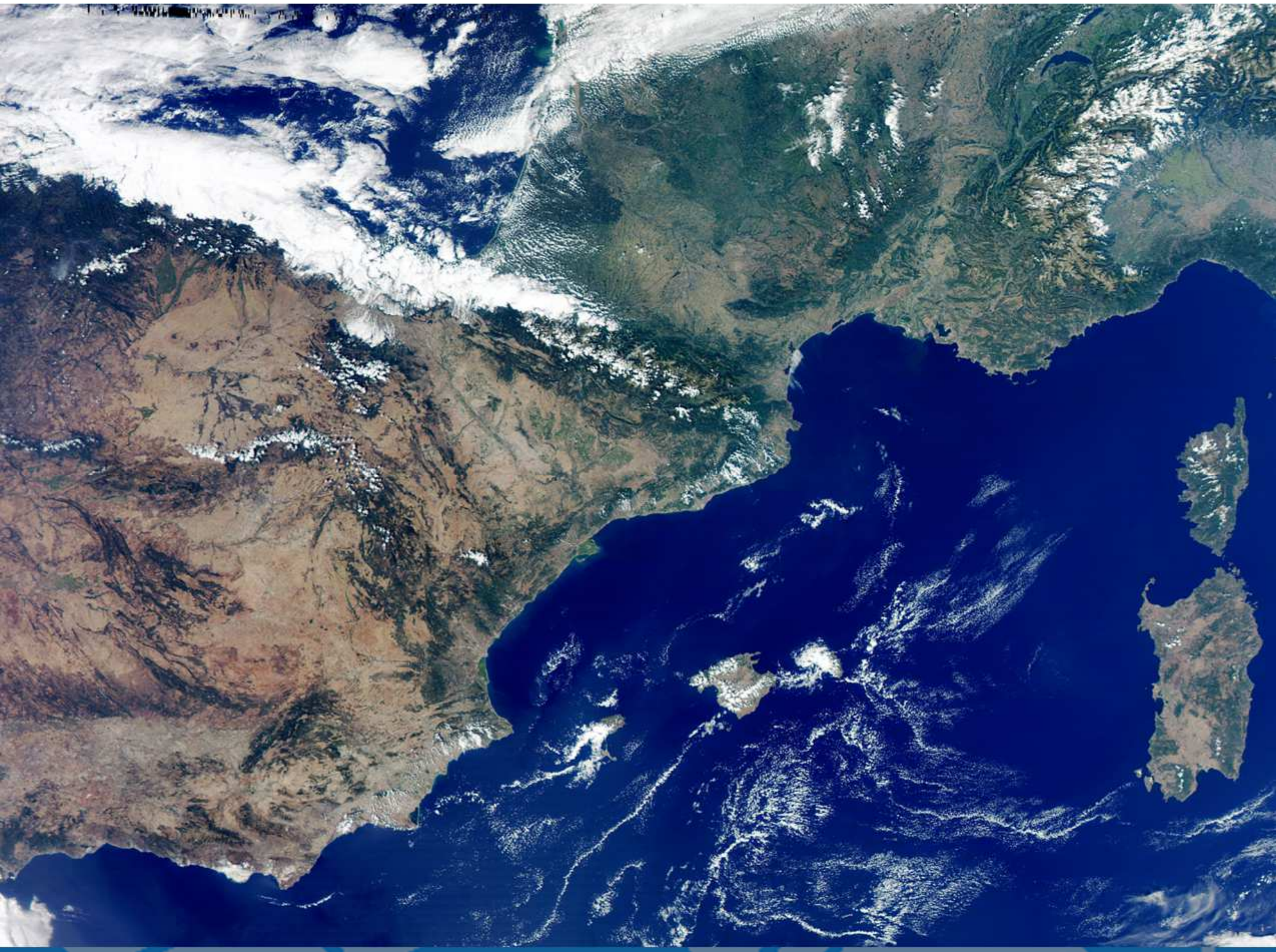
DE KUNSTMAAN

December 2020 – 47e jaargang nr. 4

Uitgave van de Werkgroep Kunstmanen



In dit nummer o.a.

Satellietontvangst in X-band

Nauwkeurig volgen van polaire satellieten

Dubbele 5V 500mA voeding

en nog veel meer

Dear member,

This pdf contains translated articles of our Dutch magazine “De Kunstmaan”.
Translation for each article is normally done by the author, using Google Translate (and manual corrections afterwards). But for sure these translations are not perfect! If something isn't clear please let us know.

Formatting is not as perfect as the paper magazine, but figures are all added.

Internet links mentioned in the articles can be found at our website; see under menu 'Weblinks' at: www.kunstmanen.net

Older magazines, from 2014 to 2018, are now also available in English; see menu “De Kunstmaan”, “Archief”.

I hope these translations will help you to understand the Dutch articles.

Rob Alblas
werkgroep Kunstmanen
kunstmanen@alblas.demon.nl

Content

Photo front page:

Recording 5 september 2020 of Aqua satellite on X-band, by Fred Jansen

Preface

Meetings

At the time of writing, the government is considering announcing stricter measures in the fight against the corona virus. Getting together seems further away than ever. Fortunately, we have telephone, email and zoom to keep in touch with each other.

We have a small club of members who attend the digital meetings, which is great. But I will not accept this as the "new normal". Maybe a meeting in March? We are then a year further ....

Paper or not

If you look at the working group's budget, you will see that a very large part (about half) is spent on shipping via PostNL . We hoped that shipping via Sandd was cheaper, but they have been taken over by PostNL and the rates are only going up. In addition, the shipping is bad: damaged or wet envelopes and it takes a long time.

The bullet is through the church, we're going digital. We know a number of variants:

- 10 Euro membership for a PDF only (both Netherlands and abroad)

- 28 Euro membership for a paper Kunstmaan (as it was) and a PDF

- 33 Euro membership for paper abroad membership

If you are very attached to paper, you can also print the PDF in A5 format. Nowadays you have affordable double-sided A4 laser printers that can print a booklet.

We are thinking about a yearbook. This includes the timeless articles, so there is no preface and satellite status. But this will only be relevant in a year.

The Kunstmaan

As a follow-up to the lecture that Fred Jansen gave on September 12, he made a description of his reception system for the 8 GHz. What is special is that with a 1.5 m mesh dish it can receive the AQUA, NPP, JPSS-1 and FY3B!

Rob made a large contribution to this Kunstmaan. For example, there is an article about adjustments to the constellation viewer.

As a follow-up to Ger Smit's article in the previous Kunstmaan, an article about downloading MSG HRIT data . xrit2pic has also been adapted for this. For example, you can make a video of the past 24 hours with a Raspberry Pi.

For the rotor builders, Rob made an analysis of an azimuth / elevation rotor setup and looked at matches with high elevations. In the tested setup, the Jaeger rotors are too slow for the high matches

Paul has scrutinized the UKW messages for us. Unfortunately, it is not yet possible to view the magazine during the meeting.

I myself wrote a small article about a dual 5V power supply. This way you can keep the digital part separate from the analog part of the circuit that you want to feed.

It remains for me to wish everyone a Merry Christmas and a Happy New Year. The first meeting of the new year is on January 9, unfortunately digital again. We only have to postpone the New Year's reception for a year.

Ben Schellekens

Chairman of the Kunstmanen Working Group

PS: We remind you to pay your contribution before 2021. See above for the rate.

Receiving weather satellite data in X-band

1. Introduction

With the increasing quality of weather satellite data, more bandwidth is needed to transmit these data. This has led to a shift in transmission frequency to be able to use the required bandwidth. The increased spatial resolution and the increasing number of spectral bands in use requires transmission speeds up to 65 Mbit/s. In this article I describe my attempts (and the results) to make this possible using a satellite dish of reasonable size (1.5 m) – according to the pioneer of this work, Jean-Luc Millette, too small to be able to succeed.

2. What is needed ?

2.1. The antenna

As mentioned in the introduction, successful amateur reception of weather satellite images from missions like AQUA, Terra and Fengyun-3D (FY3D) has only been achieved in recent years. Successful reception of so-called AmateurDSN[1] signals was already possible before. Especially pioneers like Paul Marsh (@uhf_satcom on twitter), Scott Tilley (@coastal8049 on twitter) and Edgar Kaiser (@df2mz on twitter), have achieved great results. This work however focusses on the reception of a carrier signal whereas for decoding weather satellite imagery the full signal needs to be received with sufficient quality. The equipment required however, has significant similarities, such that re-use of successful ideas is a wise thing to do.

One of the first elements in the signal processing chain is the required dish with a motorized tracker (there are true heroes out there who do the satellite tracking by hand, however I wanted to be able to work under all weather conditions and at night. For receiving L-band HRPT data, a 1 m dish with a Yaesu G5500 tracker was sufficient. Once I had this working and started to think about X-band reception, it quickly became evident you need a larger dish, but the higher frequency (8 vs 1.7 GHz) also sets tighter requirements on the tracking system, and a better rotor system was needed. While the existing system of Jean-Luc Millette used a 2.4 m dish, I decided to set myself a challenge and wanted to try with a 1.5 m dish with a fine mesh reflector (to avoid excessive windload) and a SPID RAS/HR rotor and try to get things to work. The mesh has good reflectance up to 10 GHz.

2.2. The signal processing chain

In the process of assembling a signal processing chain I looked at different available designs and ideas, but the most promising one looked to be the design of Scott Tilley (Fig. 1), also as it is made up of components that are relatively easy to find and purchase. The design of Jean-Luc Millette based itself on a no longer available DVB-S LNB and thus was not suited for my purpose.

---

[1] The acronym DSN stands for the NASA Deep Space Network, a worldwide network (Goldstone, Californie – Canberra, Australia – Robledo, Spain) of satellite dishes to enable reception of signals from deep space missions to Mars, Jupiter and beyond. The European Space Agency (ESA) uses its own ESTRACK network for this purpose.

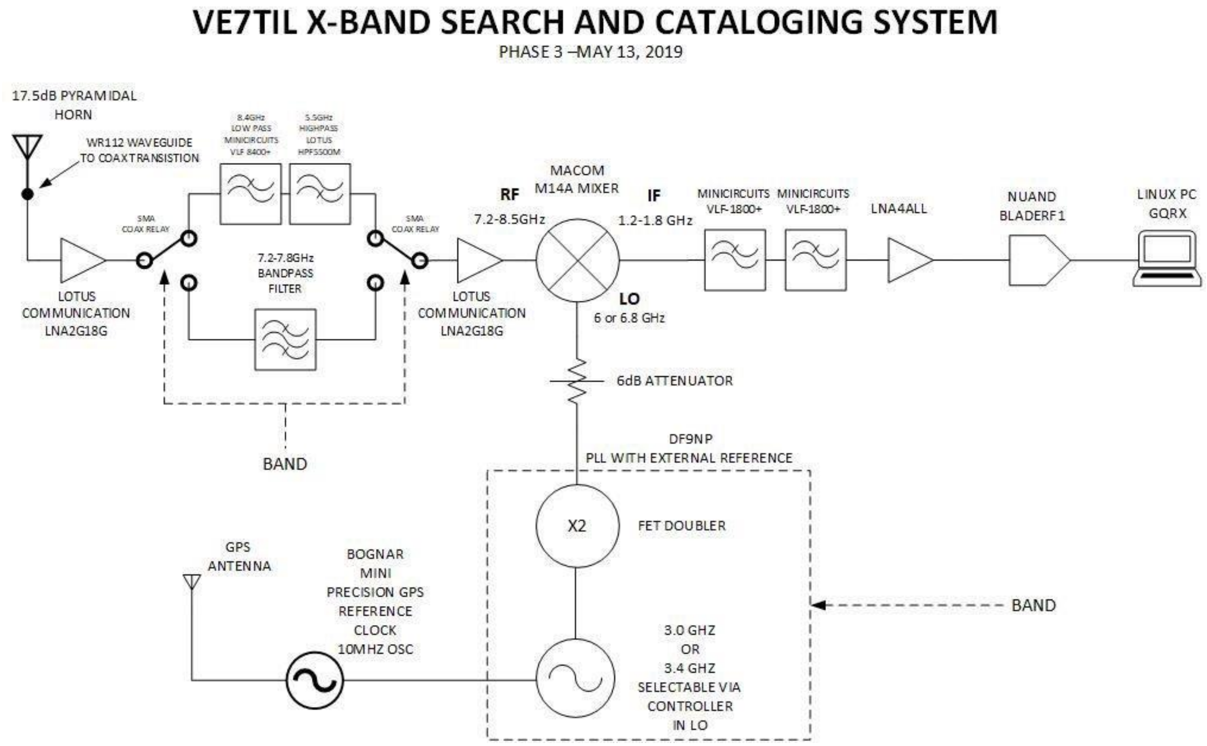


Fig. 1. Signal chain as used by Scott Tilley, which served as inspiration for designing my own system.

The design was changed by me to cater for availability of components and to optimise the noise figure.

- Instead of a horn feed I use a feed based on Jean-Luc's design, using a Delrin depolarizer. This feed was made bij Fons Buitelaar and later replaced by a septum feed he made (the advantage being that it has separate connections for left and right circular polarization).
- On ebay I was able to purchase an APT4-04000800-0710-D4 (Fig. 2) and used this to replace the LNA2G18G as the APT4 noise figure is only 0.7 dB.
- The band pass filter (BPF) I use is made of a VHF-6010+ and a VLF-8450+ (both from mini-circuits)
- Behind the BPF is a second LNA: LNA2G18G
- The Local Oscillator (LO) I use I got from Dieter Leupold's website and provides a 6.3 or 6.5 GHz LO signal – it is connected to the mixer via a VHF-5050+ filter to filter out the LO subharmonics at 3.15 and 3.25 GHz. De LO is driven by the 10 MHz signal from a Leo Bodnar mini-GPS receiver.
- The mixer is a M14A passive mixer (ebay)
- I tried using a LNA4ALL amplifier between the mixer IF exit and the SDR but this added nothing to the quality of the signal and was removed.
- The IF is connected to the SDR via a VLF-2250+ low-pass filter by mini-circuits.

Figure 3 shows the feed + LNA as mounted at the dish. The test setup was shown on the cover of the September 2020 Kunstmaan. Figure 4 shows the first measurements with this setup using a 1 m offset dish to receive the signal from the geostationary SYRACUSE 3B military satellite at 7705 MHz.

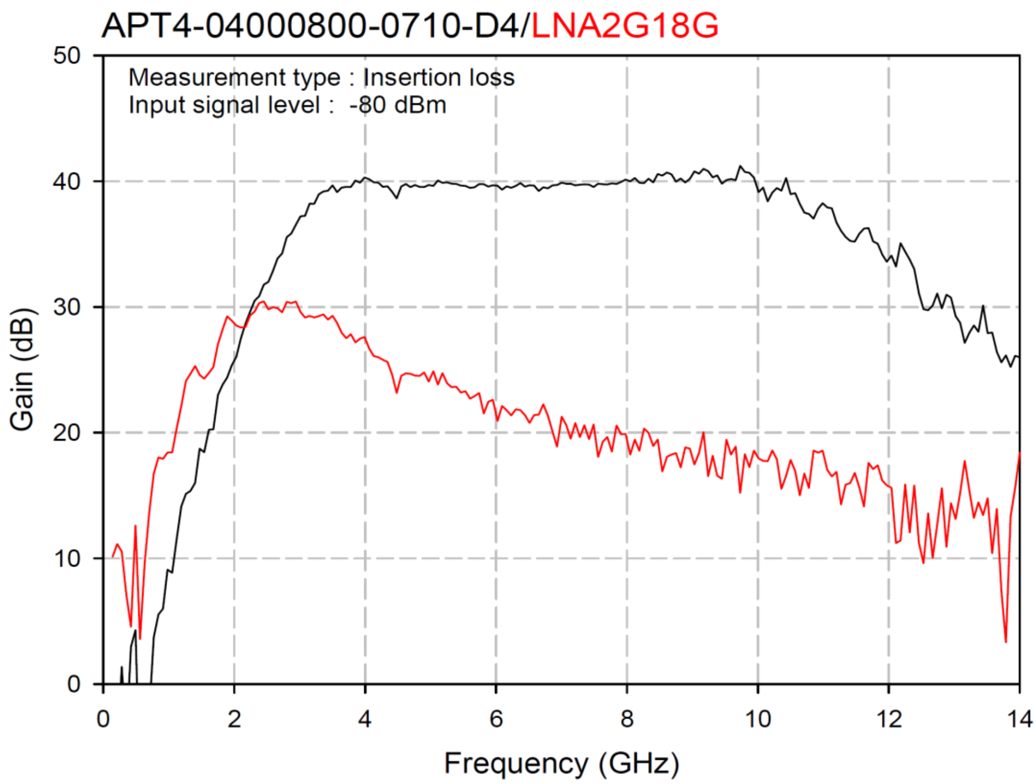


Fig. 2. Gain measurements of the APT4 and LNA2G18G LNA’s by Fons Buitelaar.

Fig. 3. Feed and LNA as mounted at the dish.

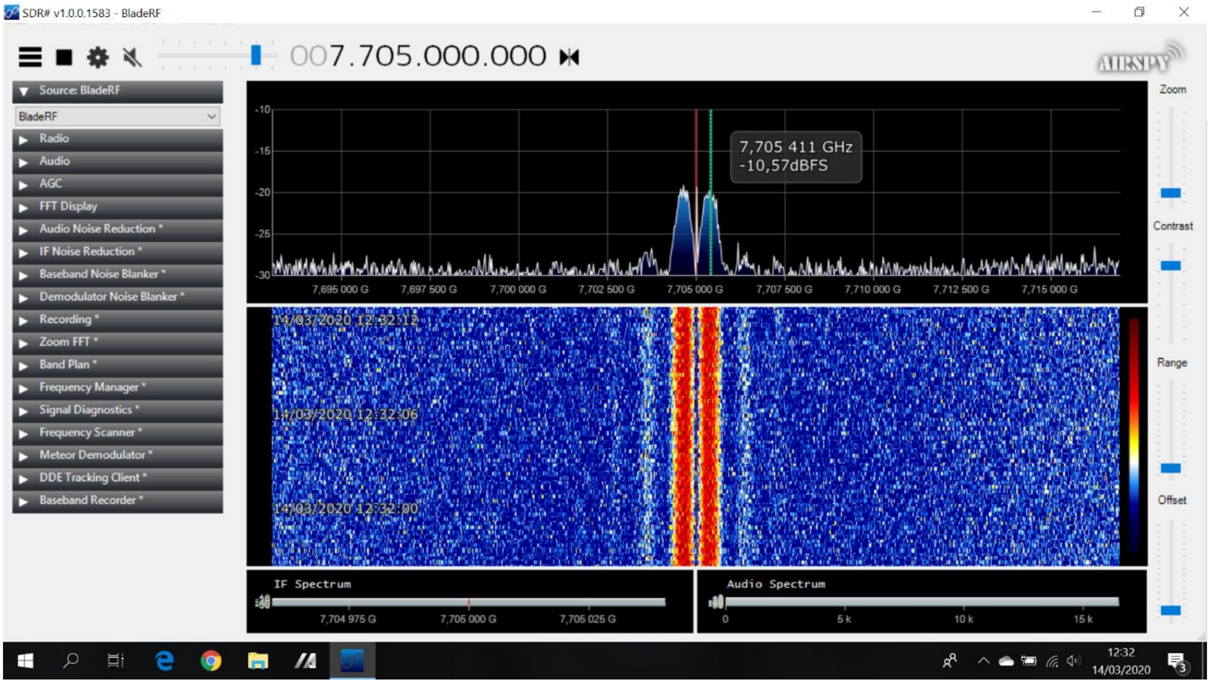


Fig. 4. First measurements in X-band (March 2020) using a 1 m offset dish and a BladeRF as SDR – SYRACUSE 3B at 7705 MHz

2.3. Tracking in reality

I decided to write my own tracking software to both achieve maximum flexibility and solve an old problem in most existing tracking software. The majority of the existing tracking software calculates the position of a satellite, aims to that position and once the satellite has moved by a (user defined) angle it points again. The disadvantage of this is that the moment the peak response of the dish arrives at the satellite position, this is already moving on and you are effectively only using one side of the response of the dish (Fig. 5). This can be partially fixed by looking ahead in time in the calculations. However, at low elevation the required look ahead time is of the order of a few seconds, while at high elevation (over 40 degrees) this look ahead time is only a few tenths of seconds. Besides this the angle used to decide if the dish needs to repoint most of the time uses the difference in Azimuth and Elevation angles as the criterion, while one should use the dot-product of the direction (= vector) the dish is pointing at and the vector pointing to the current position of the satellite. The new dish, tracker and software was first successfully used for L-band HRPT reception after which I started collecting the necessary equipment to move to X-band.

Now that the signal reception and processing chain were ready to receive and record the (still too weak) AQUA signals at 8160 MHz, the next step was to improve signal strength. One of the first steps was to check the position of the feed by looking at the shadow it throws on the dish when tracking the sun (Fig. 6). The signal strength while tracking AQUA however, remained rather low and variable. After purchasing a fairly precise inclinometer (0.3 deg accuracy) this quickly identified that one of the root causes was that the central pole supporting the setup is not exactly vertical. I introduced a correction for this in the tracking software and this significantly improved things.

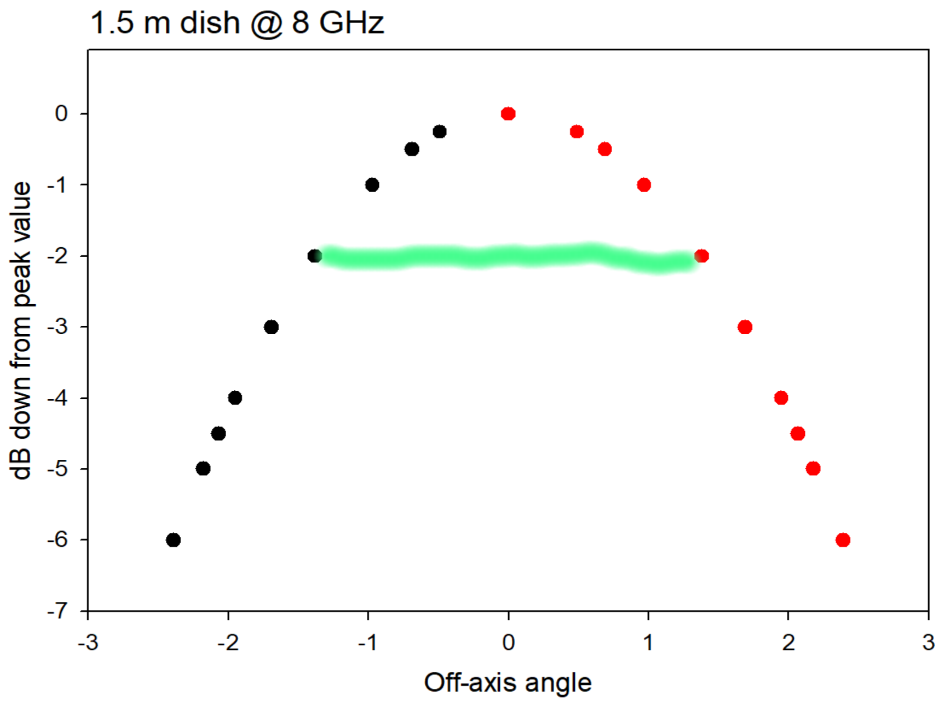


Fig. 5. Response of a 1.5 m dish at 8 GHz (from satsig.net). The red dots indicate the part of the response over which the satellite moves (see text) using “regular” tracking software (to illustrate I have assumed a re-pointing every 2.5 degrees). The signal thus varies from 0 to -6 dB attenuation. If however the software were to point 1.25 degrees ahead of the satellite and then wait for it to move by 2.5 degrees, the peak of the response is fully used, as indicated by the part above the green line. This would lead to a maximum attenuation of less than -2 dB.

Fig. 6. Shadow of the feed on the dish while tracking the sun. Using this, both the tracking software and the alignment of your system can be checked.

A second problem found was that despite having a local high-precision GPS driven NTP server, the windows built-in time synchronization was not good enough to keep the time within 0.7 seconds from UTC. After installing the Meinberg NTP client software this problem disappeared and time keeping now always is accurate to within a few milli-seconds from UTC.

2.4. The Software Defined Radio (SDR)

Although it would have been logical to buy the LimeSDR, as Jean-Luc also uses it, I decided to buy a BladeRF as this also has an adequate bandwith and also as the LimeSDR has a doubtful reputation amongst many people active in L-band and X-band reception. In principle the BladeRF worked fine (only the plastic housing is, from an RF perspective, not really convincing), but is not supported by the HDSDR software but is supported by the SDR console and SDR# software. What became painfully obvious after a number of measurements is that whatever I tried in optimising the RF chain or the feed the signal to noise ratio of the SYRACUSE 3B 7705 MHz signal barely improved. That's when I decided to directly compare SDR console and SDR# with the same signal and setup (Fig. 7). For all later optimization comparisons I decided to use SDR console with the central carrier of SYRACUSE 3B at 7705 MHz as it is very stable and thus allows good comparison.

A problem with the BladeRF turned out to be the presence of 'fake' signal peaks (spurs) in the RF spectrum. Even when I was able to receive a rather good AQUA signal, an analysis by Jean-Luc showed that the presence of these fake peaks in the signal made the demodulation of the signal nearly impossible. Because of this problem, and because the demodulation software requires an 8 bit .WAV file – for which I needed HDSDR – I decided to change course and use a LimeSDR after all. The installation of this under windows was very smooth and easy.

3. Results !

One of the final problems turned out to be that after tracking a satellite pass, the dish does not return to the same reference position. This is not caused by mechanical problems but most likely by a problem in the MD-02 controller – a problem reported on social media by quite some users of this controller. The solution I adopted is that before every satellite pass I point the dish at SYRACUSE 3B and check the reference signal level. If not correct I adjust the dish pointing for maximum signal and reset the MD-02 controller to indicate the right value.

All of this led to the successful reception of the first images from AQUA, SUOMI NPP, JPSS-1 and FengYun 3B (Fig. 8) – everything made possible by the software from Jean-Luc Millette and a lot of practical and moral support by Fons Buitelaar.

What I have not been able to achieve yet is to receive images from FengYun 3D – its signal appears to be weaker and more broadband than FengYun 3B. There still is a lot to do and to optimize, but the first successful steps have been made and I get the satisfaction of having achieved this with only a 1.5 m dish.

Besides the improvements in signal reception there are big challenges in implementing the different ways of Bow-Tie correction into my map projection software (see "De Kunstmaan" – June 2020). In order to achieve this I have to directly read the decoded data into my software and doing this is not being helped by an almost complete absence of documentation for most of the satellites – in particular the FengYun series.

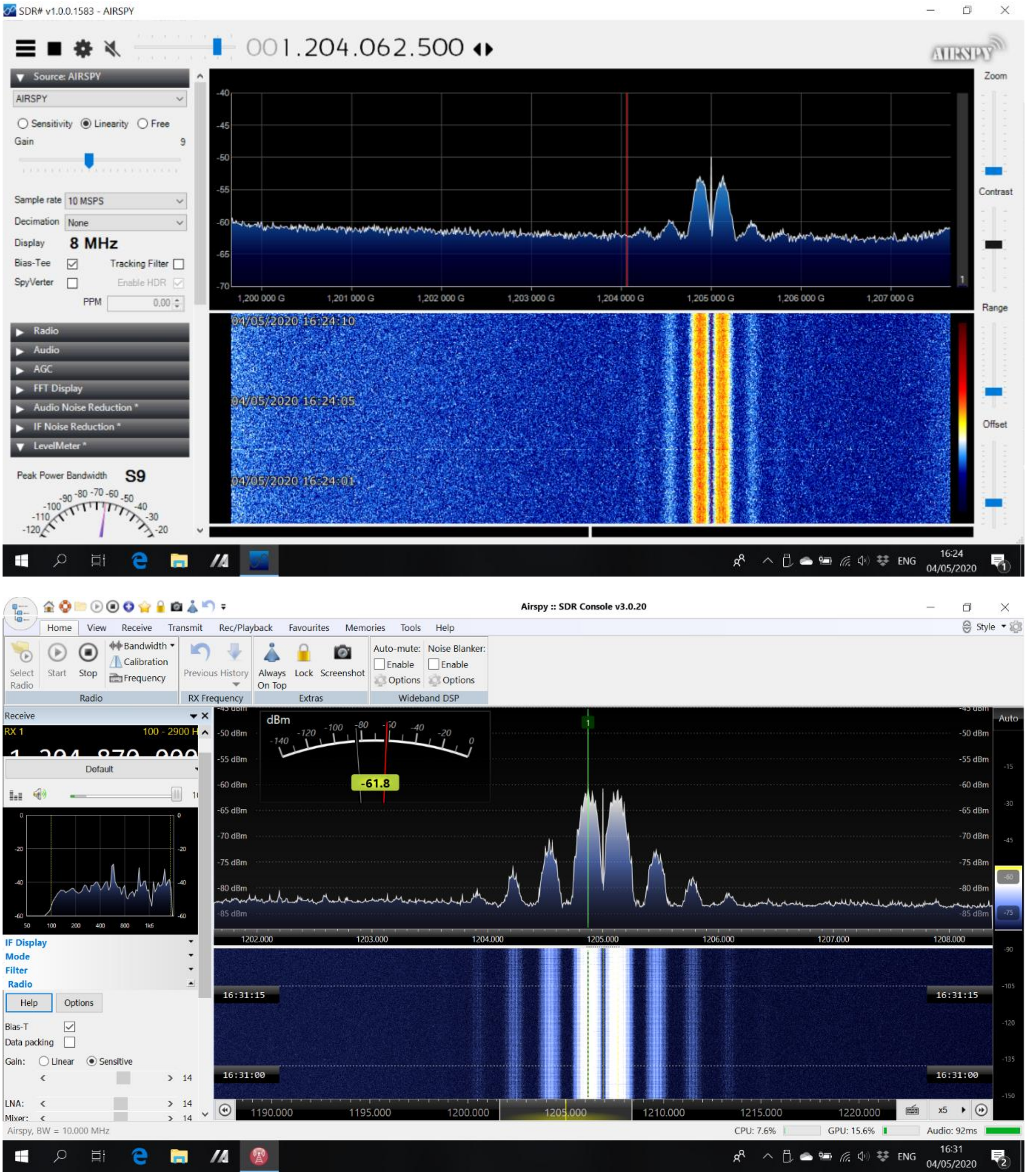


Fig. 7. Identical signal and signal chain tested with both SDR# (top) and SDR console (bottom). While in SDR# the first “side lobes”stick out some 10 dB above the noise floor, in SDR console this appears to be about 20 dB.

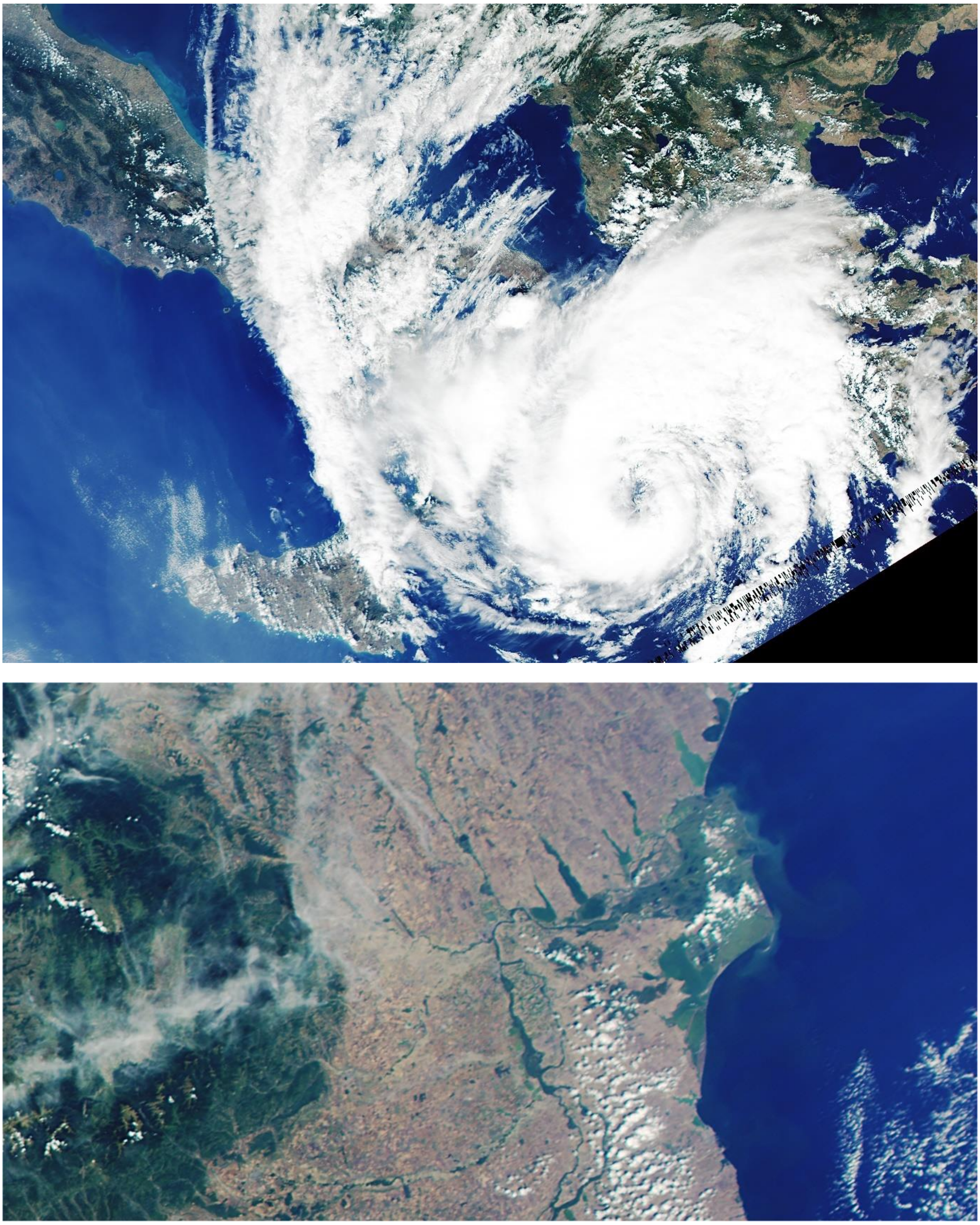


Fig. 8. Crop of an AQUA image of the mediterranean (top) showing part of Italy and Greece as well as a so-called “medicane”. At the bottom an FY3B X-band image crop of the danube delta in Romania. Both images are in “true colour” by combining several wavelength channels.

Analysing accuracy of tracking polar satellites

For reception of polar satellites in the X band, the accuracy of the tracking system is much more important than for reception of satellites in the L band. This year's magazine “de Kunstmaan” no. 2 has already paid attention to this ([1]). See also the article by Fred Jansen, elsewhere in this magazine.

The opening angle of the antenna is decisive; it is much smaller for the X band.
The opening angle (- 3dB point) can be calculated with:

$\theta = k \lambda / d$ (see [2])
Here, λ is the wavelength, d is the diameter of the antenna and k is a factor that depends, among other things, on the exact shape of the dish and the feed.

Ideally, k = 1 (θ in radians).
With θ in degrees and frequency $f = 0.3 / \lambda$ this becomes:
$\theta = 17.2 / f / d$
In practice, with a non-ideal antenna, the following formula is often used:

$\theta = 21 / f / d$

In [3] one can find a calculation tool in which a table can be generated with different efficiencies, diameters and frequencies.

For 1.7 GHz with a 100 cm dish, θ = 12 degrees; the -3 dB point is then at an angle of about +/- 6 degrees.

For 8 GHz, with an antenna diameter of 150 cm , this is +/- 0.9 degrees, so the accuracy should be almost a factor of 7 better.

Much stricter requirements must therefore be imposed on the aiming accuracy, in which the stiffness and stability of the entire construction certainly also count heavily. A larger, so heavier dish that must be more accurately aimed, whereby the bending of the construction at a low elevation must be much less than that 0.9 degree.

Also the targeting itself with rotors, including accurate calculation of elevation / azimuth and control of the motors, deserves much attention. With an elevation / azimuth-rotor system, an extra difficulty is the high needed azimuth rotation speed if a satellite with high elevation pass needs to be followed, although the azimuth-accuracy at very high elevation is less critical (at 90° the azimuth doen’t matter at all). In addition, the speed range of the azimuth motor must be much larger than that of the elevation motor. With a high pass, the rotation speed at the start and end of a pass is very low (almost 0); if the elevation approaches 90 degrees, the required rotational speed increases dramatically. Preferably you want to let the rotors run evenly, without 'hiccups', to prevent 'shaking' of the dish as much as possible. With an azimuth rotor this means on the one hand a low gear behind the motor (being able to rotate the dish evenly very slowly), but on the other hand a high gear to allow the dish to rotate fast enough at high elevations, with maximum motor rotation speed. It should also be remembered that you cannot run a motor as slowly as you want; namely, friction has to be overcome. If the voltage is too low, a DC motor “hangs” until the friction has been overcome, after which it may run too fast. In that respect, a stepper motor could be better handled, but it must also be able to run fast enough.

An X / Y system would be easier to realize in terms of required rotational speed: the variation in rotational speed is much lower here (see [2]). On the other hand, an X / Y construction is more complicated from a mechanical point of view, whereby the top rotor also has to be “dragged along” by the bottom rotor. Further, the distance of antenna to the first rotor is larger, so a larger moment, thus more prone to bending of the whole construction at lower elevations, and a higher load on that rotor . The “law of conservation of misery” abounds here!

In order to be able to test the tracking in practice, I have played a bit with a elevation / azimuth-rotor system with Jaeger motors (without antenna), made available by Harry v. Deursen, and a motor controller of Peter Smits. These are DC motors. I have expanded the controller, which is realized in an Arduino, so that the current position, requested position and speed are returned to the PC. In Xtrack ([5]) this info is captured and made visible; it can also be put in a log file. The amount of information per sec. is equal to the speed at which commands are sent to the controller, so at a command interval of 0.1 sec. you also get 10x the target / error information per second. See fig. 1.

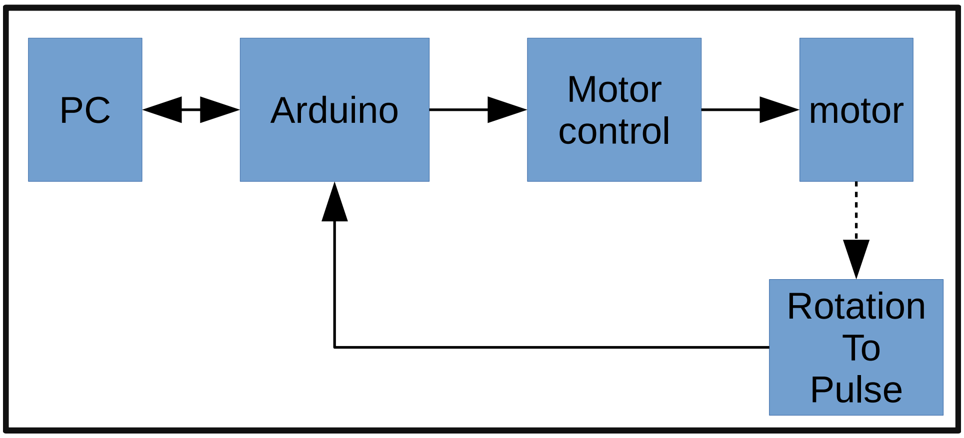


Fig. 1. Control and feedback.

Arduino software.

The first version of the Arduino software was written by Harry Arends. I have rewritten it more or less; the most important updates are:

- speed (almost) set stepless instead of 4 fixed speeds

- for the two motors one set of functions is written, which is invoked 2 times instead of separate functions for each motor
- nearly all variables are defined local, instead of global variables
- added: feedback of position rotors, using the same serial port used for sending commands to the Arduino.

The latter goes in the following format:
pos = [12.3, 78.0] req = [13.2, 78.0] spd = [50.0]
in which:

- pos: position; azimuth, elevation of the rotor
- req: azimuth, elevation 'request', so the desired position
- spd: the speed of both rotors in %

The Arduino software looks like this (in the “loop ()”):

- readCommand(&gotoval): capture new position from PC via USB, the positions will be in gotoval.ax and gotoval.ey (azimuth or x respectively elevation or y)
- rotor_goto(gotoval.ax, &AX_rot) : control of rotor nr. 1 (azimuth in this case)
- rotor_goto(gotoval.ey, &EY_rot) : control of rotor nr. 2 (elevation in this case)

Here 'gotoval' is defined as a record with which the values sent by the PC are passed on to function rotor_goto(); AX_rot and EY_rot are also records defined as type ROTOR. This record contains all the data of a motor, including the Arduino-pin by means of which the motor is activated, the received pulses from the feedback circuit etc. As a result, it is possible to control both motors with a single defined function, which is instantitated 2 times.

The three parts mentioned above will run through one after the other as fast as the Arduino can. In between, the pulses from the pulse generator of the rotor are counted; this is done via an interrupt.

The rotor_goto() function works as follows (the desired angle, from tracking software running on the PC, is entered as an argument as gotoval.ax or gotoval.ey respectively):

- translation counted pulses (from pulse generator via interrupt) into angle position of rotor
- determine difference between desired and current angle, this is the error
- determine required rotor speed; large error result in high rotation speed (proportional control, see fig. 2)
- control motor with PWM (Pulse Width Modulation)

Note that with a proportional control the rotor cannot follow perfectly by definition; the higher the required speed, the greater the error angle must be in order to reach that speed. Fig. 2 shows the transfer of the speed control.

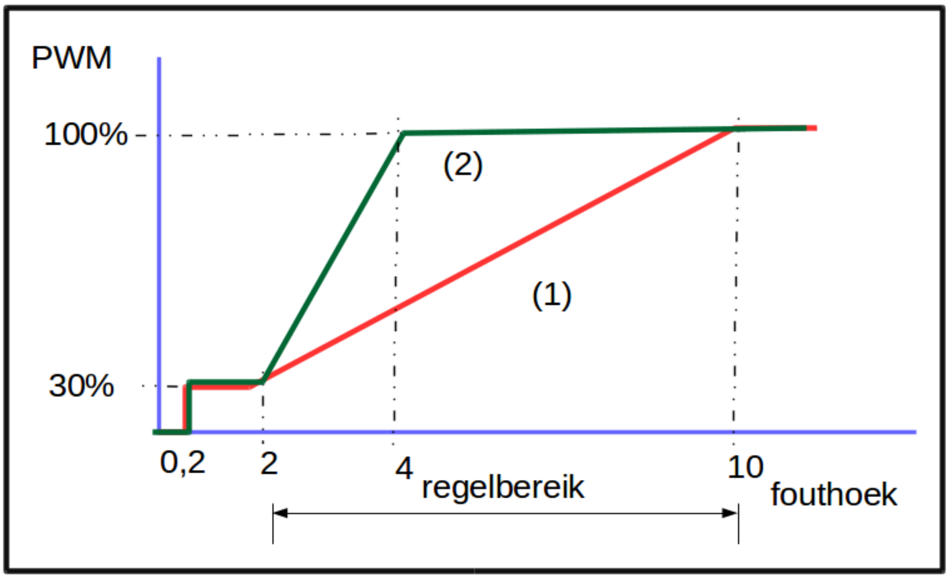


Fig. 2. Proportional control

In case of a small error angle, the motor will still not run. So a “blind spot” has been deliberately applied (here 0.2 degrees). This is to prevent that when the motor has reached the desired position, is stopped but overshoots a little bit, the motor is put “in reverse” to run back, then overshoots again, and so on . That would result in the dish flapping back and forth.

A somewhat larger error angle, just past the dead angle, gives a jump to the minimum speed, so that the friction is immediately overcome.

Then the speed gradually increases to a certain error angle where the speed is maximum. By setting this error angle to a low value (see curve 2, compared to curve 1) , the curve will run steeper, and the error angle will decrease, but the control will then become more “aggressive”. In extreme cases the motor is set immediately to the maximum value, but this will require a lot of of the whole construction. Also the overshoot upon reaching the end position, will be large.

By choosing the right steepness in combination with calculating and sending new positions in a short repeat time (eg. every 0.1 sec. instead of every sec.) one can try to let the dish move as smoothly and as accurate as possible. If that fails, other measures will have to be taken (eg proportional-integrating regulation). Another possibility is to have the dish lead by half an opening angle. The dish is then aimed a bit in front of the satellite instead of at the satellite. The satellite then runs, as it were, from one side to the other of the opening angle, leaving more time to reorient the dish. I have now built a first version of this feature in xtrack [5] (Preferences, tab Rotorconfig, Look ahead).

Actual error angle

The error angles of elevation and azimuth are irrelevant separately. What is important is the combined actual error angle. At an elevation of 90 degrees, the azimuth angle does not matter; then the dish only rotates about its vertical axis. The more the elevation deviates from 90 degrees, the greater the influence is of an azimuth error angle.
The combined actual error angle can be calculated using the so-called dot product of the two angles. A little math:
First we calculate from the elevation and azimuth the position of the satellite (from the orbit calculation) and the position to which the antenna is pointed (from the pulse generators of the rotor system):

x1 = cos (azi1) * cos (ele1)
y1 = sin (azi1) * cos (ele1)
z1 = sin (ele1)

x2 = cos (azi2) * cos (ele2)
y2 = sin (azi2) * cos (ele2)
z2 = sin (ele2)

We get a coordinate for both positions: (x1, y1, z1) and (x2, y2, z2). These are coordinates in a three-dimensional coordinate system, on a perfect sphere that represents the earth. The radius of the earth does not matter (here normalized to 1); only the angular positions is what matters.

The dot product is calculated from these positions:
phi = acos (x1 * x2 + y1 * y2 + z1 * z2)
A check: If the elevation of the satellite 'ele1' = 90 degrees then it is easy to see that the azimuth no longer matters: cos(90) = 0, so x1 and y1 are 0. The dot product is then only depending on an error in the elevation (z1 = sin(90) = 1, z2 = sin (ele2), so the error angle is then: phi = acos (sin(ele2)).

Use xtrack to determine the error angle

To study the behavior of rotors, a number of extra features have been built into xtrack. The first is the possibility to simulate a pass, without actually having to wait for a satellite pass. To do this, choose 'Predict' in the xtrack menu:

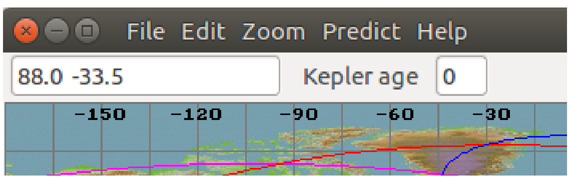


Fig. 3. Menu in XTrack.

This gives a window as in fig. 4 . On the left you can choose from which day / time and for how many days transfers must be calculated. The satellites selected in the main window are used for this.
After clicking "Predict" at the right a list of passes is shown, also the maximum elevation is shown. One of the passes can be selected.
By clicking on 'Set time' at the bottom left, the time will be set at the start of the pass.
Then, after clicking on 'Simulate', the pass is followed as if the satellite is actually passing.

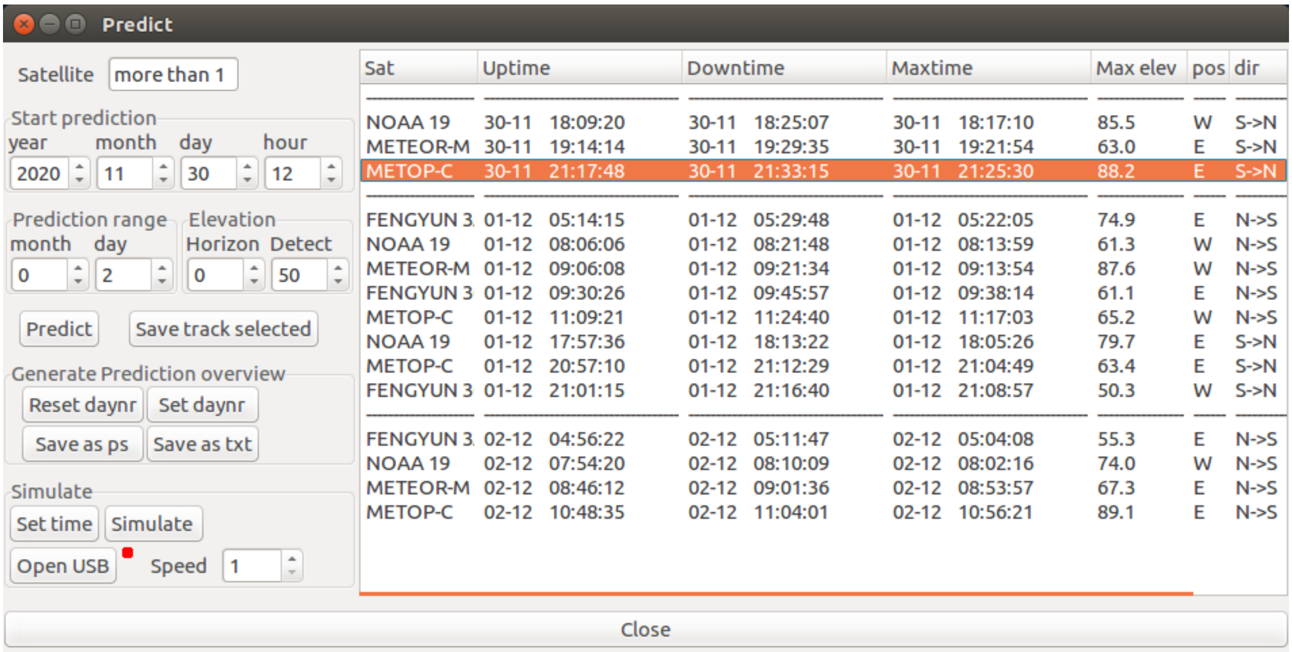


Fig. 4. Predict window.
If necessary, the time can be accelerated with 'Speed'. You can also stop the simulation by clicking on 'Simulate' again, and then resume it. In this way, a pass can be run through multiple times, so that, for example, multiple variations of the Arduino software can be compared with each other. The above possibility has been in Xtrack for some time now.
The second option is the debug window: menu Help-> Debug.

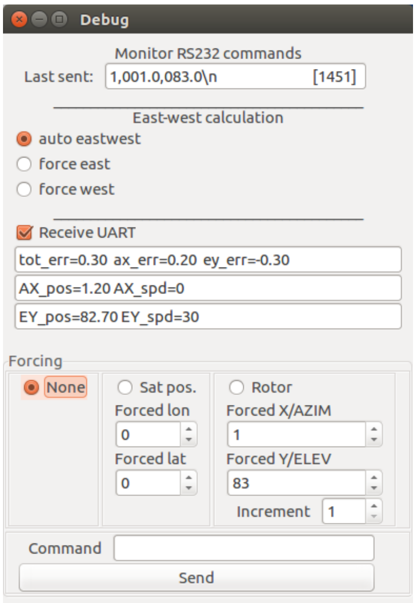


Fig. 5. Debug window.

This includes a number of things:

- Forcing the satellite position: Sat pos.
- Forcing the rotor to a certain position: ' Rotor'
- forcing east or west passage

For simulation ‘Forcing’ has to be set to 'None' (which is default).

If 'Receive UART' is checked, messages from the Arduino will be made visible:

- tot_err = the total angular error (dot product, as described)
- ax_err: error of azimuth rotor
- ey_err: error of elevation rotor
- AX_pos, AX_spd: current position (angle) and current speed azimuth rotor
- EY_pos, EY_spd: same for the elevation rotor

If xtrack is started with option '-log <filename>' then the above results will come in a log file, with lines:
AX_pos = 194.50 AX_req = 194.60 AX_spd = 0 EY_pos = 177.70 EY_req = 177.40 EY_spd = -30
AX_err = -0.10 EY_err = 0.30 tot_err = 0.32

Results

In fig. 6, the results of an pass with maximum elevation of 85 degrees is displayed. A bit zoomed in; the graph starts approx. 5 minutes after start of the pass.

- The top graph shows the azimuth and elevation of the satellite.
- In the middle the error angle is shown: azimuth, elevation and resulting error angle.
- At the bottom the rotation velocity in %

What is striking in the middle figure is the big error of the azimuth around maximum elevation: the big red bump. The error here is 23.8 degrees. There are hardly any errors in the elevation; that rotor always runs very slowly.

The small green bump indicates the resulting error angle, which is significantly lower than the azimuth error: 2.8 degrees . Only this resulting error angle is important.

What is also striking is the sawtooth in the azimuth line. This is the result of setting the minimum speed too low; this rotor has some friction problems.

In the lower graph, which shows the speed of both rotors, this can be seen in the peaks in the (red) azimuth-line; the engine must then be turned to a slightly higher speed to overcome the friction. This is obviously not a desired effect. I deliberately left it in to show the effect of too much friction or setting the minimum speed too low.

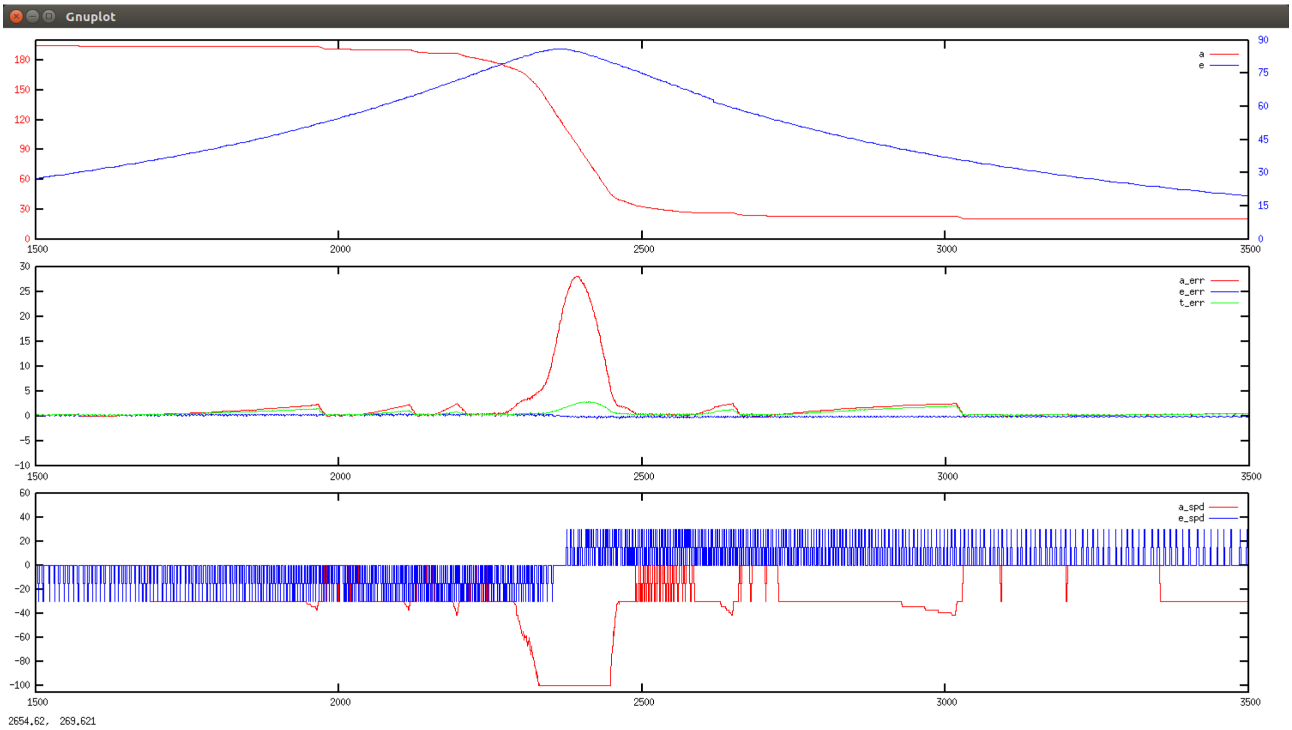


Fig. 6. Results pass with max. elevation 86˚. Top to bottom: position, error and speed.

The speed graph doesn't looks very nice, jumping between zero and minimum speed. This is a result of the fact that the motors can't run slow enough, which results in repeated turning on-off of the motors. In the blue elevation graph it can be seen that the speed first shows pulses negative (elevation rotor going from 0 to 90 degrees), and then shows positive pulses (rotor going from 90˚ back to 0˚).
Furthermore, it is clearly visible in the azimuth graph that this rotor must run at maximum speed for a while and therefore cannot follow the satellite. As said, this is not as bad as it looks like; the resulting error angle is all that matters.
More to the right, it can be seen that the azimuth motor is getting power but is not running, until the azimuth starts to lag too far behind, giving the motor more power. The tension rises slightly, then the friction is overcome and the motor continues to run at low speed to the correct position.

Fig. 7 shows the result with a passage with maximum elevation of 50˚. Here we see that the azimuth motor also keeps up well, apart from that sawtooth due to too high friction. You don't see that sawtooth halfway through because the motor runs smoother in that area; also, the motor runs a little longer without stopping in between.

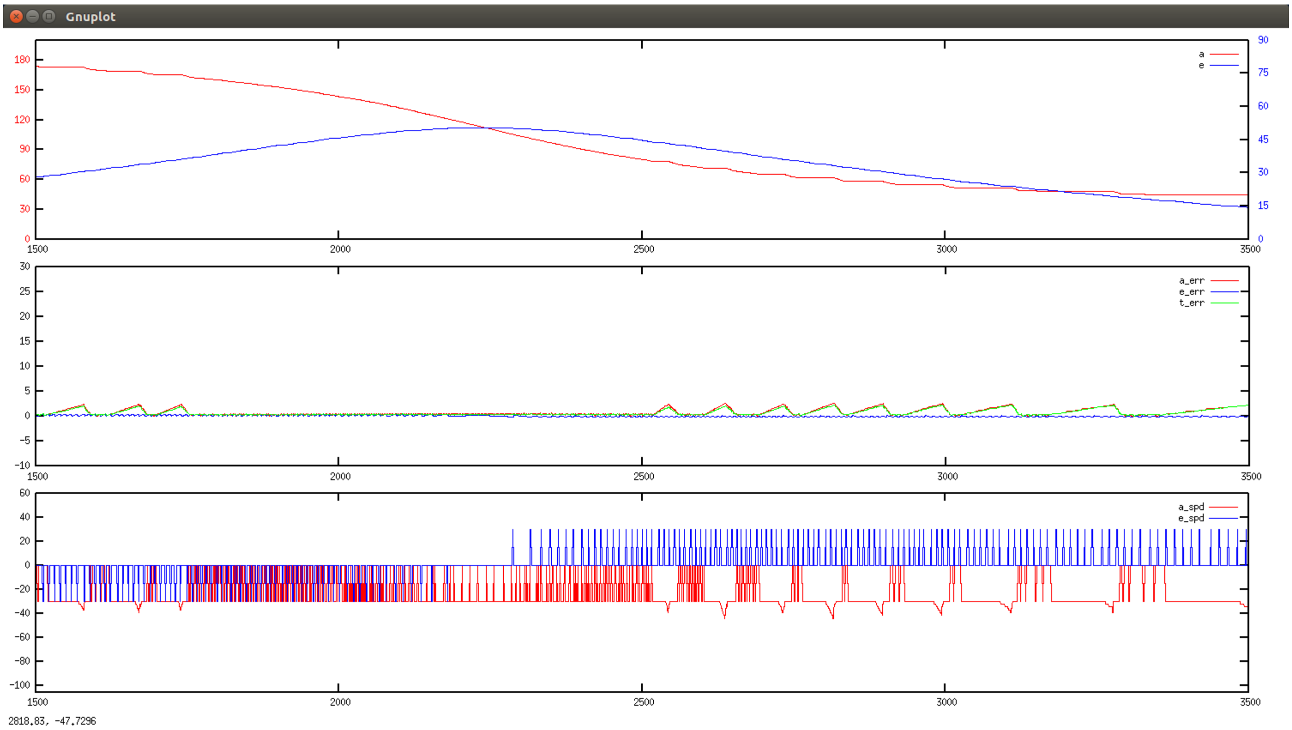
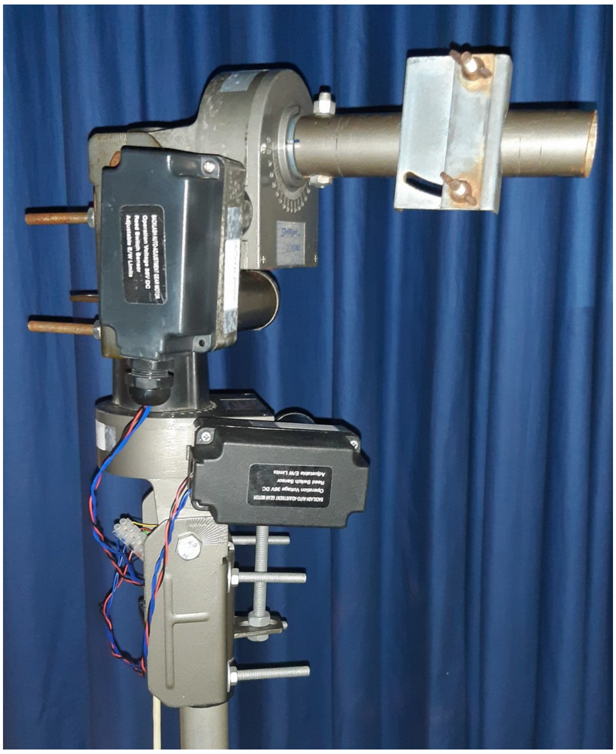


Fig. 7. Results pass with max. elevation 50˚; from top to bottom position, error and speed.

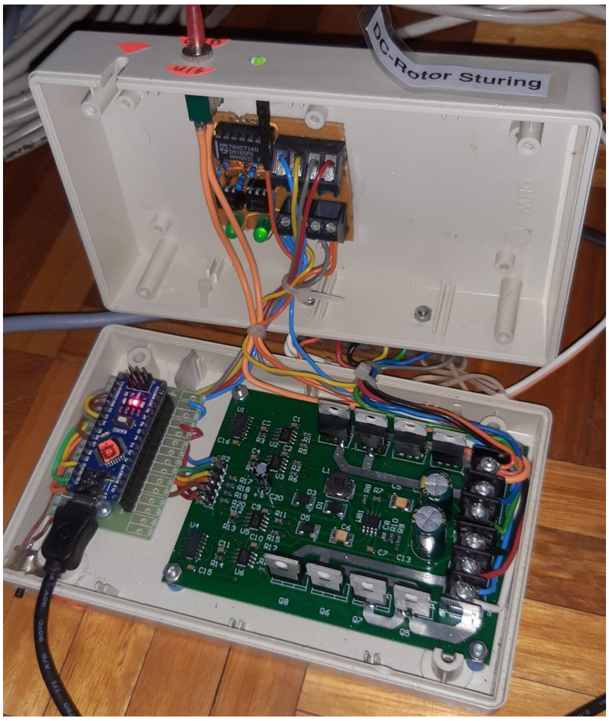
Conclusion

With the additional capabilities of xtrack combined with the feedback from the Arduino software the behaviour of the rotor system can be properly viewed. The maximum resulting error is still too big with the setup used here: 2.8 degrees. This cannot be improved with “looking ahead”; the rotor is simply too slow for high passes. The Jaeger rotors might be suitable for passes that are not too high. However, this would have to be tested in a more realistic configuration (with dish and construction as would be used for 'real' tracking); the variable loads, especially at different elevations, have not yet been included here.

Incidentally, this article was not about testing the Jaeger rotors for suitability (there are also different types), but more about developing a method with which measurements of the aiming accuracy can be made and different configurations (software and hardware) can be tested, without actually receiving a satellite.



The two Jaeger rotors mounted as elevation / azimuth



Controller: Arduino with double H-bridge

Reference

[1] Discussion about the rotor control, the Kunstmaan no. 2, p. 17.
[2] Parabolic antennas: Beam width
[3] Beamwidth calculator for satellite dishes
[4] Optimal tracking of polar satellites. The KM, June 2013, p. 70
[5] xtrack tracking program

Constellation Viewer: a few adaptions

A constellation viewer is described in “de Kunstmaan” 2019, no. 1 [1], which can be used in combination with the QPSK receiver to visualize the constellation of the received QPSK or BPSK signal. This viewer has been realized with an STM32 board (“Blue Pill”) and an OLED display.
Rob Hollander recently came up with a number of questions about extras to be added. That prompted me to take another look at the code. I discovered that the brightness can be increased by a factor of 2; mistake in the code ... This is easy to adjust yourself in constel_stm32.h:

```
#define LUM_MAX 31
```

must be:

```
#define LUM_MAX 63
```

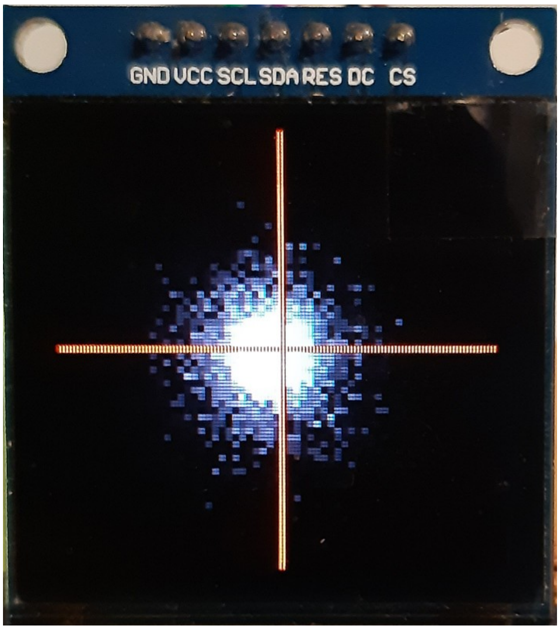
The code on github has now been changed, see [2]
I also included a number of other wishes from Rob:

- Start-up screen, displays text for approx. 5 sec., Including version number.
- Rotate what is shown on the screen so that the screen itself can be mounted upside down. This can be useful if this way the connections come out better. This is not so much necessary for the constellation diagram itself, but if text is displayed. The possibility was actually already there, but it was somewhat hidden. In str2oled.ino it now says at the top:
  `#define UPSIDE_DOWN false`
  Changing to 'true' turns the text on the screen upside down.
- Add axis cross. This can now be done by applying PB15 to GND. With the pins PA11, PA12 and PA15 the color of the coordinate can be adjusted:

PA11	**PA12**	**PA15**	colour
Open	gnd	gnd	red
gnd	Open	gnd	green
gnd	gnd	Open	blue

  And all other combinations. The brightness can be adjusted, if desired, in function add_overlay, in xy2oled.ino

Furthermore, the color and brightness of texts can now be set (in the code). As an example, see the call to str2oled () in constel_stm32.ino.



References

See http://www.kunstmanen.net, web links.
[1] Constellation viewer for QPSK receiver. de Kunstmaan 2019, no. 1
[2] Code for constellation viewer; see github, working group artisans

Meteosat without antenna: continued

In the previous Kunstmaan [1], Ger Smit described how he retrieves and processes weather satellite images from the Internet. Until now I had not looked at that, but at the moment I can not always receive the data via Eumetcast when I want to. Enough reason to take a look at the way of the Internet.

Until now you could download LRIT data via ftp; that's 5 channels, compressed as Jpeg. Unfortunately, that compression is done quite strongly, so that when you zoom in, pictures don't look very nice (to say the least). This is also because there are just 8 bits per pixel instead of the original 10 bits.

This ftp service will be discontinued next year. But there is something much better in its place: the full data from MSG HRIT, so all 12 channels, uncompressed, both “full globe” and RSS. Furthermore, data from METOP and Sentinel-3, among others, are also available in this way.

The only drawback is that the files to be downloaded are quite large: for MSG4 that is around 170 Mbytes. Compared to what is sent by satellite:

- 112 files: 11 channels with 8 segments + 1 channel with 24 segments
- prologue and epilogue files

This is all retrieved from the internet in a single zipped file . The file is in 'MSG Native' format; after unzipping, all channels are present in it uncompressed. Each 4 pixels of 10 bits per pixel are coded in 5 bytes.

Eumetsat itself has developed a number of tools to visualize the data:

- Data Tailor and
- Eumetview.

Undoubtedly nice programs with which you can do a lot, but I still want to go more towards Raspberry Pi. Those PCs consume very little and can therefore be left on continuously without too much electricity costs.

In the meantime I have adapted xrit2pic so that these files can also be processed; for the command version there is also one that works on RPi.

What then is needed is a script to extract the data from the Internet. A bit like Tellicast, with which you pull the data “out of the air” via a receiver.

Eumetsat provides a number of instructions for this in the form of python scripts. I have further elaborated this so that new data can be retrieved automatically. It then works in the same way as when receiving the data via Eumetcast / Eurobird, whereby an internet connection is required instead of receiver / dish / Tellicast software.

Uh… “werkgroup Kunstmanen (working group satellites) ???

Well, what's called… the “real” job is of course receiving the weather satellites themselves. With Eumetcast it goes via a TV-satellite channel, which is actually not that “real” anymore, and via the Internet the entire receiver is lost. But of course the data still comes from weather satellites.

What is needed:

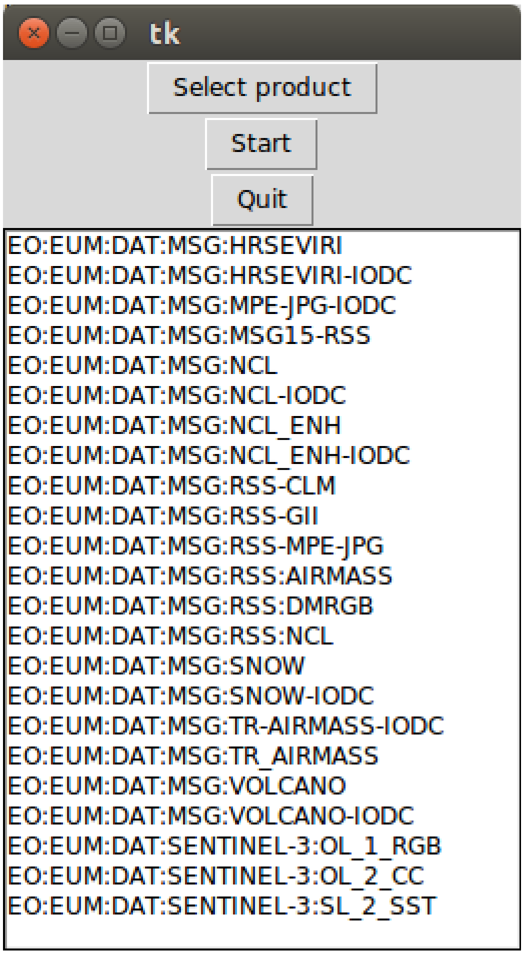
I've made some Python scripts; see [2]; one of them can already be “played” immediately:

- eci.py

If this script is started with option -g:

python eci.py -g

then an overview of data appears:



The list is taken from the Eumetsat site. This is a part of the available data. At the top we see:

- EO: EUM: DAT: MSG: HRSEVIRI, this is MSG full globe (15 min interval)
- EO: EUM: DAT: MSG: MSG15-RSS: this is the Rapid Scan service (5 min interval)

A product can be selected from this list; Clicking on 'Select product' will put the name of the product in a file: ***eci.prd*** . This is then used by the actual download script to retrieve the correct data. The advantage is that once the desired product has been selected, the gui is no longer necessary; the download script can then work in the background.

Currently, this script can be used to select one product at a time, but it should be possible to extend it to multiple products.

To actually download data some preliminary matters have to be to settled, because an authorization request has to be asked to Eumetsat to have access to this data. It goes like this:

- Request access to EOPortal. Some forms have to be filled out for this. People who have already worked with Eumetcast will already have this.
- Upon approval you will receive a “username / password” with which you can log in to the EOPortal. By selecting “Datastore” you can make the so-called “consumer key” and “consumer secret” visible via a menu. This procedure is described in detail on my website ([2])
- copy these 2 lines into a file “eci.key” so that it says something like:

```
abcdefghijklmnopqrstuvwxyz78
qqqqqrrrrrssssstttttuuuuu678
```

So there are 2 lines with 28 characters each.

Finally, the download script has to be adjusted: eci_dwnld.py

- path_download: the location where the retrieved and zipped data will be stored
- path_unzipped: the location where the extracted data will be stored. This is also the location that you should refer to with xrit2pic as “source directory”.

Note: these 2 locations must be different! They can also be placed anywhere else. With an RPi it may be wise to connect an external disc and store the data there; to prevent wearing out of the SD card by the many writes. Incidentally, I think that will be better than expected.

In this script you can also enter the consumer_key and consumer_secret, instead of putting them in a separate file 'eci.key'. The advantage of keeping it separate is that you can easily share your script with others without revealing your keys.

Starting is now done via: (at least for Linux you can ommit ‘python’)

- python eci.py -g (via gui, click “Start”), or:
- python eci_nogui. py

In both cases an infinite loop in procedure 'run_download ()' (eci_dwnld.py) is completed. To do this, a choice is first made from the collection, based on the chosen product (eg EO: EUM: DAT: MSG: HRSEVIRI): selected_collection_id . Then, in the loop:

- The current time (UTC) is determined: ***tm***
- A time span is defined: from 0:00 to 23:59 on the current day: ***start_date*** , ***end_date***

- A list of the last 10 available files is retrieved: ***found_data_sets*** . Because starting "current time" and end 23:59 these are always the 10 newest files.
- A token is requested, using the keys you entered in eci.key .: ***access_token***
- The first 2 files (the 2 newest) will be retrieved, but first it will be checked if the file was already retrieved: ***os.path.isfile ()***
- Download is done using wget
- the file will be unzipped: ***zipObj.extractall ()***
- waits for 4 minutes for the loop to run again.

Remarks:

- The repeat time is determined with ***minute_waits*** . It should be slightly lower than the expected time between 2 new files. So, for example RSS, a new file every 5 minutes: set minute_waits to a maximum of 4. A lower rehearsal time is also allowed, but causes unnecessary internet load.
- They will be always a check for the last 2 files to see whether they are retrieved. Sometimes files are not immediately available; for example with RSS, if there are no new files available on the server for more than 5 minutes, then 2 files will become available at the same time. This number is determined by ***nr_download*** , and can be adjusted if desired.
- The check is done to prevent files from being downloaded unnecessarily. Old zipped files may therefore only be removed after “nr_download” newer files have been retrieved.
- For the same reason unzipped files should be placed in a different location from the zipped ones; xrit2pic may move files to another location (the 'done'-location).
- The script deliberately did not use Python 's download option. The problem is that the entire file (ie 180MByte) will then be stored in memory before unzipping it. That causes problems on computers that are not too big. With the separate program ***wget*** it all works much better.

The scripts are available on my website [3]. It is still experimental, so there may still be some changes. The RPi I am using is a very old one, from 2011. It works fine, so there is no need to buy the latest model.

A possible setup

For example:

This program runs continuously on a RPi without head (so no monitor, keyboard, etc., only an Ethernet connection); xrit2pic makes a movie of the last 24 hours; old data is automatically discarded. This can therefore continue to run “forever”, without intervention.

At set times you can transfer the movie as it is via ethernet to a PC and watch it.

I have already extensively tested the continuous generation / update of a film with xrit2pic in gui-mode on a laptop, with data from Eumetcast (so via receiver); that works fine. This still has to be tested with xrit2pic on a RPi, with data from the internet and xrit2pic in command mode.

Other possibilities

Xrit2pic can be used to process received or retrieved MSG data from the Internet, but there is also another interesting possibility: pytroll / satpy [4]. This is a set of Python scripts that can be used to edit satellite data from MSG, METOP, etc. I haven't played with it yet, but a lot is being done here and there. Well worth a look.

Last remarks

Scripts have been adapted; no endless loop, but instead crontab can be used.

The very old RPi I have can manage a single channel to process (e.g. making a movie), but it is not suitable for composites (this RPi has just 180 Mbyte RAM...) For composites use a RPi with 2Gbyte RAM or more.

The python scripts can be used with any processing tool, not necessarily xrit2pic.

See for newest info my website ([3])

References

[1] de Kunstmaan no. 3, 2020, p. 13

[2] Procedure for requesting 'keys'

[3] EumetCastinternet scripts

[4] Pytroll / Satpy

Dual 5V / 500mA power supply

Warning: this circuit is connected to the mains light and carries life-threatening voltages. Build the power supply in a plastic housing and mount the PCB with internally threaded nylon spacers. Also do this prior to testing. Use proper power cords and an approved power supply.

Summary

A double 5V power supply is presented. This might be used for separately feeding a receiver and a decoder.

Preface

During the design and construction of the QPSK receiver it appeared that the power supply is very important for the proper functioning of the receiver. Therefore, on the nutritional input of the receiver include a coil of 100uH to interference that 5V power supply through the inner cup t to be suppressed. Many power supplies are now connected, they provide more fault d an "old fashioned" power based on eg. The 7805.

Switching power supplies

Below are a few measurements of switching power supplies. When first I Ikea Koppla tested. It has three outputs, with a maximum load of 2400mA per port (3400mA for all ports together). I also tested a diet from Phihong .
The load was a 15 Ohm resistor, resulting in a current of 330mA. When connecting a coil in series, the result is much better.

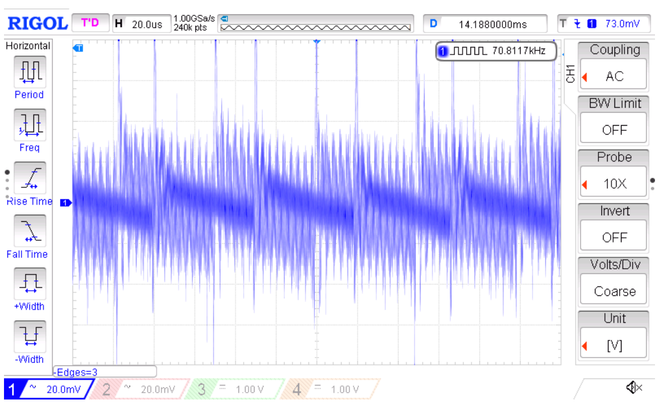


Fig 1 Ikea Koppla 330 mA. The time base is 20us. We can see that the peaks having an amplitude of 8 0 m V . The outliers go to 180mV. The scale is 20mV / div.

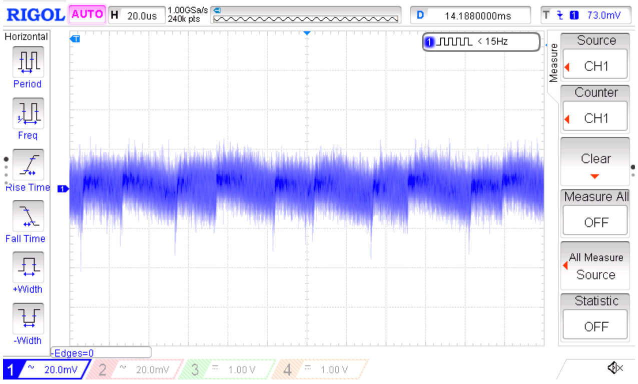


Fig 2 Now with a 100uH coil in series. D e supply is a lot cleaner . Scale distribution and time base as in figure 1.

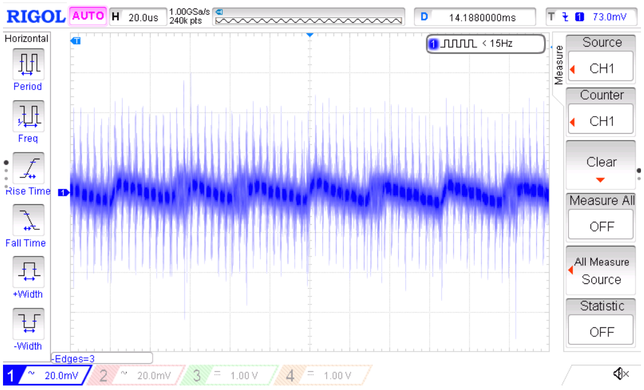


Fig 3 Phihong without coil. Scale and time base as in figure 1.

There are many comparisons between USB power supplies on the internet , some better than others. Another point is the safety of the USB power supplies . Some Chinese foods are of poor quality .

Linear power supply

The disturbances that you see with the switched power supplies you do not have with a linear power supply. A linear power supply has a:
- transformer
- rectifier
- flattening capacitors
- voltage regulator

If the reception will improve through a linear power supply use is questionable . As shown in figure 2, the 100uH coil removes a lot of junk, such a coil is also in the receiver.
This power supply would be useful for powering a digital part of a circuit (microcontroller , etc.) where a lot of interference from coming and analog part.

The power supply

Transformer
I chose a toroidal transformer for the transformer . The advantage of a toroidal transformer compared to a block transformer (with lamellae) is a much smaller stray field and a higher efficiency.
The disadvantage of a toroidal transformer is that it is slightly more expensive than a block transformer. A toroidal transformer cannot be short-circuited due to its higher efficiency .
The transformer has two secondary windings with which I can make two power supplies that are galvanically separated from each other. One output can go to the receiver and the other to the decoder.
I have chosen a model that can be attached to a print, available from Reichelt .

Rectifier
Nothing exciting: a double-sided rectifier consisting of 1N400 1 diodes . Given the low voltage, all diodes from the 1N400x series are suitable.

Flatten elco
I chose 4700uF. I will experiment whether a lower value is also possible. The diodes are then less loaded when the power is switched on. At that moment a short short circuit current comes.

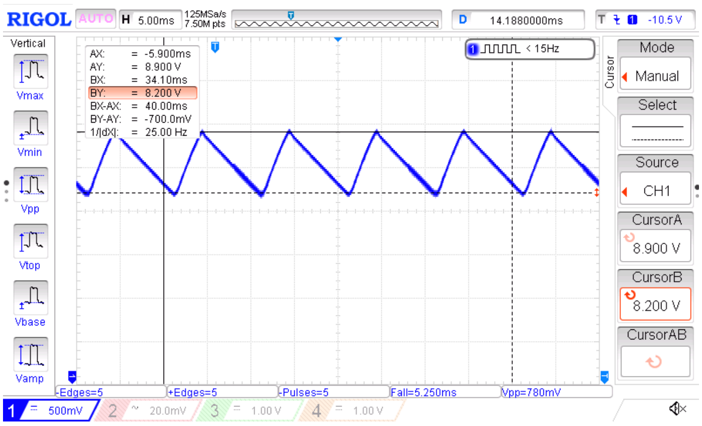


Fig 4, the voltage across a smoothing capacitors at 5 00mA load . With a minimum of 8, 2 V remains over above the dropout voltage .

Voltage regulator

This is an LM317 with which the output voltage can be in ge set d . The advantage is that the dropout voltage (minimum difference between input and output) is lower than with the 7805. Furthermore, the noise is low. They are equipped with a heat sink because they still have to lose some heat.

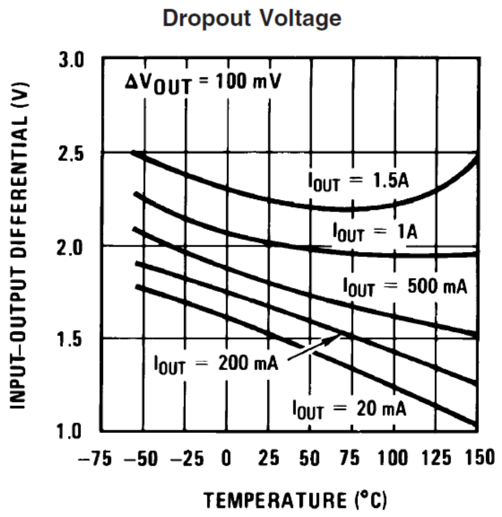


Fig 5 Dropout voltage of the LM317

Scheme

The fuse is of the type 200mA fast. The transformer is the type RKPT 2507 (the 7 V version) . A transformer with a higher voltage is not possible because the heat sinks cannot dissipate the heat . In practice you cannot go above 600 mA.

Construction

The print has a size of 10 x 10 cm and can be produced very affordably in China. The construction of this power is fairly straight forward . Pay attention to the correct orientation of the diodes . The transformer is mounted to the PCB with a bolt. The resistors are in SMD 0805 version. This is very easy if you want to solder resistors in parallel.

When installing in the housing, pay attention to earthing and ventilation. There is a lot of heat coming from the heat sinks that has to go away.

Furthermore, the power cord must have strain relief or opt for a proper power supply.

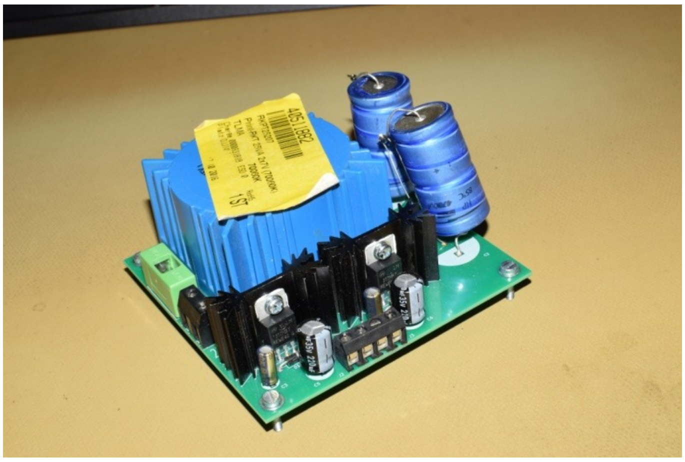
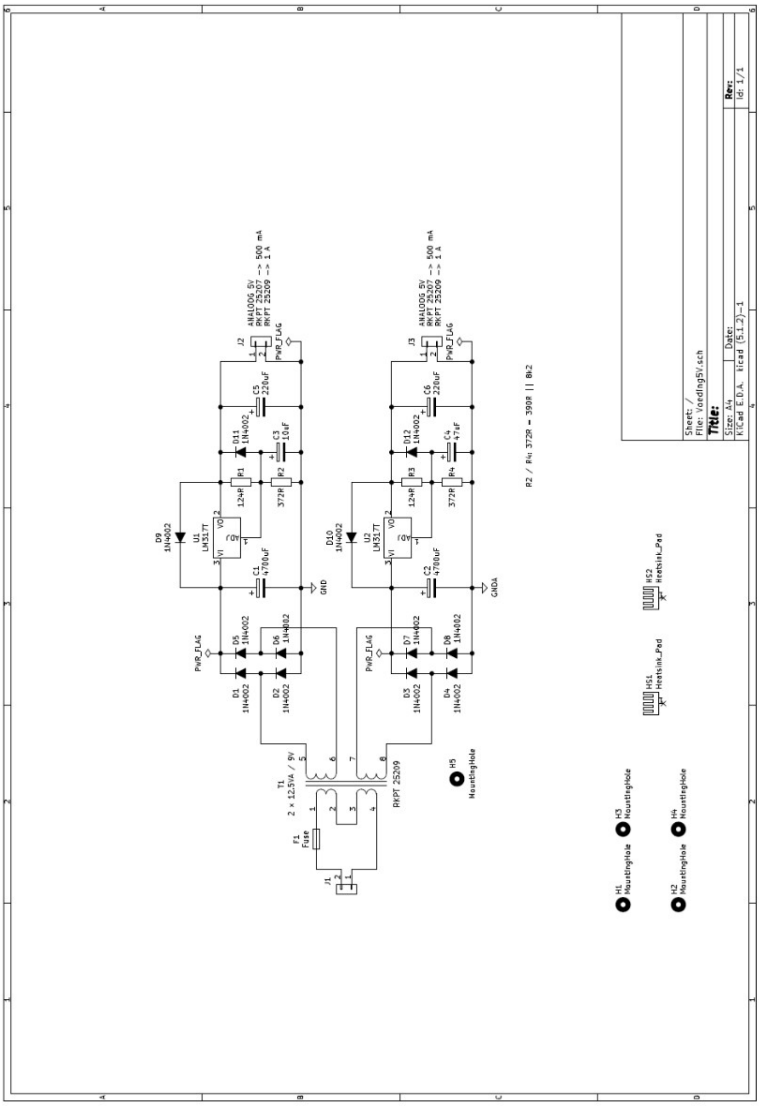


Fig 6 The built-up nutrition. The capacitors are still of the axial type.



UKW-BERICHTE

Paul Baak

Summary
In this article a concise review of articles published in the 3th edition of 2020 of the German magazine UKW-Berichte. We have a subscription to this magazine.

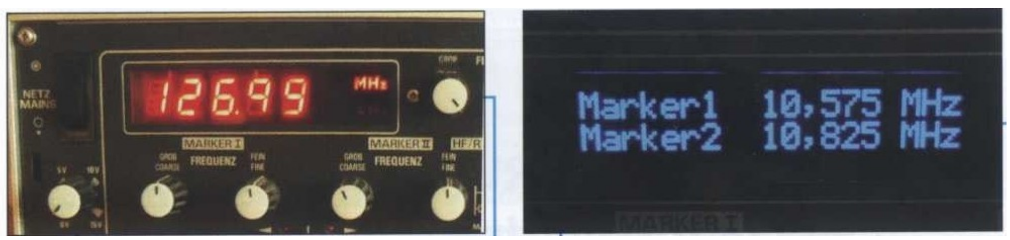
Here is an overview of UKW report 2020 Heft 3. There is still a small backlog in publications. We find 6 articles, plus the regular overview with internet links. Our club has a subscription to this magazine. Please indicate whether you appreciate this subscription! Positive or negative, it doesn't matter, as long as your board hears something. The latest editions are available for reading at meetings on the library table; as this is not viable now, you may contact the librarian directly.

Wolfgang Schneider describes an electronic fuse for a maximum current of 40 A. The intended application is to fuse LDMOSFETS that replace tubes in transmitter stages. The circuit is made around Infineon's BTS50085, a specially designed IC for this purpose.

Olaf Schilperoort, product manager at Eupen Kabelwerke, presents cable radiation as a solution to some 5G requirements. Slots are inserted in the outer jacket to ensure uniform radiation along the length of the cable. This enables uniform reception in difficult areas such as elevator shafts or tunnels. Radiant cables have been used for a long time; now also for 5G.

Gunthard Kraus writes in several pieces the third part of his series about the NanoVNA. The emphasis this time is on measuring long cables; then on the new nanoVNA-V2 which runs up to 3 GHz and on another new version, the SAA-2N version with 4 inch screen, housing, N-connectors and accessories for $88.

Dieter Barth describes the renovation of the Grundig wobbler/measuring generator AS5F [1][2] so two markers can be used. This device itself dates from about 1980 and runs up to 120 MHz. The approach consists of adding an LS7060 counter, an ATMEGA microcontroller and an LCD-display. The code can be requested from the author.



(the display before and after the conversion of the AS5F)

Under the heading Ultrakurz we find some new products (preamplifiers and antenna) that are outside our area of interest (i.e. frequencies of interest).

In Fundstelle Internet we find internet links. I mention here: the versions of simulation program MicroCap 10, 11 and 12 have been released for free; the latest QO 100 downconvertor, a site with antenna experiments; a site with HF amplifiers; the reception of weak signals.

UKW-Berichte [3] is a German language edition, now without an English version that previously existed under the name VHF communications. Including shipping from Germany, the magazine costs 34 Euro per year as of the year 2020. Individual copies and volumes (DVD) are also available.

links:

[1] Grundig wobbler AS5F, beschrijving
[2] Grundig wobbler AS5F, manual
[3] Berichte/ABO-UKW-Berichte

Member meeting report November 14 , 2020.

Opening by the chairman.

It's another zoom session, unfortunately. Not everyone likes this way; it is to be hoped that we will soon be able to have a 'real' meeting again.

Setting the agenda

No comments / remarks

Board affairs: PDF membership

This has already been discussed last time. This will take effect from 2021. A PDF subscription costs 10 euros per year; for a 'paper' subscription, the subscription will be increased to 28 euros because of the strongly increased shipping costs. For abroad this will be 33 euros.

We still think after a third option: a yearbook containing all the "timeless" articles from the past year (not eg satellite status and foreword.). With both “paper” subscriptions you also get the PDF version every quarter. This will be notified by email.

Satellite status (Arne)

There is not changed much; see elsewhere in this Kunstmaan.

There was a “Solar outage” in October (sun in a straight line with geostationary satellite and receiving antenna); check your LNB, whether it is still intact. A wet dish reflects more sunlight, so also heat to the LNB, which can cause a crack in the plastic housing.

Fred Jansen reports that GOES13 is now at 61.5 degrees east ; it has been renamed EWS-G1. This is now “visible” from the Netherlands, just above the horizon, so in principle it can be received. This satellite transmits in the L-band, 1691 MHz. According to Wikipedia, this is a replacement of MSG1 that is in roughly the same position and is almost out of fuel.

Fred also reports that the Chinese FY3B is not only active in the L-band, but also on the X-band (7778 MHz), with a very strong signal (stronger than FY3D which only broadcasts in the X-band).

Any other business

Fred Jansen: Reports that he can receive FY3B, and Aqua (X-band); with FY3D it has not yet been possible to decode it.

Job: is busy making printouts for an LNA. He uses UV through a LCD.

He also got a network analyzer running , which runs up to 20 GHz .

Rob Alblas has expanded xrit2pic to also process FY3D, MTG (not yet launched) and GOES16 / 17. This concerns data that is distributed via Eumetcast.

Further he is busy in retrieving / processing of satellite data over the internet (see the article of Ger Smit in the previous KM). At the moment this is still possible via ftp, which is MSG LRIT data, but that will disappear. In its place, one can now download MSG HRIT (also RSS) via an http link. That gives many more options and much better quality, but files to download are very large (all channels in a single file). The data format is “MSG Native”; this is now also supported by xrit2pic.

One of the possibilities is to download the data with a continuously running Raspberry Pi and, for example, make a movie out of it, which can then be viewed at set times via a PC.

Peter Kuiper: Has problems with the network connection of a PLUTO PC; this stops weekly. Problem may be that the network address is not confirmed often enough. Ben has had problems like that too; resetting the modem every night with a timer helps.

Peter Smit: has found a new motor control. It consists of a double H-bridge, so suitable for 2 motors. A few things have already been tested with this controller it works well.

Wim Bravenboer has a question about the availability of brass to make a horn for an LNA (0.5 mm thick). Mentioned is https://dhzstore.nl, and https://ijzershop.nl.

He also has an idea to use a small motor to set the distance LNA to dish . Nobody has experience with this yet.

Robert Langenhuysen announces a new Raspberry Pi Model P400. This has also been written about in Elektor. It is striking that the processor remains pretty cool.

Peter Smit : asks if with the last "Kunstmaan" a members list is sent. That is the case, but it contains only the email addresses, not full address.

Fred Jansen reports on tracking 8 GHz satellites that the PC clock determined via the Internet is not always accurate enough. This is because the clock is only adjusted once per hour. The deviation may then be too much for tracking 8 GHz satellites .

About the calibration of a satellite tracking system: that can be done on the limit switches, but Fred uses a satellite Syracuse 3B, at 7710,5 Mhz. Which is very strong and the orbit data (it is a geostationary satellite but it also moves a bit) are known. (This satellite is at about 5 degrees west.) By adapting elevation and azimuth several times to aim the dish at the maximum signal, the matter can be calibrated in this way.

Closure

Rob Alblas then gives an overview of the rotor system he is working on. It is based on an Arduino. The software for this is made generic enough that it can be used for both elevation / azimuth and X / Y systems.

Finally, Ben shows the schematic and print of a rotor controller he designed. Characteristics are: motor driver, and the rest (including processor) are separated by means of opto-couplers. For motor control, the Chinese board of Peter Smit (dual H-bridge) can be used, or a board based on the L289N.

Rob Alblas

(secretary AI)

satellietstatus

Arne van Belle, per 13 december 2020

POLAIR	APT (MHz)	HRPT (MHz)	Overkomst
NOAA 15	137.620	1702.5	ochtend/avond, zwak en soms sync problemen
NOAA 18	137.9125	1707.0	vroege ochtend/namiddag
NOAA 19	137.100	1698.0	middag/nacht
FengYun 3A	geen	1704.5	AHRPT 2.80 Msym/s
FengYun 3B	geen	1704.5 X-band	AHRPT 2.80 Msym/s
FengYun 3C	geen	1701.3	AHRPT 2.60 Msym/s
FengYun 3D	geen	7820.0 X-band	middag MPT 30 Msym/s
Metop-A	uit(137.100 LRPT)	1701.3	LRPT/AHRPT 2.33 Msym/s
Metop-B	geen	1701.3	Alleen AHRPT 2.33 Msym/s
Metop-C	geen	1701.3	Alleen AHRPT 2.33 Msym/s
METEOR M N2	137.100 LRPT	1700.0	LRPT/MHRPT
METEOR M N2-2	uit(137.100 LRPT)	1700.0	LRPT/MHRPT schade
NPP	geen	7812.0 X-band	HRD 15 Mbps
JPSS-1/NOAA 20	geen	7812.0 X-band	HRD 15 Msym/s

GEOSTATIONAIR	APT (MHz)	(SDUS)/PDUS (MHz)	Baanpositie
MET-11 (MSG-4)	geen LRIT	1695.15 HRIT	0 graden, operationeel
MET-10	geen LRIT	1695.15 HRIT	9.5 graden O, RSS
MET-9	geen LRIT	1695.15 HRIT	3.5 graden O, standby
MET-8	geen LRIT	1695.15 HRIT	41.5° graden O, IODC
GOES-E (no. 16)	1686.6 GRB	1694.1 HRIT	75.2 graden W via Eumetcast
GOES-W (no. 17)	1686.6 GRB	1694.1 HRIT	137.2 graden W via Eumetcast
GOES 14	1691 LRIT	1685,7 GVAR	105 graden W, Backup
GOES 13 / EWS-G1	1691 LRIT	1685,7 GVAR	61.5 graden W, Nu Space Force
GOES 15	1691 LRIT	1685,7 GVAR	128 graden W parallel met GOES 17
Elektro-L2	1691 LRIT	1693 HRIT	14.5 Graden W, via Eumetcast
Elektro-L3	LRIT	HRIT	76 Graden O, Operationeel
MTSAT-1R	1691 LRIT	1687.1 HRIT	140 graden O, Backup voor MTSAT2
MTSAT-2	1691 LRIT	1687.1 HRIT	145 graden O, via Eumetcast
Himawari-8	geen LRIT	geen HRIT	140.7 graden O, via HimawariCast
Himawari-9	geen LRIT	geen HRIT	140.7 graden O, Backup voor 8
Feng Yun 2E	-	-	86.5 graden O, Backup
Feng Yun 2F	-	-	112.5 graden O, Backup
Feng Yun 2G	-	-	99.5 graden O
Feng Yun 2H	-	-	79 graden O
Feng Yun 4A	1697 LRIT	1681HRIT	99.5 graden O, Operationeel

GOES-13 is nu overgedragen aan Space Force en hernoemd naar EWS-G1. Hij is nog wel te ontvangen en te decoderen.

werkgroep
kunstmanen

De werkgroep is opgericht in 1973 en stelt zich tot doel:
Het bevorderen van het waarnemen van kunstmanen
m.b.v. visuele, radiofrequente en andere middelen

www.kunstmanen.net