Drahtlose Netzwerke

Project work

Receiving and Decoding APT Data with RTL-STR

at the

University of Applied Sciences (HTW) Berlin
Faculty of computer science IV
Applied Computer Science

Submitted by:	Enrico Gamil Toros de Chadarevian
Matriculation number:	s0567011
Date of submission:	03.03.2021

Contents

Chapter 1

Introduction

The 4th of October 1957 almost 64 years ago, as of the day writing this, the first ever artificial satellite Sputnik 1 was launched [NASa].
Almost 3 years later on the 1st of April 1960 TIROS-1 was the first successful weather satellite launched by the National Aeronautics and Space Administration (NASA) [NASb].
10 Years later a total of 120 artificial satellites were active and orbiting the earth [Con].
During this time a combination of highly specialised hardware, software and people was needed to be able to receive and analyse the data being transmitted by these satellites.
Using as example weather prediction, one of the first and most important application of satellites, we can see how complicated and rather long this was. During the 60's the required steps to be able to receive forecast were many.
First the meteorologists at the U.S Weather Bureau received pictures directly from weather satellites, these would be then analysed and resulted to hand drawn "nephanalyses" (cloud depiction charts). These were then transmitted again although on a different transmission medium to forecast centres, to be then shared with everyone.
Problem of this approach, was that the data often reached the centres too late to be of any use [OAb]. Since we don't need weather predictions to know if it rained 1 hour ago, this was problematic.
To solve this a new way of transmitting the data was developed, the weather satellites would directly broadcast the data to the centres using "a format that could be received and reproduced by relatively inexpensive ground station equipment" [OAb]. One of the new direct Readout Services was the Automatic Picture Transmission (APT), initially introduced in the 1960s.

Around the year 2000 for about 200 dollars, you could buy a Hamtronics R139 Weather Satellite Receiver, a machine specialised to capture and decode APT data. A Personal Computer (PC), sound card and a “simple antenna” was still required to view and receive APT data [For]. Although relatively low-cost, this was still mostly restricted to professionals and dedicated amateurs at the time.

Nowadays components are getting cheaper and cheaper and PCs more performant and capable.

So one question arises, how hard is it to capture and analyse data from weather satellites as of 2021? Furthermore, what type of equipment is needed? Can programming languages be used to decode the signal without the use of specialised hardware?

Given that an official document describing how to decode APT does not exist, most of the information found on this paper are based on different articles and guides found online and personal experience.

The goal of this paper is to help people write their own APT decoder as I did myself, link to my open-source project can be found under the link referenced here [Chab].

Additionally this paper will be discussing the basics to record live APT data from NOAA 15, 18 or 19, all of this as low-cost as possible. It won’t focus on the actual interpretation of the weather data or my actual implementation of an APT decoder, but rather on the needed steps and knowledge to implement one.

Chapter 2

APT

2.1 NOAA 15, 18 and 19

Currently there are 3 active weather satellites transmitting APT, the NOAA 15, NOAA 18 and NOAA 19. All of them are operated by the National Oceanic and Atmospheric Administration (NOAA).
NOAA 15 was launched the 13th of May 1998 its APT transmission frequency is 137.62MHz. NOAA 18 was launched the 20th of May 2005 and transmits APT at 137.9125MHz, while NOAA 19 was launched the 6th of February 2009 its APT transmission frequency is 137.1MHz. It is fascinating to see that these satellites are still working despite their age. Sadly, since NOAA has decided to switched to Low-Rate Picture Transmission (LRPT) for its new series of polar orbit satellites these will be the last Satellites using APT [NOA].

2.2 APT

APT was developed specifically for the transmission of weather images by satellite. “The analog APT signal is derived from the original digital data and is multiplexing so that only two of the original channels appear in the APT format" [OAb].
The original data consists of 5 high resolution pictures, each picture or channel derives from a different sensor capturing different ranges of wavelengths. The High-resolution picture transmission (HRPT) transmits all of them at higher resolution. While APT

transmits only two, which are selected from the ground control. The result is always two pictures side by side showing the same view of the earth with two different spectral bands. “The signal itself is a 256-level amplitude modulated 2400Hz subcarrier, which is then frequency modulated onto the 137 MHz-band RF carrier” [Sigb]. The required bandwidth should be about 40 KHz, more on this topic will be discussed in Chapter 3 section 3.7. Data transmitted using APT does not only consist of images but also contains: Telemetry data, sync data and space and minute markers. Figure 2.1 shows the APT Frame Format.

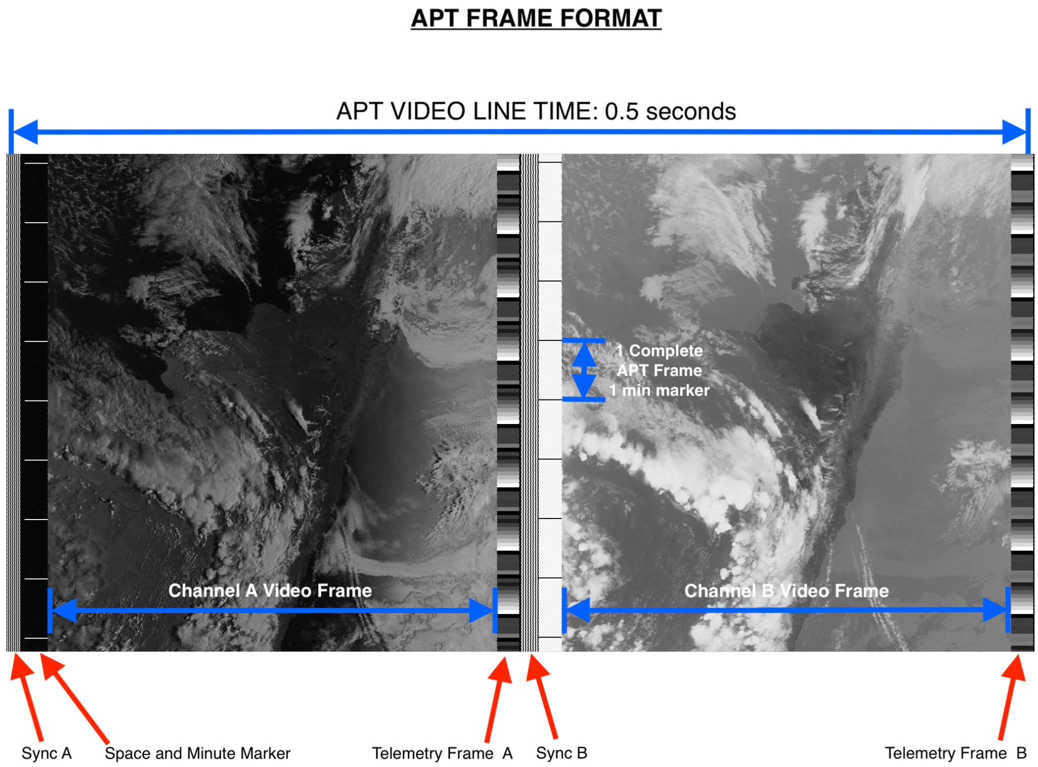

Figure 2.1: APT Frame Format [Chaa]

APT is transmitted line by line, with each line being 2080 pixels long and each pixel corresponding to 4km of the earths surface. "NOAA still images, audio files and video generally are not copyrighted" [OAa]. So anyone should be able to capture and share them for non-commercial use.

Chapter 3

Receiving APT

3.1 Sat tracking

To be able to receive and capture APT data, one needs to predict when the next satellite pass will occur.
To do so different programs exist, depending on the Operating System or specification one might prefer one over the other, personally I used WXtoImg since it has a function to print a list of the next passes [And]. Other options that I could find are: Gpredict [Cse], Orbitron [Sto] or the browser based application N2YO [N2y].
Regardless of what software is used to predict the next pass, what is usually needed is Two-line element set (TLE) data, a format able to store information about satellites and used by programs like WXtoImg to calculate prediction about satellite passes based on your location. Additionally, the current position is needed, as well as what satellite you want to track. In our case NOAA 15, 18 and 19.
The result should be a list of the next passes bound on your position for these satellites. Since NOAA satellites are polar orbiting satellites they always pass from south to north or from north to south, this should be shown in the result. Other important information that should be listed in the result are: time and duration of the pass, used frequency and Maximum Elevation (MEL) sometimes also noted as Elevation (EL). Generally speaking the higher the MEL, the stronger the Signal (90 degrees means the satellite will pass directly overhead).

Depending on the location of your ground station this information might be really important. For example if your ground station is surrounded by tall mountains or buildings and the satellite passes with a maximum elevation of 10 degrees above horizon you most likely won't be able to capture a clean signal. This is due to the fact that object like mountains or building saturate the signal.
The Duration of the pass is also key since 2 lines are transmitted per second with a pass of a minute a maximum of 120 lines can be captured, resulting in a 2080x120 image. Usually a good pass will have a window up to 15 minutes for a total of 2080x1800 pixels. Another important information that might be useful when deciding which pass to capture, is that the transmitted image is in "real time" meaning if you capture a pass during daytime the image will be different in comparison to capturing it during night or dusk. Passes recorded during daylight might look "better".

3.2 Ground Station

Where you decide to locate your ground station or simply said where you locate your antenna, is going to be in my experience one of the biggest factors. As previously mentioned, the amount and quality of the signal you will be able to capture is greatly influenced by the surrounding of your ground station.
Generally speaking, you want the antenna to be as elevated from the horizon as possible. Also depending on your country and position there might be other signals using the same frequency. For example, even though the frequencies from 137.035KHz to 137.850KHz should be reserved in the EU for weather Satellites I could notice a few times air traffic control transmitting at or near 137.100KHz and resulting in a few minutes of strong interference. Other possible sources of interference are Computers, Pagers, Power Lines or even other Satellites in some situations. A really good documentation about this topic can be found under the link referenced here [Sew]. This to say that picking a good location for your ground staton greatly influences the results.

3.3 Antenna

To receive a signal an antenna is needed. An antenna converts electrical current into electromagnetic waves to transmit signals and vice versa to receive them. NOAA APT weather satellites broadcast as already mentioned at frequencies around 137MHz, with a right hand circularly polarised signal.
Many low-cost options exist one of them being the V-Dipole antenna which is the option I personally tried. The other options won't be discussed here, but a really good and in-depth guide can be found under the link referenced here. [Rtlb]. The design I went for is the V-Dipole Antenna. An in-depth Guide to make your own can be found referenced here [Adab]. Also a simple schematic can be seen in Figure 3.1.

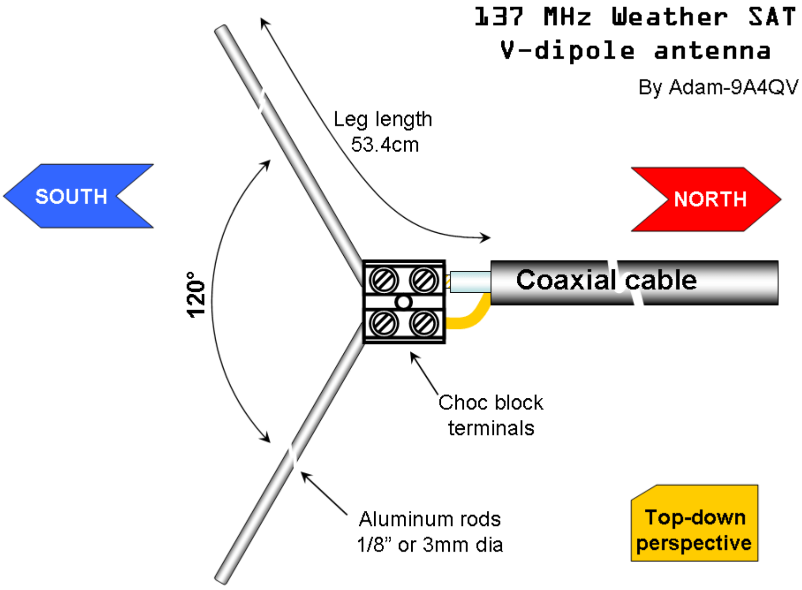

Figure 3.1: 137Mhz V-dipole Antenna [Adaa]

I decided to use this design since it is relatively easy and cheap to make. Additionally it does not require a motor to turn the antenna since it's optimized to capture signals travelling in front, above and behind the antenna and since polar orbiting satellites always travel North to South or vice versa, we can take advantage of this fact simply by orienting the antenna North to South as shown in Figure 3.1.
The version I made had aluminium rods with a diameter of 6mm and the RTL SDR

directly connected to the choc block. I wouldn't suggest to use thicker aluminium rods then 3mm, but in my case I couldn't find any other and it turned out to still work. While removing the coaxial cable turned out to be a good decision. Making the travel distance of the signal from the antenna to the RTL SDR as short as possible and so less prone to degradation. To connect my computer to the RTL SDR I used a 3m long USB extension cord so that the computer could stay further away, since the output of the RTL SDR is a digital signal, signal degradation should not be a problem. My version of the V-Dipole antenna can be seen in Figure 3.2.

Figure 3.2: Own version of V-dipole Antenna [Chaa]

3.4 Software Defined Radio

"Software-defined radio (SDR) is a radio communication system where components that have been traditionally implemented in hardware (e.g. mixers, filters, amplifiers, modu-

lators/demodulators, detectors, etc.) are instead implemented by means of software on a personal computer or embedded system" [con21].
The main advantage being that compared to traditional receivers, SDR is more flexible since it digitalises the signal "directly" so that any application can use it. Improving or implementing radio protocols is also easier since changes or additions just need to be made to the software, without the need of any change to the hardware in most cases.

3.5 RTL-SDR

"RTL-SDR is a very cheap [...] dongle that can be used as a computer based radio scanner for receiving live radio signals in your area [...]" [Rtla]. In my case I bought one of the cheapest versions a NESDR Mini with a R820T chip with frequency range from 25MHz to 1750MHz. For the purpose of receiving APT this is perfect.

3.6 RTL-SDR Software

There are many options today when it comes to the actual software SDR# [Air] being probably the most known. Depending on Operating System and preference one might use one over the other, here are some of the alternatives I could find (all free at the moment): HDSDR (Windows) [Ben], SDR++ (Windows and Linux) [Rou] and CubicSDR (Windows, Linux, OSX) [Cjc].

3.7 Receiving the Signal using RTL-SDR

Independently from what SDR Software one is using a few adjustments should be taken before recording the APT Signal.
In my experience the first one being the "calibration" of the RTL frequency offset. The easiest and fastest way of doing this is to tune in the RTL-SDR to a known radio station and adjust the offset (often called ppm) until the known radio station frequency matches the frequency on the RTL-SDR.

In addition radio should be set to WFM and the input rate can be lowered to about 1.00Msps to lower "unnecessary" Central processing unit (CPU) workload.
According to the official documentation "The signal deviation of the TIROS series transmission is +/- 17 KHz" [Rtlb] so in perfect conditions a bandwidth of 34KHz should be used (double the transmission frequency). Although if not correcting for the doppler effect, for example through the use of software like Gpredict, you should expect a frequency shift of about +/- 4.5 KHz during an overhead pass [Rtlb]. So, adding 4.5KHz to the bandwidth should mitigate for the doppler effect.
At last turn off Auto Gain if enabled and increase as much as possible the RF Gain without introducing too much noise, this usually takes a bit of trial and error.

3.8 Recording a Pass

Once the Ground Station is setup all is left to do is to wait for a good pass of a Satellite at the right frequency and record the transmission.
This process can be also completely automated through the use of Software such as Gpredict radio control and specific plugins to connect the the radio control to the SDR Software. If all was done correctly a distinctive "tick-tock" sound should be able to be heard. A sample can be found under the page referenced here [Siga]. The sound you can hear is actually the sequence used to sync the lines later on. Something else to look for during the pass is the Signal to noise ratio (SNR) the higher the better, in my experience I had a SNR of about 30dB.

Chapter 4

Decoding APT

4.1 Preparing the Signal

Recordings are usually saved as Waveform Audio File Format (WAV) files with a sampling rate of 11Khz, this is often referred as I/Q Data (the APT signal represented in the time-domain).
The audio file needs to be sampled at 20800Hz. Depending on how the audio was recorded it might be stereo or mono. Since only one channel is needed a conversion to mono should be performed.
Different File Formats store the channels in different ways, so this operation is dependent on the format. Wav File usually store channels by interleaving the bytes for each sample. The number of bytes per channel is defined by the bits per sample.
With a WAV of 16 bits (2 bytes) per sample and a total of 2 channels, you would need to take 2 bytes, skip 2 bytes, take 2 bytes, and so on. A representation of the interleaving bytes can be seen below:

Sample 1	CH1	CH1	CH2	CH1	Sample 2	CH1	CH1	CH2	CH2

The result is an array only consisting of one of the Channels. By doing so we possibly halved the number of bytes needed to be further read.

4.2 Demodulating the signal

The Signal is amplitude modulated. To get the picture we need to demodulate the signal. One way is to get the envelope of the amplitude modulated signal.
This is possible by applying the Hilbert transform. The result of the Hilbert Transform is the analytic signal of the original signal as an array of complex numbers. The analytic signal has no negative frequency components. If you remove the imaginary part of the analytic signal you get the input signal as seen in Figure 4.1.

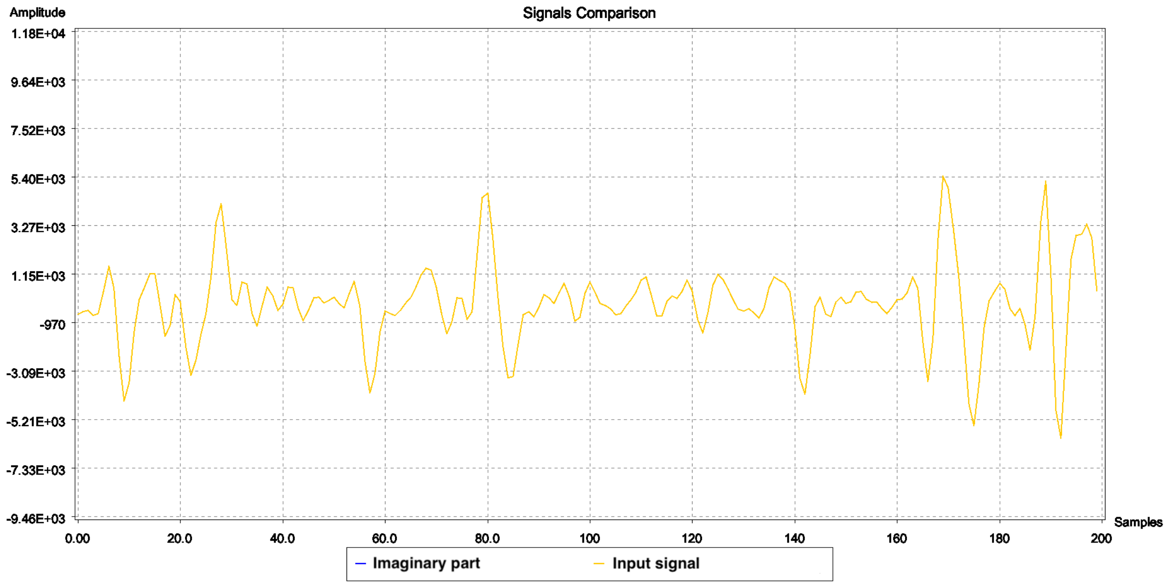

Figure 4.1: Imaginary part of analytic signal and input signal [Chaa]

The imaginary part of the analytic signal overlaps with the input signal.
While the amplitude enveloped is given by the magnitude of the analytic signal. An example of the amplitude envelope can be seen in Figure 4.2.

Figure 4.2: Amplitude envelope of example signal [Chaa]

4.3 Mapping the signal

We now have the envelope of the signal with values ranging from 0 to the arbitrary maximum amplitude of the input signal.
To make the following steps easier it is suggested to map the result values to a new range. Mapping the values from 0 to 255 which is the range used by the RGB color model, is the most common practice.
Since the maximum and minimum values of the signal envelope is going to greatly effect the mapping of the values to a new range, it is advised before doing so, to clean the signal of any possible noise or interference. For example an interference like seen in Figure 4.3 could lead to a "false" maximum and so affecting the contrast of the image.

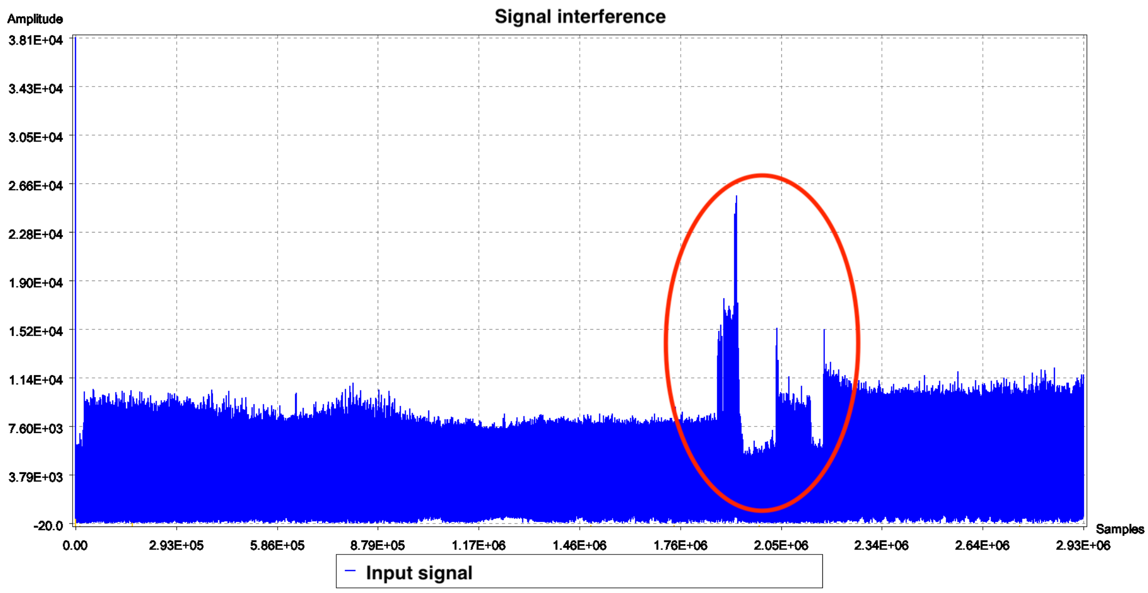

Figure 4.3: APT signal with interference [Chaa]

4.4 Drawing APT

With the values mapped to a new range it is now possible to create an image out of the signal.
Since one frame consists of 1040 pixels, one line consists of 2080 pixels, because two frames are contained per line.
Starting from the top left corner of the image we draw each pixel and start a new line every 2080 pixels. Each value of the signal corresponds to the brightness of a single pixel. Since the sensors taking the images use wavelengths outside of the visible light spectrum, the image will be in greyscale. Using the RGB color model, white consists of the triplets (255, 255, 255) while black (0,0,0).
So, for each value we draw a pixel with the same amount of red, green and blue to create a pixel in greyscale. This process is repeated until the end of the signal. Minding the fact that the last line could be shorter than 2080 pixels, in this case the rest of the line needs to be filled with "empty" pixels. The result should look like similar to Figure 4.4.

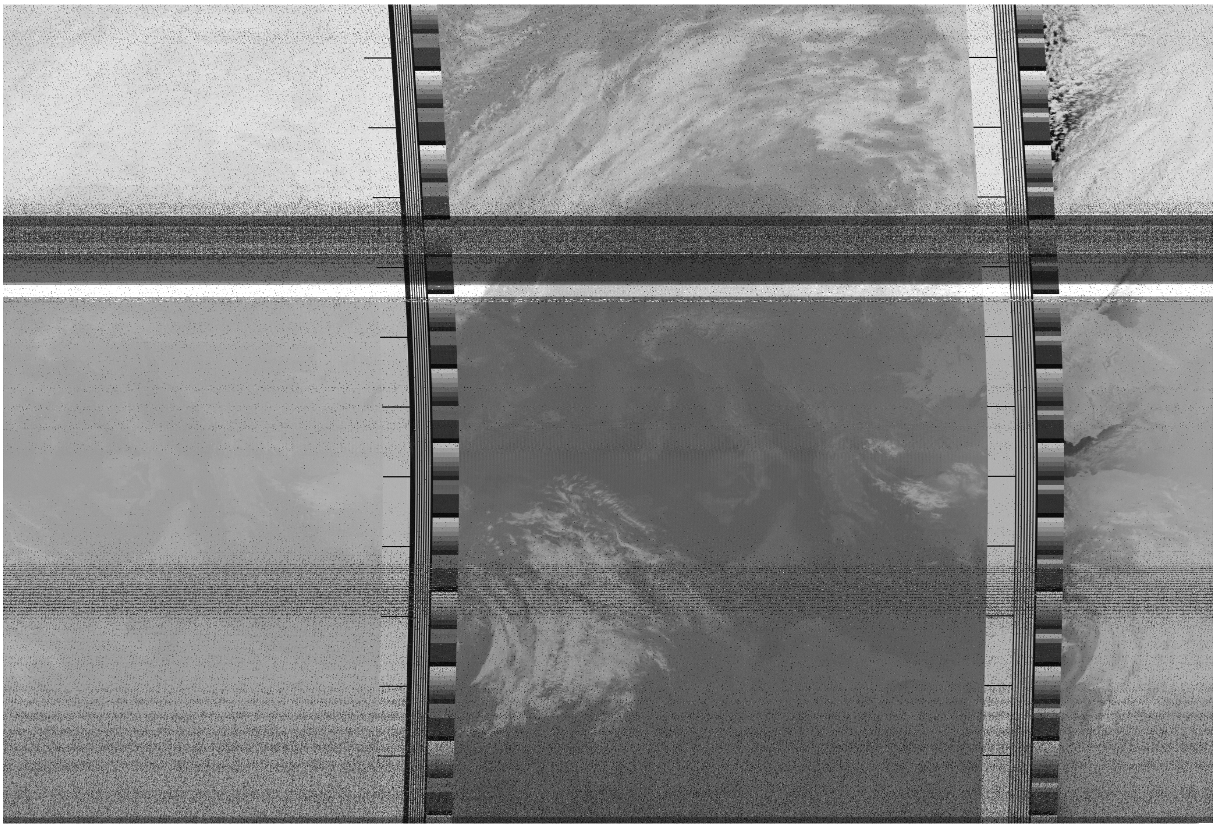

Figure 4.4: Not aligned APT Image [Chaa]

4.5 Syncing the Image

Looking at Figure 4.4 we can notice that the start of the frame is not aligned, but shifted by an offset. In this case this happened, because I started recording the signal not at the start of the line, but somewhere in the middle. Furthermore, if the receiver like in my case does not correct for the amount of shift caused by the doppler effect, the image will bow as seen in Figure 4.4.
One way of correcting this is using cross correlation. The sync sequence is known, the signal swings between two white pixels and two black pixels for seven cycles. This can be seen in Figure 4.5.

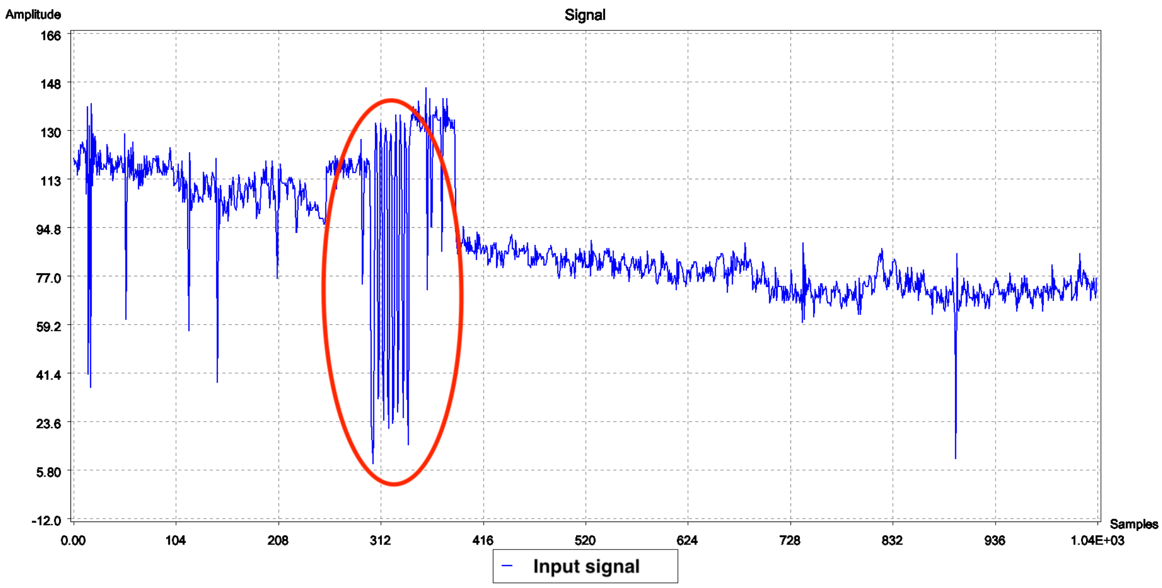

Figure 4.5: Example signal showing sync pulse [Chaa]

The sync pulse should look like the following sequence of pixels, where "B" stands for a black pixel while "W" for a white pixel:
B, B, W, W, B, B, W, W, B, B, W, W, B, B, W, W, B, B, W, W, B, B, W, W, B, B, W, W.
By cross correlating the received signal and the known sync pulse it is possible to find the start of the line and correct the offset for each of them. The result of the cross correlation is a value between -1 and +1 where -1 is a negative correlation and +1 is a positive correlation.
One way would be to search the sync pattern in the entire signal and calculate the offset for each line, this would work but be inefficient most of the times. The offset changes little line by line so most of them can be omitted. Furthermore only one sync pattern per line is needed to calculate the offset, so after finding one the next should be skipped. An example, where every positive cross correlation above 0.9 was marked with a red line can be seen in Figure 4.6.

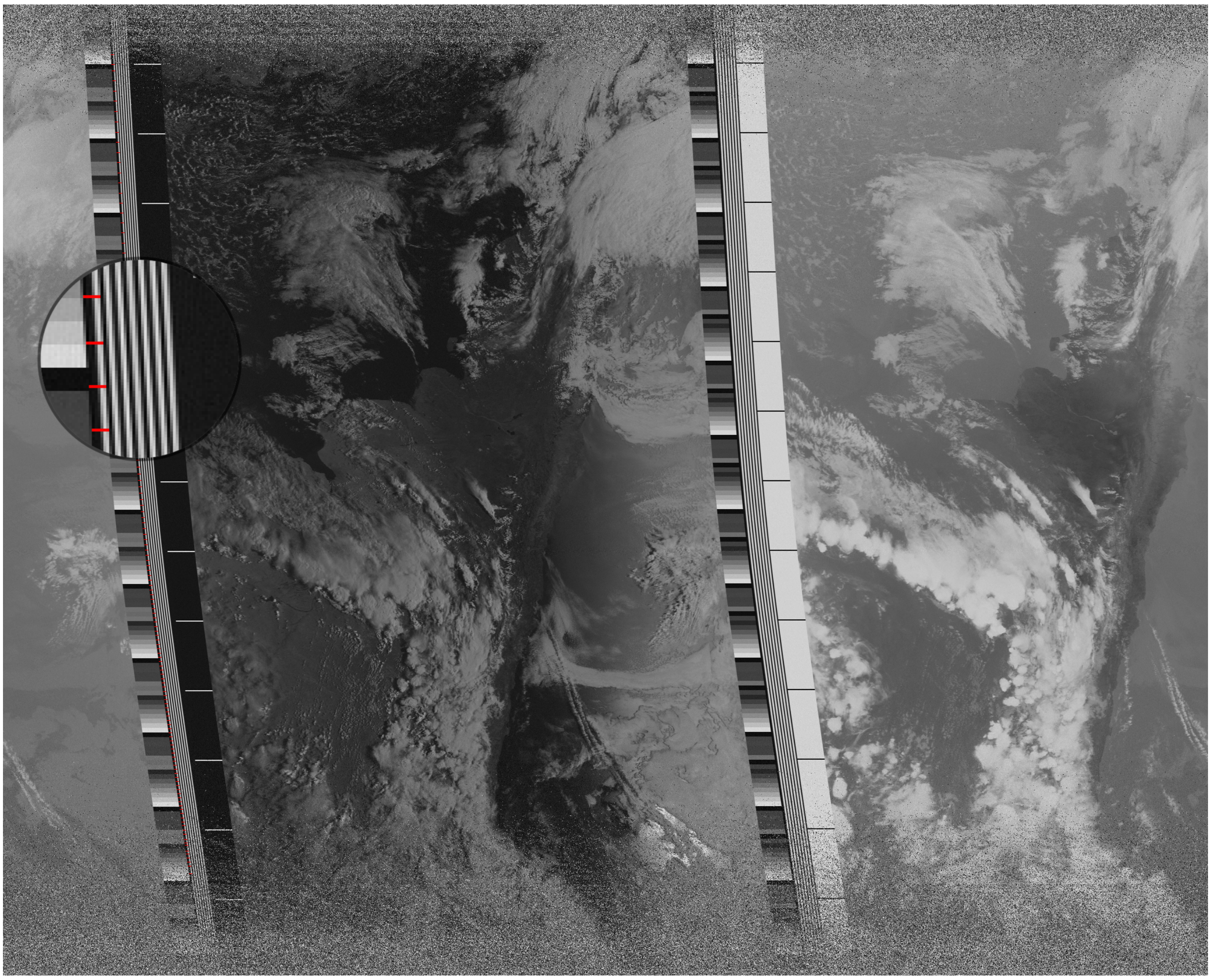

Figure 4.6: Example of cross correlation detection [Chaa]

As seen in Figure 4.6, most sync lines were identified by the cross correlation, the offset can be calculated for these lines. After correcting the offset the image can be drawn, this time aligned as seen in Figure 4.7.

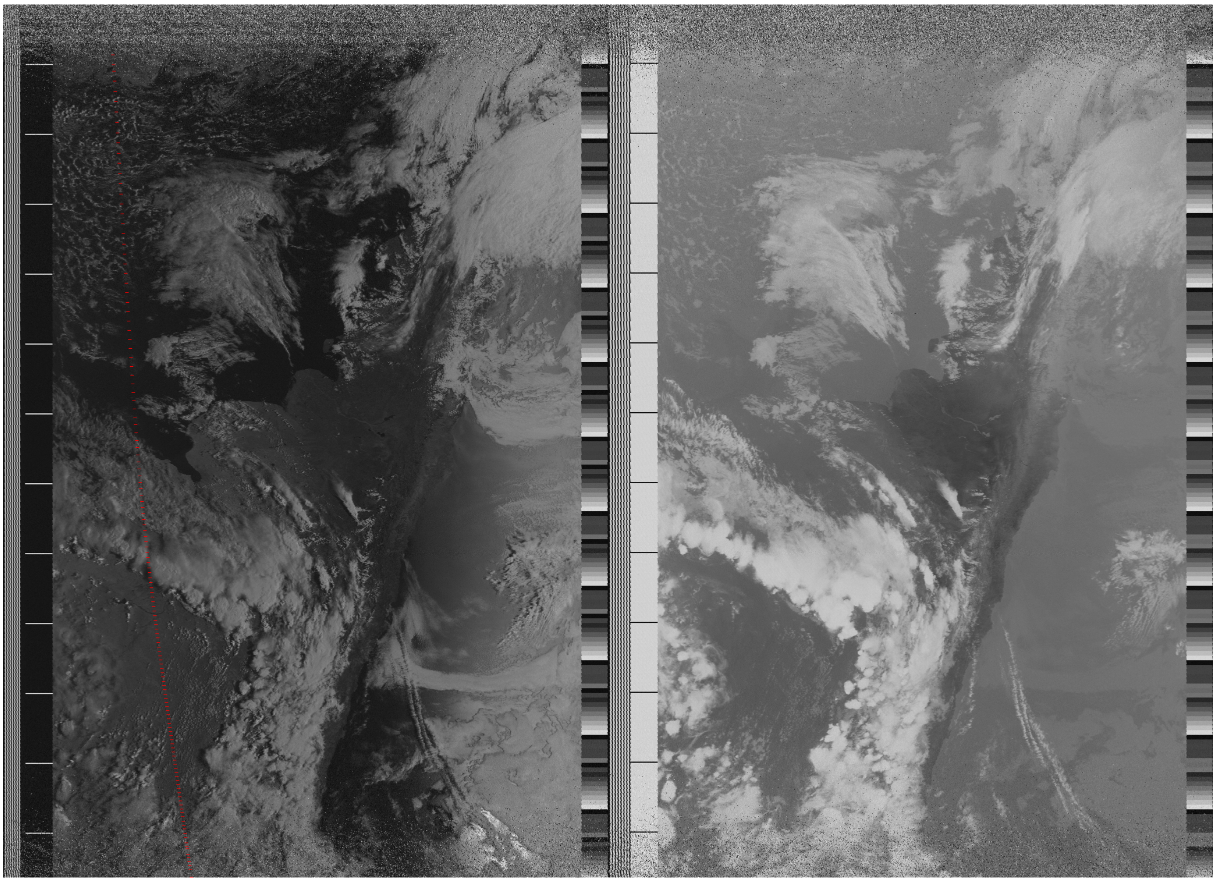

Figure 4.7: Example of synced APT Image [Chaa]

The image is now properly aligned as seen in Figure 4.7. The image at this point might be rotated by 180 degrees, this happens because the satellites can pass from north to south or from south to north. This information is not stored into the APT, so the decoder can't rotate them for you. At last my own result using my own apt decoder can be seen in Figure 4.8.

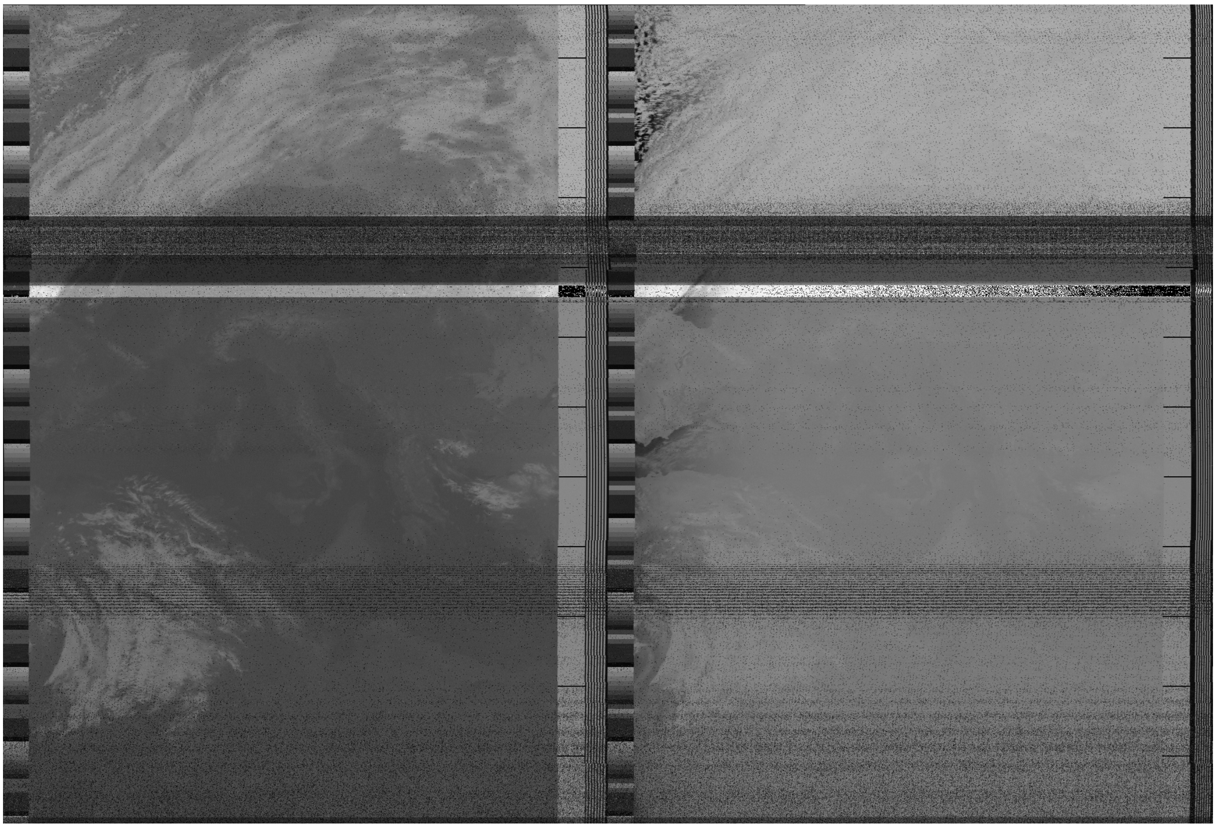

Figure 4.8: My result [Chaa]

List of Figures

Glossary

artificial satellite An artificial satellite is an object that people have made and launched into orbit using rockets [Sci]. 1

doppler effect The change in frequency of a wave in relation to an observer who is moving relative to the wave source [Wika]. 10

multiplexing In telecommunications and computer networks, multiplexing is a method by which multiple analog or digital signals are combined into one signal over a shared medium. [Wikb]. 3

polar orbit A polar orbit is one in which a satellite passes above or nearly above both poles of the body being orbited on each revolution. [Wikc]. 3

transmission medium A transmission medium is something that can mediate the propagation of signals for the purposes of telecommunication. [Wikd]. 1

List of abbreviations

Bibliography

[Adab] Adam-9A4QV. *DIY 137 MHz APT Weather satellite antenna*. URL: https://www.dropbox.com/s/6fpfn2p9filc9ol/DIY%20137MHz%20WX-sat%20V-dipole%20antenna.pdf?dl=0 (visited on 07/03/2021).

[Air] Airspy. *Airspy*. URL: https://airspy.com/ (visited on 07/03/2021).

[And] Craig Anderson. *WXtoImg*. URL: https://wxtoimgrestored.xyz/ (visited on 07/03/2021).

[Ben] Alberto di Bene. *HDSDR*. URL: http://www.hdsdr.de/ (visited on 07/03/2021).

[Chab] Enrico Gamil Toros de Chadarevian. *APT Decoder*. URL: https://github.com/enricoDec/Apt-decoder (visited on 08/03/2021).

[Cjc] Cjcliffe. *CubicSDR*. URL: https://github.com/cjcliffe/CubicSDR (visited on 07/03/2021).

[Con] Union of Concerned Scientists (UCS). *UCS Satellite Database*. URL: https://www.ucsusa.org/resources/satellite-database (visited on 07/03/2021).

[con21] Wikipedia contributors. *Software-defined radio — Wikipedia, The Free Encyclopedia*. 2021. URL: https://en.wikipedia.org/w/index.php?title=Software-defined_radio&oldid=1008886485 (visited on 07/03/2021).

[Cse] Alexandru Csete. *Gpredict*. URL: http://gpredict.oz9aec.net/ (visited on 07/03/2021).

[For] Steve Ford. *Hamtronics R139 Weather Satellite Receiver*. URL: https://www.arrl.org/files/file/Technology/tis/info/pdf/Hamtronics%20Weather%20Receivers.pdf (visited on 07/03/2021).

[N2y] N2yo. *N2yo*. URL: https://www.n2yo.com/ (visited on 07/03/2021).

[NASa] NASA. *Sputnik 1*. URL: https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=1957-001B (visited on 07/03/2021).

[NASb] NASA. *The Television Infrared Observation Satellite Program (TIROS)*. URL: https://science.nasa.gov/missions/tiros (visited on 07/03/2021).

[NOA] NOAA. *Future NOAA Polar Orbiting and Geostationary Satellite Systems*. URL: https://web.archive.org/web/20150502135907/http:/noaasis.noaa.gov/NOAASIS/ml/future.html (visited on 07/03/2021).

[OAa] National Oceanic and Atmospheric Administration. *Image Licensing and Usage Info*. URL: https://www.omao.noaa.gov/find/media/images/image-licensing-usage-info (visited on 20/03/2021).

[OAb] National Oceanic and Atmospheric Administration. *User's Guide for Building and Operating Environmental Satellite Receiving Stations*. URL: https://noaasis.noaa.gov/NOAASIS/pubs/Users_Guide-Building_Receive_Stations_March_2009.pdf (visited on 07/03/2021).

[Rou] Alexandre Rouma. *SDRPlusPlus*. URL: https://github.com/AlexandreRouma/SDRPlusPlus (visited on 07/03/2021).

[Rtla] Rtl-Sdr. *About RTL-SDR*. URL: https://www.rtl-sdr.com/about-rtl-sdr/ (visited on 07/03/2021).

[Rtlb] Rtl-Sdr. *RTL-SDR Tutorial: Receiving NOAA Weather Satellite Images*. URL: https://www.rtl-sdr.com/rtl-sdr-tutorial-receiving-noaa-weather-satellite-images/ (visited on 07/03/2021).

[Sci] Science Learning Hub. *Artificial satellites*. URL: https://www.sciencelearn.org.nz/resources/269-artificial-satellites (visited on 23/03/2021).

[Sew] Alan Sewards. *Interference to APT/WEFAX Weather Satellite reception*. URL: https://www.satsignal.eu/wxsat/PagerProblems-Sewards.pdf (visited on 07/03/2021).

[Siga] SigidWiki. *APT Sound*. URL: https://www.sigidwiki.com/images/f/fb/NOAAAPT_Sound.mp3 (visited on 07/03/2021).

[Sigb] SigidWiki. *Automatic Picture Transmission*. URL: https://www.sigidwiki.com/wiki/Automatic_Picture_Transmission_(APT) (visited on 07/03/2021).

[Sto] Sebastian Stoff. *Orbitron - Satellite Tracking System*. URL: http://www.stoff.pl/ (visited on 07/03/2021).

[Wika] Wikipedia contributors. *Doppler effect — Wikipedia, The Free Encyclopedia.* URL: https://en.wikipedia.org/w/index.php?title=Doppler_effect&oldid=1012798446 (visited on 23/03/2021).

[Wikb] Wikipedia contributors. *Multiplexing — Wikipedia, The Free Encyclopedia.* URL: https://en.wikipedia.org/w/index.php?title=Multiplexing&oldid=998829098 (visited on 23/03/2021).

[Wikc] Wikipedia contributors. *Polar orbit — Wikipedia, The Free Encyclopedia.* URL: https://en.wikipedia.org/w/index.php?title=Polar_orbit&oldid=1003708999 (visited on 23/03/2021).

[Wikd] Wikipedia contributors. *Transmission medium — Wikipedia, The Free Encyclopedia.* URL: https://en.wikipedia.org/w/index.php?title=Transmission_medium&oldid=1007724391 (visited on 23/03/2021).

Image sources

[Adaa] Adam-9A4QV. *137Mhz V-dipole Antenna.*

[Chaa] Enrico Gamil Toros de Chadarevian.