

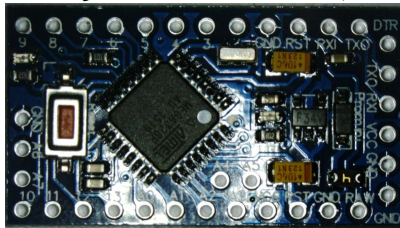
DER 3 EURO ALTIMETER

Version: SOFEX

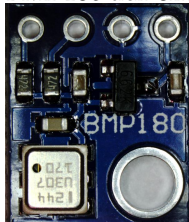
1 MATERIAL

Um den Altimeter zu Bauen braucht ihr folgende Teile, die ihr alle im bereitgestellten Kit findet:

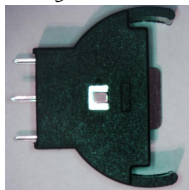
- Arduino pro Mini 8Mhz 3.3V mit einem atmega328 (es ist wichtig die 8Mhz 3.3V Version mit einem atmega328 zu haben und dass die A4 und A5 Pins verdrahtet sind)



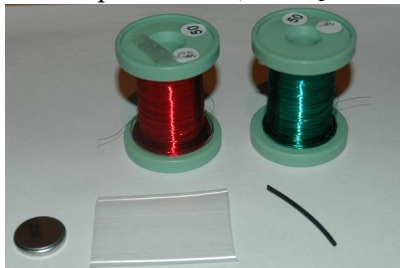
- BMP180 auf einer einer Platine



- cr2032 Batterie-Halter



- cr2032 Batterie
- löthar Lackdraht (dem Kit liegen 2 rote., 2 schwarze, 1 grüner und 1 gelber Draht bei, außerdem noch ein kleines Stück gelb isolierter Draht)
- Schrumpfschlauch (Breite 36mm und ein kleines Stück 1mm)



Außerdem braucht man noch die folgenden Werkzeuge (manche braucht man nur sehr kurz, es ist daher kein Problem wenn nicht alle Leute diese dabei haben):

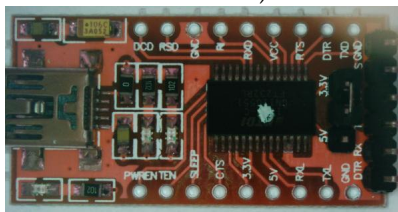
- LötKolben (mit einer Temperatur hoch genug um die Lackschicht auf dem Lackdraht wegzubrennen) + Lötzinn
- Seitenschneider
- Cutter-Messer



- Heißklebepistole + Heißkleber



- Heißluftgebläse um den Schrumpfschlauch zu schrumpfen
- optional: Multimeter um die Verbindungen zu testen
- 3.3V FTDI USB to UART (TTL) Adapter (Ich werde welche mitbringen, braucht ihr also nicht)



Es muss ein 3.3V Adapter sein, ein 5 V Adapter zerstört zwar nicht den Arduino, kann aber den Sensor zerstören (am besten benutzt man einen bei dem man die Spannung mit einer Steckbrücke wählen kann)! Es ist auch hilfreich wenn er einen Reset-Pin DTR für den Arduino hat, das ideale Pin-Layout ist DTR TXo RXi VCC GND GND, dann kann man den Adapter direkt an den Arduino anschließen. Die billigsten USB to TTL Adapter die man für 1€ auf ebay findet sollte man nicht nehmen, die haben oft falsche FTDI chips. Am besten sind die roten mit einer Steckbrücke für die Spannung (findet man auf ebay oder aliexpress unter 'FTDI USB to TTL')

2 DEN ALTIMETER BAUEN

2.1 Modifikationen um Strom zu sparen

Wenn man Lipos anstelle der cr2032 benutzt, sollte man diese Modifikation nicht durchführen.

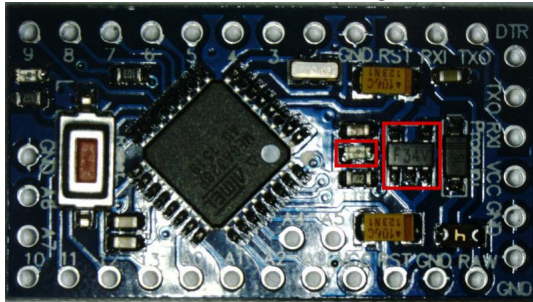
Diese Modifikation braucht man nur um Strom zu sparen, sie reduziert den Stromverbrauch auf der Rampe und in der Luft von $>20\text{ mA}$ zu 3.5 mA und nach dem Gipfelpunkt zu $18\text{ }\mu\text{A}$. Die Batterielaufzeit ist damit 2.5 h wenn man dem Arduino erlaubt unterhalb von 2.7 V zu laufen kann diese auf 40h erhöht werden. Dafür muss man die Brown-Out-Detection des Atmega herabsetzen. Den auf den Boards des SOFEX Lehrganges ist das bereits

geschehen.

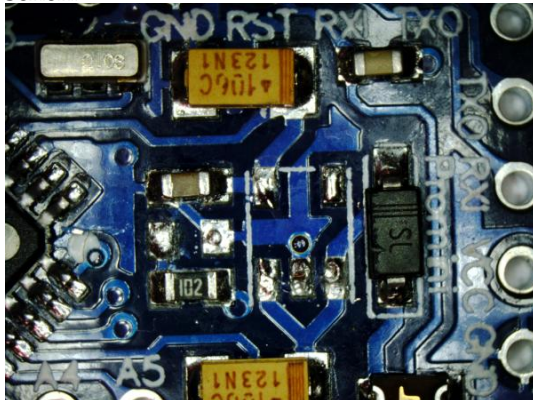
Dafür braucht man einen AVR Programmer, man kann auch einen zweiten Arduino benutzen. Man muss in der Datei `boards.txt` unter *Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328* die Zeile `pro.menu.cpu.8MHzatmega328.bootloader.extended_fuses=0x05` durch `pro.menu.cpu.8MHzatmega328.bootloader.extended_fuses=0x06` ersetzen. Und dann den Bootloader neu laden, eine Anleitung dazu findet sich hier <https://www.arduino.cc/en/Hacking/MiniBootloader>.

Die Änderungen entfernen den Verpolungsschutz des Arduinos und erlauben ihm unter 3.3 V zu laufen.

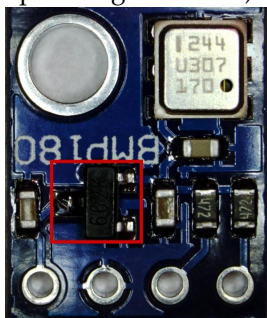
Auf dem Arduino muss man den Spannungsregler und die Power-LED entfernen. Dies ist die Veränderung die am meisten Strom spart:



Schneide die Beinchen des Spannungsreglers und der LED ab, mithilfe des Cutter-Messers, bis man beides entfernen kann. Dann entfernt man mit dem Lötcolben den Rest der Beinchen von den Löt pads. Danach sollte es so aussehen:



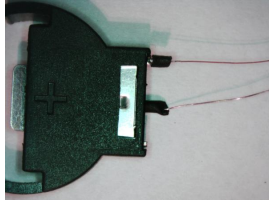
Auf der Platine des BMP180 muss man auch den Spannungsregler entfernen (das spart kaum Strom erlaubt dem BMP180 aber mit geringerer Spannung zu laufen):



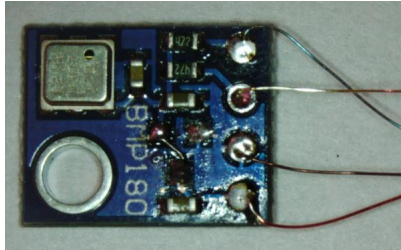
Um später die Lötösen für die Spannungsversorgung benutzen zu können, muss man den VIN Pin mit dem ehemaligen VOUT pin des Spannungsreglers verbinden, dafür den kleinen gelb isolierten Draht abisolieren und auflöten.

Danach sollte es so aussehen:

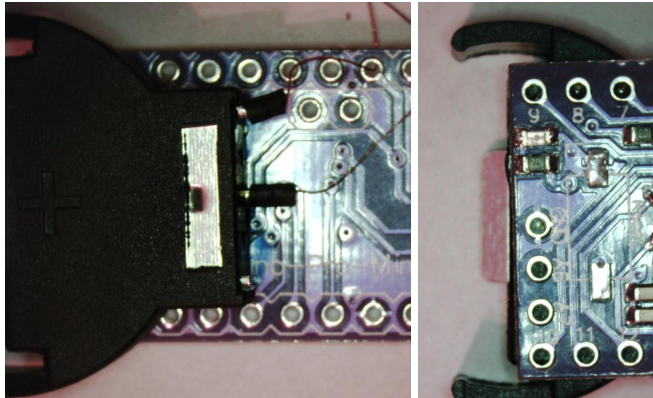
Nun fangen wir mit dem Batteriehalter an. Schneide eines der positiven Anschlussbeinchen ab, löte ein rotes Stück Lackdraht an den Pluspol und ein schwarzes an den Minuspol und isoliere beide Anschlüsse mit einem kleinen Stück Schrumpfschlauch.



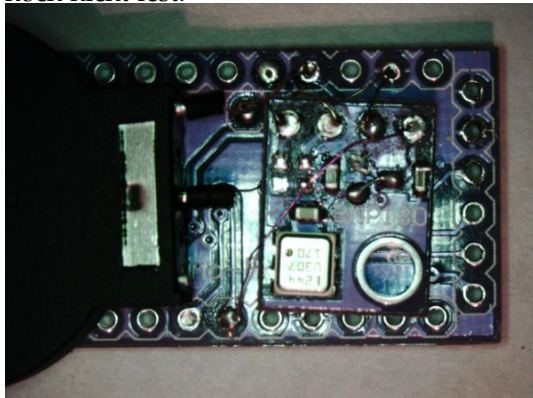
Als nächstes Löten wir 4 Drähte an den BMP180:
Schwarz an den Minuspol (GND), Rot an den Pluspol (VIN), grün an SDA und gelb an SCL.



Im nächsten Schritt kleben wir den Batteriehalter mit Heißkleber an die Rückseite des Arduino, (an die Seite an der der Schalter war). So dass der hauptteil des Batteriehalters bündig mit dem Arduino abschließt und der Batterie-Pulspol weiter von der Platine weg liegt.



Nun setzen wir den bmp180 auf die Rückseite der Platine kleben ihn aber noch nicht fest.



Nun fädelt man die Drähte durch die Lötösen auf der Arduino Platine bmp180:

BMP VIN (rot) an arduino digitaler Pin 3

BMP GND (schwarz) an arduino GND an der langen Seite der Platine
nicht an den UART

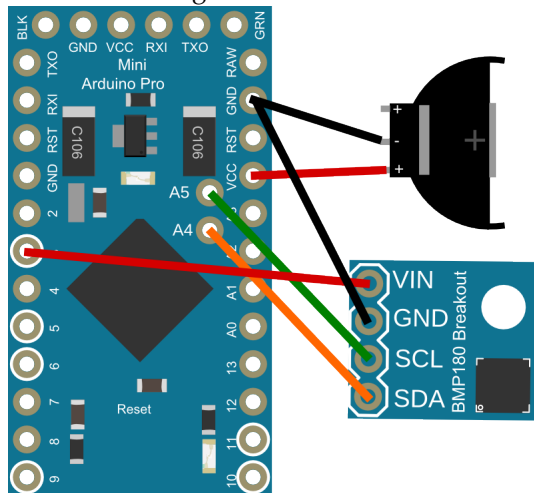
BMP SCL (grün) an arduino A5

Batterie:

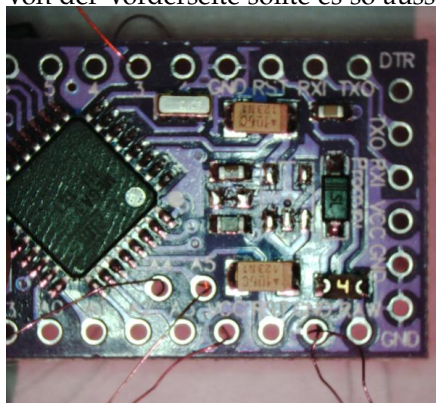
Batterie + (rot) an arduino VCC (3.3V) an der langen Seite der Platine
nicht an den UART

Battery - (schwarz) an arduino GND an der langen Seite der Platine
nicht an den UART

Die Verdrahtung sollte so aussehen:



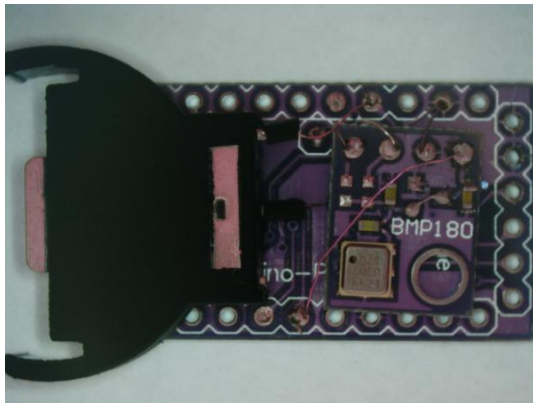
Den BMP VIN Pin an einen digitalen Pin am Arduino anzuschließen, ermöglicht es dem Arduino den BMP180 abzuschalten um Strom zu sparen. Von der Vorderseite sollte es so aussehen:



Nun klebe den BMP180 mit Heißkleber fest, benutze genug Heißkleber um eine Isolierung zwischen den beiden Platinen zu schaffen und lass die UART pins an der kurzen Seite des Arduinos frei.

Ziehe die Drähte nun angemessen kurz und löte sie fest. Löte lang genug um die Isolierung durchzubrennen. Wenn ein Multimeter zur Hand ist kann man die Verbindungen jetzt überprüfen.

Am Ende sollte es so aussehen:



Wenn alles funktioniert kann man die überstehenden Enden der Drähte abschneiden und alle Lötstellen vom Fluss säubern.

3 PROGRAMMING

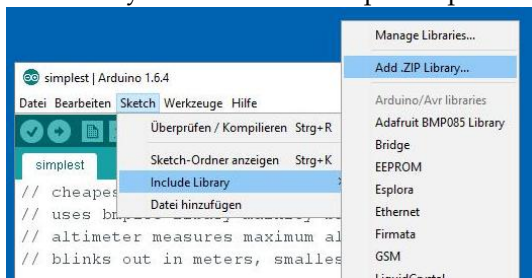
Der Altimeter basiert auf einem Arduino und kann mit der Arduino IDE programmiert werden.

Kopiere die IDE und installiere sie:

<https://www.arduino.cc/en/main/software>

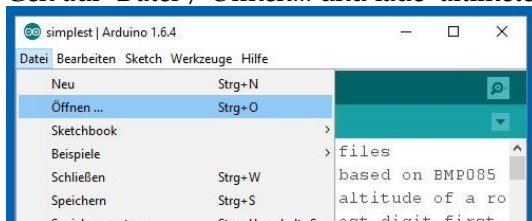
Kopiere außerdem das Programm für den Altimeter und die Bibliothek für den BMP180.

Um die Bibliothek zu installieren geh auf 'Sketch / Include Library / Add .ZIP Library ...' und wähle 'bmp180.zip'.

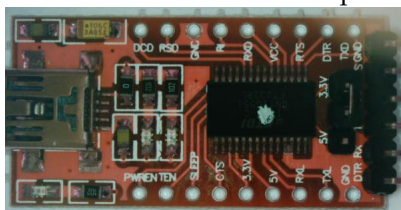


Bei Problemen mit der Installation findet man hier Hilfe <https://www.arduino.cc/en/pmwiki.php?n=Guide/Libraries>

Geh auf 'Datei / Öffnen...' und lade 'altimeter.ino'.



Schließe den FTDI USB Adapter an den USB Port an.

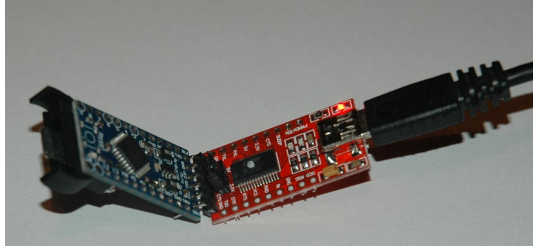


Es muss ein 3.3V Adapter sein. Das pin Layout am Adapter sollte zum arduino pro mini passen:

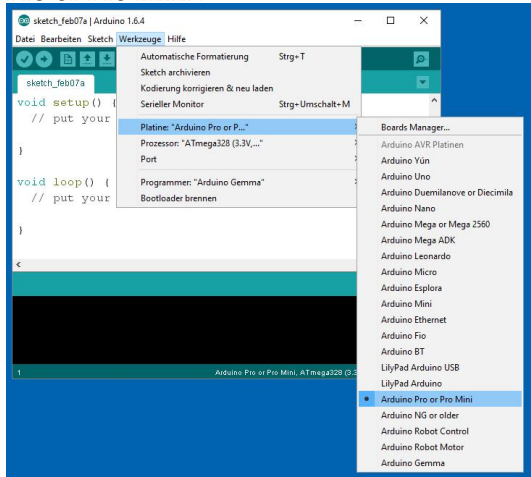
DTR TXo RXi VCC GND

manchmal sind TXo und RXi so beschriftet wie auf dem Arduino, manchmal ist die Beschriftung auch vertauscht (entsprechend ihrer Funktion auf dem Adapter) Der Adapter sollte eine Steckbrücke haben, steckt diese einfach in

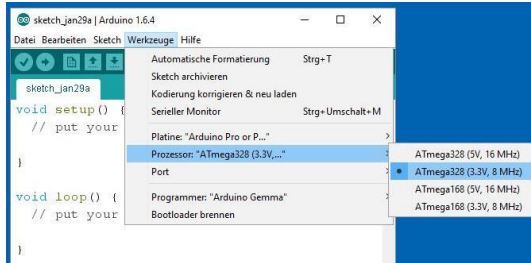
die passenden Löcher vom Arduino und kipp den Adapter so dass die Stifte Kontakt mit den Lötösen auf der Platine haben. (Mit dem roten Adapter sollten die Vorderseiten des Adapters und des Arduino in die selbe Richtung zeigen)



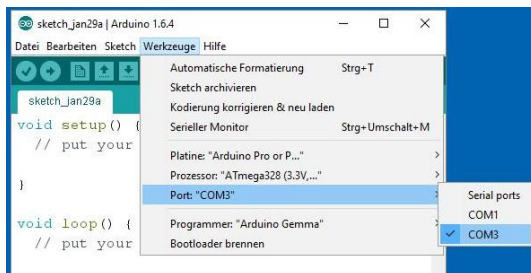
In der arduino IDE wählt unter 'Werkzeuge' als 'Platine' den 'Arduino Pro or Pro mini'.



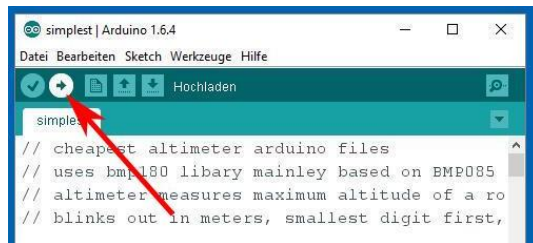
Als 'Prozessor' wählt ihr 'Atmega328 (3.3V 8Mhz)'.



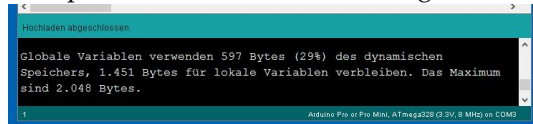
Und unter 'Port' wählt ihr den Seriellen Port des Adapters.



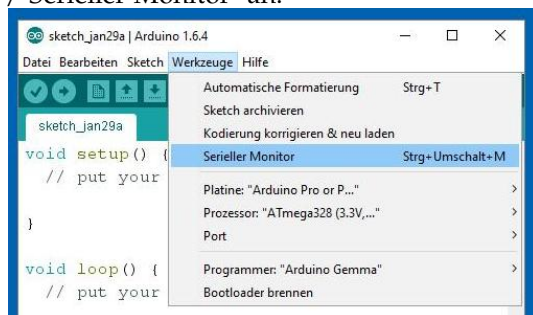
Falls ihr euch nicht sicher seid, welcher Port zum Adapter gehört entfernt das USB Kabel. Der Port welcher verschwindet gehört dem Adapter. Ladet das Programm auf den Arduino, indem ihr den rechten Pfeil anklickt (zweites Symbol von links).



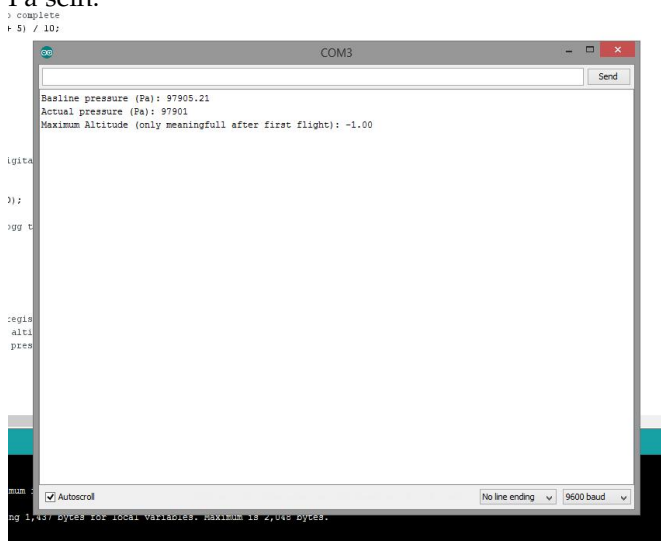
Überprüfe dass das die Stausanzeige 'Hochladen abgeschlossen' anzeigt.



Um zu überprüfen ob der Altimeter funktioniert, schließen wir ihn jetzt an den FTDI-Adapter an und wählen den richtigen Port und klicken 'Werkzeuge / Serieller Monitor' an.



Wähle 9600 baud in der rechten unteren Ecke, auch wenn es schon ausgewählt ist, dass startet den Altimeter neu. Nach ein paar Sekunden wird der Altimeter den Umgebungsdruck ausgeben. Überprüft ob das Ergebnis sinnvoll ist, es ist in Pascal und sollte nah am Atmosphärendruck von 101300 Pa sein.



Der Altimeter gibt auch die Maximalhöhe des letzten Fluges aus, das ist natürlich nur sinnvoll falls schon ein Flug stattgefunden hat.

Der Altimeter ist nun fertig zur Benutzung.

Setzt die Batterie in den Halter ein, achtet auf die Polarität, falschrum einsetzen kann den Altimeter zerstören. Bei Einsetzen der Batterie blinkt er einmal kurz. Nach 5 Sekunden wird der Altimeter anfangen die letzte gemessene Höhe in Metern auszugeben. Er fängt mit der letzten Ziffer (Einser) an, 10 mal blinken steht für 0, nach einer 2 Sekündigen Pause folgt die nächste Ziffer (Zehner) und so weiter.

Wenn die Batteriespannung zu gering ist wird der Altimeter anfangen sehr schnell zu blinken. Der Altimeter hat eine 30 Sekunden Verzögerung um den Einbau in die Rakete zu ermöglichen. Nach diesen 30 Sekunden misst er ein paar mal den Druck um die aktuelle Bodenhöhe zu bestimmen. Dann wird der Altimeter sehr kurz alle 5 Sekunden blinken um anzuzeigen, dass er eingeschaltet ist. Der Altimeter misst den Druck mit 64 Hz, berechnet die Höhe und wendet das Kalman Filter an. Nach dem der Start detektiert wurde (beim überschreiten von 20 m Höhe) wird die maximale Höhe kontinuierlich ermittelt. Nachdem 100 Werte mit sinkender Höhe vom Filter ausgegeben wurden wird festgestellt, dass der Gipfelpunkt überschritten wurde. Dann wird die Maximalhöhe im EEPROM gespeichert um sie nach der Landung auszugeben.

Dann schaltet der Altimeter den BMP180 aus und schickt den Prozessor in den Schlafmodus um Strom zu sparen. Der Verbrauch ist dann 18μ , was dem Altimeter erlaubt 1 Jahr in dieser Phase zu bleiben bis die Batterie leer ist.

Nach der Bergung entfernt man die Batterie vom Altimeter und setzt sie wieder ein (Polarität beachten), der Altimeter wird dann die letzte Maximalhöhe ausblinken. Um den Altimeter auszuschalten muss man die Batterie entfernen.