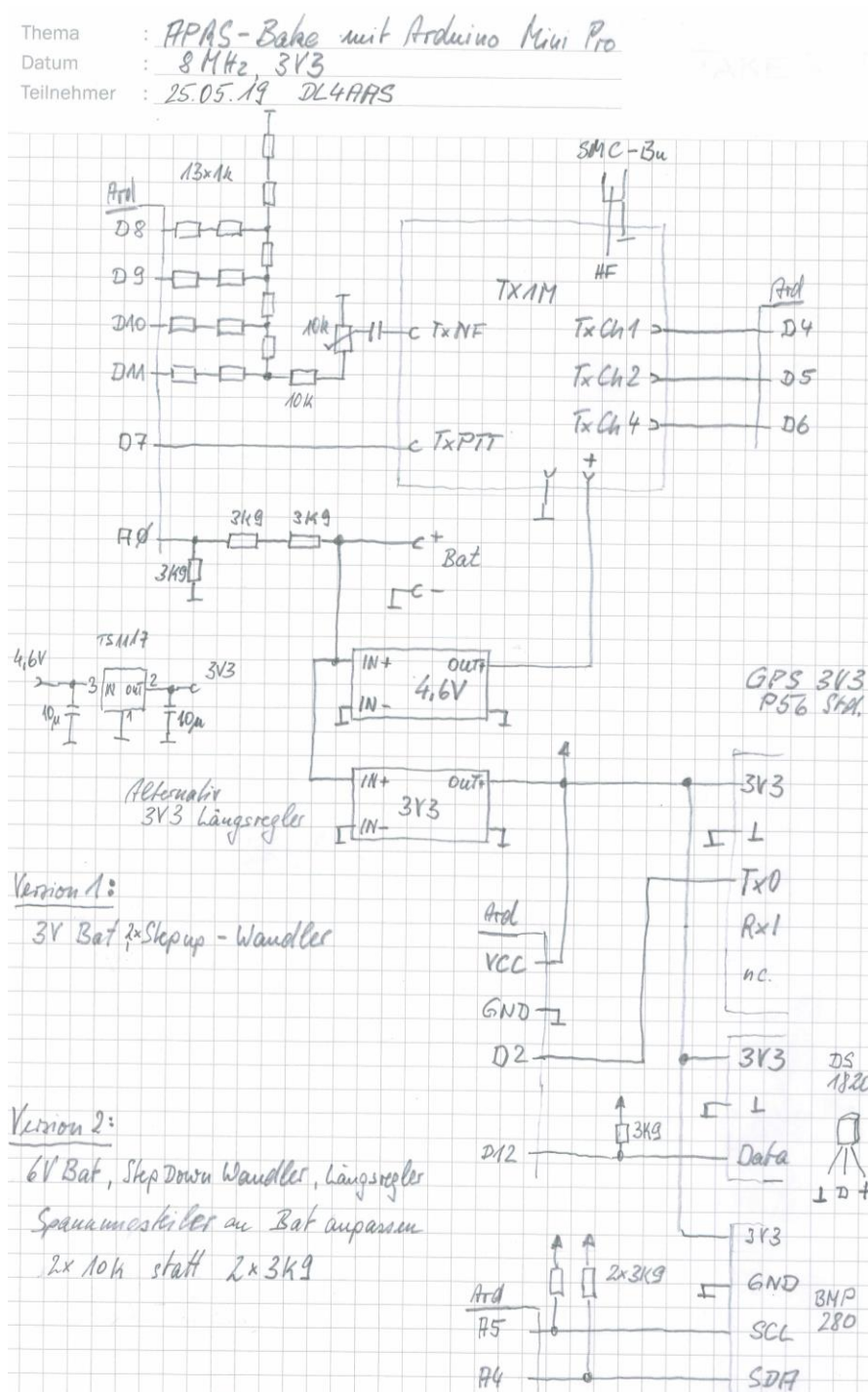


## Die APRS-Bake DLOTTM-2

Einfach sollte sie sein und mit Standard-Bauteilen und auch fertigen Bibliotheken aus dem Arduinobereich auskommen. Als Betriebsspannung habe ich 3V3 gewählt, um Leistung zu sparen. Demnach wurde der Arduino Pro Mini<sup>1</sup> auch nur mit 8MHz getaktet – ohne erkennbare Folgen. Der Schaltplan liegt derzeit nur als Handskizze vor, verwendet wurde Version 2:



Das Sendemodul, ein TX1M<sup>2</sup> von Radiometrix, fällt aus dem Rahmen. Es ist relativ teuer und auch groß, war aber vorhanden und erprobt, Frequenzwechsel per Pin inklusive. Als Antenne kam diesmal ein einfachster Dipol zu Einsatz: je 51,7cm Draht an Seele und Masse, nach unten und oben aus dem

<sup>1</sup> Arduino Pro Mini: <https://arduino-projekte.info/arduino-pro-mini/>

<sup>2</sup> TX1M: <https://www.radiometrix.com/node/180>

Gehäuse geführt und mit dem Tragseil verbunden. Das Sendemodul benötigt mindestens 4,5V, was sich auf das Design der Spannungsversorgung auswirkte: Die Batteriespannung, die zwischen 7 und 28V<sup>3</sup> liegen kann, wird durch einen Schaltregler<sup>4</sup> auf 4,5V heruntersgesetzt, die 3,3V entstehen durch einen nachgeschalteten Längsregler (Version 2 im Schaltplan). Da die Hauptlast der Energieversorgung mit ca. 60mA durch den GPS-Empfänger entsteht, könnte auch ein zweiter Schaltregler sinnvoll sein. Es stellte sich heraus, dass der verwendete Regler bei Spannungen unter 6V nicht mehr richtig arbeitete, deswegen mussten 3 Batterien CR123A verwendet werden – und damit war ausreichend Kapazität vorhanden. Dabei ist auch zu berücksichtigen, dass die Batteriespannung in der Kälte einbricht.

Der Hauptzweck der Bake ist das Wiederfinden der Nutzlast, weswegen auf Einfachheit und damit Funktionssicherheit Wert gelegt wurde. Der GPS-Empfänger stammte aus vorherigen Projekten und nutzt einen älteren ublox-Chipsatz. Er wurde nicht auf die Platine integriert, wegen der Austauschbarkeit und auch wegen der mechanischen Freiheiten, die für die Positionierung der Antenne hilfreich ist.

Über GPS-Empfänger und Ballonnutzung wurde schon viel geschrieben, die Höhentauglichkeit ist nicht bei jedem Modell gegeben. Zu 100% ist es nicht sicher, aber auch der diesmal verwendete könnte damit seine Probleme haben: Bei ca. 16300m Höhe fiel die Positionsmeldung aus, nur die Telemetriedaten wurden weiter gesendet. Ursache könnte aber auch die Kälte sein, an einer eher wärmeren Stelle der Nutzlast waren es -7°C. Dagegen spricht aber, dass die Positionen zu einer Zeit wiedereinsetzten (bei 11160m Höhe), als es noch kälter war. Auch ein Software-Thema kann natürlich nicht ausgeschlossen werden. Auf jeden Fall lieferte das GPS-Modul noch Zeitinformationen, denn an diesen hängt auch die Telemetrie-Aussendung. Die Anbindung an den Pro Mini geschieht mit 9600Bd und einem Soft-UART, verwendet werden die NMEA-Pakete GGA und VTG. Diese werden von den meisten GPS-Empfänger geliefert, ggf. nach Umkonfiguration. Die Auswertung übernimmt die mächtige Bibliothek neoGPS<sup>5</sup>. Diese sucht sich die Daten auch aus mehreren Paketen zusammen und meldet sich erst bei Vollständigkeit. Sie muss dazu aber auf den verwendeten Datenstrom konfiguriert werden, um das beim richtigen Paket zu tun.

Der Sekundentakt, mit dem die Pakete eintreffen, liefert auch den Steuertakt für die weiteren Aktivitäten: Ein einfacher Scheduler, der den Wert der Sekunde nutzt, aktiviert die APRS-Aussendungen, das Abfragen der Sensoren und auch die Frequenzwechsel. Leider geht die Übersichtlichkeit etwas verloren, weil die Konfiguration über defines erfolgt. So ist es aber möglich, verschiedene Zeitpläne vorzukonfigurieren und an zentraler Stelle vor dem Übersetzen und Herunterladen einzuschalten. Denn die Bake muss sich an ein abgestimmtes Zeitschema halten, um nicht gleichzeitig mit anderen Teilen der Nutzlast zu senden. Auf dieser Basis fällt auch die Abstimmung mit anderen Baken-Entwicklern leicht.

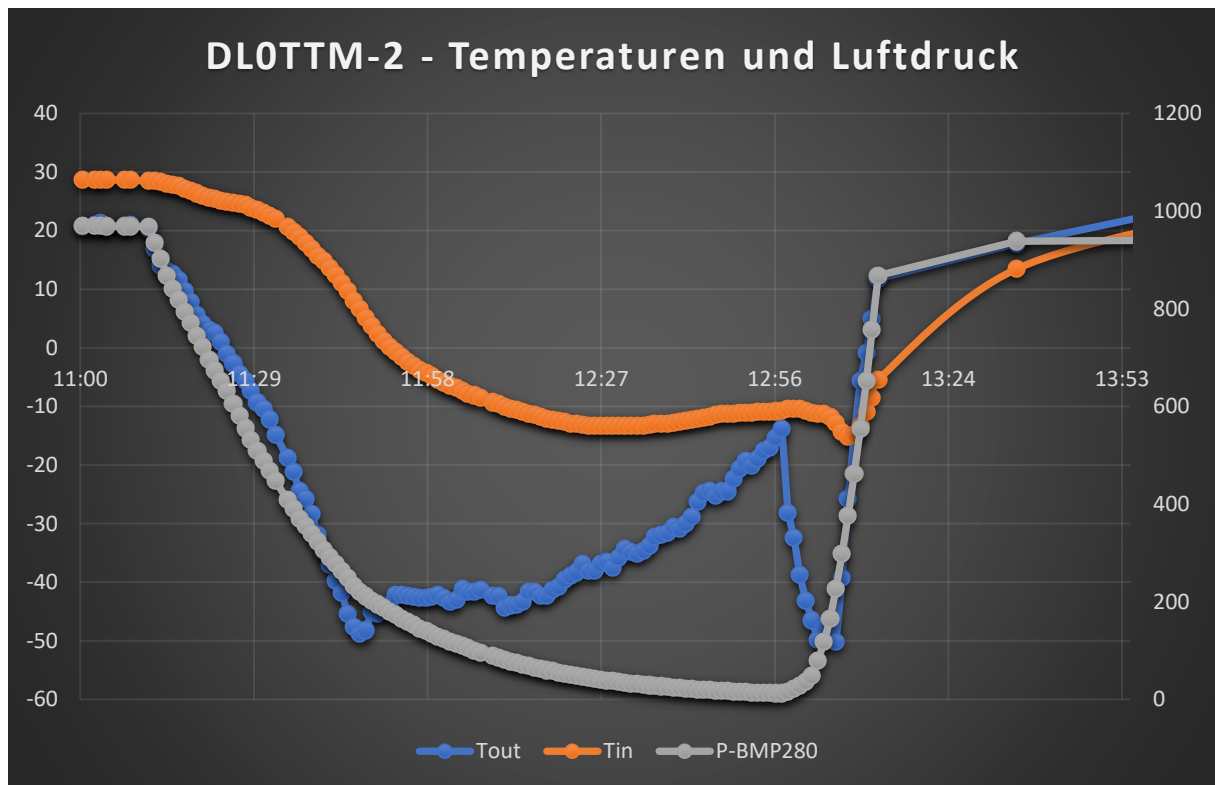
Neben der Kernfunktion ‚Position senden und übermitteln‘ hat die Bake dann aber doch etwas Sensorik bekommen: Ein 1-Wire-Temperatursensor DS18B20 für die Außentemperatur. Das Messen dieses Wertes stellt wegen des geringen Luftdrucks und der damit geringen Wärmekopplung zur Umgebung ein Problem dar. Bisher hatten wir immer viel zu hohe Werte, vermutlich durch die Eigenerwärmung des Sensors unter Spannung. Diesmal wurde die Spannung erst kurz vor dem Messen (1x pro Minute) ein- und gleich danach wieder ausgeschaltet. Die Werte sehen deutlich besser aus:

---

<sup>3</sup> Der Spannungsteiler zur Messung der Batteriespannung ist auf max. 14V ausgelegt

<sup>4</sup> StepDown-Wandler mit MP1584: <https://www.google.com/search?q=MP1584>

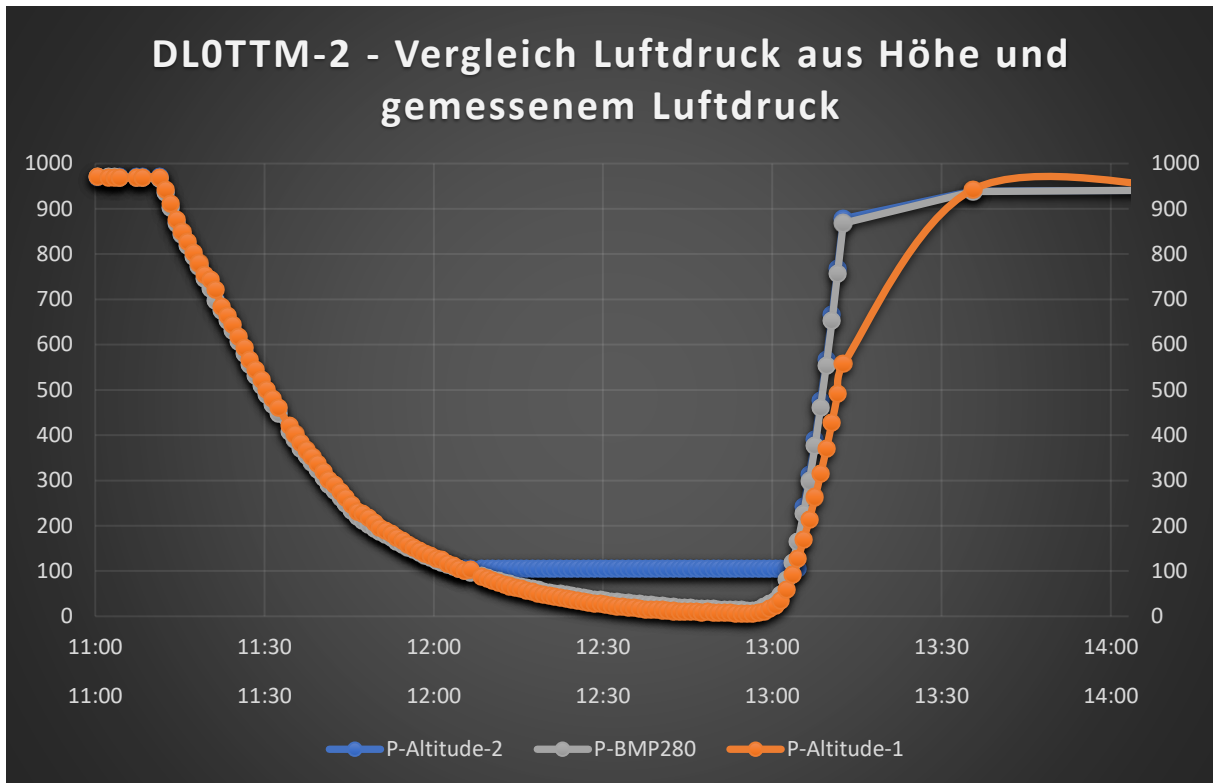
<sup>5</sup> neoGPS: <https://github.com/SlashDevin/NeoGPS>



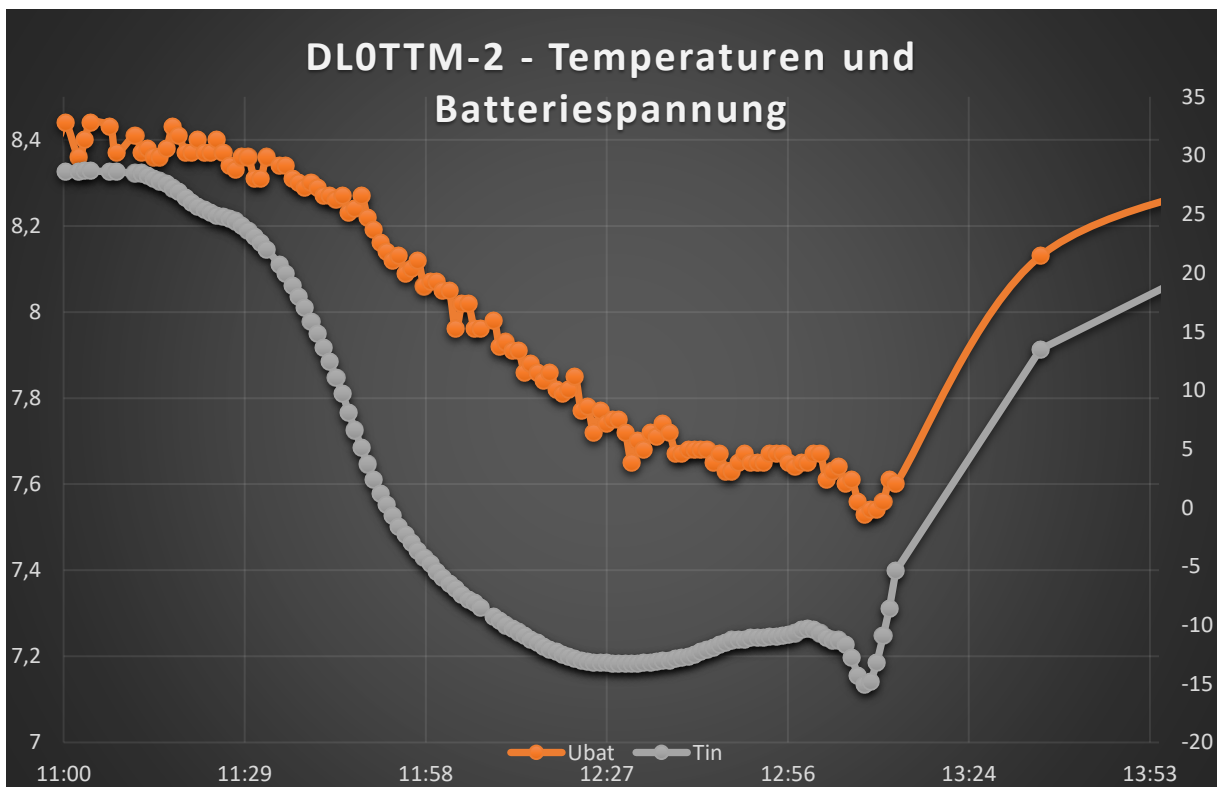
Als zweiter Sensor kam ein BMP280<sup>6</sup> zum Einsatz, wohl wissend, dass er nur bis 300mbar und damit bis ca. 9km Höhe spezifiziert ist. Die Anbindung erfolgte per I<sup>2</sup>C-Bus<sup>7</sup>. Die Ergebnisse in größeren Höhen erscheinen aber trotzdem gut und plausibel, für einen Abgleich mit der manchmal unzuverlässigen GPS-Höhe reicht es auf jeden Fall.

<sup>6</sup> BMP280: <https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf>

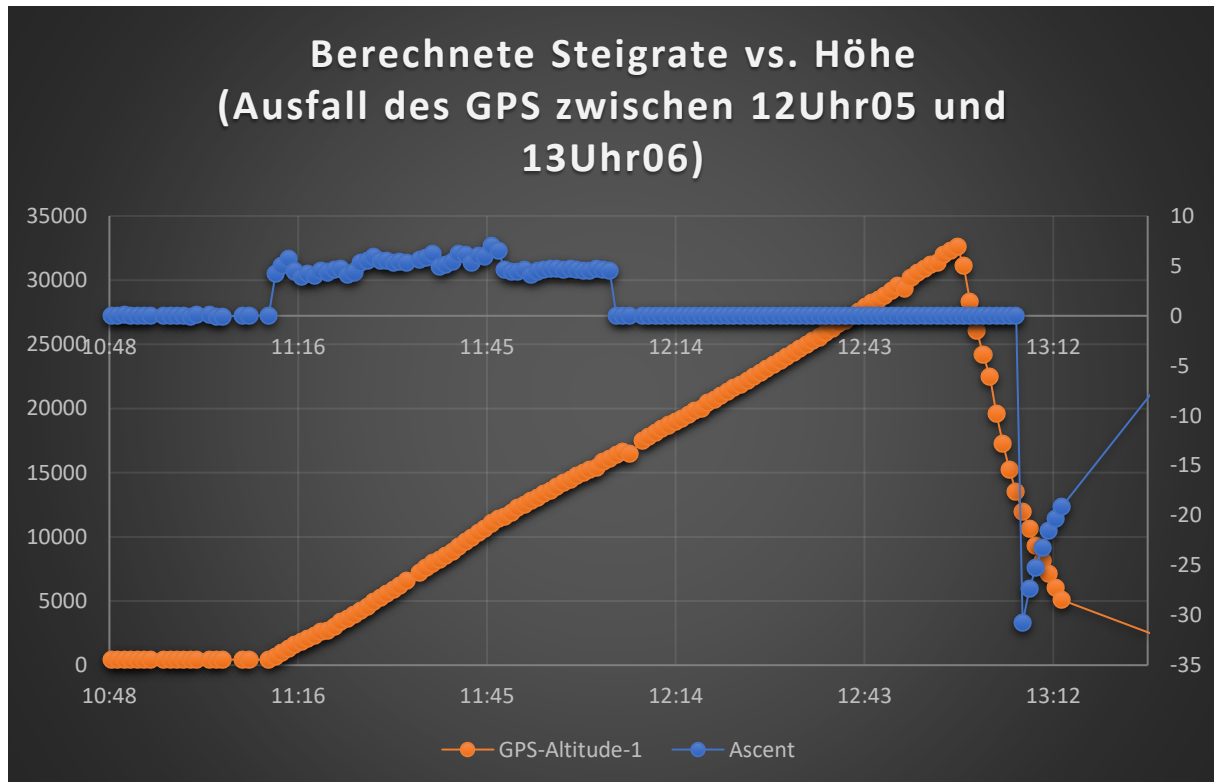
<sup>7</sup> Der Entwickler des Breakout-Moduls wählte die Nicht-Standard-I<sup>2</sup>C-Adresse 0x76, ein Scan aller Adressen brachte das zu Tage.



Der integrierte Temperatursensor misst in der Baken-Box und gibt so eine Idee über die Belastungen für die Elektronik und besonders für die Batterie. Diesmal passten Isolierung und Verlustleistung nicht gut zusammen: mit  $-15^{\circ}\text{C}$  in der Sturzphase war es definitiv zu kalt in der Box. Bei der Batteriespannung, gemessen über einen Spannungsteiler am Analogeingang, zeigt sich ein direkter Zusammenhang zur Temperatur, die Entladung fällt kaum auf:



Neben den 4 gemessenen Werten wurde per APRS-Telemetrie noch ein fünfter übermittelt: Die Steigrate, minütlich aus den Höhendifferenzen der GPS-Daten errechnet. Diese hilft bei der Anpassung der Vorhersagen und muss so nicht mühsam am Boden ermittelt werden.



Insgesamt hat die Nutzung der APRS-Telemetrie gut funktioniert. Alle Daten der vorherigen Auswertungen wurden aus den von APRSIS mitgeschriebenen Paketen extrahiert. Die sofortige Darstellung der Telemetrie auf [aprs.fi](http://aprs.fi) gibt ersten Aufschluss, aber dedizierte Auswertungen müssen dann doch im Nachgang erfolgen.

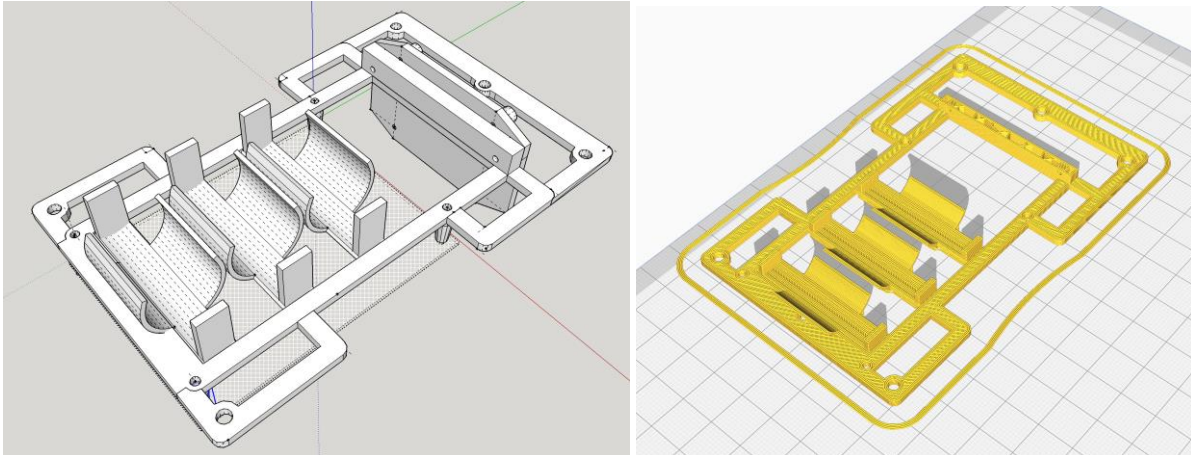
Für die Aussendung der APRS-Pakete wurde die Bibliothek `ArduinoQAPRS`<sup>8</sup> verwendet, auf die ich über Klaus, DJ700, stieß. In ihrer letzten Version arbeitet sie auch auf einem mit 8MHz getakteten Prozessor. Allerdings ist die Library wohl für sehr langsame Sendemodule wie DRA818 geschrieben, denn zum eingestellten `TxDelay` fügt sie noch 433ms hinzu – das war nicht zu überhören und wurde abgeschaltet. Außerdem muss während der Aussendung der GPS-Soft-UART abgeschaltet werden, da er das Timing stört. Die Library kann mit unterschiedlichen DA-Wandlern betrieben werden, die Bake nutzt einen R2R-Wandler an 4 digitalen Ausgängen.

Die komplette Schaltung wurde auf einer Lochrasterplatine aufgebaut. Durch Verwendung von SMD-Widerständen hielt sich nicht nur der Platzbedarf u.a. des DA-Wandlers in Grenzen, sondern verringerte sich auch die Bauzeit erheblich. Zur Verdrahtung setze ich neben blankem Draht schon seit Jahren Fädeldraht mit Lackisolierung ein, ohne Fädeldämme. Bei richtiger Anwendung (vorverzinnen!) entstehen funktionssichere und platzsparende Aufbauten.

Bei der Mechanik wurden diesmal neue Wege beschritten: Platine, GPS-Empfänger und die drei per Litze verbundenen CR123A wurden durch einen 3D-gedruckten Rahmen fixiert. Um diesen Rahmen herum wurde ein passendes Gehäuse aus Styrodur geschnitten und geklebt. Der Rahmen hatte auch

<sup>8</sup> `ArduinoQAPRS`: <https://bitbucket.org/Quon/arduinoqaprs/src/master/>

Löcher zum Führen des Tragseils durch das Gehäuse – ohne Unterbrechung. Das Ergebnis ist eine leichte, aber sehr robuste Bake, die sofort wieder eingesetzt werden könnte.



Wie geht es nun weiter mit dieser Bake? Der Ansatz, auf Standard-Hard- und Software zu setzen, hat sich bewährt und soll fortgesetzt werden. Geplant ist eine Weiterentwicklung zum Servicemodul für Ballonstarts – also keine spannenden Experimente, sondern eine solide Bake, die einfach ihren Dienst verrichtet. Ein paar Stichworte, was sich dazu verändern könnte:

- Geätzte Platine statt Lochraster
- Sender durch TRX-Modul SR\_FRS\_1WV ersetzen
- DMTF-Auswerter
- Schaltmodul, um Ballon abtrennen zu können
- Optische und akustische Signalgeber, um die Bergung zu erleichtern
- Ein- und Ausschalten der Box von außen

Auch in der bereits vorhandenen Software gibt es noch Verbesserungspotential:

- Übersichtlichere Konfiguration des Schedulers
- Notbetrieb bei Ausfall des GPS-Signals
- Signalisierung per Telemetrie, ob GPS-Fix vorhanden
- Bessere Statusanzeige per LED

Ob das allerdings umgesetzt wird, hängt vom nächsten Starttermin ab.

