

# Das Wetter im Bilde

Konrad Gralher  
Jugendforsch 2021 Thüringen

Wettbewerbssparte: Geo-und Raumwissenschaften

## INHALTSVERZEICHNIS

<b>Teil 1. „Automatic Picture Transmition“</b>	1
1. Einführung	1
2. Optimierung der Antennen	2
3. Optimierung der Software	4
Funktionsweise	4
Gnuradio	5
JuliaDSP	7
<b>Danksagung</b>	10
<b>Anhang</b>	11
Literatur	11

## Teil 1. „Automatic Picture Transmition“

### 1. EINFÜHRUNG

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



ABBILDUNG  
1. TIROS-8 im Orbit

Aktuell befinden sich noch drei Satelliten der US Amerikanischen Wetterbehörde Noaa in einem „Low Earth Orbit“ (LEO). Die veraltete Technologie des Verfahrens bietet einen erleichterten Zugang im Vergleich zu heutigen Standards. Außerdem wurden meist, wie zu der Entwicklungszeit üblich, niedrige Frequenzen genutzt. Diese erleichtern den Empfang erheblich.

In dieser Arbeit beschäftigte ich mich mit der Vereinfachung solcher Prozesse, sowohl im Empfang von Wetterdaten mit dem Bau einer Antenne als auch beim Dekodieren mittels eines optimierten Julia Programms, sodass ein Graustufenbild in Echtzeit auf z.B. dem Computerbildschirm dargestellt werden kann.

Dabei sollen Prozesse in der Dekodierung beschleunigt werden, um ein breiteres Spektrum von Meteorologen und

Wissenschaftlern zu erreichen. Zusätzlich ist der Arbeitsablauf des Empfanges von der Antenne bis zum Dekodieren noch sehr komplex, was verbessert werden soll. In der vorliegenden Arbeit gehe ich weiterhin vergleichend auf verschiedene Antennenmodelle und ihre Tauglichkeit im Bezug auf den APT Satellitenempfang ein.

Die Idee zu diesem Projekt entstand, da ich mich schon seit einigen Jahren intensiv mit Hochfrequenztechnik in Satelliten beschäftigte und zu Beginn oft frustriert von der vorhandenen Dokumentation im Internet war. Weiterhin haben mich die Antennenanforderungen förmlich erschlagen, da ich am Anfang keinerlei Erfahrungen mit Antennentechnik sowie „software defined radios“ hatte. Die Zugänglichkeit von solchen Satelliten, die keinerlei Beschränkungen wie Verschlüsselung erzwingen, sondern frei empfänglich sind, faszinieren mich. Deshalb sehe ich einerseits in solchen Satelliten ein großes Potential für Wetterstationen, Katastrophenschutz oder Regierungen mit wenig Mitteln für Wetterbeobachtungen. Diese können bisher meist nur auf Beobachtungen am Erdboden zurückgreifen, was nur wage Wettervorhersagen erlaubt. Dadurch werden potentiell gefährliche Situationen oft nicht erkannt, was wiederum schwere menschliche und wirtschaftliche Auswirkungen von Naturkatastrophen zur Folge haben kann. Andererseits können so Amateure eigene Daten erheben um genauer Wetter- und Klimaveränderungen zu erfassen. Insbesondere vor dem Hintergrund der aktuellen Entwicklungen im Erdklima können genaue Beobachtungen und standardisierte Vergleiche herangezogen werden, um schon geringe Veränderungen zu registrieren, da die Satelliten jeden Tag zweifach über Deutschland fliegen und Daten senden.

## 2. 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 Daten lassen sich mithilfe der Formel 1 und 2 berechnen. X muss dabei immer einem Vielfachen von 2 entsprechen um die Verhältnisse zu wahren.  $299792458\text{m/s}$  entspricht der Ausbreitungsgeschwindigkeit von Elektromagnetischen Wellen also annähernd der Lichtgeschwindigkeit. Das Ergebnis der Gleichung ist in Nanometern zu verstehen, so erhalten wir bei einer Mittelfrequenz von 137MHz unserer Antenne, eine Länge von insgesamt 2188266nm. Da wir den größtmöglichen Gewinn bei der Antenne erreichen wollen, nutzen wir den Divisor 2 bei Funktion 4 und erhalten so eine Länge von insgesamt 1,094133m. Jedes Element entspricht also 5,47066m.

$$(1) \quad \lambda = \frac{299792458\text{m/s}}{f}$$

$$(2) \quad \frac{\lambda}{x}$$

Um die Elemente mit dem Radio in unserem Fall einem Software Defined Radio(SDR) zu verbinden, nutzen

wir einfaches RG58 Koaxialkabel welches einen relativ hohen Dämpfungsfaktor bei hohen Frequenzen besitzt. Deshalb versuchen wir den Abschnitt zwischen dem Radio und der Antenne möglichst gering zu halten. So kann der Verlust reduziert werden und die Einstrahlung vermindert werden. Beide Elemente aus Kupferdraht können über einen handelsüblichen Kabelverbinder mit dem Koaxialkabel verbunden werden. Hierzu ist kein Löten nötig, aber es ist der Stabilität sehr zuträglich. Weiterhin muss eine gewisse Richtwirkung der Antenne bestehen um das Signal zu Rauschen Verhältnis zu verbessern. Die genauen Angaben zu dem Verhältnis der beiden Elemente zueinander, habe ich aus [1, ] bezogen.

Nach einigen Tests ist die Richtwirkung beim Empfang stark aufgefallen, so muss der Satellit in seinem Orbit verfolgt werden um die besten Ergebnisse zu erhalten. Außerdem konnte man die falsche Polarisation der Antenne bemerken, die einen geringen Verlust an Signal zu Rausch Verhältnis zur Folge hat. Das Signal wird zirkular polarisierend von den Satelliten ausgesendet, die Antenne aber ist nur linear polarisiert durch ihre flache Bauweise. Eine helixförmige Antenne würde die richtige Polarisation aufweisen, ist aber auch entsprechend kompliziert zu bauen und wurde aus diesen Gründen nicht in diesem Projekt einzbezogen.

Eine weitere Möglichkeit die klassische Dipol Antenne zu variieren, besteht darin, diese zu doppeln und in Form einer Kreuzdipolantenne zu bauen. Die getestete Kreuzdipolantenne hat sich außerordentlich schlecht für Satellitenempfang geeignet, da sie nur den Bereich über zentral Europa abdecken konnte. Ein Grund ist womöglich die schlechte Richtwirkung, diese resultiert aus den immer

gleichbleibenden Winkeln zwischen den Elementen der Antenne.

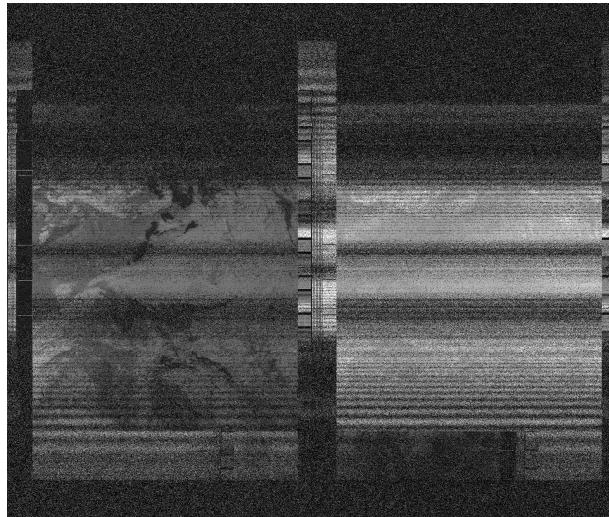


ABBILDUNG 2. Ein Überflug empfangen mit der Kreuzdipolantenne.

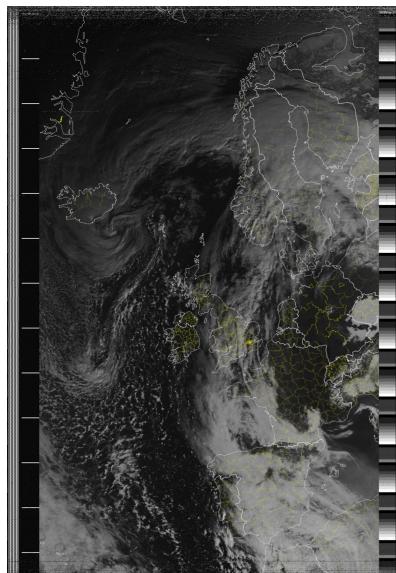


ABBILDUNG 3. Ein Überflug empfangen mit der V-Dipolantenne.

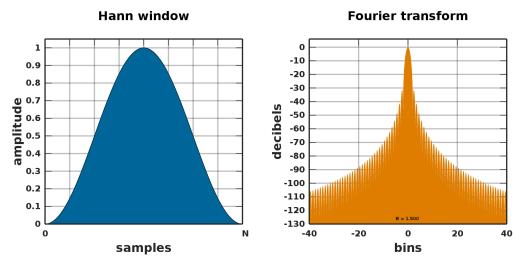
### 3. OPTIMIERUNG DER SOFTWARE

**Funktionsweise.** Die zu übertragenen 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 reichen dabei von einer Wellenlänge von  $0,58 \mu m$  bis  $12,5 \mu m$  und bieten jeweils eine Auflösung von einem Pixel auf  $1,1\text{km}$ . Kanäle 1-3A zeichnen dabei die Reflektionen der sichtbaren oder infrarotnahen Strahlen an der Erde auf. Diese Datengrundlage lässt eine differenzierte Analyse von Vegetation, Wolken, Gewässern, Schnee und Eis zu. Die verbleibenden Kanäle stellen zu einem großen Teil die erdeigene Infrarotstrahlung dar, die von Meer und 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. [2] Das Signal setzt sich aus einer Amplitudenmodulierten Niederfrequenzsignal(NF) in dem das Nutzsignal transportiert wird und dem Träger der Modulation(in diesem Fall  $2.400\text{Hz}$ ) zusammen. Das entstandene Signal wird anschließend für den Transport erneut moduliert. Diesmal wird die Frequenzmodulation aufgrund der geringeren Störanfälligkeit 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 von dem Satelliten ausgesendet. Das Signal wird dabei aufgrund der hohen Frequenz nicht an der Ionosphäre( $80\text{-}300\text{km}$ ) 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 jede Sekunde übertragen(4160 Symbole). So lässt die Position der Wolken genau bestimmen, was bei mehreren Bildern täglich eine Approximation der Wolkenbewegung ermöglicht. Außerdem lässt sich die Temperatur mit einer hohen Genauigkeit extrahieren. [3]

**Gnuradio.** Die Modulation und Dekodierung wird beim Empfang der Signale revertiert, und so das Eingangs erwähnte AVHRR Signal in durch den Übertragungsweg komprimierter Form sichtbar. Dazu habe ich die genannten Modulations- und Dekodierungsverfahren digital in GnuRadio [4, ] einem DSP(digital signal processing) Framework umgesetzt. Dazu wurde die Signale in einem Tiefpassfilter gefiltert damit hochfrequente Störungen später nicht das Bild stören. Dieser hatte eine Grenzfrequenz von  $2,4\text{kHz}$  da die FM Demodulation bereits in der SDR Steuersoftware erfolgte. Da das Signal im Hintergrund von der Software in eine Frequenzbereich mittels einer Schnellen Fourier-Transformation(eng. FFT) umgesetzt wird, habe ich eine Breitbandige Hamming Fensterfunktion hinzugefügt um dem Leck-Effekt entgegenzuwirken. Dieser entsteht durch die FFT Funktion, da diese in programmativen Implementationen keinen unendlichen Integral besitzen kann und es so zu dem Leck-Effekt kommt, da die Abstrakt definierten Enden der Funktion das selbe Ergebnis besitzen müssen.



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 die Gnuradio Implementation habe ich aus dem Quellcode von [3] errechnet. Dabei bin ich auf folgende Formel gekommen.

$x$  = Eingabeabtastrate,  
 $y$  = Ausgangsabtastrate

$$(3) \text{ Dezimationsfaktor} = \frac{x}{GCD(x, y)}$$

$$(4) \text{ Interpolationsfaktor} = \frac{y}{GCD(x, y)}$$

Die Eingangsdaten werden hierbei als WAV Datei bereitgestellt mit einer Abtastrate von  $176400(t_s)$ . Das Nyquist-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  $6225t_s$  beschränken. So reduziert man Beispielsweise den Datensatz um das

$$\approx 30.\text{fache}$$

ohne theoretisch merkbare Qualitätsverluste. Die Abtastrate von  $176400t_s$  durch das doppelte von der maximalen Nutzfrequenz, die sich aus dem Träger von 2400Hz und dem Signal(3112.5Hz) zusammensetzt. Das würde eine deutlich schnellere Laufzeit bedeuten, da unnötige Iterationen über die Daten vermieden werden können. Nachdem ich dieses Verfahren getestet habe, konnte man nur blass das Bild erkennen. Zu sehen waren nur die Telemetriestreifen, die mit der größten Signalstärke ausgesandt werden. Dieses Verhalten deutet auf eine zu geringe Abtastrate hin. Nachdem ich die Abtastrate verdoppelt habe, sind weitere Artifakte aufgetreten, aber das Bild wurde besser sichtbar. Dieses Verhalten konnte ich auf den Interpolationsfaktor zurückführen, da dieser die Daten die

schon vorher durch den Dezimationsprozess verworfen wurden versucht zu rekonstruieren. Deshalb entstehen Artifakte die es nie im Orginal Signal nie gegeben hat. Die Formel hat die Gesetzmäßigkeit das bei vielfachen der Aus- oder Eingangsabtastrate der Dezimationsfaktor konstant bleibt und sich nur der Interpolationsfaktor verändert, deswegen ist kein großer Unterschied in der Datenmenge, sondern nur in der Amplitude(Helligkeit) zu erkennen.

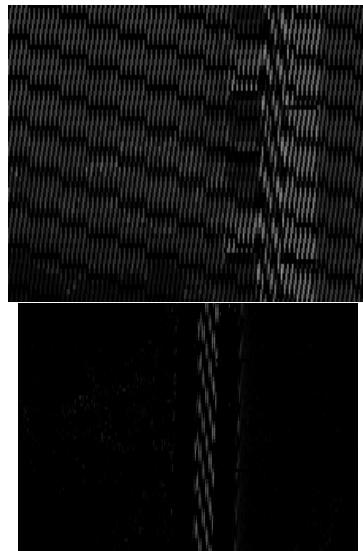


ABBILDUNG 4. Abtastraten im Vergleich  
oben:  $9600 t_s$   
unten:  $4800 t_s$

Um diesen Effekt zu umgehen, braucht man eine möglichst geringe Interpolationsrate, um dies zu erreichen habe ich die Formel(4) umgestellt, sodass der Interpolationsfaktor 1 entspricht. Daraus ergibt sich dann die Abhängigkeit, dass der größten gemeinsamen Teiler, der Ausgangsrate entsprechen muss, also ein Vielfaches von der Eingangsrate. Bei einer Eingangsabtastrate von  $176400 t_s$  bin ich zu einem Optimum bei einem Dezimationsfaktor 20 gekommen. Dieser hat eine Ausgangsabtastrate von  $8820 t_s$ .

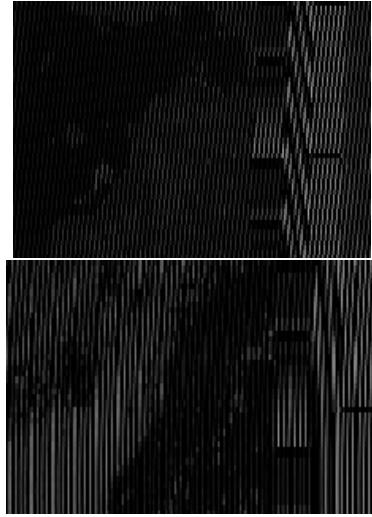


ABBILDUNG 5. Abtastraten im Vergleich oben: Eingangsabstrate unten:  $8820t_s$

Anschließend wird die Hüllkurve des Signales benötigt die in der digitalen Signalverarbeitung meist über den Betrag des Quadratursignal(komplexes Signal bestehend aus einem reellen und imaginären Teil) errechnet wird. Um das komplexe Signal zu erzeugen wird meist eine Hilbert Transformation verwendet. Der Betrag kann in Gnuradio durch den Block complex to float berechnet werden. Die verarbeiteten Daten habe ich schlussendlich in einer Art heatmap visualisiert die einer Breite von

$$\frac{\text{Ausgangsabstrate}}{2}$$

haben muss, da zwei Linien jede Sekunde übertragen werden.

**JuliaDSP.** Ich habe anschließend versucht das Gnuradio Script auch in Julia zu replizieren, um die Laufzeit weiter zu verlängern. Julia ist eine hight level Programmiersprache, die aber zugleich eine ähnliche Leistungsfähigkeit wie c oder c++ besitzt. Dazu habe ich die Daten und die Abstrate 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 Vektor belegt im Heap(meist im Arbeitsspeicher) 64Bits und ist um das doppelte genauer wie die vergleichbare Implementation in Gnu radio die eine Auflösung von 32bit liefert.

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

Bei dem Quellcode habe ich mich aufgrund fehlender Dokumentation von dem DSP(digital signal processing) Paket stark an der Arbeit von [5, Jorientiert. Um auch in dieser Implementation 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 Amplitudewert als demoduliertes Niederfrequenzsignal weiterverarbeiten. Die Filter in Julia DSP bestehen dabei aus der Filterart wie einem Bandpass in diesem Fall und einem Filtertyp einem Butterworth Filter. Der Filtertyp bestimmt dabei die Eigenschaften des Filters, so definiert er die Passbandfrequenzen und die Steigung an den Flanken des gefilterten Bereiches. 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 Neuabtastung mit einer geringeren Abstrate dabei helfen den Synchronisationsvorgang zu erleichtern. Ich behelfe mich dabei 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 Grasstufen umgesetzt und gespeichert werden können. Siehe Abb.5. Da der Julia Compiler sehr viele Überprüfungen zur Übersetzungszeit ausführt, dauert der Testprozess 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 eine vorab 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.

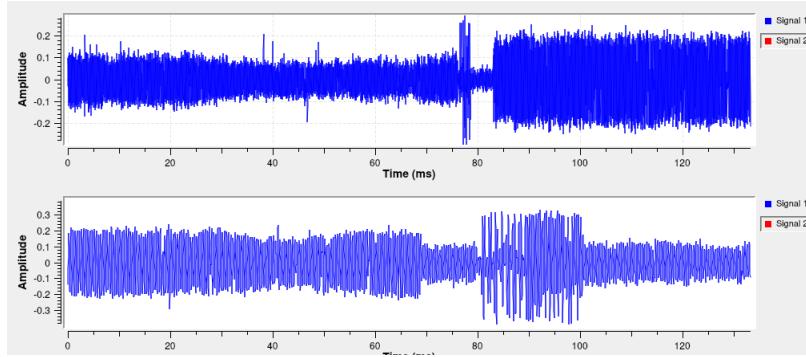


ABBILDUNG 6. Abtastraten im Vergleich oben:  $37500 t_s$   
unten:  $11025 t_s$

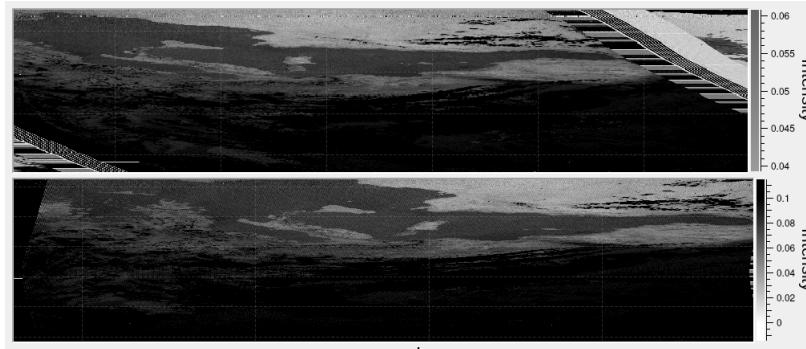


ABBILDUNG 7. Endergebnis im Vergleich oben:  $3112 t_s$   
unten:  $11025 t_s$

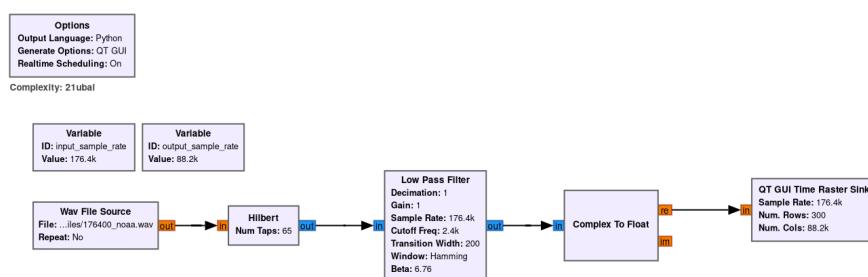


ABBILDUNG 8. Der Flowchart in Gnuradio.

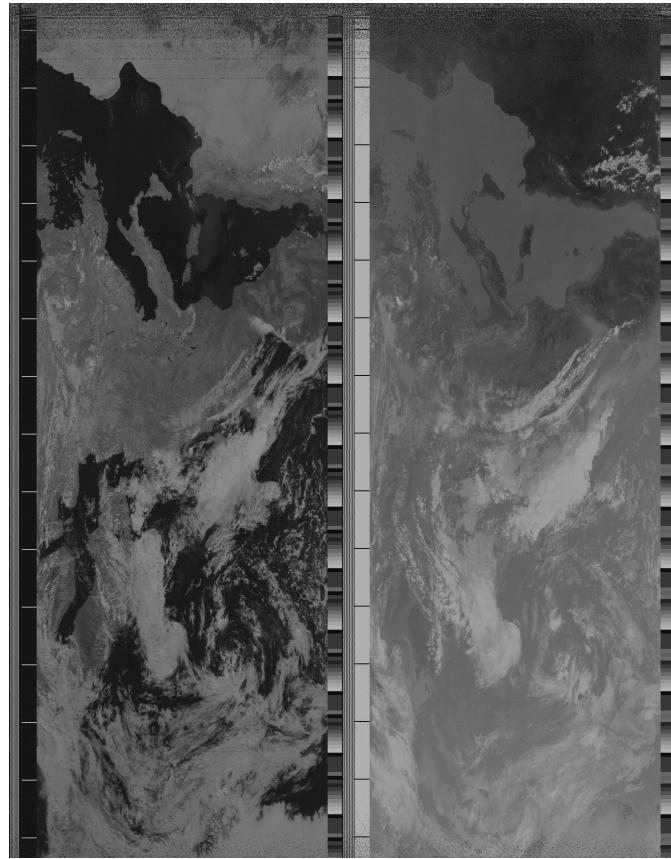


ABBILDUNG 9. Ergebnis des Julia Programmes.

### Danksagung

Ich bedanke mich bei Ullrich Müller der mich beim Bau der V-Dipol Antenne unterstützt hat und mir die Kreuzdipolantenne zur Verfügung gestellt hat.

## Anhang

### LITERATUR

- [1] 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.
- [2] 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.
- [3] M. Bernardi, “Noaa-apt how it works(englisch).” <https://noaa-apt.mbernardi.com.ar/how-it-works.html>. Abgerufen: 2021-01-01.
- [4] <https://www.gnuradio.org/get-involved/>, “Gnuradio.” <https://www.gnuradio.org/>. Abgerufen: 2021-06-10.
- [5] A. Barth, “Aptdecoder.jl.” <https://github.com/Alexander-Barth/APTDecoder.jl>. Abgerufen: 2021-01-10.