

Arduino APRS Modem

DL1STL

Wir wollten im Klub einen APRS Tracker bauen. Dabei sollte das Gerät

- individuell programmierbar sein,
- Temperatur und Spannungen messen können,
- optional auch ein Display steuern,
- ein Relais ansteuern,
- und natürlich die Koordinaten eines GPS Moduls auswerten und damit ein Funkgerät ansteuern.

Die Webseite von DJ7OO zu seinen APRS Modem: <http://www.kh-gps.de/qaprs.htm> hatte mir da richtig gut gefallen. Ein guter Tip von Bernd (DL5WB). Dort war auf der Basis eines Arduino Nano eine nette kleine Schaltungsidee für eine APRS Bake zu finden, und noch wichtiger, auch die erforderliche Software war vorhanden. Das ganze war im Original von Lukasz, SQ5RWU erstellt.

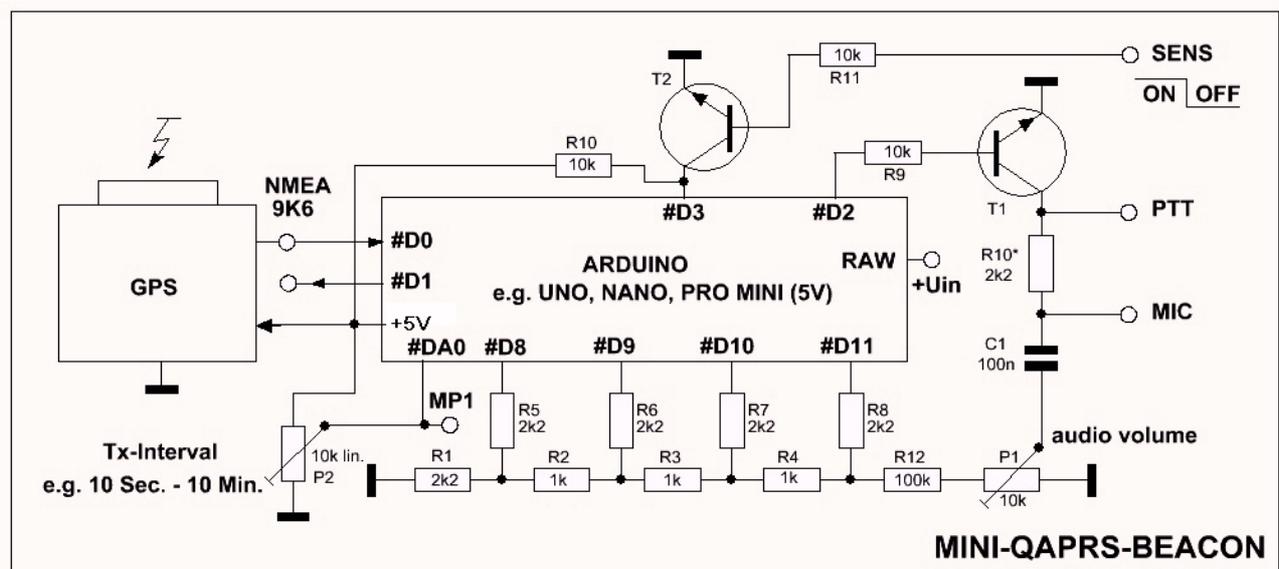


Abbildung 1: Schaltung der APRS Bake von DJ7OO

Wenn wir uns die Schaltung von DJ7OO (Abbildung 1) ansehen, ist leicht erkennbar, dass dort noch nicht alle Ressourcen des Controllers ausgeschöpft sind. Wir haben also genug Möglichkeiten, unsere gewünschten Erweiterungen unterzubringen. Also ging es an die Arbeit, aus der Vorlage der APRS Bake und unseren Ideen eine neue Schaltung zu bauen.

Als Controller wurde der Arduino Nano gewählt. Dieser wird einfach huckepack auf unsere Patine aufgelötet. Somit sind keine SMD Bauteile zu löten, und die ganzen restlichen Bauteile können mit bedrahteten Bauelementen realisiert werden. Die Leiterplatte wurde von der Topologie auch sehr übersichtlich und alles wurde auf nur einer Seite verdrahtet. Abbildung 2 zeigt die Schaltung, Abbildung 3 die Leiterplatte.

Hardware Umsetzung

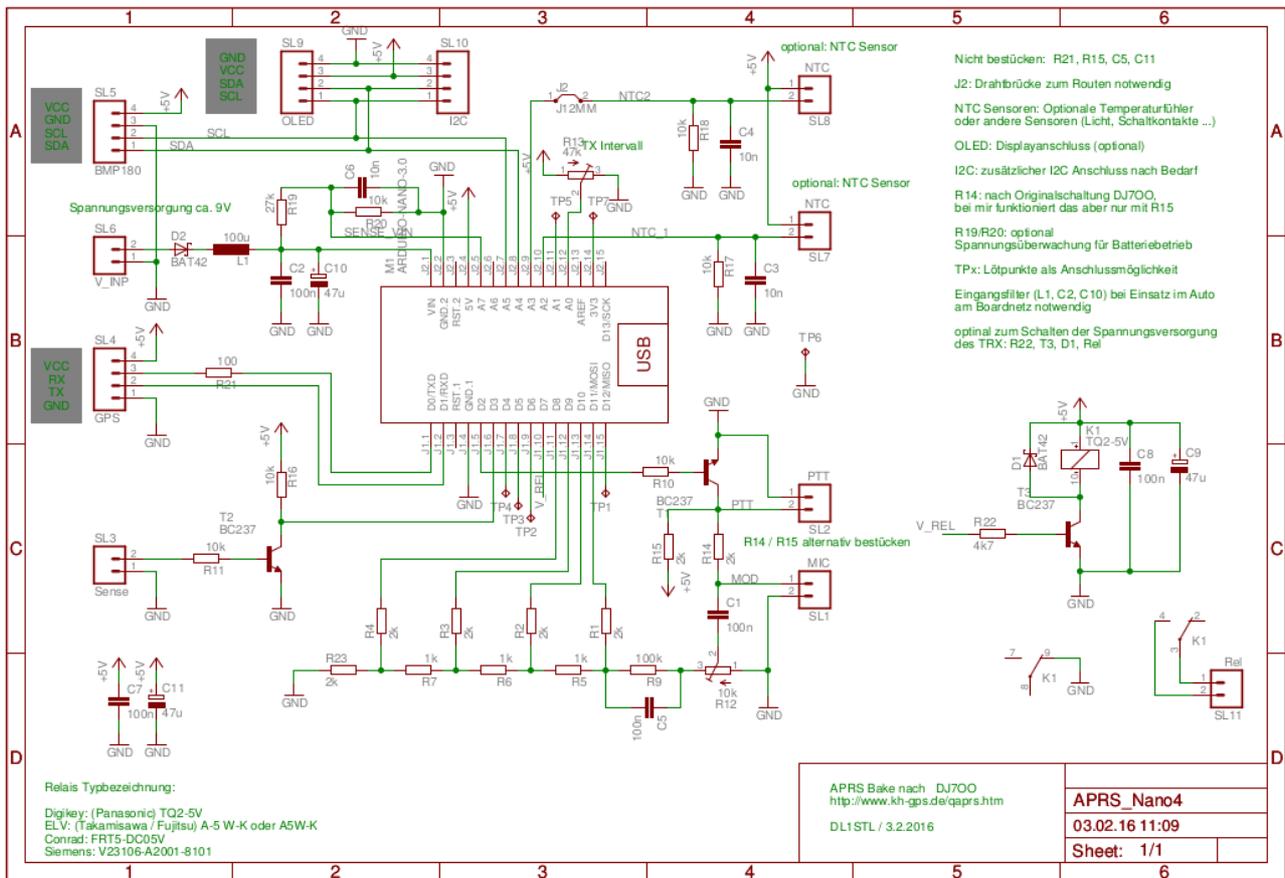


Abbildung 2: Schaltung des Arduino APRS Modem

In der Schaltung finden wir erst mal alle Komponenten der Vorlage wieder. Die Widerstände R1 bis R7 und R23 bilden hier einen einfachen 4 bit DA Wandler. Zusätzlich ist der Kondensator C5 in der Tonauskopplung dazu gekommen. Damit funktionierte die Sendung mit meinem Funkgerät besser. Die PTT Ansteuerung übernimmt der Transistor T1. Damit dieser korrekt funktioniert, muss in der Originalschaltung über die Mikrofonbuchse eine positive Spannung anliegen. Wenn das nicht der Fall ist, R14 nicht bestücken und R15 bestücken. Anders herum braucht dann R15 dann nicht rein. Neu ist die Ansteuerung des Relais über T3. Mit diesem Relais kann später beim Betrieb als Bake z.B. die Spannungsversorgung des Funkgeräts geschaltet werden. Damit wird beim Batteriebetrieb die Stromaufnahme reduziert.

Rechts oben in der Schaltung sind dann zwei Anschlüsse, die für zusätzliche Temperatursensoren verwendet werden können. Aber auch andere Sensoren sind möglich, wie beispielsweise Photowiderstände, Schalter, Feuchtesensoren u.s.w. Die Kondensatoren C3 und C4 blocken an diesen Eingängen HF Störungen ab.

Dann ist auf unserer Platine direkt ein Sensor BMP85 oder dessen Nachfolger BMP180 bestückbar. Diese sind als fertige Platine erhältlich. Wir kommen als wieder um das SMD Löten herum. Beide Module sind in der Ansteuerung über I2C hardwaremäßig und auch softwareseitig kompatibel. Zwei Anschlüsse des I2C Busses sind dann noch auf Steckverbinder gelegt. Auch hier können wieder zusätzliche Sensoren oder auch ein Display angeschlossen werden. Der I2C Bus bietet da viele Möglichkeiten.

Die externe Spannungsversorgung wird mit einer Diode gegen Verpolung abgesichert und dann noch zusätzlich mit L1, C2 und C10 gefiltert. Der eigentliche Spannungsregler befindet sich dann

auf dem Arduino Board. Beim Betrieb mit 12V gibt es thermische Probleme, so dass davon abzuraten ist. Bei 7 bis 9 V funktioniert die Spannungsversorgung optimal. Die Eingangsspannung ist über den Spannungsteiler R19 / R20 an einen Analogeingang des Arduinos geführt, um diese überwachen zu können. Der Arduino nutzt zum Messen die 5V Betriebsspannung als Referenz. Somit ist mit dieser Dimensionierung eine maximale Spannung von 18,5V theoretisch messbar. Bei Spannungen unter 7V wird der Spannungsregler nicht mehr korrekt arbeiten und somit wird die Referenz unbrauchbar. Wir können bei Messungen unter besagten 7V lediglich erkennen, dass die Versorgungsspannung zu klein ist. Die Messwerte werden dann nicht mehr stimmen.

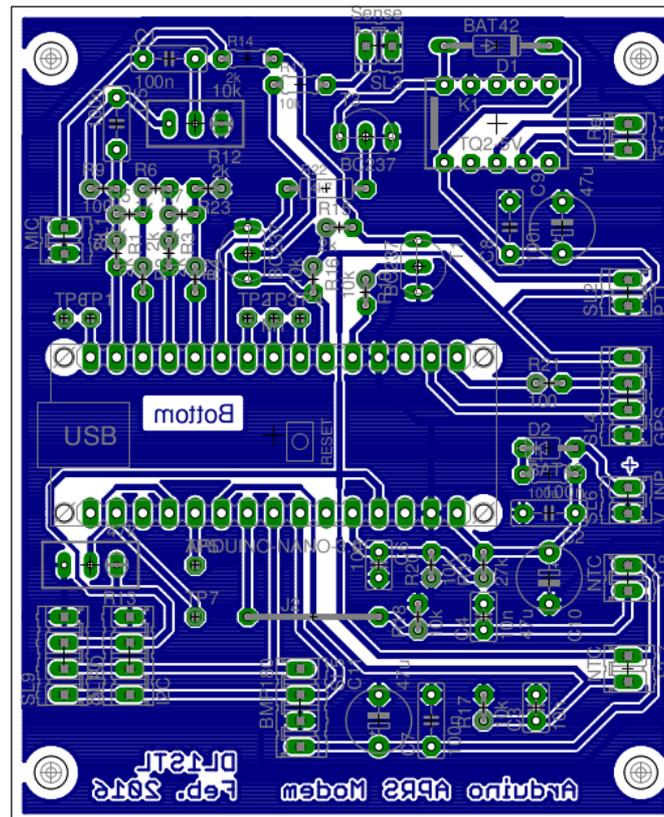


Abbildung 3: Leiterplatten Layout

Zur Störunterdrückung sind dann in der Schaltung noch mehrere Kondensatoren vorgesehen. Ich empfehle, diese auch alle zu bestücken. Den Arduino Nano kann man dann direkt bestücken werden oder mit Stiftleisten aufgesteckt werden. Damit ist dieser bei Bedarf schnell wechselbar. Der Widerstand R21 darf nicht bestückt werden.

Arduino Software

Da die Hardware des APRS Modems einfach gehalten ist, wird die Software dadurch etwas umfangreicher. Aber die Entwickler haben uns da das Leben leicht gemacht, indem alles ordentlich in eine extra Bibliothek verpackt ist.

Aber zuerst will ich die Programme zur Inbetriebnahme und des Hardwaretest beschreiben. Hierzu ist die Verbindung unserer Leiterplatte zum GPS Modul zu trennen. Zur Programmierung teilen sich dieses Modul und die Programmierschnittstelle die seriellen Schnittstelle, was dann leider zu

Problemen führt.

Zuerst ist also in der Arduino IDE unser Board und die Schnittstelle zu konfigurieren.

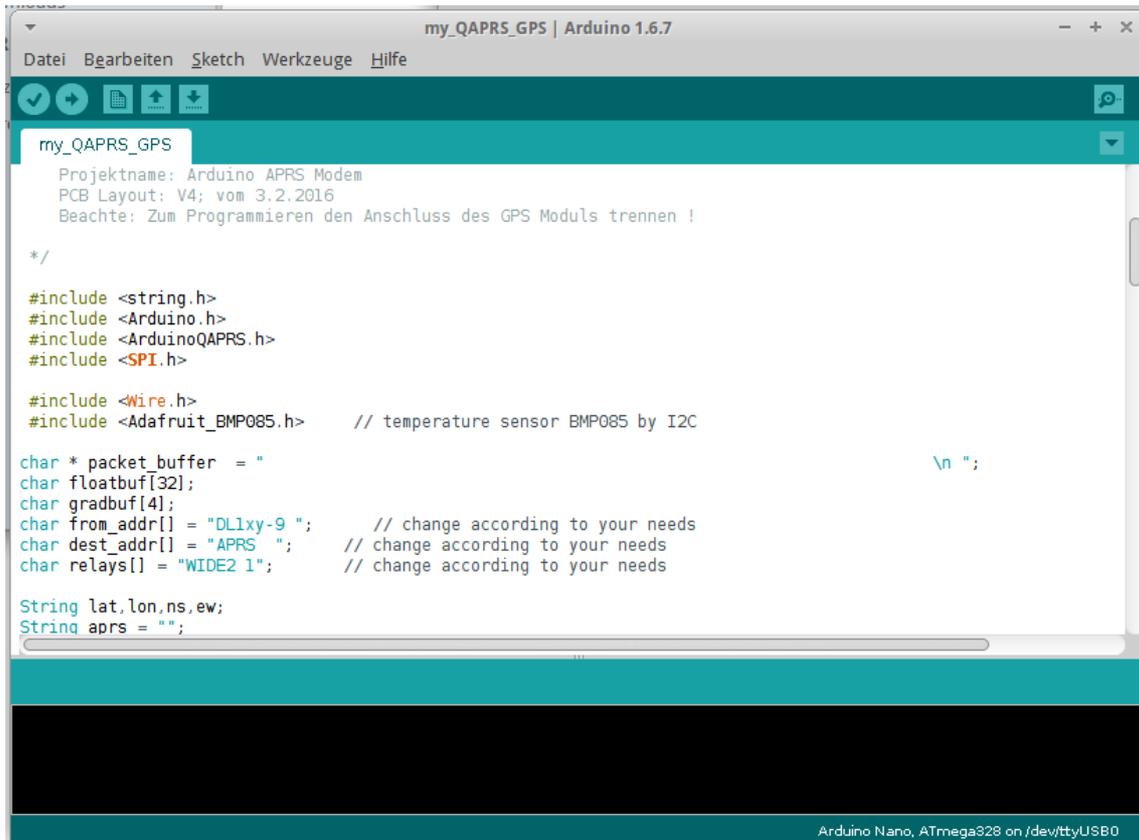


Abbildung 4: Arduino IDE

Abbildung 4 zeigt die Programmieroberfläche. Dort ist zuerst unter dem Menüpunkt *Werkzeuge* das Board und die Schnittstelle einzustellen. Mit den mitgelieferten Beispielprogramm Blink (→ Datei → Beispiele → Basic → Blink) kann ein erster Test des Arduino- Nano Boards erfolgen.

Für die einzelnen Komponenten auf den Board habe ich jeweils kleine Testprogramme erstellt. Diese haben alle die Kennung *Test* in Namen. Damit wird die Inbetriebnahme einfach und die jeweiligen Funktionen werden verständlich. Dazu ist jetzt mein Softwarepaket *qaprs_8.zip* erforderlich. Das Paket muss im Arduino Programmverzeichnis (üblich: ~/sketchbook) ausgepackt werden. Dann ist der Library- Ordner *ArduinoQAPRS* noch in den Libraryordner der Arduino- IDE zu kopieren (~/sketchbook/libraries).

Die einzelnen Programme bzw, Verzeichnisse sind jetzt:

- **ArduinoQAPRS:**
Die Bibliothek zur Ansteuerung des DA Wandlers als Modem. Sie muss, wie oben schon erläutert, in den libraries Ordner kopiert werden.
- **QAPRS_8:**
APRS Programm ohne Höhenausgabe (original Quellen)
- **QAPRS_9:**
APRS Programm mit Höhenausgabe (original Quellen)
- **my_QAPRS_GPS:**
angepasstes APRS Programm mit dem GPS Modul und dem BMP085 Temperatursensor
- **my_QAPRS_WX:**

APRS Programm ohne GPS mit BMP085 Temperatursensor zur Nutzung als Wetterstation

- Sense_Test:
Einfache Funktionskontrolle des Sense Eingangs
- BMP_Test:
Funktionskontrolle / Beispiel des BMP180 (085) Sensors
- PTT_Test:
Funktionskontrolle PTT; es wird in 10s Intervallen PTT aktiviert
- VDD_Test:
Test der Funktion der Eingangsspannungsmessung V_IN
- Test_Tone:
Ausgabe eines Test Tones über den DA Wandler. Am Mikrofonausgang ist hochohmig ein Ton hörbar.

Am Ende das Programm `my_QAPRS_GPS` oder `my_QAPRS_WX` zu installieren, je nachdem, ob die APRS Bake mobil oder an einen festen Ort verwendet wird. Im Quelltext ist dazu noch eine Anpassung des Rufzeichens und gegebenenfalls der Ausgabe erforderlich. Das Ergebnis unserer Bemühungen zeigt dann Abbildung 5. Viel Erfolg beim Nachbau.

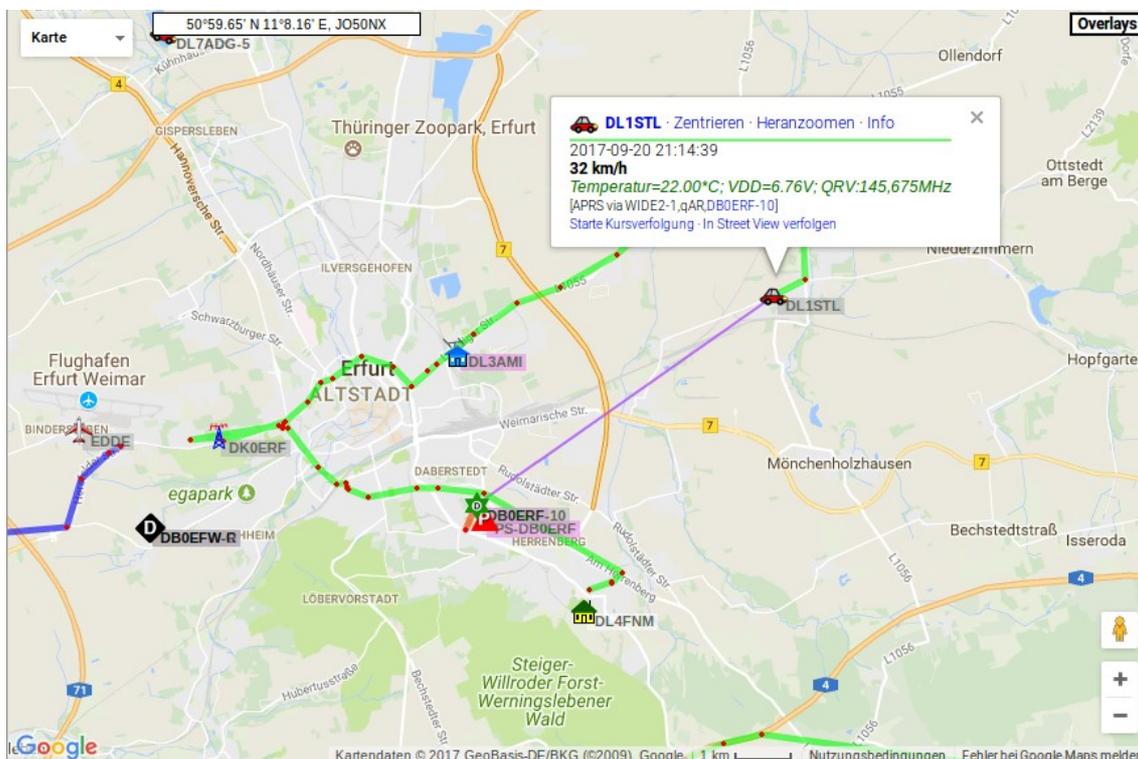


Abbildung 5: Darstellung der Ausgabe des APRS Modems von <https://aprs.fi>

Probleme:

- Einkopplung von HF in die Schaltung.

Weitere Ideen:

- Kopplung mit den 144MHz DORJI Modulen ([siehe AATiS Projekt](#))
- Verwendung der internen Referenz des AD Wandlers zur Verbesserung der Spannungsmessung