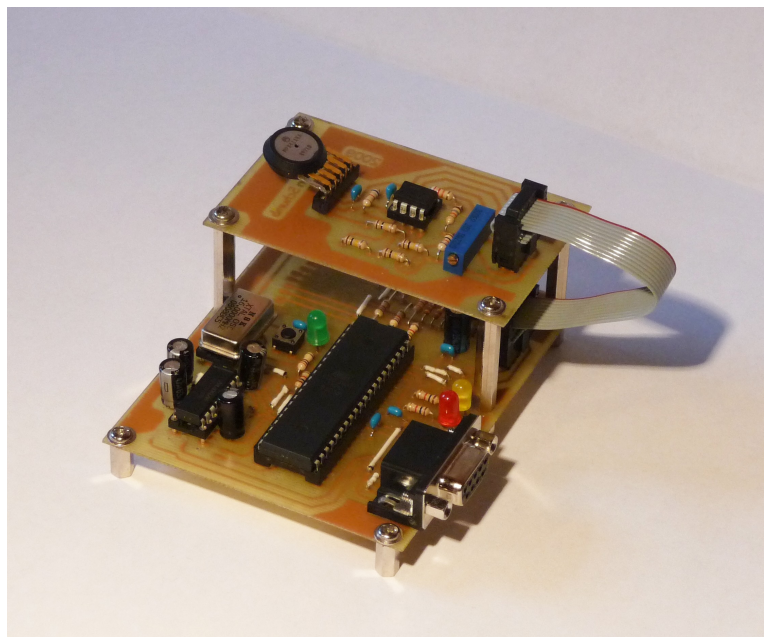


Abbildung 3: Prototyp des Höhemessers

Bei einem Fehler in der Programmierung oder einfach durch häufige Benutzung, lässt sich diese Zahl in einer relativ kurzen Zeit erreichen.

Externe Flashspeicher haben dieses Problem nicht, allerdings wäre es immer noch notwendig geworden die Werte in irgendeiner Form zur Auswertung aus dem Speicher auszulesen, zum Beispiel indem man sie nach Programmende über die serielle RS232 Schnittstelle an einen PC überträgt. Das setzt wiederum eine spezielle Empfangssoftware am PC und außerdem eine RS232 Schnittstelle, die heute leider nicht mehr standardmäßig an jedem PC vorhanden ist, voraus.

Beim der Recherche über Flashspeicher, stieß ich allerdings auf eine in C geschriebene Bibliothek, die es ermöglicht SD-Speicherkarten an den Mikroprozessor anzubinden. Mit Hilfe der Bibliothek von Roland Riegel kann man Dateien im Fat16-Filesystem der Karte erzeugen, lesen und schreiben. Das Speichern der Werte als CSV-Datei (Comma Separated Values) auf einer SD-Karte hat den Vorteil, dass man die Ergebnisse am PC einfach mit einer Tabellenkalkulationssoftware öffnen und weiterverarbeiten kann. So kann man auch auf mehreren Karten mehrere Messreihen durchführen, ohne dass es zwischendurch erforderlich wäre die Ergebnisse am PC auszulesen.

Zusätzlich lässt sich die SD-Karte auch dann noch auslesen, wenn der Prozessor aus irgendwelchen Gründen, wie zum Beispiel einen leeren Akkumulator, nicht mehr korrekt funktionieren sollte.

Die hardwareseitige Anbindung der Karte gestaltet sich ähnlich einfach wie bei einem Flashspeicherbaustein : Es sind nur relativ wenige zusätzliche Bauteile nötig. Sechs Widerstände, als Spannungsteiler wandeln den TDL Pegel des AVR's auf die Signalpegel der SD-Karte. Die Versorgungsspannung von 3,0-3,6V wird über 3 Dioden, bei denen über 0,7V abfällt aus der 5V Versorgung des AVR's gewonnen.

Trotzdem ist zusätzlich eine RS232 Schnittstelle mit Pegelwandler vorhanden, um Werte und Debug-Informationen schon während den Messungen auslesen zu können und um zusätzliche Einstellungen vom PC aus setzen zu können.

Nach dem das Grundkonzept festgelegt war und einzelne Bestandteile, wie zum Beispiel der Messverstärker oder die Anbindung der SD-Karte als Prototypen auf Lochraster aufgebaut und getestet waren, wurde ein Platinenlayout erstellt und die Platinen photochemisch selbst hergestellt.

Der Aufbau wurde in zwei Platinen aufgeteilt. Eine kleinere Platine auf der der Verstärker mit dem Sensor aufgebaut ist und eine Größere, die den Mikroprozessor und die SD-Karte enthält.

Schaltplan1

Schaltplan 2

### 3.1. Der Drucksensor

Der Drucksensor MPX4115A ist das zentrale Element des Höhenmessers.

Er misst den absoluten Luftdruck, über eine Kraft auf eine mikroskopisch kleine Referenzdruckkammer (*sealed vacuum reference*). Diese sitzt unterhalb des eigentlichen Chips (*die*). Die Kraft wird vom Chip piezoelektrisch erfasst und weiterverarbeitet.

Die Umgebungstemperatur wird über eine im Chip integrierte Schaltung kompensiert. Dadurch wird verhindert, dass die temperaturabhängige Ausdehnung beziehungsweise das Zusammenziehen des Bauteils die Messung verfälschen.

Der Wertebereich erstreckt sich von 150 bis 1150kPa, was laut Höhenformel einer theoretischen Höhe von 13511,07m bis -1080,97m entspräche.

Der Messwert wird als zum Druck proportionale Spannung von 0 bis 5V an einem Bauteilpin ausgegeben.

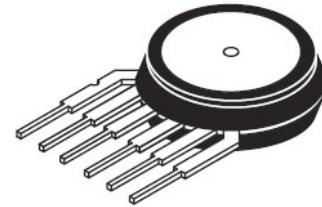


Abbildung 4: MPX5115A Drucksensor  
(Quelle : Datasheet von Motorola)

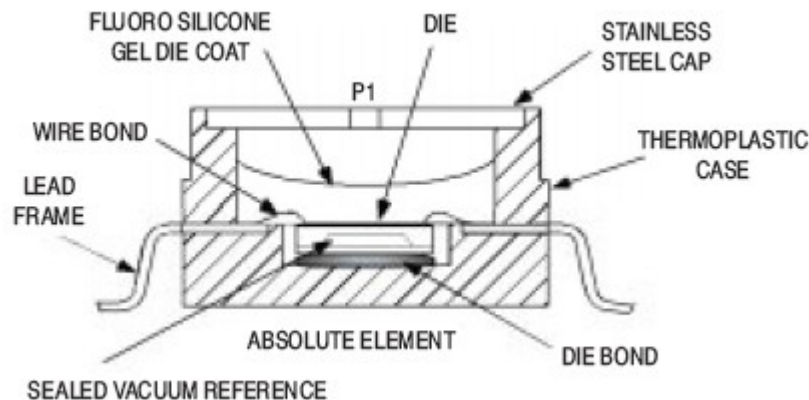


Abbildung 5: Querschnitt durch den Drucksensor  
(Quelle : Datasheet von Motorola)

### 3.2. Der Messverstärker

Da der Messbereich des Drucksensors deutlich größer ist als nötig, würde man manche Messwerte in der Realität nie erreichen. Um die Auflösung des AD-Konverters dennoch möglichst sinnvoll auszunutzen, kann man einen kleinen Bereich der Ausgangsspannung des Sensors, auf 5V verstärken.

Der Einfachheit halber wurde in diesem Fall der Bereich zwischen 4 und 3V, also 158 und 2233m über dem Meeresspiegel gewählt. Eine einfache Schaltung mit 2 Operationsverstärkern subtrahiert zunächst 3V von der Ausgangsspannung und multipliziert den verbliebenen Rest mit 5. Dadurch erhält man bei 3V eine Ausgangsspannung von 0V und bei 4V 5V. Dadurch kann man den Bereich zwischen 3V und 4V mit der vollen Auflösung des AD-Konverters erfassen. Der kleinste messbare Spannungsunterschied liegt somit bei einem Millivolt anstatt 5mV. Somit

verringert sich die kleinste messbare Höhendifferenz von 17 m auf theoretisch 2m. Damit reduzieren sich natürlich auch die möglichen Abweichungen durch Messfehler.

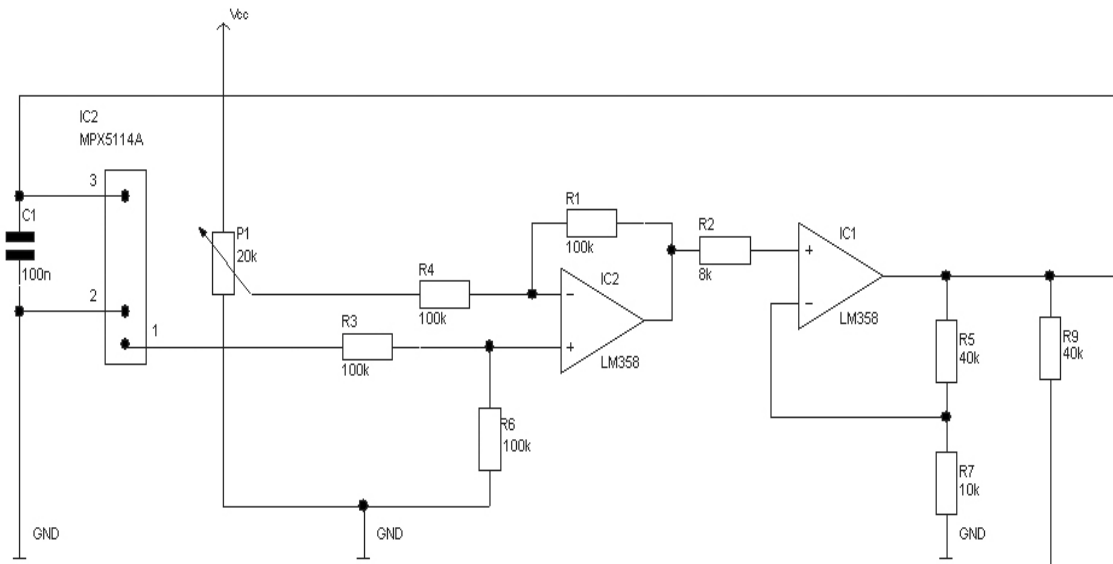


Abbildung 6: Schaltplan des Messverstärkers

Der Messverstärker wurde mit dem LM358 IC realisiert. Dieser integrierte Schaltkreis enthält zwei Operationsverstärker in einem Gehäuse, ist daher für diese Anwendung besonders gut geeignet. Mit ihm wurden zwei Operationsverstärker-Grundsaltungen, die aus dem Buch „Elektronik 3: Grundsaltungen“ entnommen wurden aufgebaut. Zunächst zieht ein Subtrahierverstärker von der Ausgangsspannung einen über einen Zehngangtrimmer einstellbaren Betrag ab. Ein Spannungsteiler aus fixen Widerständen wäre zu ungenau und würde sich nicht an veränderte Betriebsspannungen anpassen lassen. Anschließend verstärkt ein nicht invertierender Verstärker die Spannung um den Faktor 5. Die Widerstände für beide Schaltungen wurden nach den im Buch angegebenen Formeln berechnet.

Leider verhält sich der reelle Operationsverstärker nicht wie der theoretisch angenommene ideale Verstärker. Das liegt einerseits an den Fertigungstoleranzen der Bauteile, andererseits an Umgebungseinflüssen wie zum Beispiel der Temperatur.

Bei 3V Eingangsspannung erhält man zwischen 0,150 und 0,350V anstatt den gewünschten 0V. Bei 4V erhält man bis zu 5,5V. Die Abweichung zum Sollwert ist dabei nicht über den ganzen Wertebereich konstant. Sie wird gegen Ende größer. Einfaches subtrahieren genügt also nicht um die Abweichung auszugleichen. Dennoch ist die Ausgangsspannung trotz dieser Abweichungen immer noch proportional zu der Eingangsspannung. Man kann jetzt Mittels zwei Messwerten, zum Beispiel der bei einer Eingangsspannung von 3 und 4V eine lineare Funktion aufstellen, die der Eingangsspannung 3V die Ausgangsspannung 0V und der Eingangsspannung 4V 5V zuordnet. Auf diese Weise lässt sich die Abweichung mit geringem Aufwand korrigieren. Die Funktion ist Bestandteil des Programms des Mikroprozessor und dient auch dazu die linearen Abweichungen des AD-Wandlers aus zu gleichen.



### 3.3. Das Programm

Der Mikroprozessor des Höhenmessers wurde in C programmiert. Dafür wurde die Entwicklungsumgebung Eclipse mit dem AVR-gcc-Toolchain verwendet. Zum Übertragen in den Prozessor wurde das STK500-Board von Atmel benutzt. Außerdem wurden der Prozessoremulator Simulavr und ein eigens für diese Arbeit erstellter Nachbau des JTAG ICE on-Chip-Debuggers zusammen mit der Software Avr-gdb zur Fehlersuche im Programm verwendet.

Das Programm besteht im wesentlichen aus 4 Teilen. Den Bibliotheken für die SD-Karte von Roland Riegel, den Bibliotheken für die serielle Schnittstelle, für den AD-Wandler und dem Hauptprogramm. An dieser Stelle werden bloß die Quelltextteile besprochen, die von mir geschrieben wurden und wichtig für die eigentliche Höhenmessung sind, also das Hauptprogramm und die Bibliothek für den AD-Wandler.

Die Quelltexte sind auszugsweise im Anhang 7.1 und als Projekt für Eclipse auf der beiliegenden CD zu finden.

Am Anfang des Programms wird die Hardware initialisiert. Erst werden die Status Leds auf den Normalzustand gesetzt, danach wird der prozessorinterne UART für die serielle Verbindung zum PC mit 9600Baud, keiner Parität, 8 Bits mit einem Stopbit und keiner Flusskontrolle konfiguriert. Anschließend wartet der Prozessor 200ms um dem Pegelwandler für die Schnittstelle genug Zeit zu geben die Kondensatoren auf die notwendigen Pegel auf zu laden. Ohne die Vorlaufzeit wäre die erste Übertragung zum PC nicht lesbar oder würde das Terminalprogramm zum Absturz bringen. Im Anschluss daran wird die SD-Karte aktiviert.

Wenn die Hardware bereit ist, wird, nachdem ein Begrüßungsbildschirm zum PC übertragen wurde, dem Benutzer am PC zwei Sekunden Zeit gegeben eine Taste zu drücken um die Kalibrierung der Software zu starten.

Wenn eine Taste gedrückt wird beginnt die Kalibrierungssequenz. Die Kalibrierung dient dazu die Abweichungen des Verstärkers und des AD-Wandlers aus zu gleichen. Dafür muss der Drucksensor gegen ein 10gang Potentiometer mit passendem Stecker getauscht werden. Anschließend müssen mit dem Potentiometer und einem Voltmeter zur Kontrolle zwei Spannungen eingestellt werden. Die erste Spannung, die abgefragt wird ist 3V, sie dient als Nullpunkt. Die zweite Spannung 3,8V dient dazu die Obergrenze der Messbereichs fest zu legen. Eigentlich müsste der zweite Wert bei 4V gemessen werden, aber da dann die Ausgangsspannung des Verstärkers über 5V, also über dem Maximalwert des AD-Wandlers liegen würde, wird die Ausgangsspannung bei 3,8V benutzt. Die beiden Werte werden als Strings auf der SD-Karte in der Datei *cal.conf* gespeichert, damit man den Höhenmesser nicht jedes mal neu kalibrieren muss. Danach muss man den Höhenmesser über den Taster auf der Platine neu starten.

Wird keine Taste gedrückt, weil zum Beispiel kein PC angeschlossen ist, werden die beiden Werte von der SD-Karte gelesen und in den Variablen *cal\_min* und *cal\_max* abgelegt.

Nach dem die Werte gelesen wurden beginnt die eigentliche Messung. Zunächst wird auf der Speicherkarte die Datei *data.csv* erstellt. Existiert diese schon, so wird sie erst wieder gelöscht. Jetzt kann die eigentliche Messung beginnen. In einer Schleife wird mit der Funktion *readADC* ein Wert vom AD-Wandler gelesen. Die Funktion *readADC* liest 5 Werte mit 5ms Abstand aus und bildet dann den Durchschnitt. Das

soll dazu dienen das Rauschen, als minimale Schwankungen, des AD-Wandler und des Verstärkers heraus zu filtern. Nach dem Auslesen wird zunächst die 16-bit Ganzzahl, die der AD-Wandler liefert in eine Spannung von 0 bis 5V umgerechnet und in der Variablen *voltage* abgelegt. Dieser Spannungswert ist mit Messfehlern des AD-Wandlers und vom Verstärker verfälscht. Jetzt werden die beiden Kalibrierungswerte dazu benutzt die Fehler mit einer linearen Funktion heraus zu rechnen. Das Ergebnis wird in der Variablen *voltage\_cor* abgelegt.

Die Funktion ordnet dem Wert in *cal\_min* 0V und dem Wert in *cal\_max* 4V zu. Die konkrete Implementierung sieht so aus : 
$$voltage_{cor} = (voltage - cal_{min}) \cdot \frac{4}{cal_{max} - cal_{min}} .$$

Die Abweichung vom im Programm erfassten Wert zur realen Spannung wird dabei auf maximal ein Millivolt reduziert.

Dieser korrigierte Spannungswert kann dann auf die Spannung vor dem Verstärker umgerechnet werden. Dieser Wert wird durch 10 geteilt und über die 10 Schleifendurchläufe aufsummiert um den Durchschnitt zu bilden. Danach enthält die Schleife noch eine Verzögerung, so dass ein Durchlauf ungefähr eine Sekunde braucht. Nach diesen 10 Durchläufen wird aus der Spannung der Druck und aus dem Druck mittels der Höhenformel die Höhe errechnet.

Es werden also 10s lang zu jeder vollen Sekunde 5 Werte im Abstand von 5 ms gemessen, aus denen dann in zwei Stufen der Durchschnitt gebildet wird. Danach wird die Höhe errechnet. Dieses Verfahren wurde empirisch gefunden und hat sich in der Praxis hervorragend bewährt. Die Messwerte sind relativ genau. Im Stand hat man maximal 5m Abweichung zwischen zwei Messungen.

Die Messwerte werden sowohl in die Datei auf der Speicherkarte geschrieben, als auch über die serielle Schnittstelle ausgegeben, damit es möglich ist eine Messreihe in Echtzeit zu verfolgen.

Am Ende jeder Messung wird nochmal 10s bis zur nächsten Messung gewartet, eine Messung dauert also knapp 20s.

Ein Datensatz in der CSV-Datei benötigt im Schnitt 25Bytes. Mit 3 Messwerten pro Minute könnte man also mit 1GB Speicher mehrere Tage lang ununterbrochen messen.

Zum Beenden der Messphase kann man entweder wieder ein Zeichen über die serielle Schnittstelle senden, oder wenn der Höhenmesser ohne Computer benutzt wird, den Akkumulator abklemmen. Das führt dann zwar dazu, dass das Programm nicht korrekt beendet wird, aber da die Speicherkarte ungepuffert beschrieben wird sind alle Messwerte sicher gespeichert und können später problemlos ausgelesen werden.

Die Datei mit den Messwerten auf der SD-Karte kann von OpenOffice Calc oder Excel direkt eingelesen werden. Allerdings wird vom Mikroprozessor der Punkt „.“ und nicht wie von Calc erwartet das Komma „.“ als Dezimaltrennzeichen verwendet. Abhilfe kann dadurch geschaffen werden, dass man entweder Calc auf englische Sprache umstellt oder die Datei mit einem Texteditor öffnet und alle Punkte durch Kommas ersetzt. Anschließend erhält man eine Tabelle mit 4 Spalten, die erste enthält die Nummer der Messung, die zweite die Ausgangsspannung des Sensors, die dritte den Druck und die vierte die eigentliche Höhe. Die Werte können in der Tabellenkalkulationssoftware ganz normal weiter bearbeitet werden. So kann man sich zum Beispiel ein Höhenprofil als Diagramm anzeigen lassen.

## 4. Test

Nachdem die Hardware und die Software fertiggestellt waren, ging es daran den Höhenmesser zu testen. Dabei standen Genauigkeit und die Praxistauglichkeit im Vordergrund. Da mir keine Unterdruckkammer zur Verfügung stand und mir ein Test unter Laborbedingungen sowieso als wenig aussagekräftig erschien, wurde der Höhenmesser an einem echten Berg getestet.

Die Teststrecke verlief von einem Parkplatz in der Wilgartswiesener Ortsmitte, der laut TK25-Karten auf ungefähr 200m über NN liegt, auf den Hemersberger Hof, der ungefähr 530m hoch liegt.

Als Referenz wurde ein Garmin GPSMap 60Cx GPS-System mit integriertem barometrischen Höhenmesser benutzt. Der Höhenmesser stützt sich auf den Luftdruck, die eingespeicherten TK-25 Karten und das GPS, was eine sehr hohe Genauigkeit ermöglicht.

Die Bedingungen am Testtag waren nicht gerade optimal. Es war regnerisch und windig, der Luftdruck hatte schon den ganzen morgen stark um den Normalwert geschwankt.

Auf dem Parkplatz wurde das Garmin-GPS anhand der Karte geeicht und der selbst gebaute Höhenmesser ein letztes Mal kalibriert. Außerdem wurde ein Laptop angeschlossen um den Höhenmesser während der Fahrt beobachten zu können. Bei den ersten Messungen betrug die Differenz vom Garmin GPS zum eigenen Höhenmesser gerade einmal 2 Meter.

Die ersten Messwerte im bewegten Auto zeigten auch noch keine Abweichung. Erst auf den wirklich steilen Streckenabschnitten wurde eine Differenz von 20m erkennbar. Diese lässt sich aber darüber erklären, wie der Höhenmesser misst. Da der Durchschnitt über 10s gebildet wird und sich das Auto mit ca. 60km, also 16,667m/s, fortbewegt hat, sind die gemessenen Höhen die mittlere Höhe über die letzten 166m. Diese Vermutung wurde bestätigt, als der sich die Werte auf den ebenen Streckenteile wieder auf 2m genau annäherten.

Allerdings stellte sich ab 500m Höhe, also fast auf dem höchsten geplanten Punkt eine weitere Abweichung ein. Der Wert des selbstgebauten Höhenmessers stieg nicht mehr weiter an. Selbst als das GPS 530m anzeigte, wurde auf dem Laptop immer noch 503m angezeigt. Vor Ort durchgeführte Messungen mit dem Multimeter und Berechnungen mit dem Taschenrechner ergaben, dass die errechnete Höhe zwar



Abbildung 7: Abgefahrene Strecke (Aufzeichnung des Garmin GPS)  
(Quelle : Google Earth)

korrekt, die Ausgangsspannung des Drucksensor allerdings falsch war. Im ersten Moment war davon aus zu gehen, dass ein unbemerkter Fehler in der Schaltung eventuell unterwegs den Sensor beschädigt haben könnte. Kurzer Hand wurde beschlossen die Rückfahrt zum Parkplatz in Wilgartswiesen ebenfalls in die Messung mit aufzunehmen um den Sensor zu testen. Erstaunlicherweise reagierte er nach einiger Zeit wieder absolut normal. Allerdings zeigte das GPS zu diesem Zeitpunkt schon wieder 450m. Nach einiger Zeit hatten sich beide Werte wieder angenähert und in Wilgartswiesen war die Differenz wieder minimal.

Leider war für einen weiteren Test keine Zeit mehr, deshalb konnte der Effekt bei 500m nicht weiter untersucht werden.

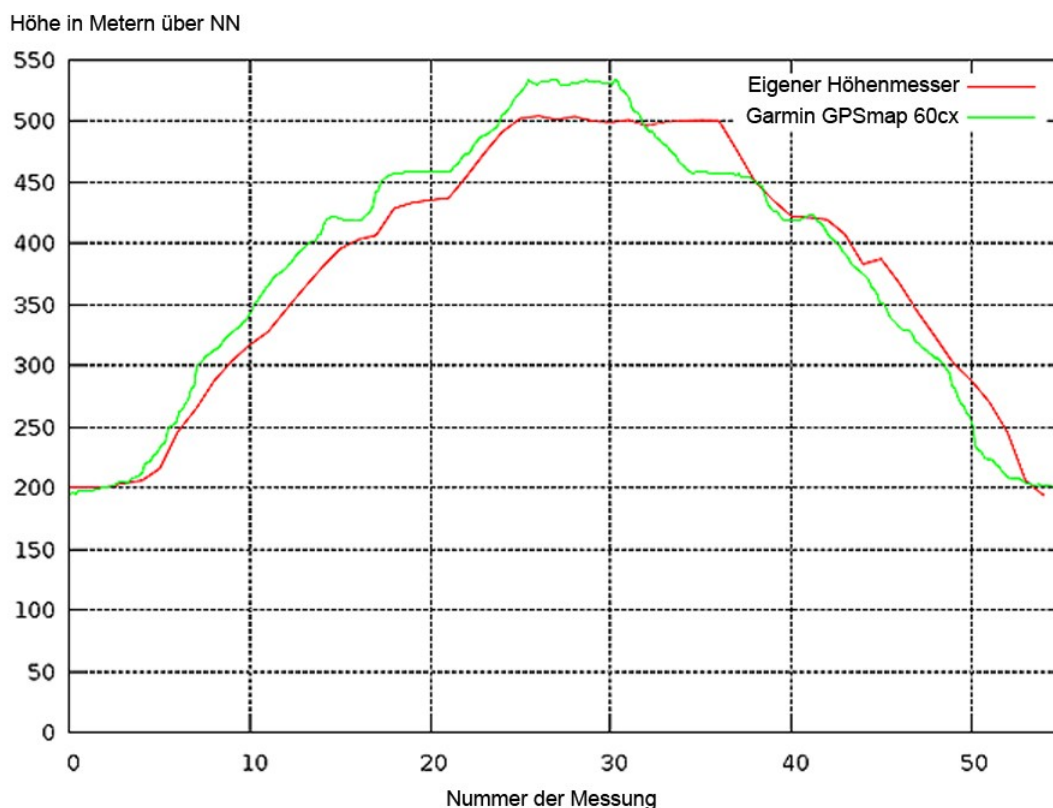


Abbildung 8: Messkurve des Test

Die restliche Messkurve entspricht allerdings, wenn man die unterschiedlichen Messgeschwindigkeiten der beiden Geräte, das Garmin GPS misst 5 mal mehr Werte, und die dadurch entstehenden Abweichungen berücksichtigt, relativ genau der des professionellen Geräts. Zwar ist die Kurve des selbst gebauten Höhenmesser durch die Durchschnittsbildung über die langen Messintervalle deutlich geglättet, dennoch sind die wesentlichen Streckenabschnitte in beiden Kurven gut zu erkennen.

Das Diagramm wurde mit der Software Gnuplot erstellt, die es ermöglicht zwei Kurven, die unterschiedlich viele Messpunkte enthalten, in einem Diagramm einheitlich darzustellen.

## **5. Fazit**

Zielsetzung dieser Arbeit war es zunächst die theoretische Grundlage für den Bau eines Höhenmessers zu schaffen und anschließend einen funktionierenden Prototypen zu bauen.

Obwohl die theoretischen Überlegungen und die Planung relativ ausführlich waren, traten in der Praxis unerwartete Probleme auf.

Der Messbereich des zur Verfügung stehenden Sensors war zu groß, der Verstärker verhielt sich anders, als vorher berechnet, der AD-Wandler des Mikroprozessors hatte ein relativ starkes Grundrauschen. Viele dieser Probleme konnte man in der Planungsphase gar nicht oder nicht im vollen Ausmaß erkennen und das Finden von Lösungen war oftmals mit großem Arbeitsaufwand verbunden. Nicht alle Fehler waren offensichtlich und meistens ließen sich Fehlerquellen nur durch einfaches Ausprobieren finden. Einige Lösungen sind auch eher provisorisch, zum Beispiel die auf der Platinenunterseite eingefügten Bauteile.

Doch trotz dieser Probleme und dem vergleichsweise primitiven Aufbau überzeugt das Endergebnis durch eine unerwartet hohe Genauigkeit. Der fertige Höhenmesser ist zwar nur ein Prototyp und hat noch viel Optimierungspotential, trotzdem konnte er in der Messung gegen ein professionelles Gerät bestehen. Alleine wenn man die Komplexität professioneller Höhenmesser, die meist über bessere Sensoren, präzisere Elektronik und eine ausgefeiltere softwareseitige Auswertung verfügen, betrachtet hätte man von dem hier erläuterten einfachen Aufbau eine wesentlich größere Abweichung erwartet. Dennoch war die Abweichung nie größer als 30m und im Stand ging sie sogar auf bis zu 2m zurück.

Das ist ein durchaus akzeptable Ergebnis.

Deshalb denke ich, dass die am Anfang dieser Arbeit gesetzten Ziele mehr als erfüllt sind.

## 6. Quellenverzeichnis

### 6.1. Luftdruck und Höhenformel

Dieter Karnetzki „*Luftdruck und Wetter*“

Delius Klasing Verlag 1999

ISBN : 978-3874121309

Seydlitz „*Geographie Rheinland-Pfalz SII*“

Schroedel Verlag 2008

ISBN : 978-507-52574-0

Hans-Jochen Bartsch „*Taschenbuch mathematischer Formeln*“

Verlag Harri Deutsch 1980

ISBN : 3-87244-239-9

IHK Braunschweig : „*Niels Bohr und das Barometer*“

[http://www.braunschweig.ihk.de/innovation\\_umwelt/nachrichten\\_2007/april\\_07/?viewMeldung=meldung\\_1175348701.15](http://www.braunschweig.ihk.de/innovation_umwelt/nachrichten_2007/april_07/?viewMeldung=meldung_1175348701.15)

Wikipedia : „*Luftdruck*“

<http://de.wikipedia.org/wiki/Luftdruck>

Fachgruppe Physik der RWTH Aachen : „*Barometrische Höhenformel*“

[http://web.physik.rwth-](http://web.physik.rwth-aachen.de/~fluegge/Vorlesung/PhysIpub/Exscript/8Kapitel/IIX6Kapitel.html)

[aachen.de/~fluegge/Vorlesung/PhysIpub/Exscript/8Kapitel/IIX6Kapitel.html](http://web.physik.rwth-aachen.de/~fluegge/Vorlesung/PhysIpub/Exscript/8Kapitel/IIX6Kapitel.html)

Wikipedia : „*Höhenformel*“

<http://de.wikipedia.org/wiki/H%C3%B6henformel>

Wikipedia : „*Thermische Zustandsgleichung idealer Gase*“

[http://de.wikipedia.org/wiki/Thermische\\_Zustandsgleichung\\_Idealer\\_Gase](http://de.wikipedia.org/wiki/Thermische_Zustandsgleichung_Idealer_Gase)

### 6.2. Elektronik

Klaus Beuth „*Elektronik 2 - Bauelemente*“

Vogel-Buchverlag Würzburg 1984

ISBN : 3-8023-0539-9

Klaus Beuth / Wolfgang Schmusch „*Elektronik 3 - Grundsaltungen*“

Vogel-Buchverlag Würzburg 1984

ISBN : 3-8023-0555-8

Karlheinz Albrecht / Max-UlrichFarber „*Elektronik mit Halbleiterbauelementen*“

Aulis Verlag Deubner & Co KG 1973

ISBN : 3-7614-0187-6

Mikrocontroller.net : Operationsverstärker-Grundsaltungen

<http://www.mikrocontroller.net/articles/Operationsverst%C3%A4rker-Grundsaltungen>

Ulrich Radig, mikrocontroller and more :: MMC-SD

<http://www.ulrichradig.de/home/index.php/avr/mmc-sd>

Datasheet für den ATmega32

[http://www.atmel.com/dyn/resources/prod\\_documents/doc2503.pdf](http://www.atmel.com/dyn/resources/prod_documents/doc2503.pdf)

Datasheet für den MPX4115A Lufdrucksensor

<http://www.datasheetcatalog.org/datasheet/motorola/MPX4115A.pdf>

Datasheet für den MAX3232 Pegelwandler

<http://www.datasheetcatalog.org/datasheet/maxim/MAX3222-MAX3241.pdf>

### 6.3. Programmierung

Microkontroller.net : „*AVR-Tutorial : ADC*“

[http://www.mikrocontroller.net/articles/AVR-Tutorial:\\_ADC](http://www.mikrocontroller.net/articles/AVR-Tutorial:_ADC)

RN-Wissen : „*UART mit avr-gcc*“

[http://www.roboternetz.de/wissen/index.php/UART\\_mit\\_avr-gcc](http://www.roboternetz.de/wissen/index.php/UART_mit_avr-gcc)

Rolant Riegel : „*MMC/SD/SDHC card library*“

<http://www.roland-riegel.de/sd-reader/>

## 7. Anhang

### 7.1. Quelltextauszüge

#### Main.c - Das Hauptprogramm

```
#include <avr/io.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include <math.h>

#include "include/uart/uart.h"
#include "include/sdcard/sdcard.h"
#include "include/adc.h"
#include "include/leds.h"

/*
 * sleep(ms) wartet ms Millisekunden
 */
void inline sleep_ms(uint16_t ms) {
    while(ms > 0) {
        _delay_ms(1);
        ms--;
    }
}

/*
 * Hauptprogramm
 */
int main() {
    uint8_t cal_active;           // Startet Kalibrungssequenz wenn 1
    uint16_t i,j,res;            // zwei Laufvariablen, Resultat der AD-Wandlung
    struct fat_file_struct* file; // Handel für die Dateien
    float cal_min,cal_max;       // Minimum und Maximumkalibrierung
    // Gemessene Spannung, Korregierter Spannungswert, Spannung vor der Verstärkung, Druck, Höhe
    float voltage,voltage_cor,voltage_pre,pressure,height;
    char tmp[10]; // Puffer für die Konvertierung von Zahlen in Strings

    leds_init(); // Leds initialisieren

    uart_init(); // Uart aktivieren

    sei(); // Interrupts aktivieren

    // 200ms warten, bis alle Kondesatoren geladen und der Pegelwandler bereit ist
    leds_set_busy();
    sleep_ms(200);
    leds_unset_busy();

    // Banner übertragen, Eye-Candy
    uart_puts_const(PSTR("\n"));
    uart_puts_const(PSTR("+-----+\n"));
    uart_puts_const(PSTR("| Barometrischer Höhenmesser |\n"));
    uart_puts_const(PSTR("+-----+\n"));
    uart_puts_const(PSTR(" by Sebastian Schumb 2009\n\n"));

    if(sdcard_init() == -1) // SD-Karte initialisieren
        return 0;
}
```



## Facharbeit : Bau eines barometrischen Höhenmessers

---

```
/*
 * Kalibrierungssequenz fuer den ADC
 */

cal_active = 0;
i = 0;
leds_set_busy();

// 2s Warten ob eine Taste gedrückt werden
uart_puts_const(PSTR(">> Press any key to start calibration \n"));
while(cal_active == 0 && i <= 20) {
    if(uart_getc() != -1) {
        cal_active = 1;
    }
    uart_puts_const(PSTR("."));
    i++;
    sleep_ms(100);
}

uart_puts_const(PSTR("\n"));

if(cal_active == 1) {
    file = sdcard_fopen("cal.conf",1); // Kalibrierung starten
    if(!file) { // alte Datei loeschen, neue anlegen
        return 0;
    }

    // Minimum einlesen
    uart_puts_const(PSTR(">> set 3.00V at amp input and press a key\n"));
    while(uart_getc() == -1);
    cal_min = readADC() * 0.00489;
    dtostrf(cal_min,6,3,tmp);

    uart_puts_const(PSTR(">> minimal : "));
    uart_puts(tmp);
    uart_puts_const(PSTR("V \n"));

    // Minimum speichern
    sdcard_fwrite(file,tmp);
    sdcard_fwrite(file,"\n");

    // Maximum einlesen
    uart_puts_const(PSTR(">> set 3.80V at amp input and press a key\n"));
    while(uart_getc() == -1);
    cal_max = readADC() * 0.00489;
    dtostrf(cal_max,6,3,tmp);

    uart_puts_const(PSTR(">> maximal value : "));
    uart_puts(tmp);
    uart_puts_const(PSTR("V \n"));

    // Maximum speichern
    sdcard_fwrite(file,tmp);
    sdcard_fwrite(file,"\n");

    uart_puts_const(PSTR(">> Values written, please reset.\n"));
    return 0;
}
else { // alte Werte einlesen

    uart_puts_const(PSTR(">> Trying to load calibration data\n"));

    file = sdcard_fopen("cal.conf",0); // Datei oeffnen, falls sie existiert
    if(!file) {
        uart_puts_const(PSTR(">> Could not load values. Please restart an calibrate.\n"));
        return 0;
    }
}
```

```
    }

    if(sdcard_freadline(file,tmp,10) < 1) { // Minimum auslesen
        leds_set_error();
        uart_puts_const(PSTR(">> Could not load minimal value. Please restart an calibrate.\n"));
        return 0;
    }

    cal_min = strtod(tmp,0);

    uart_puts_const(PSTR(">> minimal value : "));
    uart_puts(dtostrf(cal_min,6,3,tmp));
    uart_puts_const(PSTR("V \n"));

    if(sdcard_freadline(file,tmp,10) < 1) { // Maximum auslesen
        leds_set_error();
        uart_puts_const(PSTR(">> Could not load maximal value. Please restart an calibrate.\n"));
        return 0;
    }

    cal_max = strtod(tmp,0);

    uart_puts_const(PSTR(">> maximal value : "));
    uart_puts(dtostrf(cal_max,6,3,tmp));
    uart_puts_const(PSTR("V \n"));

}
sdcard_close(file);
leds_unset_busy();

file = sdcard_fopen("data.csv",1); // Datei zum speichern der Messwerte neu anlegen
if(!file)
    return 0;

sdcard_fwrite(file,"Sample#;preamp Voltage [V];Pressure [hPa];Height;\n"); // Tabellenuerberschrift

i = 1;
while(uart_getc() == -1) {

    voltage_pre = 0;
    for(j = 0; j < 10; j++) {
        res = readADC(); // 5 Werte über 25ms verteilt auslesen
        voltage = res * 0.00489; // ADC wert in Spannung
        //Korrektur der Fehler im Verstaerker
        voltage_cor = (voltage - cal_min) * (4/(cal_max-cal_min));
        voltage_pre += ( (voltage_cor / 10) + 3 )/5; //Spannung vorm Verstaerker

        leds_set_busy();
        sleep_ms(900);
        leds_unset_busy();
    }

    pressure = ((voltage_pre/5)+0.095)/0.0009; //Umrechnung in Druck in hPa
    //Berechnung der Hoehe, Wurzel über e-Funktion und Logarithmus naturalis
    height = (-exp(log(pressure/1013.25)*(1/5.255))+1)*288.155/0.0065;

    //Messungsnummer speichern und ausgeben
    itoa(i,tmp,10);
    uart_puts(tmp);
    sdcard_fwrite(file,tmp);
    uart_puts_const(PSTR(";"));
    sdcard_fwrite(file,";");
}
```

```
        //Spannungswert
        dtostrf(voltage_pre,6,3,tmp);
        uart_puts(tmp);
        sdcard_fwrite(file,tmp);
        uart_puts_const(PSTR(";"));
        sdcard_fwrite(file,";");

        //Druk
        dtostrf(pressure,6,3,tmp);
        uart_puts(tmp);
        sdcard_fwrite(file,tmp);
        uart_puts_const(PSTR(";"));
        sdcard_fwrite(file,";");

        //Höhe
        dtostrf(height,6,3,tmp);
        uart_puts(tmp);
        sdcard_fwrite(file,tmp);
        uart_puts_const(PSTR(";"));
        sdcard_fwrite(file,";");

        uart_puts_const(PSTR("\n"));
        sdcard_fwrite(file,"\n");

        //Warten bis zur nächsten Messung
        sleep_ms(1000);

        i++;
    }

    sdcard_close(file);
    sdcard_end();

    while(1);

    return 0;
}
```

### adc.c – Bibliothek für den AD-Wandler

```
#include "adc.h"

#define ADC_SAMPLE_COUNT 5

uint16_t readADC() {
    uint16_t result;
    int i;

    //ADC aktiv, Prescaler 16
    ADCSRA = (1 << ADEN) | (1 << ADPS2);

    //ADC0 kein multiplexing
    ADMUX = 0;

    //AVcc als Versorgungsspannung
    ADMUX |= (1 << REFS0);

    //ADC starten mit Dummyreadout
    ADCSRA |= (1 << ADSC);
    while(ADCSRA & (1 << ADSC));

    //mehrfach Messen und den Durchschnitt berechnen
    result = 0;
    for(i = 1; i <= ADC_SAMPLE_COUNT; i++) {
```

## Facharbeit : Bau eines barometrischen Höhenmessers

---

```
//eine Messung
ADCSRA |= (1 << ADSC);
while(ADCSRA & (1 << ADSC));
result += ADCW;
_delay_ms(5);
}
result = result / ADC_SAMPLE_COUNT;

//ADC aus
ADCSRA &= ~(1 << ADEN);

return result;
}
```

## 7.2. Bilder

In diesem Anhang sind Bilder enthalten, die nicht zu den Texten gehören, aber den Prozess der praktischen Umsetzung zu illustrieren.

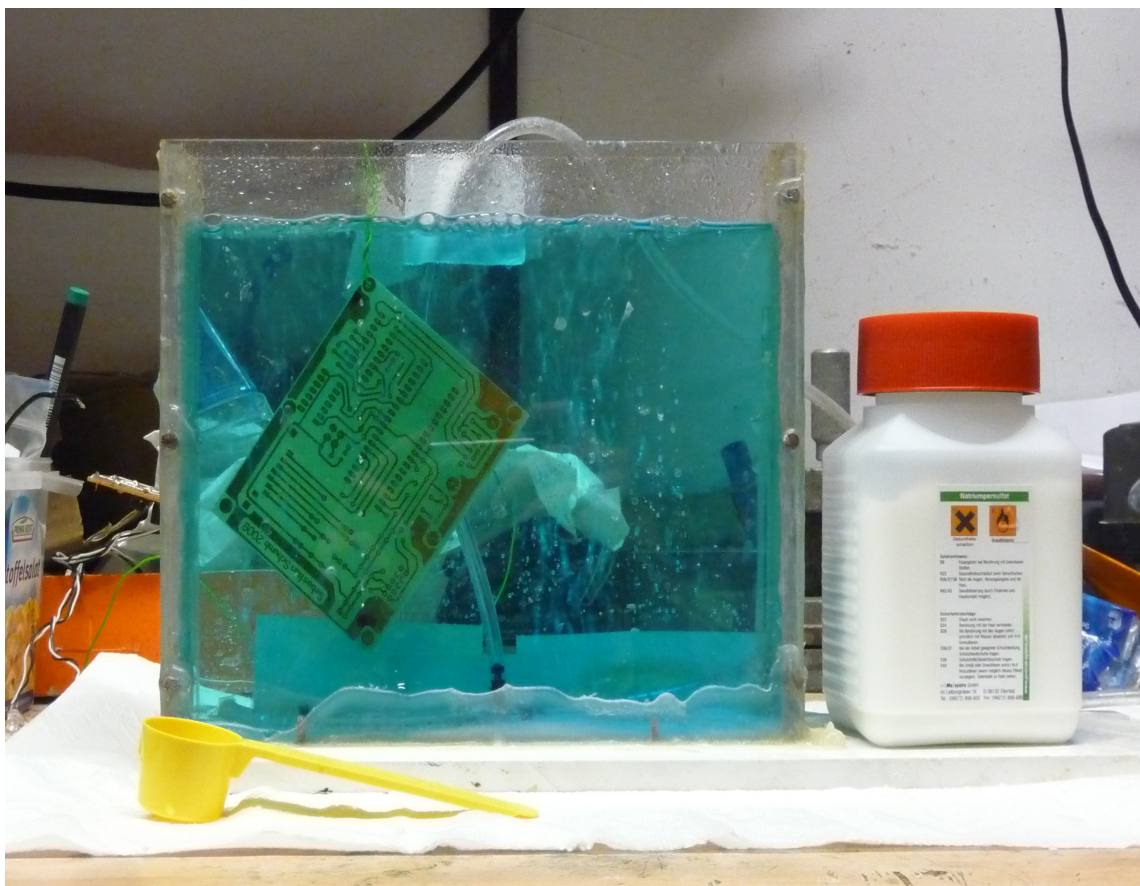


Abbildung 9: Ätzen der Prozessorplatine

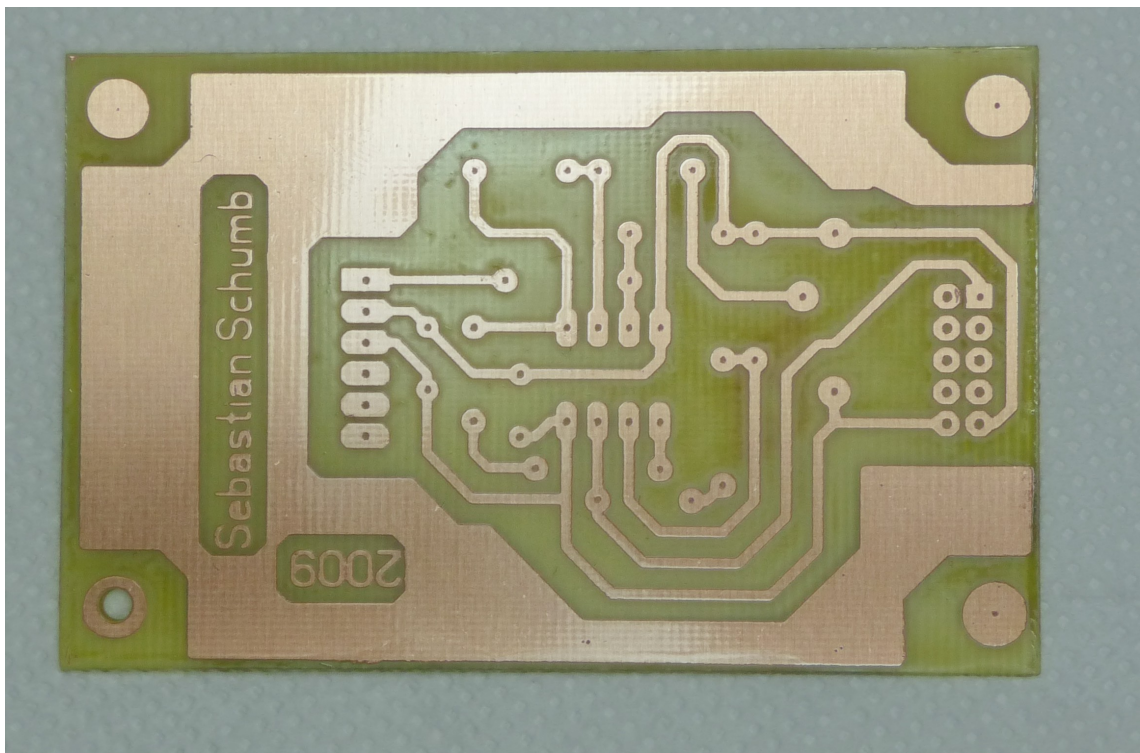


Abbildung 10: Fertige Messverstärkerplatine



Abbildung 11: Bohren der Löcher

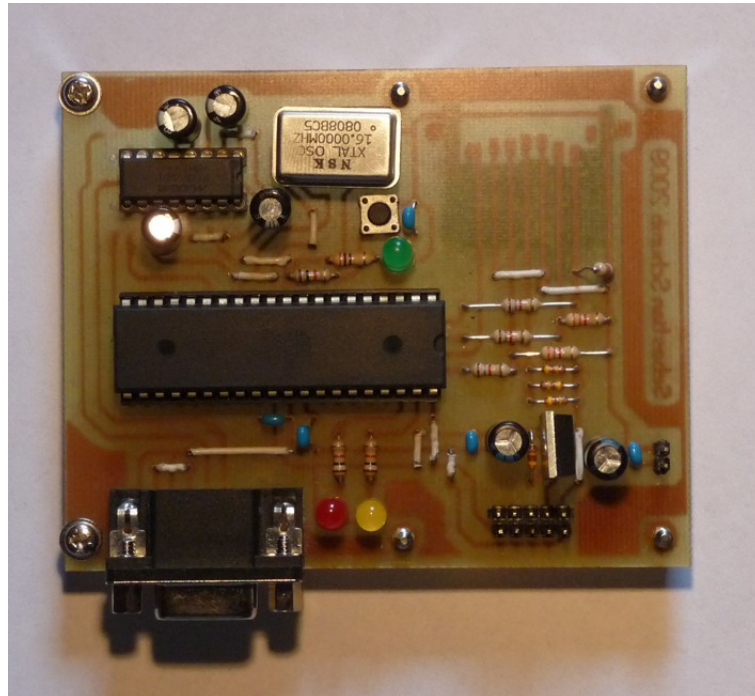


Abbildung 12: Untere Platine mit Bestückung

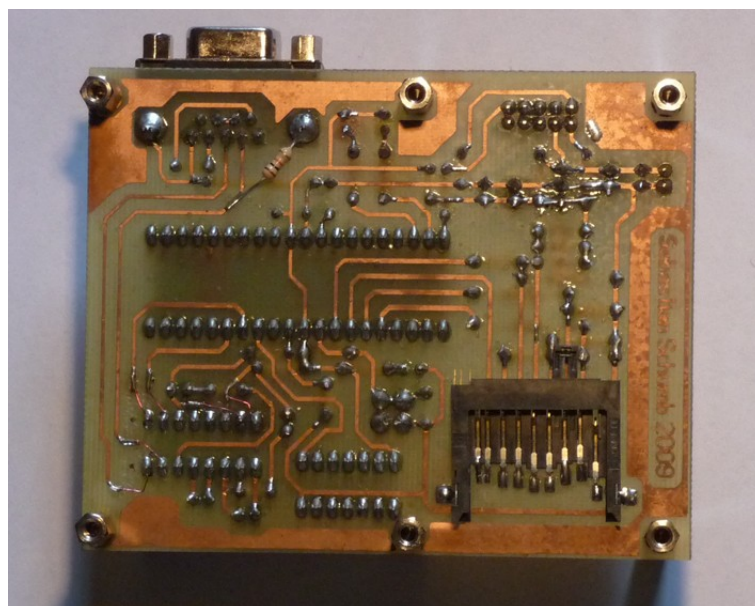


Abbildung 13: Untere Platine von Unten

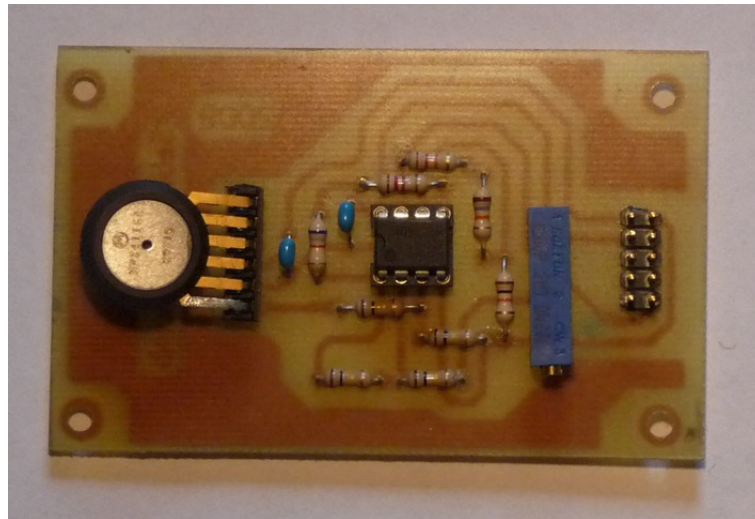


Abbildung 14: latine von oben

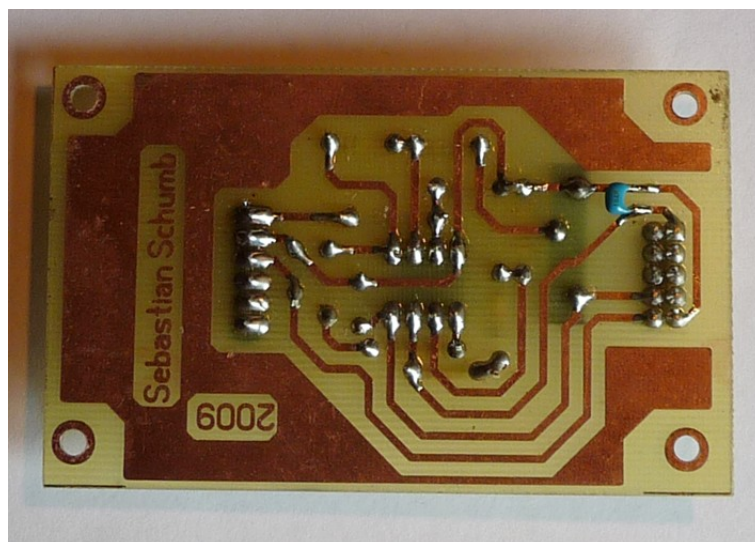


Abbildung 15: Messverstärkerplatine von unten

### 7.3. Platinenlayouts

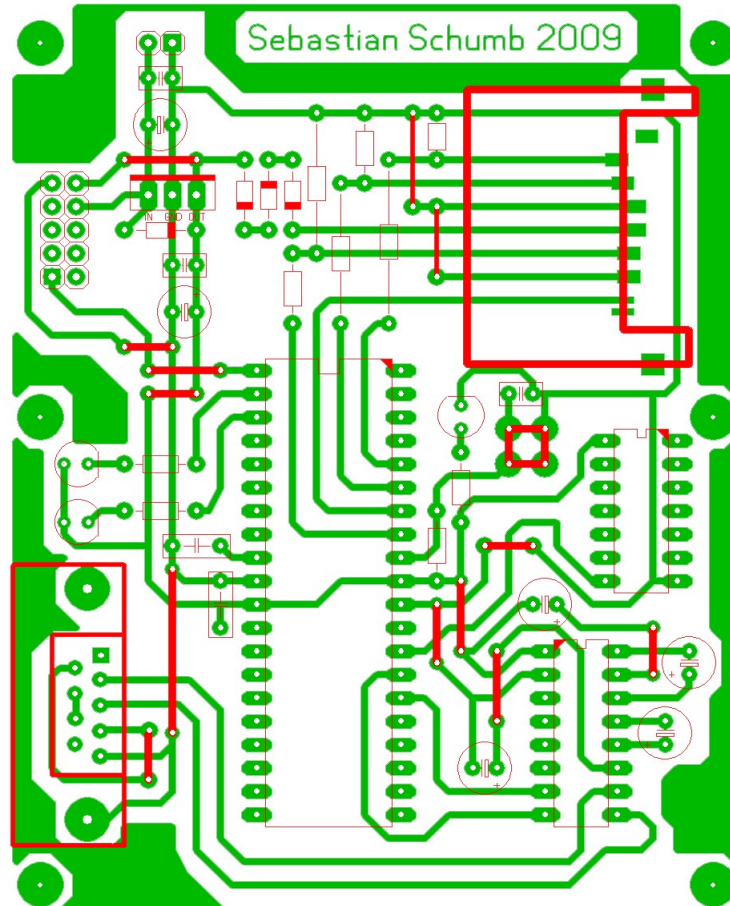


Abbildung 16: Layout der unteren Platine

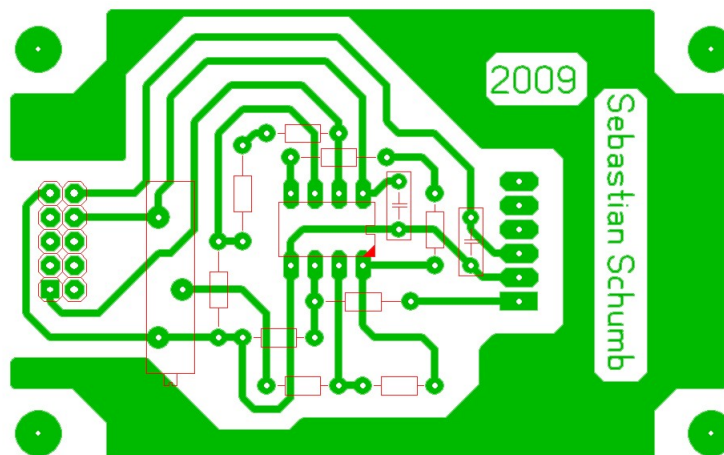


Abbildung 17: Layout der oberen Platine