uTrak APRS-Implementierung

Da die Aussendung von APRS schon ein eher komplexes Thema mit der minimalistischen Hardware ist, soll die Implementierung hier dokumentiert werden.

Beschreibung

APRS wird im Amateurfunk in Form von Packet Radio ausgesandt. Ein AFSK-Signal wird frequenzmoduliert und auf regional koordinierten Frequenzen ausgesendet. Um die Abdeckung der Verfolgbarkeit des Trackers zu erhöhen, sollen Positionsaussendungen im APRS-Format ausgesendet werden. Da der Si4060 eigentlich keine frequenzmodulierten Aussendungen unterstützt, wird die FM in Software erledigt. Die Implementierung folgt ansonsten der AX.25-Spezifikation. Konkret werden folgende Parameter der Aussendung implementiert:

- Physical Layer
  - Mark-Frequenz: 1200 Hz
  - Space-Frequenz: 2200 Hz
  - Übertragungsrate: 1200 Baud
  - NRZI encoding
  - Bit-Stuffing (nach jeder fünften „1“ eine „0“ einfügen)
- Data-Link Layer
  - AX.25-Framing (HDLC)
  - Frame Check Sequence nach CRC16-CCITT

Implementierung

AFSK-FM

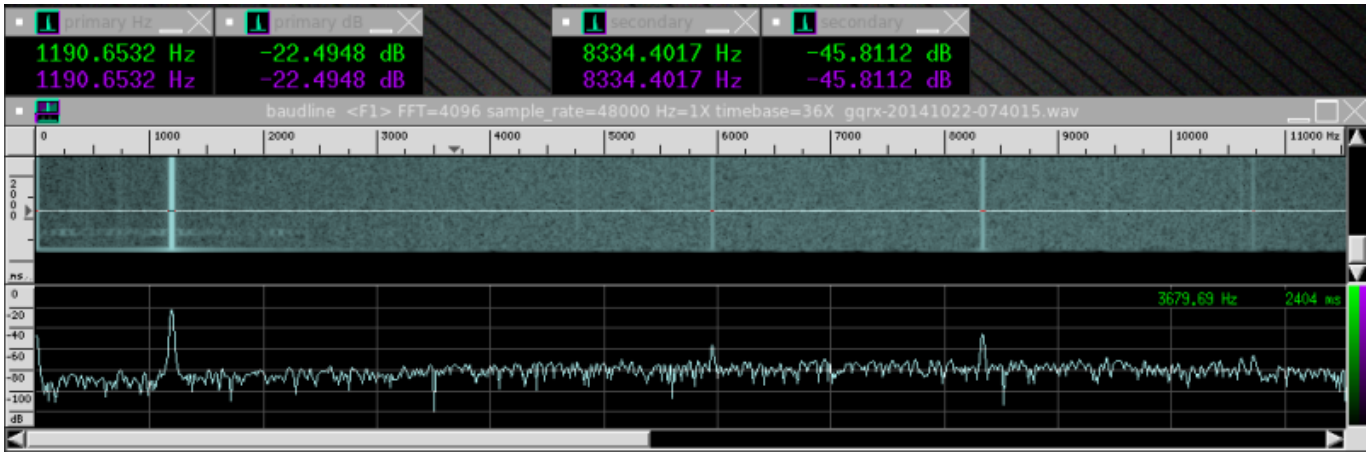
Methode 1 (alt): Digitale Modulation im Si4060-FSK-Offset-Register

Im MSP430 wird ein NCO implementiert, der mit möglichst geringem Fehler die nötigen Frequenzen erzeugt. Die Samplerate wird dabei aus dem vorhandenen Grundtakt so gewählt, dass der Fehler für Baudrate, Mark- und Spacefrequenz möglichst klein ist. Um dieses Optimum zu finden, wurde ein Matlab-Skript geschrieben, was nach Brute-Force-Methode den prozentualen Fehler für alle Einstellungen herausfindet (msp_fm.m). Ein Timer stellt die Zeitbasis für den NCO zur Verfügung, dessen Samplerate so gewählt werden sollte, dass ein üblicher FM-Demodulator die Samplefrequenz schon nicht mehr in den NF-Zweig durchlässt. Ein Wert um die 10kHz sollte dafür ausreichen. Der Phasenakkumulator wird bei jedem Takt auf Grundlage des aktuell zu übertragenden Tons erhöht (FCW_MARK bzw. FCW_SPACE). Aus einer Sinus-Tabelle, erstellt mit einem kleinen Skript, wird der nächste Sinus-Wert ausgelesen und daraufhin der Si4060 auf den entsprechenden Kanal gewechselt.

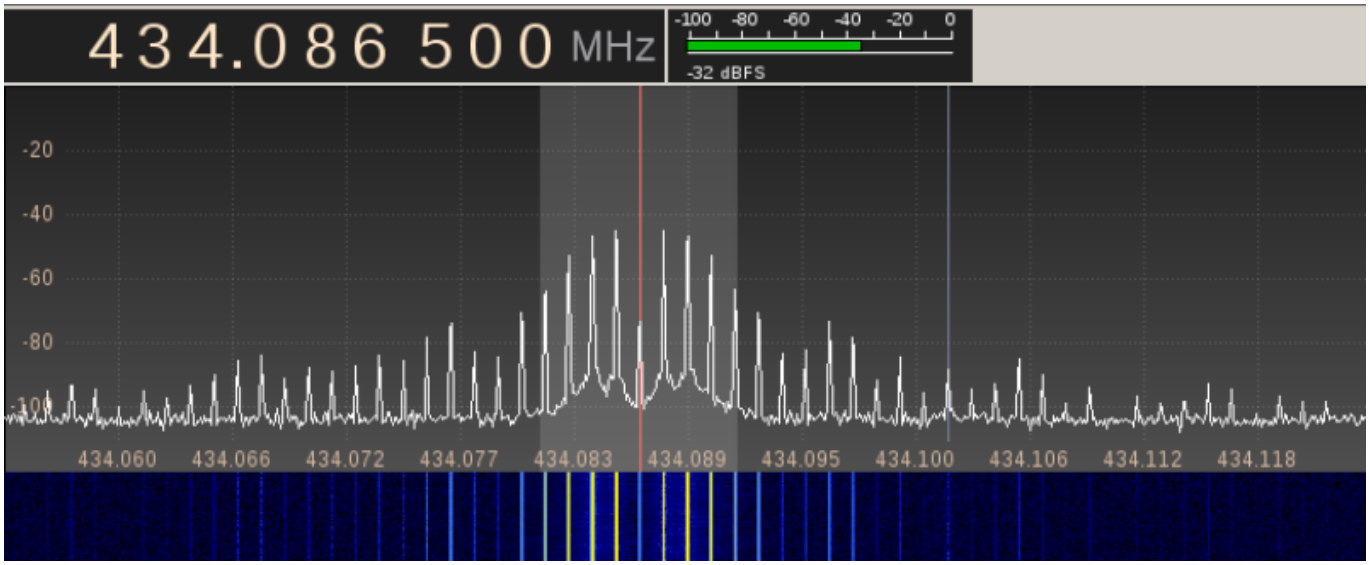
Beim Si4060 beschränkte eine wenigstens annähernd saubere FM-Aussendung bisher, dass die PLL-Register nicht während des Sendens aktualisiert werden können. Dies lässt sich umgehen, indem man statt der PLL-Register selbst (Integer/Fractional-Teiler) einfach die Offset- oder Deviation-Register des FSK-Modems beschreibt, im Ergebnis ist es das gleiche. 16 Bit „Einstellbreite“ sind vorhanden, wenn

man das Offsetregister verwendet. Der Hub sollte 3kHz nicht übersteigen, bei 30MHz Quarzfrequenz und einer PLL-Auflösung von 21 Bit hat man eine Frequenzauflösung von 14Hz im 70cm-Band, also sollte man 419 „Digits“ Hub machen. Dies schlägt sich in der Sinustabelle des NCO nieder. Der Si4060 erzeugt so sogar ein im Nahfeld verhältnismäßig sauberes Spektrum. Der UART-Interrupt muss allerdings deaktiviert werden, sollte während der Sendung eine Nachicht vom GPS kommen, zerstört dies das Timing (zu hohe Auslastung).

Als erster Test sollte ein Sinus mit 1200Hz ausgegeben werden. Das Ergebnis der Demodulation ist im folgenden Screenshot zu sehen. Mit 23dB Oberwellendämpfung ist es zwar sicherlich nicht optimal, aber weitab der zweiten Nutzfrequenz (2200Hz), sollte also nicht zur nennenswerten Verschlechterung der Demodulation führen.



Vor der Demodulation sieht das Spektrum nach einer typischen FM aus:



Methode 2 (neu): Nutzung des FIR-Filters im Si4060, Modulation mit Rechteck

Der Si4060 enthält für GFSK-Modulation ein FIR-Filter 16ter Ordnung mit symmetrischen Koeffizienten, die frei programmierbar sind. Der Ansatz war nun, die Modulation am DATA-Pin als Rechteck durchzuführen, und das FIR-Filter zur Tiefpassfilterung zu benutzen, sodass ein möglichst sauberer Sinus am Ausgang entsteht. Vorteil ist die höhere erreichbare Samplerate: Das interne Filter arbeitet durch Oversampling mit der zehnfachen Frequenz des Dateneingangs. Um sowohl 1200 Hz als auch 2200 Hz sauber abtasten zu können, wurde 4400 Hz als Datenrate gewählt, das Filter läuft also intern

bei 44 kHz. Es ist möglich, ein noch annehmbares Tiefpassfilter für diese Gegebenheiten zu entwerfen: 1200 Hz und 2200 Hz sollten kaum beeinflusst werden, 3,6kHz (3. Harmonische von 1200 Hz) sollte möglichst gut gedämpft werden.

Zusätzlich kann (mit Abstrichen im Stopband des Filters) die notwendige Preemphase mit in das Filter integriert werden. Es muss hier allerdings ein Trade-Off zwischen Stopbanddämpfung und Preemphase-Güte gemacht werden. Mit dem aktuellen Filter wird eine Dämpfung von 14dB bei 3,6kHz gegenüber dem 1,2kHz-Ton erreicht, alle anderen Oberwellen sind >30dB unterdrückt.

Erreichte Verbesserung: TNC2S dekodiert jetzt alle Pakete sauber (keine Rejected Packets mehr), MixW und QTMM dekodieren ebenfalls sauber.

Bit-Stream

Aus einem ganzzahligen Teil des Sampletaktes wird der Bit-Takt erzeugt, mit welchem die Funktion zum holen des nächsten Bits ausgeführt wird. Es wird sich einerseits um die NRZI-Kodierung gekümmert, als auch um die Vorgabe des AX.25-Protokolls, Bit-Stuffing zu betreiben. Nach der fünften „1“ wird automatisch eine „0“, also ein Flankenwechsel eingefügt und das nächste Nutzdatenbit um eine Bitlänge verzögert.

AX.25-Protokoll

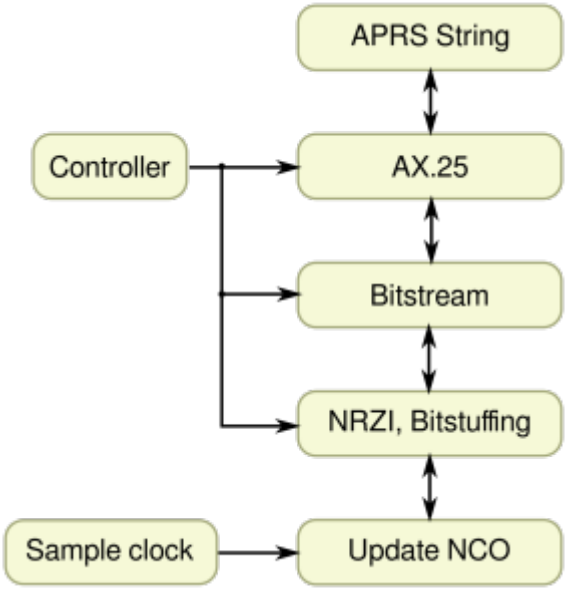
Von der Software wird lediglich ein Puffer beschrieben, in welchem die Nutzdaten liegen. Das zugehörige Längenfeld muss mitbeschrieben werden. Die Sende-State-Machine kümmert sich um Aussendung der Flag-Sequenzen am Anfang und am Ende, der Header-Informationen, der Nutzdaten sowie der FCS (Frame Check Sequence).

Im Hinterkopf behalten: APRS braucht zwar immer den gleichen Header (Source: eigenes Rufzeichen, Destination: APRS), aber für evtl. Telemetriepakete muss man das eigene Rufzeichen als Target einstellen.

APRS-Nutzdaten

In erster Instanz sollen lediglich Position und Höhe im APRS-Netz verteilt werden können. Dies wird mit einem gewöhnlichen Position Report (APRS Spec, Seite 32) bewerkstelligt. Positionen werden im NMEA-Format eingebunden (mit anhängtem N/S bzw E/W), die Höhe wird im Kommentarfeld in Fuß nach dem Altitude Identifier „/A=“ angegeben. Danach folgt noch ein fester Kurztext.

Design



Samplerate

Am Si4060 zeigte sich kein Problem bei Verwendung von ~10kHz Samplerate unter Verwendung der 30MHz-Taktquelle. Wenn allerdings die RS-92 TCXOs mit 16.3676MHz verwendet werden, scheint der Si4060 keine so hohe Aktualisierungsrate mehr zuzulassen - nicht alle Frequenzwerte werden übernommen und das Spektrum unrein. Workaround ist die Verwendung der größtmöglichen Samplerate, die durch Experimentieren als >5kHz gefunden wurde. Damit ist man zwar gefährlich nah an der Nyquist-Grenze, was sich wegen des fehlenden Filters auch bemerkbar macht (nur noch etwa 3dB Spiegelfrequenzdämpfung nach der Demodulation beim 2,2kHz-Ton). Das TNC2S (als Quasistandard bei APRS-Digis) zeigte sich demgegenüber tolerant. Es sollte aber die Ursache gefunden werden und das Problem an der Ursache behoben (insofern dies möglich ist).

Links

- AX.25-Spezifikation, Version 2.2
- APRS-Spezifikation, Version 1.0
- APRS-Symboltabelle

From:
http://loetlabor-jena.de/ - **Lötlabor Jena**

Permanent link:
http://loetlabor-jena.de/doku.php?id=projekte:utrak:aprs&rev=1437726133

Last update: **2015/07/24 08:22**