DE KUNSTMAAN

December 2022 – 49th annual nr. 4

Published by the Werkgroep Kunstmanen

This issue includes.
- Comparison of power meters
- Reception of Artemis
- Planetary gears
- and many more

colofon

President
Ben Schellekens
06 54987640
ben@towerhouse.nl

Treasurer, Membership Administration and Secretary (a.i.)
Rob Alblas
(035) 685 11 52
kunstmanen@alblas.demon.nl

Radio Observer coordinator
Arne van Belle
(078) 673 71 65
a.van.belle@hccnet.nl

Editor-in-chief
Job de Haas
kunstmanen@ph4as.nl

Library
Vacancy

Website
http://www.kunstmanen.net

Vacancy
webmaster@kunstmanen.net

Membership
Membership costs €28.00 per year and for abroad €33.00; PDF only: €10.00, effective from the first of the month of registration, thereafter per calendar year.

Use for payment:
IBAN:NL65INGB0004803163
Attn: Werkgroep Kunstmanen Hilversum
indicating the year of membership.
BIC: INGBNL2A

The next meeting will be on 14 January 2023

Photos
Front page:
Launch of the Meteosat Third Generation (MTG-1) satellite. Source: ESA

Inside pages:
All photos and images accompanying the various articles are by the respective authors unless otherwise stated.

DE KUNSTMAAN

Association organ of the Werkgroep Kunstmanen.

The Werkgroep Kunstmanen aims to promote the observation of artificial moons using Visual, Radio Frequency or other means.

This magazine is published 4 times a year and contains publications in the field of visual and radiographic observations of artificial Earth satellites.

TABLE OF CONTENTS

December 2022 – 49th annual nr. 4

All internet references accompanying the articles can be found on our website under:
<Weblinks | Links from KM>

van de voorzitter

Ben Schellekens

November meeting
The Nimeto is still in the middle of renovations. As expected, they are running out of schedule due to lack of materials and people. As a result, the large car park is not accessible. There are parking spaces next to the canteen.

The idea that we have a "fixed" classroom is nice, but the practice is different. The Nimeto has become something of a land grab of different groups meeting on Saturdays. At the next meeting, I will be there early to claim a classroom. Now at the previous meeting, we had nothing to complain about. The room was spacious and a nice screen for us to use.

As part of the cutbacks at Nimeto, coffee has become free. In return, there are no coffee cups. All students got a mug at the beginning of the school year and they have to make do with that. the Nimeto saves a lot of waste that they have to pay for.

At this meeting, no Zoom. The meeting was pleasant because you are not busy with the online stuff. Rob gave a talk on Satpy. This is an open source initiative to handle satellite data.

Vacancy
The position of webmaster/librarian is still open. The association runs on a very small group of volunteers and we would like to see reinforcement here. Those interested should contact me.

De Kunstmaan
This Kunstmaan is again packed with articles worth reading.

Rob has extended his QPSK generator to include the Fengyun-3D. Because of its high signal level, this satellite is interesting to design a receiver in hardware for this. Rob also writes about Satpy, which he gave a talk on.

Jaap managed to receive signals from the Artemis 1. The Artemis program is an international space program launched by NASA to land astronauts on the Moon again by 2024. The Artemis 2 is scheduled for 2024.

I myself wrote a story about power meters and an exercise in Matlab to simulate (O)QPSK.

Harry has written an article on planetary delays.

At the last minute of this Kunstmaan, the Meteosat Third Generation has been launched. Much more about it in this Kunstmaan here.

Contribution
From this place, we request you to pay your dues. We will keep the rates the same for next year:
€ 10 membership for PDF only (both Netherlands and abroad)
€ 28 membership for paper Kunstmaan and PDF
€ 33 membership for paper abroad membership and PDF.

It remains for me to wish everyone happy holidays and a very happy and healthy New Year. The New Year's meeting is scheduled for 14 January. We will usher in the new year with snacks and drinks. It will be unclear which room we are in, but we will hang up notices.

Ben Schellekens
Chairman Working Group Kunstmanen


Meeting Nimeto

POWER METERS

Ben Schellekens

Introduction
With a power meter, you measure power across the full bandwidth of the power meter. With a spectrum analyser, you see power at specific frequencies. In this sense, there are similarities between power meters and spectrum analysers.

A key difference is that power meters are much cheaper spectrum analysers and can measure up to much higher frequencies. For example, my spectrum analyser DSA815 from Rigol is capable of measuring up to 1500MHz, with the article power meters mentioned in this article I can measure up to 8GHz.

I have several power meters. It's fun to list them and look at the differences.

Rohde&Schwarz NRP-Z11
A professional powermeter that I managed to get hold of second-hand, now at least 12 years old. Very convenient is that it can be connected to a computer via USB. Many other powermeters have a sensor head and a separate box. Using an accompanying programme, you can read out the read power in dBm.

The NRP-Z11 has a bandwidth of 10MHz to 8GHz and a dynamic range of -67 to 23dBm.

I have not yet been able to control the NRP-Z11 remotely via GPIB.

The NRP-Z11 directly connected to the measuring transmitter

Mini-Circuits ZX47-40-S+
This power meter has a bandwidth of 10MHz to 8 GHz and a dynamic range of -40 to 20dBm. The output delivers a voltage, depending on the measured power, from 0.5 to 2.1V. You will have to do the calculation to dBm yourself.

Analog Devices AD8317
Has a bandwidth of 1MHz to 10 GHz and measures power between -70 and 20dBm. The linear range is between -40 and -10dBm.

Fig 2. AD8317 board

Measuring setup
See figure fig 3

ESP8266
This runs a web server that allows you to read data from the AD converter via Wi-Fi. The LCD display shows the IP address by which the ESP8266 is known in the local network. This IP address must be entered into the Matlab script.

LCD display
A standard OLED display controlled via I2C.

ADS1115
This is a 16-bit, 4-channel, I2C AD converter.

Power supply
The USB interface to the ESP8266 serves only as a power supply for ESP8266 and the entire I2C bus. The ZX47-40-S+ and the AD8317 are powered via a separate 5V supply.

QWIIC
The interface between the LCD display and the ADS1115 is via I2C. This is an interface with four signals: Vcc, GND, data and clock.

The supplier Sparkfun has come up with QWIIC. This is a system with 4-pin JST connectors that allow you to easily connect I2C modules together. Sparkfun supplies its own Qwiic modules, but they also have PCBs to which you can connect your own I2C modules. It is a nice system that allows you to put something together very quickly without spending a lot of time connecting the I2C bus properly.

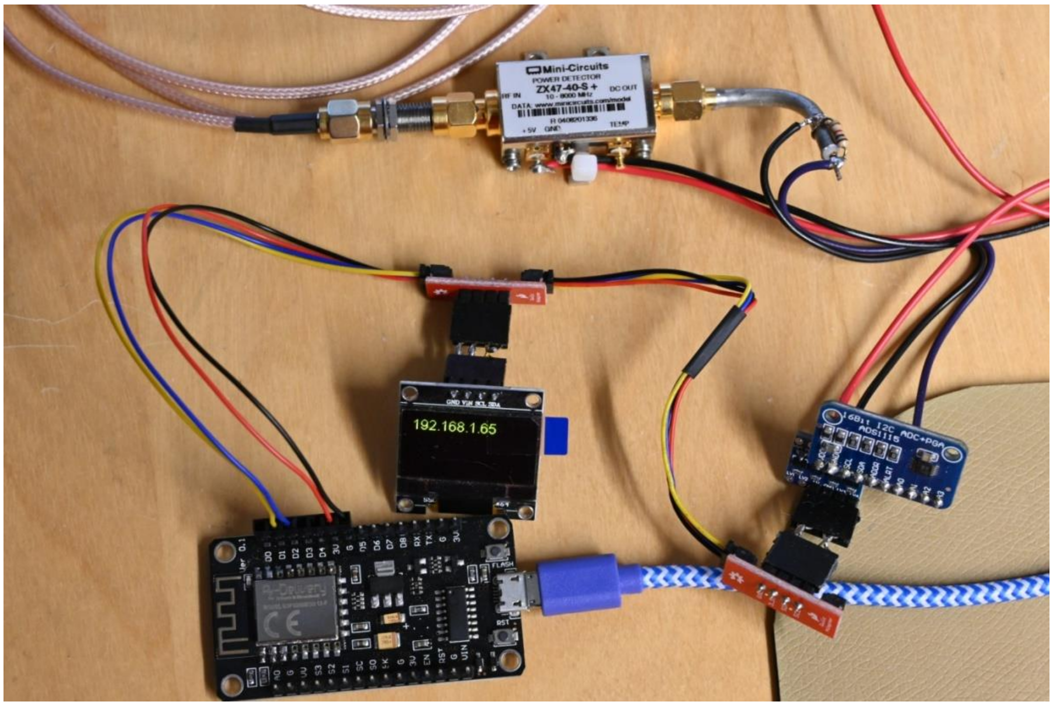
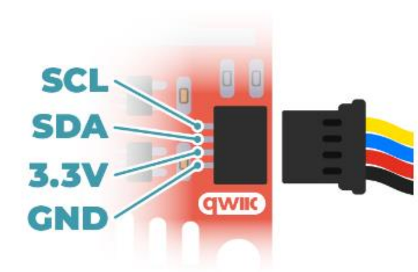


Fig 3. Measurement setup*: top of the power detector. Via a piece of rigid coax, voltage is measured with the ADS1115. Bottom left the ESP8266 and the LCD display, all connected through with Qwiic connectors.*



- Yellow = SCL
- Blue = SDA
- Red = 3.3V
- Black = GND

Fig 4a Pin layout of Qwiic

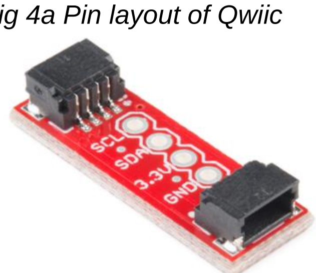


Fig 4b. Adapter PCBs for connecting any other I2C components. This must be 3V3 though! I also used these

Matlab script

I use a Matlab script to automate measurements. If you want a stand-alone power meter, you can of course do the power calculation in the ESP8266.

Line 8 shows the voltage-power ratio. This is taken verbatim from the datasheet.

In line 13, the power is calculated. I chose the offset, 38.6 for the ZX47-40-S+ and 11.6 for the AD8317, so that the readings fit well with measurements with the NRP-Z11.

Measurements

The power detector outputs a DC voltage that is inversely proportional to the measured power.

The set power from the signal generator does not come into the powermeter due to cable losses. To correct for this, I first measured the power with my NRP-Z11 mounted directly on the signal generator.

The measurements in the table on the next page were made at 2200MHz.

As you can see, the set power on the Marconi and what is read by the NRP-Z11 are close to each other. One and a half meters of coax takes 3.3dB of power. The ZX47-40 and the AD8317 were measured after 1.5m of coax.

The columns ADS1115, DMM and the ZX47-40 concern one set of measurements. There are small differences in voltage measured by the ADS1115 and a digital multimeter. Translate this to dBm and you can see that the measurements deviate at 10dBm. The curve is no longer linear here.

The last column is from the AD8317. Again, inaccurate readings at 0dBm, and unusable at 10dB.

```
1   % Reading power level
2
3   % Reading the power sensor
4   url_esp8266 = "http://192.168.1.65";
5   Message.Mes01 = "Reading";
6   Message.Mes02 = "power level";
7
8   slope = 0.025; % ZX47-40S+ 25mV/dB / AD8317 22mV/dB
9
10  result = webwrite(url_esp8266, Message);
11  voltage = ((result.ADC0_msb * 256) + result.ADC0_lsb) * 0.1875e-3;
12  voltage_str = num2str(voltage);
13  dBm = voltage / -slope + 38.6; % ZX47-40S+ 38.6 / AD8317 11.6, met ESP8266
14  dBm_str = num2str(dBm, '%.1f');
15
16  fprintf(strcat("Volt: ", voltage_str, ", dBm: ", dBm_str, "\n"))
17
18  clear Message;
```

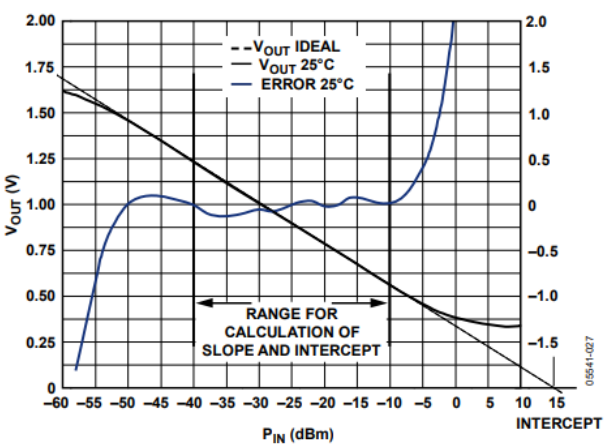
Matlab script to read out power sensor

Marconi 2024	**NRP-Z11 direct**	**ADS1115 in volts**	**DMM U3402 in volts**	**NRP-Z11 1.5m coax**	**dBm ZX47-40**	**dBm AD8317**
-30	-30.1	1.797	1.804	-33.4	-33.3	-33.4
-20	-20.1	1.551	1.560	-23.4	-23.4	-23.3
-10	-10.3	1.305	1.315	-13.6	-13.6	-13.6
0	-0.3	1.068	1.068	-3.6	-4.0	-5.3
10	9.7	0.822	0.821	6.4	5.8	-5.9

Table 1 Measurements at 2200MHz

Calibrate

For accurate measurements, a power meter must be calibrated. This calibration must be done per frequency and per power. If you are in the linear region of the powermeter then the offset changes (line 13 in the Matlab script), beyond this it becomes a trickier story with more maths.



Dynamic range of the AD8317 runs from -40 to -10. The intercept is at 15dBm, which in my case is 11.6dBm.

With the AD8317, they indicate that if you want to measure beyond 8GHz, the dynamic range drops.

Conclusion

For a few tenner you can buy the AD8317 from Amazon or eBay. If you take into account the limitations, this is a good buy. If you want to measure more than -10dBm of power then you can think of attenuators and below -40dBm of amplifiers.

RECEPTION OF THE ORION / ARTEMIS 1 LUNAR MISSION

Jaap Rusticus

Last September, I heard that the Camras foundation (Dwingeloo radio telescope) was going to participate in the project to receive/follow the Orion/Artemis lunar mission. What I only found out later was that NASA had asked on the Internet for participants with a dish of at least 9 metres. Camras had applied for this and with the Dwingeloo dish of 25 metres they became one of the 18 participants worldwide, including two individuals from Canada and the United States.

My interest was piqued. I knew the stories of Jan PA0SSB who wrote to NASA in the 1970s asking for the frequencies to be used in the Apollo moon missions. He got a neat reply with the various frequencies in the satellite downlink S-Band (2200-2300MHz). At the time, he managed to receive the signals with his self-built 6-metre rotating dish!

I had no idea about the strength of the signal. Nor did I know the frequency, but after an Internet search and a hint from Camras, the reception frequency was 2216.5MHz.

The weather was nice in September and equipment was set up outside to do experiments well before the planned launch. Equipment being stuff I had owned for a long time: dishes of 70cm, 132cm and 244cm (the latter was not used, by the way), a Meteosat Frisolac bus (feedhorn), a home-built Meteosat preamplifier that still performed pretty well even at 2300MHz and, as a receiver, an HP 8560E spectrum analyser. Soon, Inmarsat satellites were observed over the Indian Ocean. By observed, I mean seeing carriers (carrier waves) on the screen, nothing more - no demodulation or anything else.

GPS signals were also quite visible. I could not find the Meteosat satellites with the 132cm dish, but I did find a weak signal from the probably Russian Elektro-L2 weather satellite at 1693MHz (fig. 1).

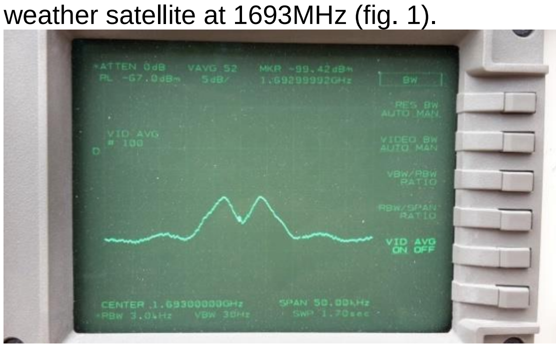


Fig. 1 Electro L2 at 1693MHz

During all these experiments, it became clear to me that the signals at telecommunication frequencies were coming in pretty strong on the bus antenna. The frequency ranges I am referring to here are roughly 750-970MHz, 1800-1900MHz, 2110-2170MHz, 2400MHz ISM band, 2596-2617MHz. These (terrestrial) signals are many times stronger than the desired weak signal (Orion) and can cause oversteering and intermodulation products in the receive chain. So I needed a filter for the satellite downlink band 2200-2300MHz. I could also have used a filter for only 2216.5MHz but I wanted to keep things somewhat universal. Unfortunately, the filters I had and might be able to modify were only about 10MHz wide and also not good for tuning. So I just started experimenting with making a good filter myself (fig. 2). This ended up working out very well. It became a so-called combline bandpass filter. This has the advantage over an interdigital filter that all trimmers are on the same side. The filter has the following properties: a pass of 2200-2300MHz, a pass attenuation of about 3dB, a variation of 2dB and an attenuation of 45dB at 50MHz and 75dB at 100MHz distance (see fig. 3). So far, I have the prototype in use. I still want to set it up in brass. For a follow-up story I guess.

Fig. 2 2200-2300MHz prototype Bandpass Filter

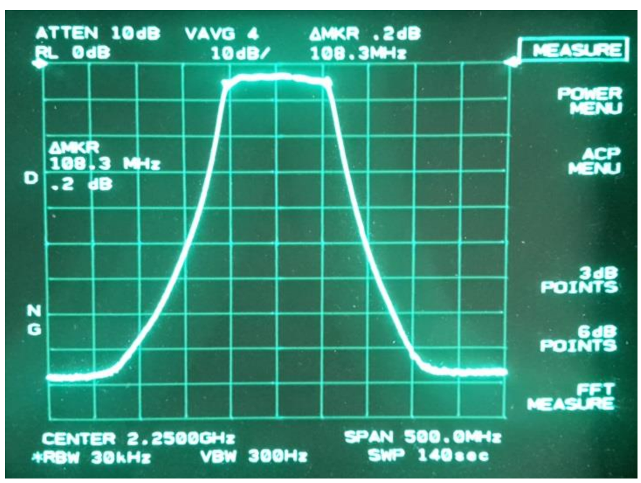


Fig.3 2200-2300MHz Bandpass Filter Plot

A word about the can (feedhorn). The Meteosat feedhorn is too big for S-band frequencies. Off to the shop to see what's on sale in the supermarket. I came home with two cans of Pineapple pieces in light syrup. I still have them. At 8.3cm internally, they were actually too small. Then I came back to my trusty tins of Unox soup anyway (especially Chinese Tomato Soup) with 9.8cm internally nicely sized. Frequency range ~1790-2340MHz. Probe length 29mm and probe distance from the rear wall of 54mm. The reflex attenuation of just the can was around 20dB, a good value.

As the launch has been postponed several times, it was heading towards October and November. Too cold and wet to go outside with equipment anymore. So still after 28 years, the Yaesu G-5600 El-Az rotor came in handy. It had never been outside before (bad huh). The corresponding stand-alone Trakbox tracker, built 28 years ago, also came out of the cupboard (ditto). By the way, I did not use that one for the Orion experiments, but this aside.

Before the launch, I was already busy with the dish (132cm) and rotor setup in the shack. Preamplifiers, filter, cables, rotor control, etc. The day before the launch, I heard that the Artemis launch would take place at 7am the next morning. Huh? I was under the impression that it was going to happen at 7 pm. So no. I worked through the night to get things in order, saw the launch at 8am on Wednesday morning and by 8.30am, the dish was outside with rotor and all on the tripod.

A word about the reception chain. At the back of the van, I had placed my Meteosat pre-amplifier with about 30dB of gain. Then a 3-metre satellite TV cable to the box at the bottom of the dish leg. A satellite TV inline amplifier, the aforementioned bandpass filter, another satellite TV inline amplifier and then 12 metres of satellite TV cable towards the shack. There follows a Bias-T to feed the amplifiers and an HP 11693A Limiter to protect the spectrum analyser. Then follows the spectrum analyser.

Reception:
Beforehand, I had an expectation/hope that if Orion was still close to Earth, I would be able to receive the signal. That Wednesday later, after the Trans Lunar Injection Burn to shoot it to the moon, it would be low in the southwest here. Near me there are all trees, so that was going to be nothing. Thursday I was late for a good attempt. The only position indication I had was to the right of the moon, approximately.... Nothing received.

Friday a good attempt - nothing received. Spent a long time spinning the rotor and searching with the spectrum analyser. A frequency quote of 2216.5MHz means the carrier could be between 2216.45 and 2216.55. That's about 100kHz and with a low Resolution Bandwidth (RBW), for a better signal-to-noise ratio, this is difficult searching. It was already almost halfway to the moon and may have become too weak for reception with such a small dish.

During these days, I tried to receive other satellites starting with the geostationary orbit. I had Az-El values from that. Pretty soon I saw quite strong signals around 40º East and 12º East. Carrier waves with sidebands at 65kHz distances. No idea what this is (fig. 4).

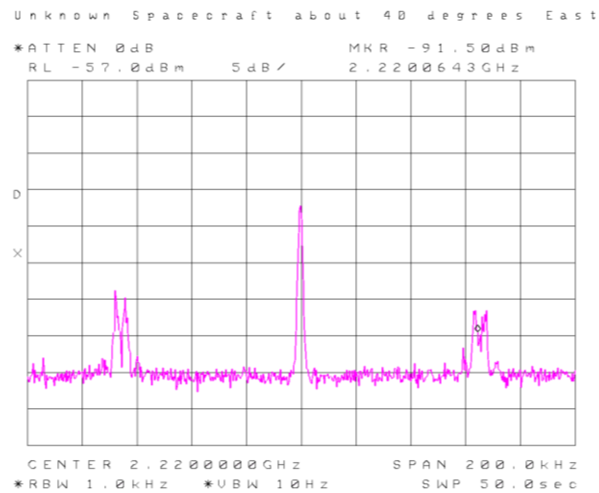


Fig. 4 Unknown geostationary satellite at about 40 degrees east

I could also see the sun noise. It was almost 3 dB higher than the noise next to the sun. A nice indicator of the sensitivity of the system (Fig. 5).

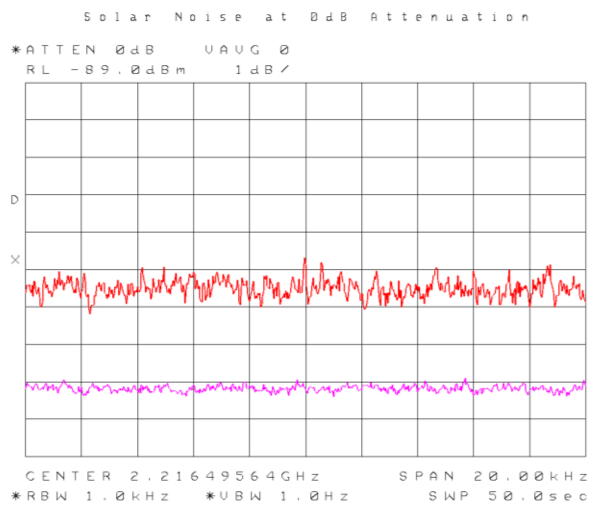


Fig. 5 Top line is noise from the sun - bottom line is the average noise not facing the sun

Saturday and Sunday. Again, no Orion. We did see a special other satellite, namely the DSCOVR satellite at the Lagrange L1 point 1.5 million km away (Fig 6).

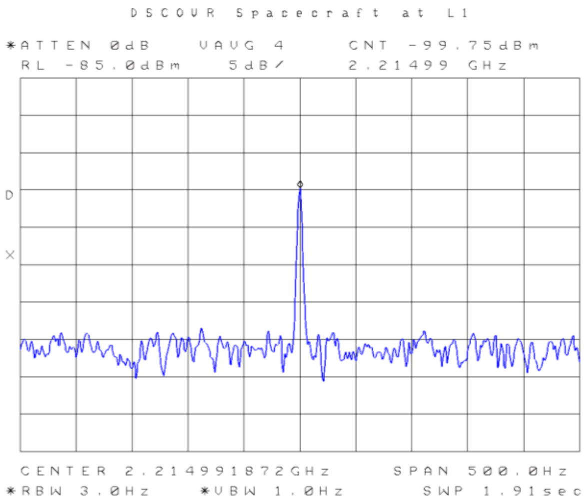


Fig. 6 DSCOVR satellite at Lagrange L1

I luckily found data on the Internet. Pretty well with a good signal of about 20dB above the noise with an RBW of 3Hz. Also, the Doppler Shift (frequency shift due to distance change between observer and source) now becomes quite visible.
The satellites I have received so far are:

ACE
- Advanced Composition Explorer - 1997

CHANDRA / CXO
- Chandra X-Ray Observatory - 1999

DSCOVR
- Deep Space Climate Observatory
- 4º-15º half-year halo job - 2015

KPLO
- Korea Pathfinder Lunar Orbiter
- also known as - Danuri - 2022

Monday. New Orion attempt.
By now, I had found a tactic to aim my dish. The way it used to be done when installing TV antennas - with one eye on neighbour's antenna.

The NASA Deep Space Network website https://eyes.nasa.gov/dsn/dsn.html lists their dishes with the satellite being received and the position of the dish. I took the data from DSN Madrid and corrected the Az-El values as I saw fit. Better something than nothing.

Later - even better; check the Camras website https://www.camras.nl/. There you will find the current antenna position at the bottom of the page. One learns by doing.

Again later - much better. On NASA's JPL (Jet Propulsion Laboratory) Horizons website https://ssd.jpl.nasa.gov/horizons/app.html#/, you can enter your own position on the globe, enter satellite of your choice, time period and step size (I chose Step 15 minutes), and view and download the Az-El data. There will probably be persons among you who knew all this for a long time. For me, it was a self-exploration and learning process.

To the point, Monday, new attempt. Orion had reached the moon by now so targeting was no longer difficult. Moon Az-El data I took from https://www.heavens-above.com/moon.aspx
Do create own login with position and press "Reset time" every now and then.

As before, pointing dish (now environment moon) and setting spectrum analyser to default frequency of 2216.5MHz. Miraculously, I immediately saw a carrier at almost 2216.500MHz. Finally and so anyway. And after correcting the dish to maximum signal quite a strong signal of about 23dB above the noise at an RBW of 3Hz. At the SPAN of 500Hz, the Doppler Shift was clearly visible. The carrier slides slowly to the left or right on the screen.

A word about the frequency stability and accuracy of the analyser. A spectrum analyser has an internal frequency source that determines both. This is, if all goes well, an OCXO - an Oven Controlled Crystal (Xtal) Oscillator. This needs to come up to temperature and be properly tuned. To avoid inaccuracy due to any deviation of the OCXO, most analysers have a BNC connector for connecting an external accurate 10MHz reference source. This can be a Rubidium source, but even these can be slightly off. Nowadays there are also many GPSDOs for sale - GPS Disciplined Oscillator - these are generally OCXOs that are set to the correct frequency by a GPS signal. When the GPS signal drops out, they remain at the last frequency for a long time. I use such a GPSDO as a reference for the spectrum analyser. The frequency reading becomes quite accurate from it.

Tuesday. Again, nothing received. Why not.

Wednesday. Got in touch with an acquaintance at Camras. After I asked him why I usually don't see Orion's signal, he replied that the same modulation technique is not always used. Usually this is a modulation technique where the carrier is suppressed and then I don't see it. Camras has 25dB more signal and can do carrier recovery using the sidebands. I always didn't understand it, but now it was clear why I usually didn't see anything.

In the days that followed, it always remained at a brief attempt, no carrier seen.
Only the following Saturday did I capture the signal twice again. By capture, I mean taking a screenshot. I do that using an HP GPIB to USB interface and the John Miles KE5FX program (fig. 7).

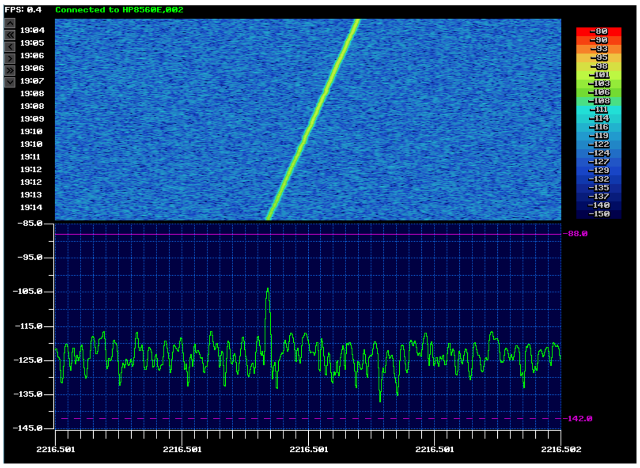


Fig. 7 Orion Return Link Carrier Spectrum Waterfall - 19.15 MET 01-12-2022 - the Doppler Shift can be clearly seen - horizontal frequency indication is not nicely indicated by the programme

The last time I saw the signal again was the day before landing. During this time, I actually saw the signal disappear due to the modulation method change. The Doppler Shift had already risen above 10kHz due to the increasing speed towards Earth.

Doppler Shift was an observation that NASA had also asked the participants to make.

For fun, I timed one screenshot to the second. The frequency is to the Hz on the screenshot and was 2216,510,821 Hz (Fig. 8).

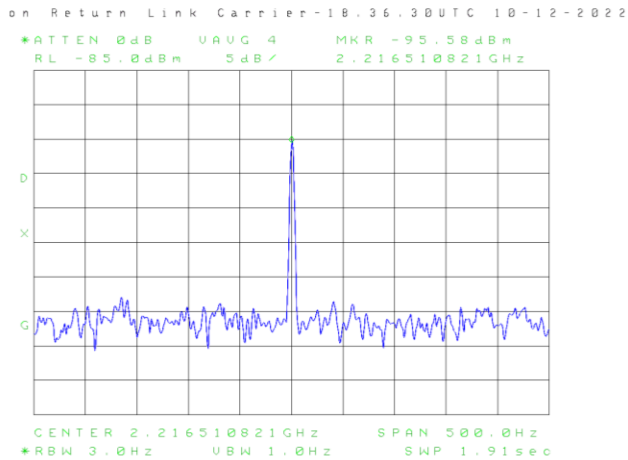


Fig. 8 Orion Return Link Carrier 18.36.30 UTC 10-12-2022

I then looked up the frequency in the current online Camras reception log at the same time. The Camras frequency was 2216,510,828 Hz. So Camras was 7 Hz too high :-)

Also attached is my list of satellites I have seen (& - last column) and would still like to see.
If you guys have any questions or further information, especially more precise frequencies, I would love to hear from you.

Jaap Rusticus
jrusticus@hotmail.com

Frequency	Name	Position	Az & El	S/N @ RBW	Details	Det.
2205 ??? ??? Hz	TDRS ????				According to NASA from 2007	
2214 992 462 Hz	DSCOVR	Lagrange L1	6º 18-11-22	22dB @ 3Hz	252Hz Doppler Shift at 30º	&
2216 499 512 Hz	Artemis Orion	E - M - E	-	27dB @ 3Hz		&
2216 500 000 Hz	TDRS Rx					
2219 999 310 Hz		GEO ~38º O	Az 142º & El 23º	49dB @ 3Hz	with sidebands + and - 65kHz	&
2223 368 977 Hz		GEO ~8º O	Az 177º & El 29º	46dB @ 3Hz	with sidebands + and - 65kHz	
2226 5?? ??? Hz		GEO ~8º O	Az 177º & El 29º		short-term carriers	
223? ??? ??? Hz	XMM Newton		-			
2231 759 862 Hz		GEO ~18º O	Az 165º & El 28º	40dB @ 3Hz	with sidebands + and - 65kHz	&
2241 744 080 Hz		GEO ~31º O	Az 150 & El 25º	47dB @ 3Hz	with sidebands + and - 65kHz	&
2245 ???? ??? Hz	SOHO	Lagrange L1	+28º to -28º		to 28º east or west of the sun	
2249 989 000 Hz	CHANDRA				highly elliptical orbit	&
2252 700 ??? Hz		GEO	Az 158º & El 27º	6dB	4 MHz wide	&
2255 5?? ??? Hz	HST				+ 2287.5 MHz	
2260 800 000 Hz	KPLO	Lunar Orbit	2 kHz	35dB @ 3Hz		&
2265 ??? ??? Hz	ISS					
2275 ??? ??? Hz	WIND	Lagrange L1				
2270 5?? ??? Hz	JWST	Lagrange L2				
2271 125 ??? Hz	LRO	Lunar Orbit				
2278 365 558 Hz	ACE	Lagrange L1		23dB @ 3Hz		&

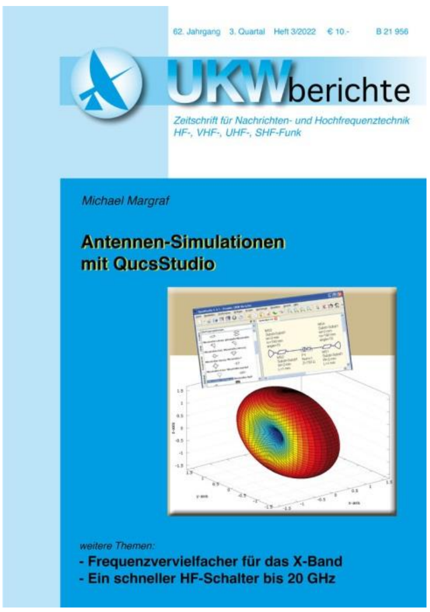
Frequency	Name	Position	Az & El	S/N @ RBW	Details	Det.
2282 5?? ??? Hz	THEMIS-A					
2282 5?? ??? Hz	THEMIS-B	Lunar Orbit				
2282 5?? ??? Hz	THEMIS-C	Lunar Orbit				
2282 5?? ??? Hz	THEMIS-D					
2282 5?? ??? Hz	THEMIS-E					
228? ??? ??? Hz	MMS-1					
228? ??? ??? Hz	MMS-2					
228? ??? ??? Hz	MMS-3					
228? ??? ??? Hz	MMS-4					
2287 500 000 Hz	TDRS Rx	GEO				

UKW-BERICHTE

Ben Schellekens

Besides the regular sections, four main articles have been published in this year's fourth UKW Notices.

The first article is by Wolfgang Schneider and he describes frequency multipliers to come out on the X-band.



The HMC369LP3E from Hittite (now Analog Devices) is a doubler with dimensions of 3x3mm. The output power is up to 4dBM. The input power can vary between -10dBm and 5dBm, 0dBm is recommended. The output frequency is between 9.9 and 12.7 GHz.

The test specimen built has Rogers RO4350B laminate, as power source a 78L05. The housing is aluminum and is available on eBay for very little, incl. LNA. Incidentally, I have had lesser experiences with these enclosures, incl SMA connectors.

A cheap LNA from eBay serves as the housing

Groundwave is suppressed by 30dB, mixing products were not measurable.

For the frequency range from 9.8 to 11.2GHz, several multipliers are available from Hittite in the HMC44x series: x4, x8 and x16.

Hittite provides with the HMC441 an amplifier for the frequency range 6.5 to 13.5GHz, gain around 15dB. In particular, he warns that capacitors, among others, must be suitable for the GHz range. Through-metalised PCBs are available from the author.

The second article is by Michael Margraf. He is the programmer of (freely available) QucsStudio and has now extended it to include antenna simulation.

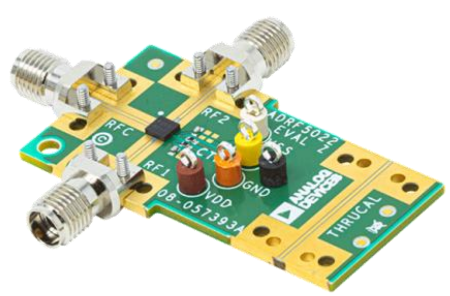
Many antenna simulation programs are based on the public NEC (Numerical Electromagnetics Code). This method is particularly suitable for wire antennas. In QucsStudio, it uses FDTD (finite-difference time-domain) and is suitable for PCB antennas, e.g. patch antennas (so-called planar structures). Among other things, he describes how to analyse a circularly-polarised patch antenna.

The built-in scripting language Octave is used for calculating and presenting the 3D plots.

In the third article, Bernd Kaa describes a diplexer (= an inverter) suitable up to 20GHz. He uses this chip to go from 0 to 15.9GHz in one sweep with his tracking generator.

Analog Devices is the manufacturer. The ADRF5020 and ADRF5021 are available for around 100 Euros. The advantage of these chips is that they can switch several nanosecond large powers.

It's IC is very small at 3x3mm, pin spacing is 0.4mm. He has designed his own PCB that fits into a 30x30mm aluminium tube.



They are gems, Analog Devices' boards.

When building circuits in the GHz range, you always run the risk of reflections when installing them in enclosures. So there seems to be special foam (MAF-19) to dampen reflections. Foam onto which ICs are pinned also works, but not as well.

The fourth article is by Alexander Meier and is a follow-up to his article in this year's first UKW Message. Now he describes the UHF connector suitable up to 300MHz. In addition, the more interesting N-connector for us, suitable up to 11GHz. N connectors are also available in 75 ohm versions, these are not compatible with the 50 ohm version! BNC connectors are suitable up to 4GHz, TNC up to 10GHz.

Up to 18GHz, the SMA are suitable. Similar to SMA is the 3.5mm connector. It has a different dielectric and goes up to 26.5GHz.

3.5mm connector, compatible with SMA

The lesser-known 7mm connectors are genderless and go up to 18GHz. He further mentions the SMP, SMB, SMC and the 7-16 connectors.

The latest article is by Jochen Dreier. He describes the control of HP attenuators going up to 18 GHz. With changeover switches, the attenuators (of 1, 2, 4, 4,10, 20 and 40dB) can be switched on. Control of these attenuators using an Arduino I described in the June Kunstmaan 2019.

In Jochen's article, the HP attenuators are set with switches

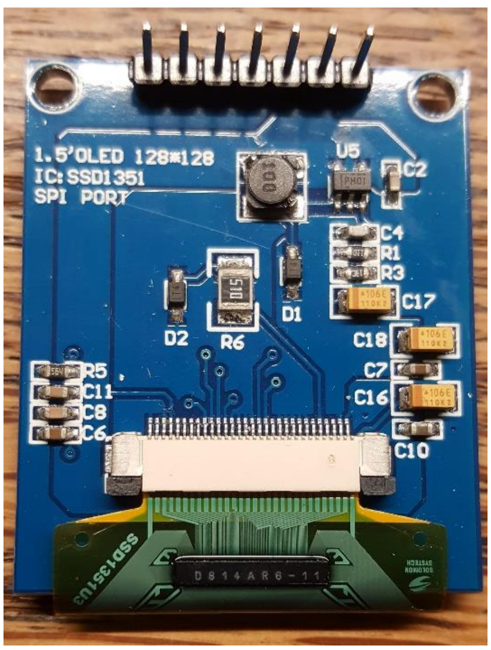
Also, the regular feature Fundstelle Internet by Gunthard Kraus. In Ultra kurz the report that the Kuhne company has been taken over by Alaris from South Africa

Please let me know if you are interested in an article.

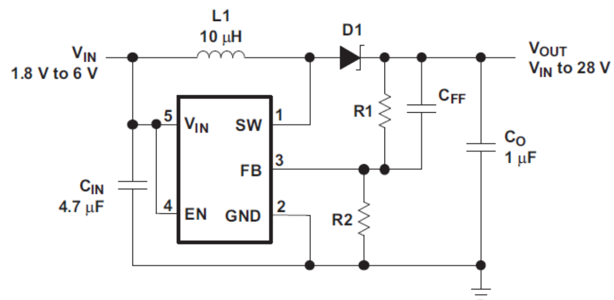
OLED DISPLAY OF THE CONSTELLATION VIEWER

Rob Hollander

There are several reasons to take another look at the constellation viewer's OLED display. First, you have to look closely to see anything at all on the display. The brightness is improbably low, so low that it is hard to imagine this OLED being a marketable product. So the second reason is that there is probably something wrong with the controls, either hardware- or software-based. This suspicion is reinforced by the fact that a lot of light can indeed come out of the display, as evidenced by the bright stripe that sometimes lights up when switched off. The suspicion that there would be something to improve on the hardware side was given to me by a comment in Tam Hanna's Nov-Dec 2018 article in Elektor Magazine: 'Controlling colour LEDs with STM32' which includes a picture of the display contained in the constellation viewer. That commented: 'When the display is powered with 3.3V, the brightness is not anything to write home about; outdoors, the display imported from China was basically unusable. In indoor applications, however, the displays are quite readable'. This comment by Tam Hanna suggests that the display could give better results with a different supply voltage.



First, let's take a look at the hardware. The OLED display is driven by the SSD1351. This is the large chip on the flexible connector of the display. The SSD1351's datasheet states under 'FEATURES' that the OLED display must be powered by VCC = 10.0V - 18.0V (Panel driving power supply). This VCC should not be confused with the PCB's VCC connection; it corresponds to VCI (Low voltage power supply) in the datasheet and should be 2.4V - 3.5V (e.g. 3.3V). The suggestion that the display would function better with a different (higher) supply voltage is nonsense; the supply voltage of the PCB should be 3.3V. But then how does the OLED display get a VCC of 10-18V? On the PCB is a PHOI (U5). This is the TPS61040 low power DC-DC converter. The application example from the datasheet is implemented on the PCB.



The voltage on FB is compared to an internal reference of 1.233V.

With the measured values of R1=126k and R2=13.2k (on the PCB, this is R3), it becomes

Vout = 1.233V x(1+R1/R2) = 13V

This voltage was also measured on the anode of D1 and this thus falls within 10 - 18V.

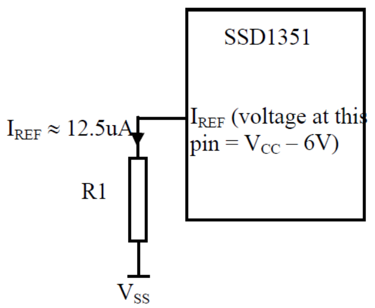
The brightness of the display is determined by the current through the segments. How much current the OLED display can handle is not known. However, the maximum current the SSD1351 can deliver per segment is 200uA (listed in the FEATURES). The current through a segment is determined by an external resistor. In the datasheet:

'For example, in order to achieve ISEG = 200uA at maximum contrast 255, IREF is set to around 12.5uA. This current value is obtained by connecting an appropriate resistor from IREF pin to VSS as shown in Figure 8-8.

Since the voltage at IREF pin is VCC - 6V, the value of resistor R1 can be found as below:

For IREF = 12.5uA, VCC =16V: R1 = (Voltage at IREF - VSS) / IREF ≈ (16 - 6) / 12.5uA ≈ 800kΩ

Figure 8-8 : I_{REF} Current Setting by Resistor Value



The SSD1351's datasheet assumes a VCC for the OLED display of 16V, while 13V is generated on the PCB. To achieve the maximum current through a segment of 200uA, to get IREF = 12.5 uA a resistor of 560k must be taken. On the PCB, this resistor R5 is indeed 560k ! Therefore, although the VCC is not 16V but 13V, the maximum segment current is still 200uA.

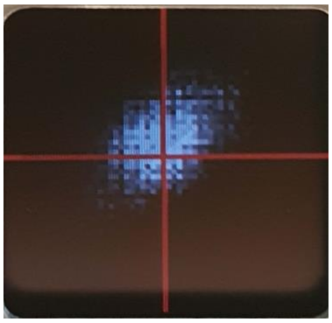
So on the hardware side, everything seems fine to get the maximum brightness; the SSD1351 can't provide more power. Then another look at the software.
The brightness can be set using commands for the SSD1351:

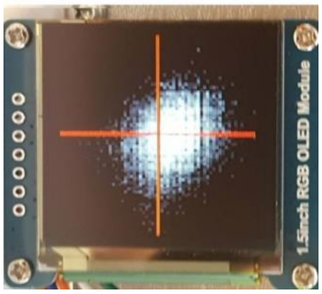
- C7 Master Contrast Current Control (for all colours). Set to 'No reduction' at RESET.
- C1 Set Contrast Current (per colour). Is not set to maximum at RESET.
- B2 Display Enhancement. Set to 'No display enhancement' at RESET.

Just to be sure, I included command C7 (MCCC), 'No reduction' in the program. This gave no improvement, as might be expected. The B2 command was used to turn on 'Display Enhancement'. This gave no change in brightness. With command C1, the Contrast Current for the three colours was set to maximum. This gave some increase in brightness.

Furthermore, there are commands to set the 'pre-charge period' and the 'pre-charge voltage level'. These are commands to allow adjustment for the capacity of the OLED displays. If the capacity is high, the period should be longer and the voltage higher. These commands were played with, but no improvement in brightness could be obtained.

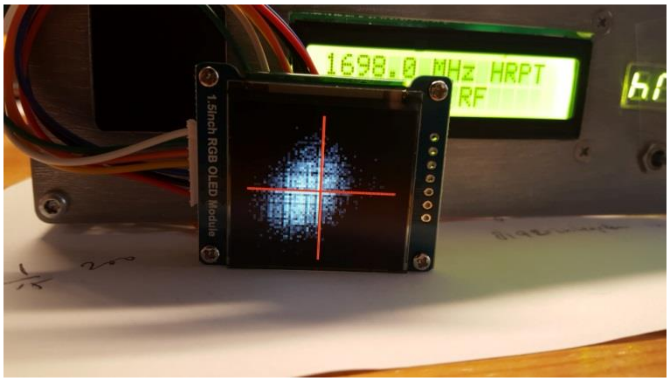
Now there are differences in brightness between copies and my copy is slightly worse than Rob Alblas's. So it seems that a display has entered the market that is improbably poor and we will have to live with the low brightness of the constellation viewer's image. There are similar displays in old-fashioned mobile phones that can be used as torches!





I was left with a nasty feeling and decided to purchase another OLED display. At Tinytronics [1], I found a display from Waveshare, similar to the one I used; also 128 x 128 RGB, SPI interface and SSD1351 driver IC. Unfortunately, the connector pins are rotated 90 degrees and the order of the 8 connections is different. That can be overcome. A connection cable with loose pins is included. With this, the display could be quickly tried out . Left the old, right the new OLED display. What a difference! A perfectly clear display

[1] [OLED display from Tinytronics]



OLED display old and new

QPSK GENERATOR IN CPLD PART 3

Rob Alblas

In "de Kunstmaan" Nos 2 and 3 ([1],[2]) I described a generator that can be used to generate the data for a QPSK generator. The formats are:

satellite	bit rate	coding	data
Metop	2x 2.3333Mb/s	Viterbi, punctioned	Metop-like
NOAA20	2x 15 Mb/s	Viterbi	Metop-like
Aqua	2x 7.5 Mb/s	no	Metop-like

This generator is primarily intended for testing receivers; at least a decoder will be able to synchronise properly on the data. The actual data is then not so important. Therefore, due to the limited space in a CPLD, the data generated is the same for all satellites.

No generator was yet available for Fengyun-3D, but due to the high signal level of this satellite, it is precisely what makes it interesting to base a receiver on.

The datarate is a bit faster; 2x 30Mb/s, and in this, unfortunately, the Viterbi coding is also realised in a rather peculiar way, just like in Fengyun-3A, 3B and 3C. Namely, here 2 Viterbi encoders were used for I and Q. Normally, a Viterbi coder already produces a double signal, which is then used for I and Q; this is how it was realised for all other satellites.

With the CPLD used, and the board used for this, there are now two problems:

- The CPLD is already quite full
- On the board is a 50 MHz crystal, a 30 MHz clock is needed for FY-3D. That's going to be rather jittery if you're going to derive that 30 MHz with a fractional divisor from 50 MHz.

With a simple approach, I was able to make the generator suitable for other data-rates as well. A separate external clock (oscillator) is then needed with the right frequency. With an extra pin, one can choose between internal and external clock. Viterbi coding and data format don't change, so that won't be right for FY-3D, but you can generate a data stream at the right rate. By the way, a decoder for the receiver that can also handle FY-3D still has to be made, but initially, with the modified generator and an "artificial satellite " ([3]), you can already start working to see if the data comes out of the receiver nicely.

In the case of an external clock, the data rate depends on the selected satellite:

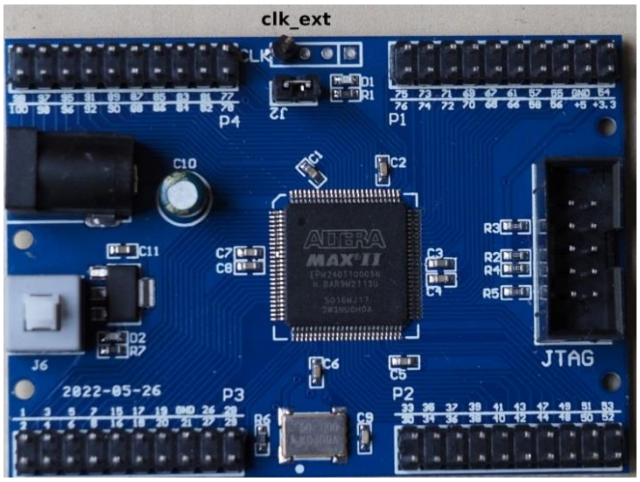
selected satellite	sub-factor	freq. for FY-3D datarate
Metop	9	270 MHz, so it won't go
NOAA20	1	30 MHz
Aqua	2	60 MHz

The most appropriate setting seems to me to be NOAA20, so with a 30 MHz external oscillator, a NOAA20-encoded signal with Metop-like data is generated at Fengyun-3D speed. So the main issue then is to generate data at the right speed.

The pinning is chosen so that you don't have to do anything for original functionality. All input pins have pull-ups, so for a functional '1' (3.3V) they do not need to be connected.

pin	function	notice.
2	I-Out	
4	Q output	
18,20	satellite selection	11=Metop, 01-Aqua, 10=NOAA20
16	choice oscillator:	'1'=internal, '0'=external
clk2	external oscillator	FY-3D: select NOAA20, osc. 30 MHz

For clk2=clk_ext, an extra pin has to be fitted, see picture. All other pins are on connector P3, bottom left.



The custom files for programming are available from the workgroup git ([4]). The programming is described in part 2 ([2]).

[1] QPSK generator in CPLD. de Kunstmaan 2022 no. 2, p. 12
[2] QPSK generator in CPLD part 2. the Kunstmaan 2022 no 3, page 12
[3] A special artificial satellite. the Kunstmaan 2021 No. 3, p. 10
[4] Github" QPSK generator

SIMULATION OF QPSK

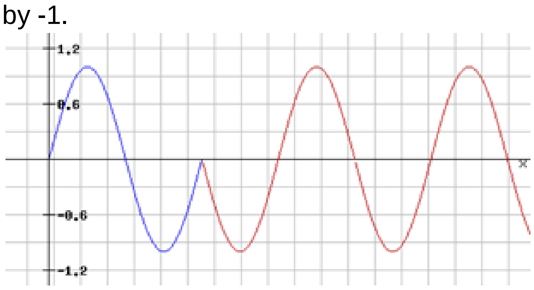
Ben Schellekens

Introduction
At the last meeting, the question of what is now the bandwidth of a QPSK signal came up during the roundtable discussion. But also the question: what is an OQPSK signal. All this in light of the question of whether it is possible to build a receiver in hardware for the 8GHz satellites. The scenario is that you have to build a separate receiver for each modulation. Hopefully, it won't be too bad.

In this article, I try to explain (O)QPSK using simulations in Matlab (or Scilab / Octave).

QPSK
Besides amplitude modulation (AM) and frequency modulation (FM), we can also use phase modulation to transmit data.

The simplest form is BPSK, in which the phase of the signal jumps 180°, so this has two phases: 0° and 180°. This is the same as multiplying the carrier wave by -1.



A BPSK signal with a phase jump of 180° With QPSK, you know four phases (0°, 90°, 180° and 270°).

The staggered phases create a wide frequency spectrum. Filtering on the transmitter side will be necessary. In the simulations below, we see the spectrum.

Another issue is that on the receiver side, it is not known what phase the signal has at any point in time. This will have to be reconstructed afterwards.

One way to overcome this problem is to see if a certain bit pattern is received and test just as many times (max 4 times) until you are "in phase". This is how it is currently done in Rob's HRPT/QPSK decoder in the FPGA. This is why it takes a while to get a lock.

Another method is by looking at the relative phase shifts in the signal. This is called differential QPSK, a topic of a forthcoming article.

What is I/Q
In the context of QPSK, you will also see the terms I and Q passing by. What do these terms mean?

Above, I talked about transmitting data by means of staggering the phase of the signal. This is technically, to say the least, challenging to achieve. They have found a solution to that and this is the so-called I/Q modulator.

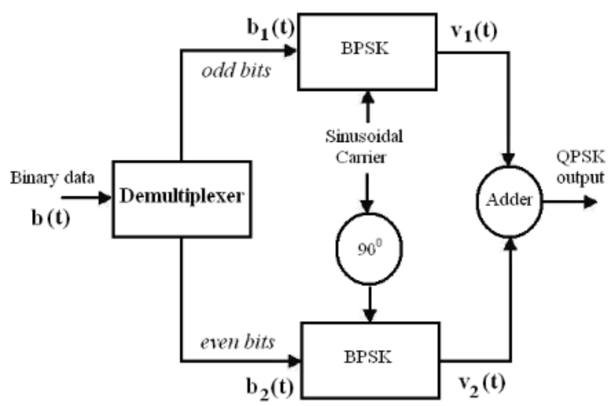
The "I" stands for "in-phase" and the "Q" for quadrature. This does not make it any clearer: something can only be "in-phase" in reference to another signal and there is none.

When talking about I/Q, one means two sines of equal frequency shifted 90° in phase with respect to each other.

By convention say the I signal is the cosine and the Q signal is the sine.

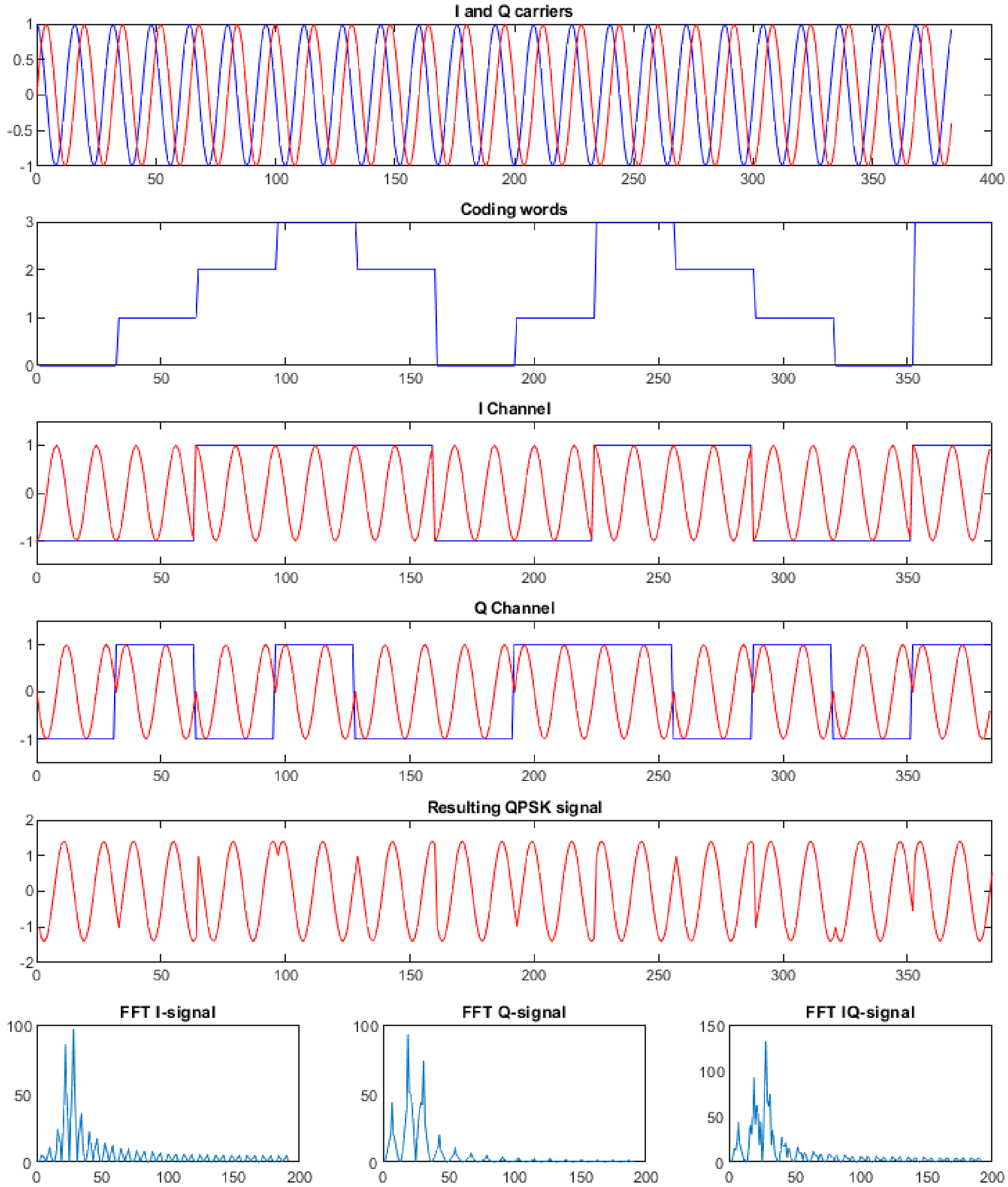
Modulate
So we have two signals I and Q. Let's zoom in on the I signal. We want to put information on this. We can do AM, FM or phase modulation. I/Q modulation always involves phase modulation. The carrier wave can be inverted, shifted 180°. Just as it is done in BPSK.
With the Q signal, exactly the same thing happens, on a 90° shifted carrier.

The QPSK signal is obtained by adding the I and Q signals together.

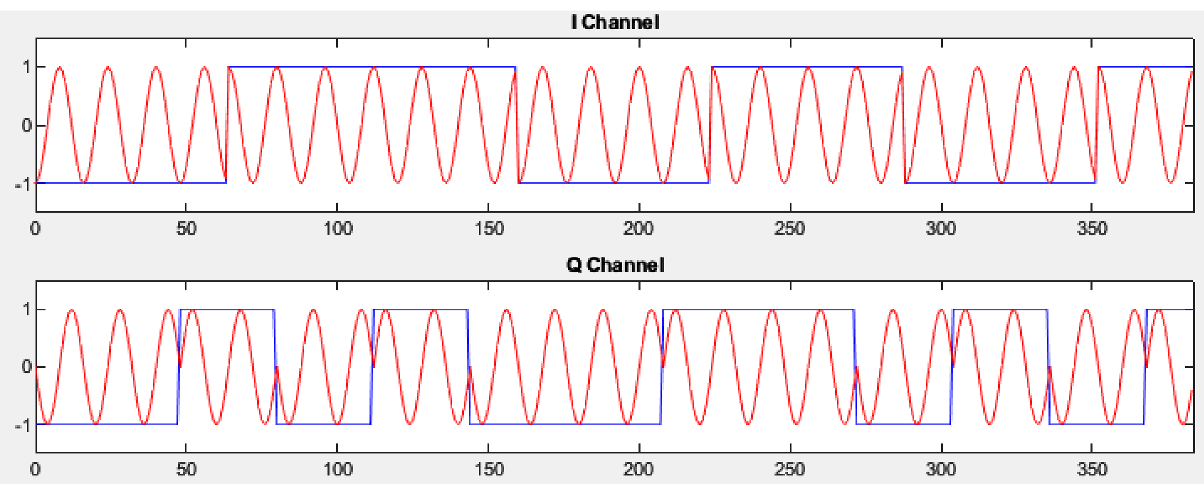


Schematic overview of a QPSK modulator

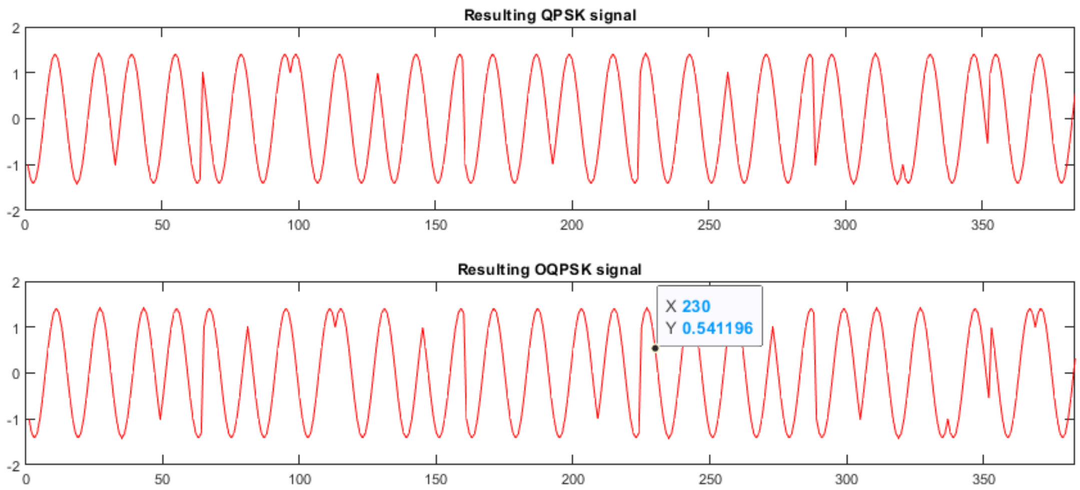
OQPSK
In a QPSK signal, the I and Q change at the same time. With OQPSK, the changes take place at different times. The advantage is that the phase jump is no more than 90°. The bandwidth is smaller.



The top graph shows the I and Q signals. The I signal is red. The third and fourth graphs show the phase shift in the I and Q due to the bit stream. The fifth graph shows the resulting QPSK signal.



Very clearly see that the Q signal has shifted half a bit.



A comparison between QPSK and OQPSK. The transitions 64, 287, 351 are weaker.

Matlab script
The Matlab script used leans heavily on what Robert Lacoste used in his book "The darker side". He wrote it in Scilab, but the lack of updates to Scilab made me switch to Matlab.

You can see that the script is very similar to the hardware of a "real" QPSK modulator. In "de Kunstmaan "of December 2021, Harm described a modulator board using the AD8346.

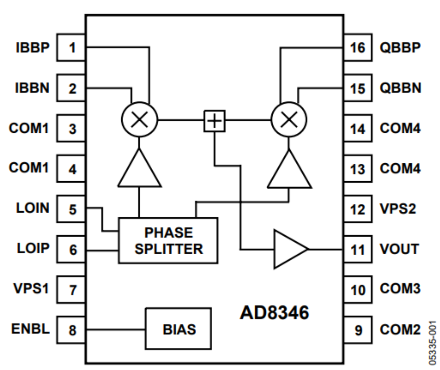
The Matlab script is at the end of this article.

In the script, the oscillator and phase splitter in the AD8346 are lines 9-11.

The script works with "words" composed of two bits: 0=00, 1=01, 2=10 and 3=11. Lines 13-15.

In lines 20-23, the words are split between I and Q. The words 2 and 3 are in the I signal, which is then a logical 1. The words 1 and 3 in the Q signal.

In line 23, the Q signal is shifted half a bit width to make the OQPSK signal.



The AD8346 with the phase splitter/shifter and two mixers

The mixers in the AD8346 are lines 25-27.

The addition of the I and Q signals is done in lines 29-30.

The spectrum
What you see is that the spectrum does not widen when you add the I and Q signal together. This is an important advantage of this modulation technique.

Conclusion
A tentative conclusion might be that a QPSK receiver in hardware can also receive OQPSK. If you lock on the I signal then you have to sample your Q signal half a bit further, this means a modification on the decoder / FPGA side.

Links / references
https://www.allaboutcircuits.com/textbook/radio-frequency-analysis-design/radio-frequency-demodulation/understanding-i-q-signals-and-quadrature-modulation/

```
% QPSK modulation

points_per_period = 16;
bit_count = 12;
carrier_periods_per_bit = 2;
nr_points = points_per_period * bit_count * carrier_periods_per_bit;
t = (0:nr_points - 1);

% Generate I and Q carriers
cw_i = cos(2 * pi * t / points_per_period);
cw_q = sin(2 * pi * t / points_per_period);

% Generate random words (from 0 to 3) sequence
%bit_sequence = randi([0 3],1,bit_count);
bit_sequence = [0 1 2 3 2 0 1 3 2 1 0 3];

% Oversample signal
bit_sequence_oversampled = kron(bit_sequence, ones(1, points_per_period * carrier_periods_per_bit));

% Extract i and q signs
sign_i = 2 * fix(bit_sequence_oversampled / 2) - 1;
sign_q = 2 * mod(bit_sequence_oversampled, 2) - 1;
sign_q = cat(2, sign_q(1:16), sign_q(1:nr_points-16)); %OQPSK-modulatie

% Apply signs to each carrier
i_signal = cw_i .* sign_i;
q_signal = cw_q .* sign_q;

% Sum I- and Q-signal
signal = i_signal + q_signal;
```

```

subplot(6,3,[1,3]);
plot(t, cw_i, 'b', t, cw_q, 'r');
title('I and Q carriers');

subplot(6,3,[4,6]);
plot(bit_sequence_oversampled, 'b');
axis([0 nr_points 0 3]);
title('Coding words')

subplot(6,3,[7,9]);
plot(t, sign_i, 'b', t, i_signal, 'r');
axis([0 nr_points -1.5 1.5]);
title('I Channel');

subplot(6,3,[10,12]);
plot(t, sign_q, 'b', t, q_signal, 'r');
axis([0 nr_points -1.5 1.5]);
title('Q Channel');

subplot(6,3,[13,15]);
plot(signal, 'r');
axis([0 nr_points -2 2]);
title('Resulting QPSK signal');

% Plot FFT
spectrum_i = abs(fft(i_signal));
subplot(6,3,16);
plot(spectrum_i(1:end/2));
title('FFT I-signal')

spectrum_q = abs(fft(q_signal));
subplot(6,3,17);
plot(spectrum_q(1:end/2));
title('FFT Q-signal')

spectrum = abs(fft(signal));
subplot(6,3,18);
plot(spectrum(1:end/2));
title('FFT IQ-signal')
```

PLANETARY GEARS 1

Harry H. Arends

This article complements Peter A. Kuiper's article "My X-Y Rotor", here a planetary gear is used. In this article, I try to clarify the operation of this type of gear. To this end, I cover the following parts, among others:

- The gears
- Gear ratio
- 'Module'

Fig .1 Planetary transmission

Introduction
Before we look at planetary transmission, we first cover some concepts:

- *Epicyclic*
  circle whose centre itself describes a circular orbit.
- *Pitch*
  The length of 1 tooth + hole. The pitch is the number of teeth divided by the circumference of the pitch circle
- *Pitch circle*
  The circle through the centre of the teeth of a gear where the imaginary contact surface between gears is located. (d1 and d2 in fig 4)
- *Module*
  Ratio of pitch-circle diameter to number of teeth (in mm)
- *Planetary gearing*
  Gear set consisting of at least 1 gear set with a carrier, sun wheel and a ring wheel.

A planetary gear is a type of epicyclic gear that can be used where we need precise motion control, such as in our home-built rotors. In general, there are many applications in industries where we need to transmit higher torque in a limited space and we need a lightweight unit to transmit power, motion and torque from drive equipment to driven equipment.

Parts

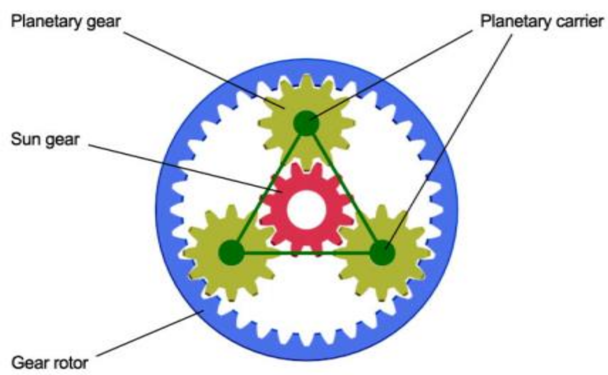


Fig 2. Planetary gearbox components

Planetary gear sets have 4 main components: sun, planet(s), ring and the carrier or carrier. The planets are mounted on the carrier. These planets engage with the sun and the carrier rotates on the same centreline of the sun. The planets also engage the Ring wheel.

Generally, the carrier , the ring or the sun will be fixed in rotation so that there will be predictable motion. In rare cases, the input and output are connected, so there is no relative motion between the sun, ring and planets.

Sizing the gears
It is important to size each of the gears so that they fit properly. There are two simple formulas to help you choose the right gear ratio.

First, all gears must have the same diametrical '*module*' to merge. Next, the next equation will relate the number of teeth on the sun, ring and planet. These variables can also represent the pitch-circle diameter of the gears, but usually one uses the number of teeth to quickly evaluate whether the configuration will work.

$$S + P = \frac{R + S}{2} = A$$

S = Sun, R = Ring, P = Planet, A = Arm(carrier)

For example, if your sun gear needs 17.5 teeth, the above calculation will not work because the number of teeth must always be an integer. This is a little harder to see when using the pitch circle diameter of the sprocket.

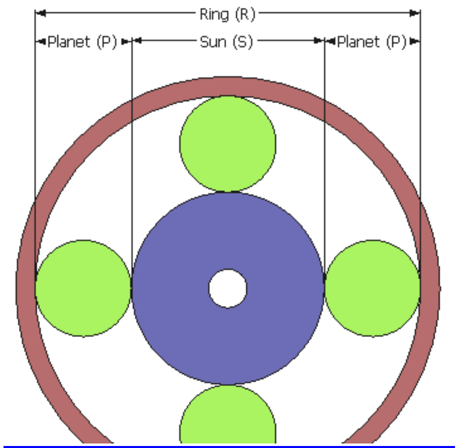


Fig 3. Sun and planet gears

This can be made clearer by imagining "gears" simply rolling (no teeth) and representing an even number of planet gears. From the picture in Fig. 3, you can see that the diameters of the sun gear, plus two planet gears, added together must equal the ring gear size.

Now imagine taking out one of the green planet wheels and rearranging the remaining wheels to be evenly spaced. Nothing changes about the number of teeth or the pitch circle diameter. There is still the same degree of transmission.

The above piece shows that the pitch circle diameter can also be used in the formula but it is better to calculate directly with teeth.

Gear Module

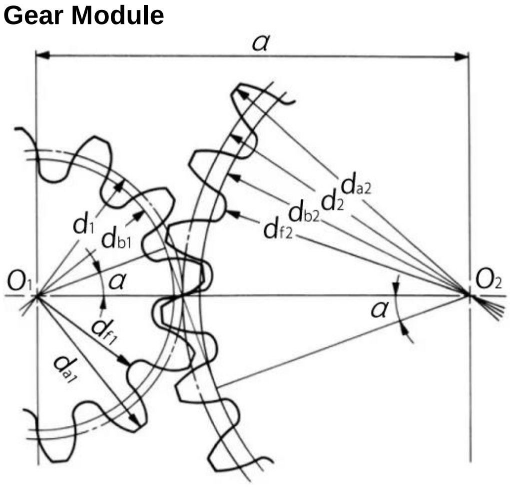


Fig 4.

Parameters of a gear

The pitch (pitch) of a gear is the number of teeth divided by the pitch circle (circumference). Generally, the size of a gear tooth is expressed as its *module*.

The sizes of gear teeth using the module system are indicated by the symbol *m* followed by numbers such as *m1, m2* and *m4*, with tooth sizes increasing as the numerical value increases

module table

Fig 5 Gears with same module

module (m)	stitch (cp)	Diametral Pitch (dp)
0.5	1.57080	50.8
0.6	1.88496	42.33333
0.8	2.51327	31.75
1	3.14159	25.4
1.25	3.92699	20.32
1.5	4.71239	16.93333
2	6.28319	12.70
2.5	7.85398	10.16
3	9.42478	8.46667
4	12.56637	6.35

For the gears to merge, the pitch, (pitch p) must be equal. The pitch of a gear indicates the size of the gear tooth, but since pitch contains the circular constant π (3.1415...), it is not a convenient number. So by dividing the pitch by π (PI), π can be eliminated. The remaining value is given the name *module* and represents the tooth size. (The unit of module is mm.)

Incidentally, the pitch circle diameter of a gear (d) can be obtained by multiplying the module (m) and the number of teeth (z). Expressed as a formula, it is

$$D = M * Z$$

In the ISO 21,200 system, the unit to indicate gear tooth size is specified as *module*. In reality, there are other units such as CP (pitch or circular pitch) and DP (diametral pitch).

The table compares the equivalent values of module (m), pitch (CP) and diametral pitch (DP). The latter is the same as the module, but per inch instead of mm: DP = 25.4/m.

Calculating planetary gear ratios

In this case, the Planet gear has 12 teeth, the Sun gear has 18 and the Ring gear has 42 teeth. So, entered into this formula:

$$R = 2P + S$$

gives 42 = 2 × 12 + 18

The number of planetary teeth is counted twice regardless of the number of planets in the transmission.

Calculating the gear ratio
Calculating the gear ratio of a planetary gear train can be tricky. Let us state the following:

Tr	Ring gear revolutions
Ts	Sun wheel revolutions
Ty	Revolutions of the planetary gear carrier (the Y-shaped thing in the previous picture, the carrier)
R	number of ring gear teeth
S	Sun number of gear teeth
P	number of planetary gear teeth

The number of revolutions then becomes:

$$(R + S) * Ty = R * Tr + S * Ts$$

Example
Now, in a planetary gear, usually one of the gears becomes stationary. For example, if we keep the ring gear in a stationary position, Tr will always be zero. So we can remove those terms from the above formula and we get:

$$(R + S) * Ty = Ts * S$$

So, as we rotate the sun gear, we can rearrange the formula to solve for revolutions of the Y carrier (carrier):

$$T_y = T_s * \frac{S}{R+S}$$

The gear ratio is thus:

S / (R+S)

Restrictions
A planetary transmission can have a lot of freedom in construction. With some limitations in the ratios, the transmission can be more robust.

If you want the planetary gears to be evenly distributed and all engage the next tooth at the same time, then both your Sun and Ring gears must be divisible by the number of planets.

If you want them to be evenly distributed, but they cannot all be in the same phase with respect to their teeth, then the sum of the ring gear teeth and the sun gear teeth must be divisible by the number of planets. That is,(R + S) divided by the number of planets is an integer.

However, if you are willing to place the planets unevenly, this restriction does not apply. However, the angle between the planet's gears relative to the solar wheel is still limited by:

$$Hoek_p = \frac{360}{R+S} * N$$

Where N is an integer. That is, the angle between planetary gears is a multiple of

$$\frac{360}{R+S}$$

Finally
In a subsequent article, I will describe the construction of a 3D printed gear and how it can be used practically. If you have any questions, my e-mail address is known to the editor.

Source:
https://woodgears.ca/gear/planetary.html

SATPY: FILMMAKING AND MORE

Rob Alblas

In the last 3 "de Kunstmaan" of this year, a number of extensions of satpy-related issues have been covered:

- KM1: custom scripts for different satellites [1]
- KM2: a gui used to "fire up" Satpy scripts [2]
- KM3: Automatically retrieve TLE from space track [3].

Besides a few tweaks/improvements, I have now added scripts that can be used to make movies. First the...

Adjustments

There is now a central file 'satpy_settings.py' that contains all configurations, such as:

- position of satpy scripts etc (EMCtools and EMCdata folders)
- viewer for the gui

The OS used (Linux or Windows) is now determined automatically.

The location where TLEs (Two Line Elements) originating from space-track are now always stored in EMCtools/pppconfig (this used to be the location where the script was started). There is also a 'max_tle_age' defined in satpy_settings.py that allows you to specify when a new TLE should be retrieved from space-track.

Furthermore, a number of buttons have been added in the gui:

- allowing multi-passes (merging multiple passes)
- whether or not to use TLEs from Spacetrack

The menus for satellite selection, area and composite have also been modified here. These are now configurable in satpy_settings.py:

Satellite menu

There is a table in satpy_settings.py, e.g:

```
data_loc=[['msg',home+'/srv/...'],
          ['mtg',home+'/srv/...'],
          ['goes',home+'/srv/...'],
```

...

The left column contains the names of the satellites, while the right column contains the location where the corresponding data resides. The variable *home* here is for Windows C:user_name>.

This table is fully customisable, including the names in the left-hand column. The names in the left column appear automatically in the menu, see fig. 1

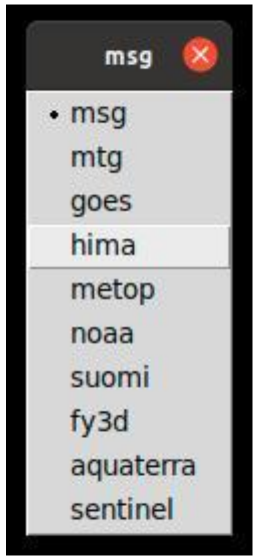


Fig 1. Satellite selection menu

Area selection menu

In satpy, very many areas-on-earth are defined, more than 200, and you can also define areas yourself. All this is a bit too much for a simple menu. Hence this can now be divided into a menu with regions and below that a sub-menu with the areas in that region. This too is configurable in satpy-settings.py:

```
region_list=[['europe' ,( 30, -20),( 70, 40)],
             ['asia' ,( -9, 56),( 51,158)],
             ['africa' ,(-38, -25),( 36, 48)],
             ['N america' ,( 4,-169),( 70,-58)],
             ['S america' ,(-55, -87),( 15,-32)],
             ['Arctica' ,( 65,-180),( 90,180)],
             ['Antarctica',(-90,-180),(-65,180)],
             ['rest' ,(-90,-180),( 90,180)]]
```

On the left the name of the area, on the right the north-west and south-east ends (latitude and longitude respectively). This table is also entirely self-configurable. The menu then looks like Fig. 2.

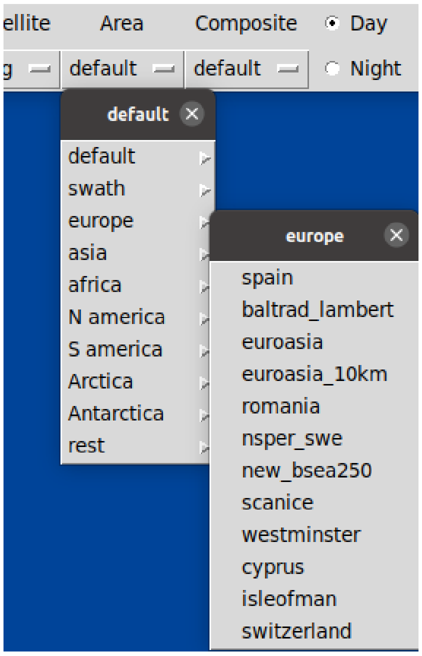


Fig. 2. 'Area' drop-down menu.

There are two added menu options:

- default: then the area defined in the relevant script is used
- swath: no specific area is defined. In the case of GEOs, you get to see the entire globe, possibly upside down! In the case of LEOs, you get to see the orbit, with no day/night or location involved, just as you get to see it when receiving directly. So a narrow and high image.

Composite selection menu

Again, the options are defined in a table in satpy-settings.py, fully configurable to your own requirements:

```
composite_list=[['default', ['default']]
                 ['MSG' ,
 ['natural_color','realistic_colors',
'hrv_clouds',...]],
                 ['Himawa' ,
 ['natural_color','true_color','overview']],
                 ....
```

As a result, see Fig. 3.

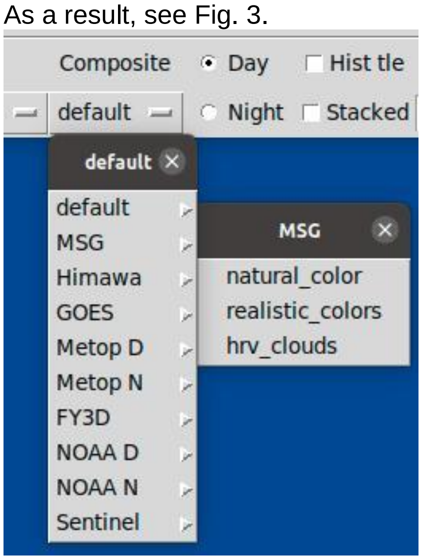


Fig. 3. Composite drop-down menu.

There is room for improvement here, so that only the composites belonging to the selected satellite are always visible. Now you have to select the right satellite in this menu to choose the applicable composites.

Making film.

In Python, it is also quite easy to create a movie from individual jpg files generated by the satpy scripts. I have created some functions that make this easy to do. The first creates a movie from a list of jpeg files:

make_movie(jpg_files,ofile,fps=2,codec='DIVX')

- jpg_files is a list of jpeg files
- ofile is the name of the file where the film will be placed
- fps=number of frames per second, which defaults to 2
- codec indicates the desired encoding, here DIVX by default.

A small Python script still needs to be built around this, eg:

```
jpg_files=glob('*.jpg')
# create list of all jpg files
make_movie(jpg_files,'movie.avi')
# generate fil in 'movie.avi'
```

The second expands a film with a jpeg file, i.e. with a single frame:

extend_movie(ifile,efile,ofile,maxnrfrm=0,fps=0,codec='DIVX')

- ifile: pre-existing file
- efile: name of jpg to be added
- ofile: result. ofile may be equal to ifile.
- maxnrfrm: limit for the expanding film, in frames. 0 means: unlimited. For MSG, with 4 jpgs per hour so 96 per day, make the film maximum 1 day by entering for maxnrfrm=96.

Example:

```
extend_movie('msg.avi','pic.jpg','msg.avi',96)
```

As can be seen, input and output file may be the same. If the file does not already exist then it is created, with 1 frame consisting of the file 'efile'.

These functions have been used in a complete script that completely automatically creates a movie as data comes in: *live_movie_msg.py*

In this file, modify a location *'path_to_data'*, the location where data enters.

A *'watchdog' is* then activated that monitors this location. As soon as a file arrives, action is taken:

- check whether it is the right MSG data
- check whether all files belonging to the relevant satellite and day/time are present (114 for MSG, 44 for MSG-RSS)
- see if the last incoming file is still growing.
- fire up MSGx.py, with the right parameters, out of which comes a jpg file
- use function extend_movie() as described above to create or extend the film

What could possibly be added is deleting used files, and defining a maximum number of frames. This script could then run 'forever', constantly updating the film over e.g. the last 2 days while never filling up the disk space.

This script can be used for MSG ordinary, RSS or IODC, but can also be adapted for e.g. GOES or Himawari.

Satpy with Internet data

Apart from through Eumetcast, Eumetsat data can also be downloaded from the Internet. See "de Kunstmaan" 2021, no. 1 [4]. This data can also be edited with Satpy, with the same capabilities My scripts currently support all MSG satellites and Metop. If required, this can also be done on a Raspberry Pi: downloading data and editing it with Satpy. For Metop, as far as the Pi is concerned, there is a limitation. Unfortunately, the files are very large: almost 500 MByte to download, and after unpacking it is almost 1 GByte! Unfortunately, an ordinary Pi cannot handle that, not even the biggest one with 8 GByte of RAM on board. A possible solution is the Orange Pi [5], which is available with 16 or 32 GByte of RAM. However, I have not tested this.

For installing Satpy on a Raspberry Pi, I did provide some instructions on my website ([6]). To make it completely simple, a complete . iso-file could be made, which you simply put on an SD card and then plug into your Pi. I have yet to figure out how to do that with a compact iso-file, so you don't get a file with the same amount of bytes as the original SD card (e.g. 16 GByte).

Scripts by Ernst

The scripts I created are derivatives of Ernst Lobsiger. I have added/adapted some things that I think simplify use, such as options to make a satellite selection within the same species (such as MSG2,3,4) and choose an area ('area') instead of making a separate script for each area. And I also find the use of space-track to automatically handle TLEs a great improvement. Meanwhile, Ernst has created a new version (v4.0) that will be (or has already been) released soon. This script set is called SPS (Satpy Processing System). Quite a few improvements have been made in this, but no additional options have been added. It did become possible to create a set of composites, instead of generating them one by one; this saves a lot of computer time.

Ernst's scripts are aimed at continuous editing of recent data; then it's a matter of setting it up once and then running it. If you want to be able to 'browse' more, then it gets a lot trickier. Furthermore, Ernst does not plan to support Internet data although that is certainly very easy to fit for MSG. That makes an energy-efficient option more difficult, where you only run a Raspberry Pi continuously to retrieve and edit data. Even for people who cannot receive Eumetcast, there is then no solution. It is up to each to make a choice which scripts to use (they can also be used interchangeably if set up properly). Basically, they are the same; I may still adjust my scripts by starting from the new SPS version.

Latest news: Ernst's new script set is now available, V4.1. Available for download from my web-site: [7] (http://www.alblas.demon.nl/SatPy/index.html)

[1] Weather satellites with Satpy. the Kunstmaan 2022, no. 1, p. 19
[2] A viewer for Satpy. the Kunstmaan 2022, no. 2, p. 4
[3] Positioning satellites with Satpy. the Kunstmaan 2022, no. 3, p. 5
[4] Meteosat without antenna (part 2). the Kunstmaan 2021, no. 1, p. 13
[5] Orange Pi: http://www.orangepi.org/
[6] Satpypi: Satpy on a pi
[7] Version v4.1 SPS of Ernst

MEMBERSHIP MEETING REPORT 19 NOVEMBER 2022

Rob Alblas

1 Opening.
We are in a different classroom again, which is a surprise every time.

Parking is a problem today, it is very crowded. It can be free next to the containers on the side.

The coffee is free, though you have to grab a (used) mug 'somewhere' and wash it, or bring your own. Disposable cups are out of favour.

Due to the renovation, it is not yet clear how things are with Wifi/Ethernet now, classrooms have Ethernet connections, but it is not clear if it is working yet.

2 Adoption of agenda.
No changes.

3 Governance matters.
We still need a librarian and webmaster, need them badly.

4 Satellite status
There will be some changes to Eumetcast, especially for HVS-2. Dish, LNB and frequency do not have to be adapted for basic and HVS-1.
See further for satellite status elsewhere in this KM.

5 Any other business
Job is working on bandpass filters; he calculates them, has them made at OSHPARK and evaluates the result, on the basis of which adjustments are then made to come closer to the desired result. Indicative price: per design of a small PCB about 10 euros, you will always get 3 identical PCBs. It can be done more reliably (right the first time) but then it is also a lot more expensive.

Timo shows a so-called logper antenna (broadband directional antenna), with a range of 0.1 to 11 GHz.

Elmar has found Windows-10/11 drivers for the old opera receiver we used to use for Eumetcast. That can then be used only for TV, not for Eumetcast.

Peter Kuiper has been away for a while, wondering where we are now in terms of reception. So we are working on 8 GHz, especially preamplifiers. The current QPSK receiver cannot be used, it is too narrow-banded. Pluto has a new SDR, which is more broadband: 60 MHz. That could be an alternative to the LimeSDR, which is no longer made.

Peter is now working on an X-Y rotor, with BLDC motors; they are small but very strong. With gears, you can "lift" 10 kg or more. This makes making such a rotor a lot easier.

Rob Alblas modified the receiver control so that it is also usable for the UV916.

Harm de Wit wonders if we can use an existing prefix for QPSK/8 GHz. The problem is limited availability. That's always the problem with construction: what parts can you use that have a somewhat longer lifespan/availability? So that those parts are still available at the time we have something that can be rebuilt.

It would be nice if, with the club's 50th anniversary (2023), we could show something in this area.

A discussion follows on how to proceed. Ben prefers a hardware receiver for the 8GHz.

Job suggests we focus on FY3D; it has a strong signal and many channels. It does have a large bandwidth (45 MHz), but then it should work for satellites with lower bit rates (such as NOAA20, Aqua/Terra) after that.

Rob Alblas is going to look at making a bitstream at FY3D speed (2x30 Mb/s), for the artificial-satellite so we have a test signal.

Harm looks at the HF part.

Rob Hollander revisited the constellation viewer to see if the brightness of the OLED could be increased. Longer/faster integration gives higher brightness, but also a "blotchy" picture. On the module's PCB is a DC-DC converter, which generates 13V from 3.3V. The case is sized to achieve the highest brightness. With a software setting, the brightness can be increased a little more.

6 Closure

Rob Alblas
secretary AI

satellietstatus

Arne van Belle, December 18, 2022

POLAR	APT (MHz)	HRPT (MHz)	X-BAND (MHz)	Remark
NOAA-15	137.620	1702.5		Morning/evening,weak/sync problems
NOAA-18	137.9125	1707.0		Early morning/afternoon
NOAA-19	137.100	1698.0		Afternoon/night
FengYun 3A	no	Off(1704.5)	no	AHRPT 2.80 Msym/s
FengYun 3B	no	Off(1704.5)	no	AHRPT 2.80 Msym/s
FengYun 3C	no	Off(1701.4)	7780	AHRPT 2.60 Msym/s
FengYun 3D	no	no	7820	RHCP 30MS/s QPSK
FengYun 3E	no	no	7860	RHCP 38.4MS/s QPSK
Metop-B	no	1701.3	7800	Only AHRPT 2.33 Msym/s
Metop-C	no	1701.3	7800	Only AHRPT 2.33 Msym/s
METEOR M N2	137,100 LRPT	1700.0		LRPT/MHRPT
METEOR M N2-2	off(137,900 LRPT)	1700.0	8128	LRPT/MHRPT
AQUA			8160	RHCP 7.5 Mbps no FEC
TERRA			8212,5	RHCP 7.5 Mbps no FEC
SUOMI NPP(jpss)			7812	RHCP 15 Mbps
NOAA-20 (jpss-1)			7812	RHCP 15 MHz FEC ½
NOAA-21 (jpss-2)			7812	RHCP
ARKTIKA-M1			7865	RHCP BPSK 30.72MS/s
OCEANSAT-2			8300	RHCP 42.4515 Mbps

GEOSTATIONAIR	LRIT/GRB (MHz)	HRIT/GVAR (MHz)	Orbital position/status
MET-12 (MTG-I1)			Just launched
MET-11 (MSG-4)	no	1695.15 HRIT	0 degree, operational
MET-10	no	1695.15 HRIT	9.5 degree E, RSS
MET-9	no	1695.15 HRIT	45.5° degree E, IODC
GOES-E (no. 16)	1686.6 GRB	1694.1 HRIT	75.2 degree W via Eumetcast
GOES 17	1686.6 GRB	1694.1 HRIT	Drift to 105 degree W storage mode
GOES-W.(no 18)	1686.6 GRB		137.0 degree W via Eumetcast
GOES 14	1691 LRIT	1685.7 GVAR	105 degree W, Backup
GOES 13 / EWS-G1	1676 SD	1685.7 GVAR	61.5 degree E, Now Space Force
GOES 15	1691 LRIT	1685.7 GVAR	128 degree W
Electro-L2	1691 LRIT	1693 HRIT	14.5 Degree W, 7500 MHz & via Eumetcast
Electro-L3	1691 LRIT	?	76 Degree E, Operational
MTSAT-1R	1691 LRIT	1687.1 HRIT	140 degree E, Backup for MTSAT2
MTSAT-2	1691 LRIT	1687.1 HRIT	145 degree E, via Eumetcast
Himawari-8	no LRIT	no HRIT	140.7 degree E, via HimawariCast
Himawari-9	no LRIT	no HRIT	140.7 degree E, Backup for 8
Feng Yun 2G	-	-	99.5 degree E
Feng Yun 2H	-	-	79 degree E
Feng Yun 4A	1697 LRIT	1681 HRIT	99.5 degree E, Operational
Feng Yun 4B	1697 LRIT	1681 HRIT	7500 MHz X-band
SYRACUSE 3B		7705MHz LHCP	Only for test signals 5,2W

On 17 December, it was reported by David Taylor on GeoSubscribers that Meteor-M-N2 LRPT shows a carrier but not QPSK on 137.100 MHz

werkgroep kunstmanen

Founded in 1973, the working group aims to:
Promoting the observation of artificial moons
by visual, radio-frequency and other means

www.kunstmanen.net