

Das Wetter im Bilde

Konrad Gralher
Jugendforscht 2021 Thüringen

INHALTSVERZEICHNIS

| | |
|--|----------|
| Teil 1. „Automatic Picture Transmition“ | 1 |
| 1. Einführung | 1 |
| 2. Optimierung der Software | 1 |
| 3. Optimierung der Antennen | 6 |
| Anhang | 8 |
| Literatur | 8 |

Teil 1. „Automatic Picture Transmission“

1. EINFÜHRUNG

Das Übertragungsverfahren „Automatic Picture Transmission“ kurz APT ist ein analoges Bildübertragungsverfahren welches zuerst auf dem TIROS-8(Start 1963) Satelliten zur Anwendung kam und meist für Wetterbilder genutzt wird.



ABBILDUNG
1. TIROS-8 im Orbit

Heute befinden sich noch drei Satelliten der US Amerikanischen Wetterbehörde NOAA in einem „Low Earth Orbit“(LEO). Die veraltete Technologie des Verfahrens bieten dabei die Möglichkeit zugänglicher im Empfang sowohl als auch in der Demodulation gegenüber heutigen Verfahren zu sein. Deshalb beschäftigt sich diese Arbeit mit der Vereinfachung solcher Prozesse sowohl im Empfang als auch beim Decodieren. Dabei sollen Prozesse in der Decodierung beschleunigt werden um für ein breiteres

Spektrum von Meteorologen zugänglich zu werden. Zusätzlich ist der Arbeitsablauf des Empfanges von der Antenne bis zum Decodieren noch sehr komplex was verbessert werden soll. Es sollen weiterhin verschiedene Antennenmodell auf ihre Tauglichkeit im Bezug auf den APT Satellitenempfang getestet werden. Ich bin auf dieses Projekt gekommen, da ich mich schon seit einigen Jahren intensiv mit Hochfrequenztechnik in Satelliten beschäftige und ich zu Beginn oft frustriert von der vorhandenen Dokumentation im Internet war. Weiter haben mich die Antennen Anforderungen förmlich erschlagen, da ich am Anfang keinerlei Erfahrungen mit Antennentechnik sowie „Software defined Radios“ hatte. Auch fasziniert mich die Zugänglichkeit von solchen Satelliten die keinerlei Beschränkungen wie Verschlüsselung erzwingen. Ich sehe in solchen Satelliten deshalb ein großes Potential bezogen auf Wetterstationen oder Regierungen mit wenig Mitteln für Wetterbeobachtungen die meist nur auf Beobachtungen am Erdboden zurückgreifen müssen und so nur wage Wettervorhersagen treffen können, was zu potentiell gefährlichen Situationen führen kann. Außerdem können so Amateure eigene Daten erfassen um genauer Wetter- und Klimaveränderungen zu erfassen.

2. OPTIMIERUNG DER SOFTWARE

Die zu Übertragenden Daten stammen zumeist aus einem „Advanced Very High Resolution Radiometer“(AVHRR). Da das APT Verfahren nur einen geringen Datendurchsatz ermöglicht, werden die hochauflösenden Datenströme von dem AVHRR Instrument herunterskaliert sowie auf Graustufen umgesetzt und

eignen sich deshalb für einen Transport über das APT Verfahren. Das AVHRR der Serie 3 ist in der Lage zwischen sechs verschiedenen Spektralbändern zu differenzieren. Die Bänder reichten dabei von einer Wellenlänge von $0,58 \mu\text{m}$ bis

12,5 μm und bieten jeweils eine Auflösung von einem Pixel auf 1,1km. Kanäle 1-3A zeichnen dabei die Reflektionen der sichtbaren oder Infrarotnahen Strahlen an der Erde auf. Diese Datengrundlage lässt eine Analyse der Vegetation, den Wolken, Gewässern, Schnee und Eis zu. Die verbleibenden Kanäle stellen zu einem großen Teil die Erdeigene Infrarotstrahlung dar, die von dem Meer sowie dem Land ausgestrahlt wird und genaue Temperaturwerte liefert. Die verbleibenden Instrumente sind für diese Arbeit von keiner Relevanz, da sie nicht in den entsprechenden Übertragungsverfahren ausgeliefert werden. [1] Das Signal setzt sich aus einer Amplitudenmodulierten Niederfrequenz(NF) in der das Nutzsignal transportiert wird und dem Träger der Modulation(in diesem Fall 2.400Hz) zusammen. Das entstandene Signal wird anschließend für den Transport erneut moduliert, diesmal wird die Frequenzmodulation aufgrund der geringeren Störungswahrscheinlichkeit gewählt. Der Träger entspricht jetzt der gewünschten Frequenz im VHF Bereich $\approx 137\text{ MHz}$) und das Signal wird über eine Helix Antenne zirkular polarisiert mit wenigen Watt ausgesendet. Das Signal wird dabei aufgrund der hohen Frequenz nicht an der Ionosphäre(80-300km) reflektiert und kann somit in die Erdatmosphäre eindringen und auf der Erde empfangen werden. Es werden insgesamt 2 Linien der beiden Kanäle und zusätzliche Telemetrische Daten übertragen. So sind maximal 4160 Pixel in Grustufen pro Sekunde möglich. Anhand dieser Daten werden einfache Vorhersagen des Wetters ermöglicht. So ist die Position der Wolken genau zu bestimmen. Außerdem lässt sich die Temperatur zuverlässig bestimmen. Naturereignisse wie Vulkanmausbrücke bleiben allerdings lange unbemerkt auf den schlecht aufgelösten Bildern des APT

Übertragungsverfahrens. [2] Dieser Prozess wird beim Empfang der Signale revertiert und so das Eingangs erwähnte AVHRR Signal wird mit einigen Telemetriedaten sichtbar. Ich habe deshalb versucht diesen genannten Prozess selber umzusetzen und dabei eine möglichst gute Laufzeit und Nutzererfahrung meines Programms zu erreichen. Ich habe dazu Versuche in Gnu Radio durchgeführt, in denen ich anhand von Blöcken den Prozess evaluiert habe. Die einzelnen Blöcke in Gnu Radio repräsentieren dabei Python- oder wahlweise C++ Code. Die Darstellung in Gnu Radio ähnelt sehr einem Fließdiagramm was die Prozesse grafisch gut und verständlich darstellen kann. So kann Wissenschaftlern weltweit geholfen werden besser die Abläufe und Hintergründe der Aussendungen zu verstehen. Als zweiten Schritt soll das Flowchart in Julia einer sehr performanten Programmiersprache abgebildet werden um das Programm in der Laufzeit noch effizienter zu gestalten. Zur Hilfe nahm ich mir dabei das Flowchart von Alexandre Rouma [3], sowie die Julia Bibliothek [4,] zum Verarbeiten der Signale gleichwertig zu Gnu Radio.

Grundlegend bei Digitaler Signalverarbeitung ist die Abtastrate. Da Signale in Form eines Vektors digital übertragen werden können nur eine begrenzte Anzahl an Werten innerhalb von einer Sekunde übertragen werden. Deshalb setzt sich die Einheit in der die Abtastrate angegeben wird aus einer Sekunde und den Datenpunkten innerhalb dieses Zeitfensters zusammen(t_s).

Die entsprechende Abtastrate für das Flowchart habe ich aus dem Quellcode von [2] errechnet. Dabei bin ich auf folgende Formel gekommen.

$$\begin{aligned} x &= \text{Eingabeabtastrate,} \\ y &= \text{Ausgangsabtastrate} \\ (1) \quad \text{Interpolationsfaktor} &= GCD(x, y)/y \\ (2) \quad \text{Dezimationsfaktor} &= GCD(x, y)/x \end{aligned}$$

Die Eingangsdaten werden hierbei als WAV Datei bereitgestellt mit einer Abtastrate von $37500(t_s)$. Das Shannon-Abtasttheorem beschreibt dabei, dass ein Sinusförmiges Signal bei doppelter Abtastrate von der Eingangsfrequenz komplett repräsentiert werden kann. Diese Eigenschaft können wir uns auch zunutze machen und die Abtastrate auf theoretisch $11025t_s$ beschränken. So reduziert man Beispielsweise den Datensatz um das dreifache ohne merkbare Qualitätsverluste. Die Abtastrate von $11025t_s$ durch das doppelte von der maximalen Nutzfrequenz, die sich aus dem Träger von 2400Hz und dem Signal(3112.5Hz) zusammensetzt. Das bedeutet eine deutlich schnellere Laufzeit da unnötige Iterationen eingespart werden können. Ein ähnliches Verfahren wird auch bei analogen Empfängern verwendet um leichter das Signal verändern zu können, weil Oszillatoren mit einer hohen Frequenz die Tendenz besitzen stärker in der Frequenz zu schwanken. Abb. 2 ist zwar etwas Zeitversetzt zeigt aber anhand der Stärke der Farbe, dass bei gleichbleibender Qualität weniger Abtastpunkte benötigt werden. Um noch mehr Abtastpunkte einzusparen wird das Signal gegen Null mithilfe eine Signalquelle gemischt um so eine maximale Arbeitsfrequenz von 3112.5Hz zu erhalten. Da das Signal auf beiden Seitenbändern aufmoduliert ist können wir uns auf das obere beschränken und so eine Abtastrate von 6225Hz nutzen. Wir erhalten nachdem wir die Formel

angewand haben einen Interpolationsfaktor von 83 und einen Dezimationsfaktor von 147. Dabei ist zu beachten das wir keine Abtastpunkte löschen können, da wir sonst die Frequenz verändern und so alle nachfolgenden Signalverarbeitungsschritte anpassen werden müssten.

So konnte zum ursprünglichen Algorithmus die Abtastrate um den Faktor 3,5 reduziert werden, was die Anzahl der Iterationen im darunterliegenden Code um den gleichen Faktor mindert. Die Dateigröße bei einer Aufnahme mit $3750 t_s$ sollte auch die Dateigröße der Eingangsdatei maßgeblich reduzieren. Bei genauerer Betrachtung der Resultate ist ein leichter Qualitätsverlust erkennbar. Dieser Resultiert womöglich aus der Nutzung der theoretischen Mindestabtastrate, nicht berücksichtigt wurden bei den Berechnungen mögliche Ungenauigkeiten wie der Dopplereffekt der die Aufnahme verzerrt oder Effekte die bei der Digitalen Signalverarbeitung entstanden sind. Außerdem werden Artefakte von dem Sinusförmigen Signal mit einer Frequenz von -1.4MHz in das Nutzsignal gemischt. Um Trotzdem eine unveränderte Bildqualität zu erreichen hat sich der dreifache Wert des theoretischen Minimums als beste Balance zwischen Qualität und Leistung herausgestellt. Ich habe anschließend versucht das Flowchart auch in Julia zu replizieren. Dazu habe ich die Daten und die Abtastrate aus der WAV Datei importiert und so einen Array aus Floats mit einer Größe von jeweils 64Bit bekommen. Der Wertebereich reicht dabei von 1.0 bis -1.0.

$$\mathbf{W} = [-1.0, 1.0]$$

Jeder Zahl in dem Vector belegt im Heap(meist im Arbeitsspeicher) 64Bits und ist um das doppelte genauer wie die vergleichbare Implementation in Gnuradio die eine Auflösung von 32bit liefert.

$$\begin{bmatrix} -0.35938596758934294; \\ -0.7111117893002106; \\ 0.25470748008667254; \\ 0.5893124179815058; \end{bmatrix}$$

Als Demodulationsalgorithmus benutzte ich bei dieser Implementation den in der DSP Bibliothek integrierten Hilbert Transformationsalgorithmus. Dieser Bei dem Quellcode habe ich mich aufgrund fehlender Dokumentation von dem DSP(digital signal processing) Paket stark an der Arbeit von [5,]orientiert. So nutze ich die Synchronisationsfunktionen um das Bild anhand der Telemetriedaten auszurichten. Um an das Niederfrequenzsignal aus der Amplitudenmodulation zu kommen, verwende ich die Hilbert Transformation. Diese setzt das Signal in ein komplexes Signal um, welches aus einer Amplitude und einem Phasenwinkel besteht. Da die Hochfrequenz nur in den Phasenwinkel umgesetzt wird, können wir den Amplitudenwert als demoduliertes Niederfrequenzsignal weiterverarbeiten. Die Filter in Julia DSP bestehen dabei aus einem Filterart wie einem Bandpass in diesem Fall und einem Filtertyp einem Butterworth Filter z.B. Der Filtertyp bestimmt dabei die Eigenschaften des Filters, so definiert er die Passbandfrequenzen und die Steigung an den Flanken. Da wir das Eingangssignal filtern um keine unnötigen Störungen in der Hilbert Transformation zu erzeugen beschränken wir uns mithilfe des Bandpasses auf den Bereich von 400Hz bis 4400Hz. Nachdem das Signal demoduliert wurde kann ein resample dabei helfen den Synchronisationsvorgang zu erleichtern. Dabei behelfe ich mich der Funktion 1 und 2. Der verwendete Code um die Arrays in eine richtige Anordnung zu bringen habe ich komplett von [5,] übernommen. So entstehen mehrdimensionale Vektoren die in Graustufen umgesetzt und gespeichert werden können. Siehe Abb.5. Da der Julia Compiler sehr viele Überprüfungen zur Übersetzungszeit ausführt, dauert das

testen sehr lange was den Einstieg erschwert. Außerdem werden die Abhängigkeiten schon vorab kompiliert was zusätzlich die Übersetzungszeit verlängert. Die Benutzung wird aber erleichtert da keine Speicherlecks mehr möglich sind und die Laufzeit durch Kompilierung verbessert wird. Aber die Bildergebnisse wurden deutlich verbessert außerdem konnte eine deutliche Reduktion der Laufzeit festgestellt werden. So erreicht Julia C ähnliche Leistungen was sich deutlich in der Auslastung von Systemressourcen zeigt.

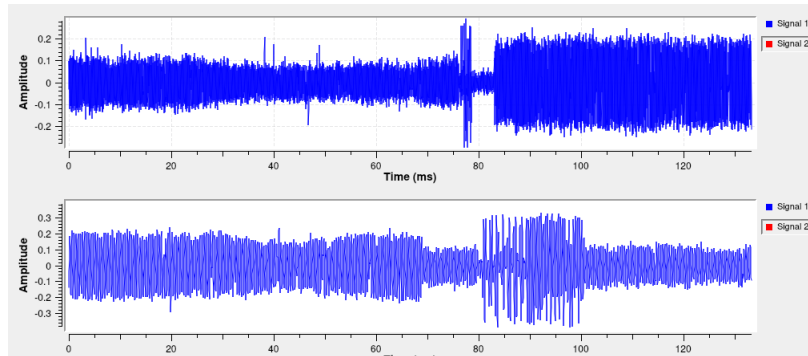


ABBILDUNG 2. Abtastraten im Vergleich oben: $37500 t_s$
unten: $11025 t_s$

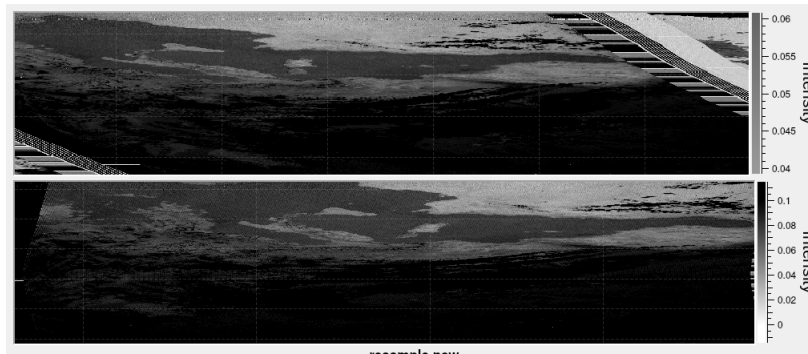


ABBILDUNG 3. Endergebnis im Vergleich oben: $3112 t_s$
unten: $11025 t_s$

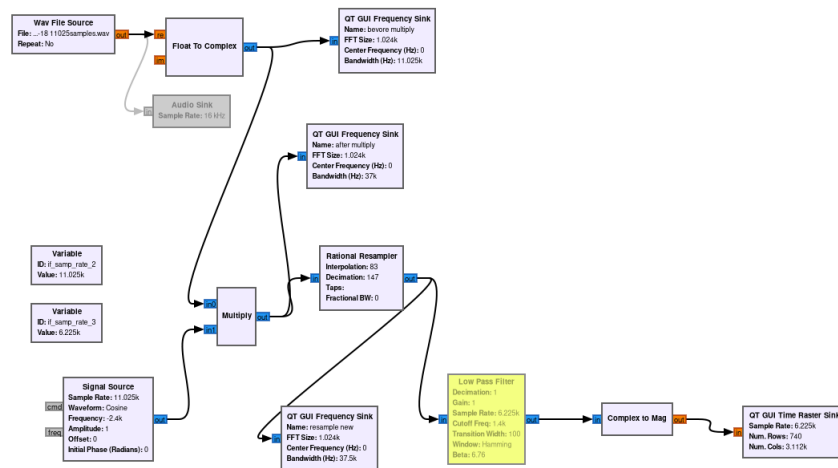


ABBILDUNG 4. Der Flowchart in Gnuradio.

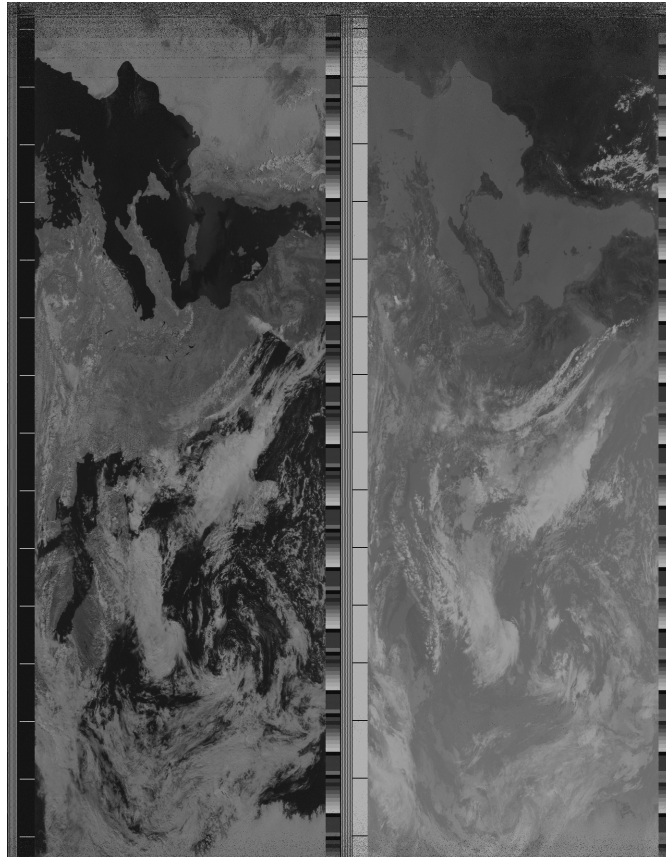


ABBILDUNG 5. Ergebnis des Julia Programmes.

3. OPTIMIERUNG DER ANTENNEN

Einer der wichtigsten Faktoren ist neben der Empfindlichkeit des Empfängers die Charakteristiken der verwendeten Antenne. In dem folgenden Abschnitt habe ich verschiedene Antennen verglichen und sie auf Tauglichkeit sowie ihre Anwendbarkeit geprüft. Die erste Bauweise die ich analysiert habe ist die sogenannte V-Dipol Antenne die eine starke Ähnlichkeit zu einer gewöhnlichen Dipol Antenne besitzt. Ich habe mich aufgrund der besseren Reproduzierbarkeit dieser Arbeit dazu entschieden die Antenne selber zu bauen. Die nötigen Anweisungen konnte ich dem Material [6,] entnehmen. Die Länge des

Kabels konnte ich mithilfe Die Materialliste beschränkt sich auf nur wenig Zubehör was positiv zu erwähnen ist. Theoretisch reicht ein kurzes Koaxialkabel nach Möglichkeit mit einer Impedanz von 50Ω . Weiterhin werden zwei Kupferdrähte nach Möglichkeit mit einem Durchmesser von über 1mm und Schraubklemmleisten mit zwei Ein- und Ausgängen benötigt. Zu Beachten beim Bau der Antenne ist die Verzinnung der Kabelenden sowohl der Schirm als auch die innere Seele. Beim Bau der Antenne unterstützte mich Ullrich Müller. Der Unterschied besteht dabei in dem Winkel zwischen den Beiden Elementen der

Antenne. Generell werden die APT Signale mit einer rechtsdrehende Zirkulation ausgesendet. Deshalb eignet sich eine Kreuzdipol oder eine Helix Antenne am besten.

Anhang

LITERATUR

- [1] K. Kidwell, "Noaa klm user's guide(englisch)." <https://web.archive.org/web/20041204200739/http://www2.ncdc.noaa.gov/docs/klm/index.htm>, 2004. Abgerufen: 2021-01-01.
- [2] M. Bernardi, "Noaa-apt how it works(englisch)." <https://noaa-apt.mbernardi.com.ar/how-it-works.html>. Abgerufen: 2021-01-01.
- [3] A. Rouma, "Apt flowchart in gnuradio)." <https://twitter.com/ryzerth/status/1343933680184664066/photo/2>. Abgerufen: 2021-01-03.
- [4] J. DSP, "Dsp.jl)." <https://github.com/JuliaDSP/DSP.jl>. Abgerufen: 2021-01-10.
- [5] A. Barth, "Aptdecoder.jl)." <https://github.com/Alexander-Barth/APTDecoder.jl>. Abgerufen: 2021-01-10.
- [6] A. 9A4QV, "Diy 137 mhz apt weather satellite antenna)." <https://www.dropbox.com/s/6fpfn2p9filc9o1/DIY%20137MHz%20WX-sat%20V-dipole%20antenna.pdf?dl=0>. Abgerufen: 2021-01-10.