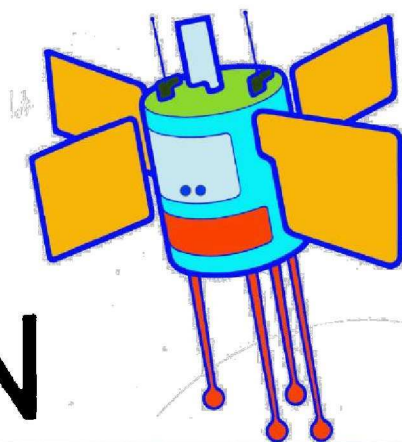


Oktober 2014
41^e jaargang nr. 3



DE KUNSTMAAN



In dit nummer o.a.:
"METEOR-N2
Weersatellieten in Vietnam deel 13
Pulsteller
en nog veel meer.....

Dear member,

This pdf contains some translated articles of our Dutch magazine “De Kunstmaan”.

Google Translate is used; none or very few corrections are done afterwards. Results may be sometimes incorrect or hard to understand, but mostly I think it is usable.

Figures and pictures are partly left out. Please use the “paper” (Dutch) magazine together with these translations.

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

Please check also our web-site, which has now a translated version:

www.kunstmanen.net

(middle of home page, “English version”).

Rob Alblas

werkgroep Kunstmanen

kunstmanen@alblas.demon.nl

Content

	page KM	page in this pdf
From the Chairman	98	4
Report meeting 13 sept.	101	6
Weather satellites in Vietnam (13)	103	8
The new METEOR-satellite	104	9
From the library	109	15
Pulsteller met Arduino	110	16
1000 divider – part 2	117	25
Sketch of display board of the WRX-1700	120	30
First Image	127	41
Satellite status	132	49

From the Chairman

My HRPT receiving system is never finished. Every time you think of something new that you think this makes the system better. After all, it is your hobby and you go back to the hallway. The result of my efforts is described in the article "My HRPT Part 2", the first part was three years ago, how time incredibly fast.

What is beginning to gnaw is that I want to receive LRIT the MSG-3 Although this is transmitted by direct reception EUMETCat is still the best there is. The first results can be read in the article "First Image".

Very nice is that recently a new Russian Meteor was launched. The number of polar weather satellites that we can receive direct now stands at four. Rob describes in his article which is different to the signal from the satellite.

Lately I've been busy with frequency counters and the like. Whereby a stable frequency source very desirable is. I'm glad I was able to make it., With the help of Timo Then you also want to be able to measure high frequencies with the new design of a 1,000-divisor is this possible. The only thing missing was a decent frequency counter. I had a PIC-based counter, but of course it was not accurate enough. I recently owned by a nice (used) HP 53181A counter. Lots of buttons and options that I agreed to go investigate. At the next meeting in November, I will take it with you. Up slowly but surely the measuring equipment in order, time and knowledge is lacking. Will this ever change?

Because it is not set aside to purchase equipment of reputable news for me, I went looking for suppliers of used equipment. So I came to Helmut Singer, not far over the border in Aachen (Germany). Once a month they are open and you can look around what they all have. An indescribable amount of equipment you did not know that it exists. Look at their website and you get a good idea of what they have.

That we're a special association shows that we regular membership requests from get abroad. Now our leaflet in Dutch and therefore not as accessible to our foreign members. Rob and Fred have investigated whether you can translate with online tools from Google or translation features in Word. Simply an article in English This is going amazingly well! For our foreign members we're going to make a digital Kunstmaan which is translated with Google. In the paper Kunstmaan a small summary in English is so you still get a quick impression of what an article is about. No argument for foreigners to join our working group. More non member And we need them, see the efforts of Martin in the First Image article.

Occasionally, you come across a book that you think that I should have. It is the book with the (recruiting for us) entitled "The Great Satellite World Atlas." In this book, the maps are combined with satellite imagery. It's really a beautifully executed book with high resolution images of the Ikonos and Landsat satellites. There are also MODIS and AVHRR images between. What a pity that is

not specified by the satellite they are made. At satellite photos Recordings of NOAA, I could not discover. For the money, you do not have to leave. Original price was 60 Euro, but as with many books now available. 20 Euro at NRC For 1.95 they deliver the atlas of more than three pounds at home.

The turnout at the last meeting was not so high, many were still on holiday or had because of the nice weather other pursuits. Arne gave a fascinating lecture which EUMETSAT is now with the introduction of DVB-S2. Despite being a few months old signal is no longer broadcast the reception technique is not yet fully crystallized. This week (September 17) in Darmstadt a meeting of the test users where the current state of affairs will be discussed. We are curious because this is for the receipt of EUMETCast of great importance.

Let us know what you are doing. Hoogs t likely this is also interesting. Other members We would like to include your story in the Kunstmaan because we reach more members than through the meetings. The next meeting is on 8 November. A week earlier, we stand on the Day for the Radio Amateur in Apeldoorn. See you then!

Report meeting September 13, 2014.

1. Opening

Ben makes a number of announcements:

- If something needs to be scanned from the library than can be passed. To the librarian (Paul Baak) He does it at home and send the results by email.
- Newsletter: The find links in the newsletter do not all work. There it is looked at.
- The newsletter is now sent as a reminder to every meeting. It is indicated by the attendees preferred to do. Shipment 14 days before each meeting
- Handbook: we are working on a new set of articles, the second supplement. All attendees have some interest.

2 ALV Report of May 10, it was published in the July Kunstmaan.

There are no comments or remarks on the record.

3 Adoption agenda

No additions.

4 Administration

There are no changes.

5. Satellite Status (Arne)

See later in this KM.

6 Any other business

Ben: took his PC so attendees HRPT decoder can reprogram the latest decoders.

Peter Smits took his big rotor. New is a foot to set him up. This makes this hefty rotor relatively easy to transport and firmly set. On The rotor is adapted to establish. 2.5m a dish
Ask attendees: Who can weld aluminum? No response.

Elmar: has some things in his "shop".

Oa also some things from the estate of Joseph Jongen, such as a HRPT receiver and decoder.

Ben let his built-sight (complete with HRPT decoder). He also has a GPS time standard.

7 Closure

The next meeting will be held on 8 November.

Lecture: DVB-S2 and EUMETCast by Arne

Brief summary of this presentation.

As known broadcasts EUMETCast via Eurobird-9 with the DVB-S system will stop at the

end of this year. That will be replaced by Eutelsat 10 (10 degrees East), using the DVB-S2 system. At this time, the two systems operate in parallel.

Reason for this change: there is a need for more and more data. Among others, the Sentinel, successor to Envisat, will also be broadcast. Via EUMETCast The bandwidth is so high that it can no longer using the old DVB-S system.

With the new system, it is possible to send as a high resolution (16PSK). Via the same satellite both lower-resolution data (via 8PSK) This will be constantly alternated; the receiver must be able to keep this switch.

For the higher bit rate (16PSK) a larger antenna is needed, but with a smaller antenna, the bits can still be received with a lower bit rate (8PSK) anyway.

Halfway through 2015, the high-resolution portion (HVS through 16PSK) be connected. There are currently being tested for various receivers. It is advisable not to order any receiver; Some receivers offer more problems. One particularly when receiving the mixed high and low resolution goes wrong, with decreasing signal strength (eg by rain.). Then, the receiver is unable to decode, until the conversion 16PSK / 8PSK is fixed to 8PSK; then, the low-resolution signal again usable. In other words, at high signal both high and low resolution can be received with lower signal strength is all gone, still lower signal strength may be low resolution again be decoded. That's not right; with decreasing signal strength must be high-resolution loss, and decoding of the low resolution continue.

Another problem is that on some PCs provide certain recipients problems, and will work on other PCs again.

Conclusion: The conversion of DVB-S to DVB-S2 is more difficult than expected. The final changeover may be delayed for a few months, but basically it's on 1-1-2015 ended with DVB-S.

Rob Alblas
(Secretary AI)

I was this summer especially busy with my motorbike. That is of Vietnamese quality. After extensive considerations, I decided to look ahead to perform. New but mostly a series of adjustments to the existing round. There is a lot of time to sit down but the final result is reasonable. I'll stick to look around but not fanatical.

Through all these activities, there is not much done to the weather satellites. Yet an update.

By Hai, my regular welder is a side piece welded to the existing standard, where the totem pole-rotor fits.

Visitors have taken transistors for the interface to the rotors from the Netherlands. I can make the solder station hot.. The problem is that my old two-channel oscilloscope, I've bought via Marketplace is to the eternal electronic hunting grounds. Only the backlight still works. So when problems in built circuit I have a problem. I have got a scoop, you can connect to a PC in order. They sell them in Vietnam in one case in Hanoi, but then I lost almost double what I pay if I order it in England. Now fingers crossed that he actually gets here. Next time more.

Dimension-4

I have on my PC weather satellites in a while the Chronos program run at regular times, the internal clock is set equal to a whole series of servers. The disadvantage of this freeware program is that frequent the web browser start-up to demonstrate the program website (and recommend the paid version on). I suppressed or again with a script, but remains annoying. Meanwhile I have after a hint on the astro forum.

Dimension-4 downloaded from

<http://www.thinkman.com/dimension4/>

This works perfectly and I immediately got rid of the annoying popup screens. You can also view the history: a graph with defects. For me that well within the 0.5 sec.

Everything

Space and astronomy

2017 test flight SLS / Orion, and in 2021 the first astronauts into space. Nice movie including flight simulation.

http://www.astromart.com/news/news.asp?news_id=1421

Reliable forecasts

On <http://www.astroforum.nl/showthread.php/154537-Waar-is-een-betrouwbare-quot-wolkenverwachting-quot-te-vinden?p=657797#post657797> is found with a lot of links on which site the most reliable cloud is expected to be found an extensive discussion. Who makes a program based on MSG recordings?

The new METEOR satellite.

Rob Alblas

Summary.

This article discusses the data format of the Russian polar satellites Meteor: N1 and N2. Calibration is not covered.

Introduction.

In 2009, Russia has a new polar weather satellite launched the Meteor-M N1. Both frequency band (1700 MHz) and bit rate (665,4kb / s) correspond to the NOAA, so does not need to be adjusted. Receiver

The data format is completely different (AHRPT); this I some time ago the HRPT decoder adjusted. This satellite does not transmit from APT.

Recently a modified version launched the Meteor-M N2. There will be more launched a total of six between 2014 and 2021 (launch year). The satellite transmits the same information as the N1 (of which there is only 1 has been launched), but in a slightly different form. Both decoder and software (program WSAT) had to be adjusted.

An important difference between the METEOR NOAA's hand and on the other is the extra channel for near-infrared that METEOR has extra. In order to maintain the data rate, the horizontal resolution must be reduced: from 2048 to 1540 pixels.

The instrument aboard the NOAA's for the images as a result is the so-called. **Advanced Very High Resolution Radiometer**, or simply **AVHRR**. In METEOR instrument called **multispectral imager scanning radiometer**, or simply MSU-MR.

The table below gives an overview of the various spectral channels from NOAA and METEOR.

The spectral sensitivity of 5 of the 6 channels are similar to those of NOAA; the additional channel is sensitive to near-infrared (around 1.7 microns)

spectral sensitivity	Meteor (N1, N2)	NOAA
0.5-0.7 microns	x	x
0.7-1.1 microns	x	x
1.6-1.8 microns	x	
3.5-4.1 microns	x	x
10.5-11.5 microns	x	x
11.5-12.5 microns	x	x

Table 1: Channels of Meteor and NOAA.

With this additional channel are the three channels in / near the visible spectrum of Meteor consistent with the MSG geostationary satellites. By mapping these channels onto blue, green and red easily fine "false color" plates can be made.

Data Structure METEOR.

The structure of the NOAA data is entirely based on 10 bits per word. All the bits relating to a complete line are in a frame, preceded by a synchronization word. That word is recognized by the decoder, with which the beginning of a line is detected. It is then a question of counting to determine which bit belongs to which channel and x-position. A frame (not more than a set of bits) contains at the NOAA 110900 bits, of which $5 \times 2048 \times 10 = 102400$ bits for the AVHRR channels, the rest is calibration data, the sync word (60 bits) and so on.

In METEOR data structure is very different. This is based on 8-bit words (ie, bytes) with a frame length of only 256 bytes (Meteor-N1) or 1024 bytes (Meteor-N2). This means that there are several frames are needed for a complete line.

The number of bits per pixel is also METEOR 10 Here a special schedule should be used to transmit. Those 10 bits over bytes

The table below shows some differences between NOAA and the METEORS merged.

	NOAA	METEOR N1	METEOR N2
bits per frame	110900	256	1024
frames per line	1	50	12.5
sync words (hex)	284 16F 35C 19D 20F 095	1A CF FC 1D	1A CF FC 1D
X resolution	2048	1540	1572 (chan 1-3). 1568 (chan 4-6.)
number of channels	5	6	6

Table 2 Characteristics of the two NOAA and METEOR's

The N2 has a slightly higher resolution compared to the N1

Structure N1.

The structure of the N1 data for a line (6-channel) is as follows:

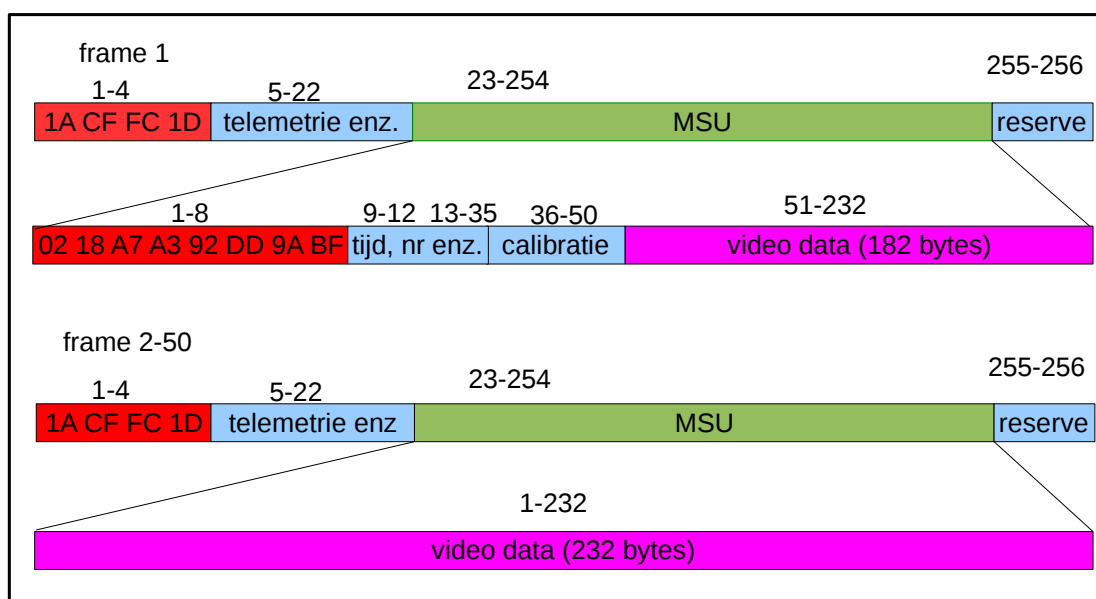


Fig. 1 Structure N1.

The data is divided into 50 frames. The synchronization word 1A 1D FC CF is used by the decoder to recognize. The MHRPT-N1 frames Since there are 50 frames, the first frame of a line to be recognized. Hence, the second 8-byte sync word which is present only in the frame that contains the first MSU. Recognition of this 2nd sync word is handled by the software.

Because of this synchronization word, as well as the calibration data, there is in this frame is less space for the "video data": 182 bytes in the first frame, and 232 bytes in the frames 2 t / m 50 together that $182 + 49 \times 232 = 11550$ bytes.

Per channel that $11550/6 = 1925$ bytes. Since each pixel requires 10 bits, there is place for $1925 / 10 \times 8 = 1540$ pixels, which is exactly equal to the horizontal resolution.

For the assignment of the 10-bit data bytes in the following scheme is used:

byte no.	bit 7	bit 6	bit 5	Bit 4	bit 3	bit 2	1 bit	bit 0
1	ch1 [9]	ch1 [8]	ch1 [7]	ch1 [6]	ch1 [5]	ch1 [4]	ch1 [3]	ch1 [2]
2	ch1 [1]	ch1 [0]	ch1 [9]	ch1 [8]	ch1 [7]	ch1 [6]	ch1 [5]	ch1 [4]
3	ch1 [3]	ch1 [2]	ch1 [1]	ch1 [0]	ch1 [9]	ch1 [8]	ch1 [7]	ch1 [6]
4	ch1 [5]	ch1 [4]	ch1 [3]	ch1 [2]	ch1 [1]	ch1 [0]	ch1 [9]	ch1 [8]
5	ch1 [7]	ch1 [6]	ch1 [5]	ch1 [4]	ch1 [3]	ch1 [2]	ch1 [1]	ch1 [0]

Table 3: Set of 4 pixels of the first channel is stored in 5 bytes.

In five bytes as the 10 bits of four pixels from one channel housed. Here are four pixels in 5 bytes of channel 2, then track channel 3, 4, 5 and 6 after 30 bytes the party begins again for pixels 5 t / m 8.

After 385 of these 30-byte sets all channels transmitted.

Note that the number of bytes in a video frame (182 or 232) is not divisible by 30 This means that a frame does not contain an integer number of such sets.

To sum up, the receiving of the serial bit stream as follows:

- The decoder receives the serial bit stream and looks for the 32-bit synchronization word. With this, both the beginning of a frame as the byte-boundaries in the bitstream are known. This is transferred to the PC via USB.
- The data is put in a file, beginning with the start of a frame. (That is in general not the first frame of a line!)
- Software scans the file the first frame with the 64-bit 2nd sync word.
- Then it's a matter of counting, to fish. Desired data out

Structure N2.

As mentioned, the METEOR N1 a loner. After this, the data format changed, but the basic information is almost equal.

The frame length is increased from 256 to 1024 bytes. For a line 4x less frames are therefore now needed: $50/4 = 12.5$. Was the N1 for a line a whole number of frames needed for N2 there are frames in which data from two consecutive lines are (note the broken number "12.5" on).

What is the reason for the change in N1 to N2, I do not know. The efficiency is higher,

making the X resolution could be from 1540 to 1572 pixels per line. Slightly increased Not really worth it, I guess; it is only 2%.

The structure is more complex; the body can better be interpreted in two strokes.

The MSU-blocks containing video data are to the other frame in a slightly different place. To keep these frames apart they are here 'even' and 'odd' called frames, although the name is not to be taken literally.

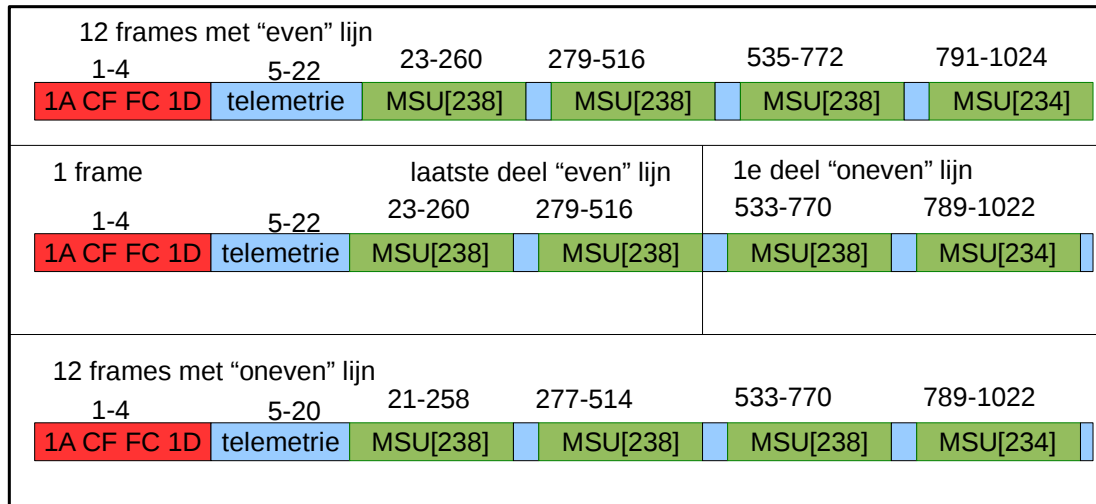


Fig. 2 Structure N2 data.

Altogether 25 frames contain two lines: an even and an odd line. Somewhere in the middle will contain the end of an even line and the beginning of an odd line (or vice versa) a frame .Where that separation is exactly is not fixed; the position of the MSU with the first pixels in a line is also not fixed. In the figure above is that the first MSU in a frame, but it might as well have one of the other MSU's 3 may be.

The first MSU again contains an 8-byte sync word, so, depending on the line at two positions of an MSU can begin; see Fig. 3.

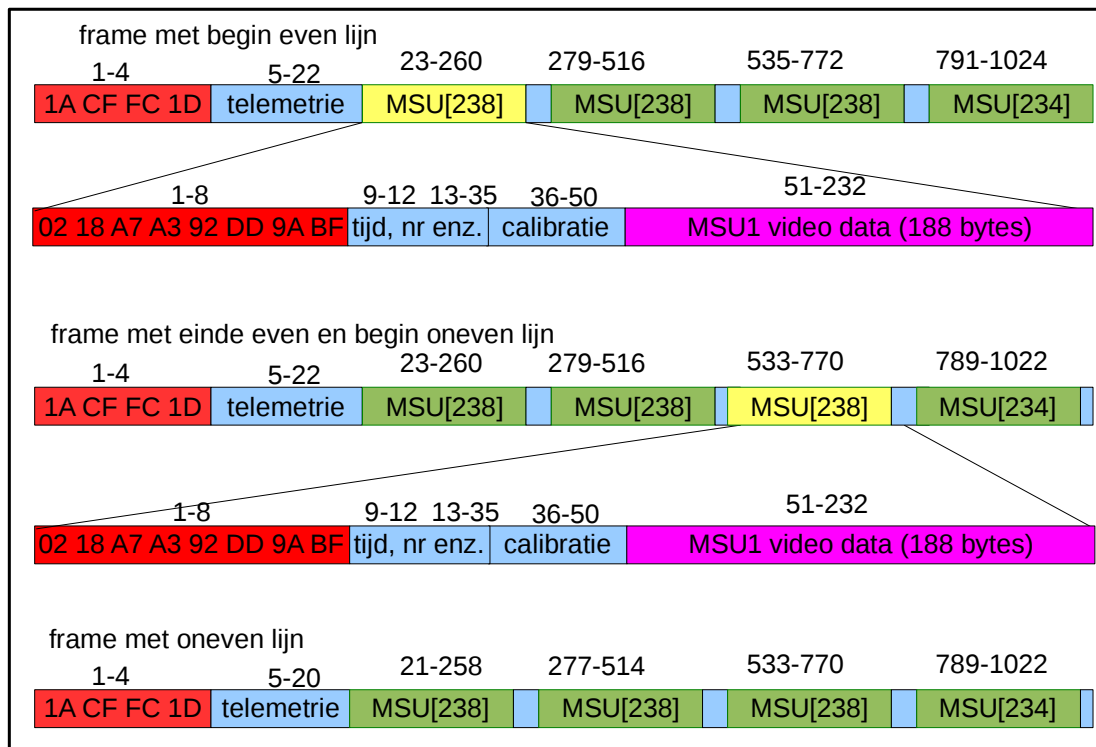


Fig. 3 Even / odd lines in Meteor N2 data.

Note: This is just one example here too! The 1st MSU of a line can be located at each of the 4 positions in a frame, as mentioned earlier.

Also, it can be seen than in the transition "to go to an odd line" in this example, the last 2 MSU's are still at the "even" positions are (starting at byte 23 and 279), after which the first MSU of the odd line is located at position is 533 (instead of 535).

There are a total of eight positions in a frame that can stand the first MSU of a line:

- "Even" line: 23, 279, 535 or 791
- "Odd" line: 21, 277, 533 789

After a 1st MSU follow 49 cubes with the remaining data for a line, spread over several frames. That MSU contain only video data, no time / calibration data. Together contain 50 MSUs 11850 bytes, which corresponds to $11850/6/10 \times 8 = 1580$ pixels per line for each channel. In reality, there are 1572 or 1568 pixels per line, as mentioned earlier.

The extraction of the 6 channels is done in exactly the same manner as in the N1-satellite (see Table 3).

The gathering together of the data of a line, starting from any point in the file received may proceed as follows:

- Find the 8-byte sync word in consecutive MSU's 2 positions.
- If this is found, then it is also known whether it is an "even" or "odd" line is concerned.
- read the 49 following MSU's in, and make a series of 11850 bytes.

- Extract Now the desired channels according to Table 1.

Errors in N2 data

Unfortunately, the received data of the Meteor N2 appears to contain errors. These errors manifest themselves in vertical "interference" lines, at certain positions. This seems to be a mistake in the satellite.

The errors are only in the "odd" lines; if they are beaten on the picture is perfect. Because the errors are in fixed locations, it is possible for them to "correct", that is to say, be replaced by a good adjacent pixel. Therefore, it is still possible to have (with a number of X-positions, the loss of horizontal resolution) the full vertical resolution. The "WSAT" program, which now also can handle the Meteor data, it contains correction facility. By default, they are disabled; possible the error is recovered, or have future Meteors not this error. In Preferences you can choose between:

- No correction (now vertical jitter)
- Only use the 'even' lines (half vertical resolution)
- The faulty pixels in odd lines replaced by neighboring good pixels

Incidentally, this gets the name "odd / even" still a bit of meaning. 'Odd' in English 'odd', which also means "strange," "weird". Hence ...

Literature

The data structure of Meteor-N1 is described on the following web site:

http://planet.iitp.ru/english/spacecraft/meteor_m_n1_structure_eng.htm

Autumn has begun. The parliament is back at work and your librarian too. Unfortunately there is no najaarsGeo, but the Space of NVR makes up for it. Space Weather, aktueel we almost had a beautiful auroras in our country and here we find a nice description of space weather and a mission proposal. All beyond our reach, we can no satellite near Venus observe. An article on Dr. Ockels. He is often recalled as the inspiration with emotion and words; Here we find a technical framework for what he did and then still more lives. To me.

How do propulsion rockets. I've sometimes wondered how those things work, performing delicate maneuvers. That can be difficult with a big engine. Now, here's the explanation: cold gas systems supplied by Moog Bradford from our Brabant. It works with nitrogen and xenon under high pressure and then wegsputten very small quantities. Not yet milligram per second. This is going to be used when landing on the Rosetta comet Churyumov / Gersimenko. It should be very careful otherwise you're right back also on the way back.

I was a year or so ago whether it once that is done to the space debris around the Earth. I am served here on my beck and call. A whole project called e.Deorbit is set up by ESA to capture the whole thing and send. Further down or up Touching is losing they already know from experience, so they are already thinking magnets, harpoons, nets and more. The spacecraft that will do is itself not small, because the waste, sometimes complete tanks, is not a light material. One is not out yet, but it has to be done. The great apostasy comes sometime unscheduled back and has a chance to survive. Atmosphere Despite gasoline prices, I do not tank a cubic meter in my garden.

Finally, 6 pages with Space Chronicle, full launches, rendezvous and more. Fun for Arne. It strikes me that there are countries engaged in satellite operations which I think you have not got better things to do. But who I am. Nice are a few photos: a satellite that makes a selfie (also there), a satellite that another down into the lunar orbit photographing (LRO peeping Ladee) and from ISS: a picture of the re-entry of Progress M22M. Understood under the ISS.

In Electron July a sequel to the article from April about SDR (not be confused with the LIMA project I mentioned last Kunstmaan). Samples and filters. Too complicated to summarize. Words in a pair of That'll work with the monitor for 10GHz spectrum: a pair of 555's and a tuner can. Surprisingly simple. Just keep in mind if we have 6 GHZ go. 5,

The Electron of August: a method for measuring antennas based on solar noise. You do need a chain. Amplifiers It is always a difficult technical issue, even in the execution. That includes SDR: even in this episode yet another article. SDR agree still very large.

Then as the day of the Radio Amateur: go there parking fees are levied. 3 pasteurized. You have been warned. As a fundamental miser I'll go take a walk.

Pulse counter with Arduino

My rotor control consists of the 8052 board with the rotor driver designed by Harrie. This design is based on DC motors in which the position of the antenna over the pulse generators is fed back. As a check on the antenna position I separate scalars used. For a correct position determination, you may not lose pulses or interference received too many pulses. My pulse counter is an old design from the Elektor built from discrete components.

Because the pulse counters a good indication that the rotor rotates I want to build into the final housing of the rotor control. However, they are quite large, a printout of 8 x 7 cm and does not fit behind the front panel.

Time to see if a Arduino with an alternative to the pulse counters can be made. My current counters are equipped with LED displays and this shows clearly good. I want to keep it.

Without special equipment costs to control four LED displays seven segment quite a few connections to the Arduino. Fortunately, there are LED driver chips through multiplexing up to 64 LEDs at once, there can control time: regular are LEDs on and off put. This happens so fast that they seem to burn, constantly in reality Flashing.

Maxim is a well-known supplier of the MAX7219 and the MAX7221 LED driver chip. In Reichelt cost them chips between five and six Euro. At Mouser they cost around ten Euro. If you look on eBay than you buy four Euro a full kit, incl. LED displays, print and the MAX7219. I have ordered a kit, my experiences are another story. Given this wide price you read on the Internet stories over the authenticity of the MAX7219, in spite they seem to be working. [1]

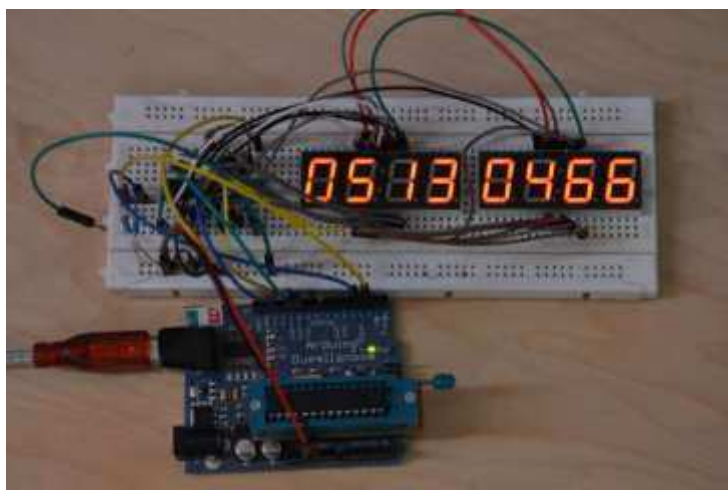


Figure 1 The pulse counter on a breadboard

Difference between the MAX7219 and the MAX7221

The MAX7219 and MAX7221 are very similar chips with only minor differences.

Connectors of the chips are the same. For the sketch described in this article does not

matter what chip you use. Thus the MAX7221 supports standard SPI protocol, the MAX7219 deviates little from it. For the Arduino This does not matter. In addition, the MAX7221 less electromagnetic interference. For some applications, this is of importance. Because the LED outputs are nice to chatter this provides a lot of pollution. Do not use this type of display therefore receivers. The MAX7221 is more expensive than the MAX7219.

Note: Only Common Cathode type displays can be used!

Libraries

For the MAX7219 and the MAX7221 is a special Arduino library "ledcontrol" written. For this pulse counter, it is not necessary to use the library. We only use the standard SPI library to communicate. The MAX7219

How does the counter work

The pulse counter operates on the basis of interrupt UPTs. I will try to explain without getting too technical to its operations.

The Arduino has to perform many tasks in the pulse counter:

- Recognize when a new pulse arrives.
- Determine whether the signal is a pulse of the X or Y-rotor.
- The corresponding counter count.
- Send Codes make it to the MAX7219.

The Arduino However, only carry one thing at a time. This is in contrast to an FPGA that can perform. Multiple tasks at the same time

The normal way to see if a pulse is received is to give a command by the Arduino: read the value of the digital pin 5. By continuing to do this and to check whether the value has changed to get there back or a pulse is received.

However, the Arduino can only do one thing at a time, when the Arduino data to the MAX7219 to sending is can not see if there is a pulse has arrived. So You miss pulses.

For this problem exists a nice solution. interrupts not only the Arduino but the processors in our computer use interrupts. The Arduino Due, two interrupts inputs are available: 0 interrupt on pin 2 and 1 interrupt on pin 3.

In your sketch you define interrupt UPTs by saying. If a certain pin a value changes, then do X. Then the Arduino will be task where it is doing stop immediately and again get started with the handling of the interrupt request. Is the interrupt request handled then the Arduino continue where it has been. Doing

What happens if two interrupt requests at a time is coming in time? In our case, there is an interrupt from the X rotor inside and at the same time of the Y-rotor. This can cause unpredictable situations, you might be missing pulses. We have, moreover, have a fraction of a millionth of a second in which this should happen. It is therefore important that the interrupt - routine is as short as possible, with minimal instructions.

The setup loop, you see the following back, I describe a nly the X-rotor:

```
attachInterrupt (0, xCount, RISING);
```

This means that Interrupt 0, which sits on pin 2, the subroutine xCount will start when there is a transition from 0 to 1, so we look at the rising edge of the pulse.

Below is the subroutine xCount:

```
void xCount ()
{
  xcounter ++;
}
```

This routine increases xcounter the variable by one. Shorter than this value can not.

Initialization of the MAX7219

The commands the direction of the MAX7219 be carried out with the subroutine sendCommand:

```
void sendCommand (int command, int value)
{
  digitalWrite (slave select, LOW); // By chip select pin
  10
                                     (Slave select) to convert low
  SPI.transfer (command); Send // registry
  SPI.transfer (value); // Send the data
  digitalWrite (slave select, HIGH); // Enter chip free again
}
```

16 bits are actually written in the various registers. See Table 2 in the datasheet [2] for the different registers. The entire operation of the MAX7219 is controlled table from this registry.

Before we can send numbers to the MAX7219 chip needs to be initialized. The information can be found on page 7. In the datasheet

I could not figure out, and have not tried it, or the order of the commands difference makes.

```
sendCommand (12.1); // Normal mode (default is
shutdown
                                     mode);
```

Register 12 is described in Table 3., The value "1" means normal operation and "0" is the shutdown mode.

```
sendCommand (15.0); // Display test
```

Register 15, the display test registry and is described in Table 10. If 'one' then all elements of.

```
sendCommand (10.8); // A mean brightness (0-15)
```

The clarity with which the displays are shown egmenten state in register 10, the value 8 is located in the middle. Brightness is determined by this register, but also by the resistance at pin 18 The value in this register determines the pulse-width control, the resistance value determines the current (see later in this article).

```
s endCommand (11.7); // 7221 digit scan limit command
```

In register 11 is the number of digits that the MAX7219 t get triggered. The value 7 means eight pieces. In table 8 it further toegelich t.

```
sendCommand (9,255); // Decode command, standard 7-  
segment
```

displays

Table 4 n ste options for decoding s - register 9. State here the value "00" then each segment can be controlled individually. The value 'FF' eight 7-segment display can be controlled. Table 5 contains a description of the font.

Showing a number

To display a number I made a separate function. Functions are useful because you can call. z e in your sketch multiple times They also make sure that you sketch clutter. Below is the function:

```
// Function to display four numbers on the 7-segment display  
void display number (int position, int number)  
{  
  for (int i = 0; i <numberOfDigits i ++)  
  {  
    byte character = number% 10;  
      sendCommand (position - i, character);  
      number = number / 10;
```

```
}  
}
```

The first line starts with "void", this means that the function is not where the returns. We do give it two values: the position and number. The total display consists of eight 7-segment displays. With the position you determine where the number (to the left) to be displayed. During the pulse counter this is the value 8 (all times right) or 4.

Then we come in a for-loop right. In this loop are different instructions several e times (four times) run.

In particular, the % sign in the instruction "character = number % 10;" needs to be clarified, it is called the modulo d. When two whole numbers by shared together then gives the rest to the modulo. Some examples:

```
x = 7 % 5; // X 2
```

```
x = 59 % 5; // X contains four
```

```
x = 5 % 5; // X contains 0
```

```
x = 45 % 5; // X contains four
```

In our example, we have 4325 pulses $x = 4325 \text{ modulo } 10 = 5$ will contain. This is the most right-hand number.

On the next line, we will send the number to the MAX7219. Within the registry are positions 1 to 8 reserved for the eight 7-segment displays.

On the last line we divide the number by 10 (the next time the for-next loop) to the next rightmost number to processor; n.

The main loop

We're almost there. In the main program (in Arduino "loop" ending) does not happen so much. We have the following code:

```
display number (4, xcounter);  
if (xcounter > MaxCount)  
{  
  xcounter = 0;  
}
```

The first line sends the number of pulses to the display, in this case at the fourth position. The part below if ... checks if the counter does not reach the maximum (9999). If so, then the counter to 0 gets it. If not do this it getting your very strange characters on your display.

Program

The chart below even the overall sketch. The basis of the sketch comes from the Arduino Cookbook [3].

```
/*
XY pulse counter with the MAX7219 / MAX7219

Connections
Arduino digital 2 -> Pulses for X-rotor, use a pull / up resistor
Arduino digital 3 -> pulses from the X-rotor, using a pull / up resistor
Arduino Digital 10 -> Pin 12 MAX7221
Arduino Digital 11 -> Pin 1 MAX7221
Arduino Digital 13 -> Pin 13 MAX7221
*/
#include <SPI.h>
const int slave select = 10; // Select the MAX7219
const int numberOfDigits = 4; // Number of numbers per rotor
MaxCount const int = 9999;

xcounter int = 0;
ycounter int = 0;

void setup ()
{
  // Initialize the MAX7219 via SPI
  SPI.begin ();
  pinMode (slave select, OUTPUT);
  digitalWrite (slave select, LOW);

  // The MAX7219 must show, 7-segment data, see also the data sheet
  sendCommand (12.1); // Normal mode (default is shutdown mode);
  sendCommand (15.0); // Display test
  sendCommand (10.8); // Gemiddeldse brightness (0-15)
  sendCommand (11.7); // 7221 digit scan limit command
  sendCommand (9,255); // Decode command, standard 7-segment displays

  digitalWrite (slave select, HIGH);

  // Set the Interrupts
  attachInterrupt (0, xCount, RISING);
  attachInterrupt (1 yCount, RISING);
}

void xCount () {
  xcounter ++;
}

void yCount () {
```

```

        ycounter++;
    }

// Function to display four numbers on the 7-segment display
void display number (int position, int number)
{
    for (int i = 0; i <numberOfDigits i++)
    {
        byte character = number% 10; // Determine the rightmost number
            // Send digit number as command, first digit is one command
        sendCommand (position - i, character);
        number = number / 10;
    }
}

void sendCommand (int command, int value)
{
    digitalWrite (slave select, LOW); // By chip select pin
    10
                                     (Slave select) to convert low

    SPI.transfer (command); Send // registry
    SPI.transfer (value); // Send the data
    digitalWrite (slave select, HIGH); // Enter chip free again
}

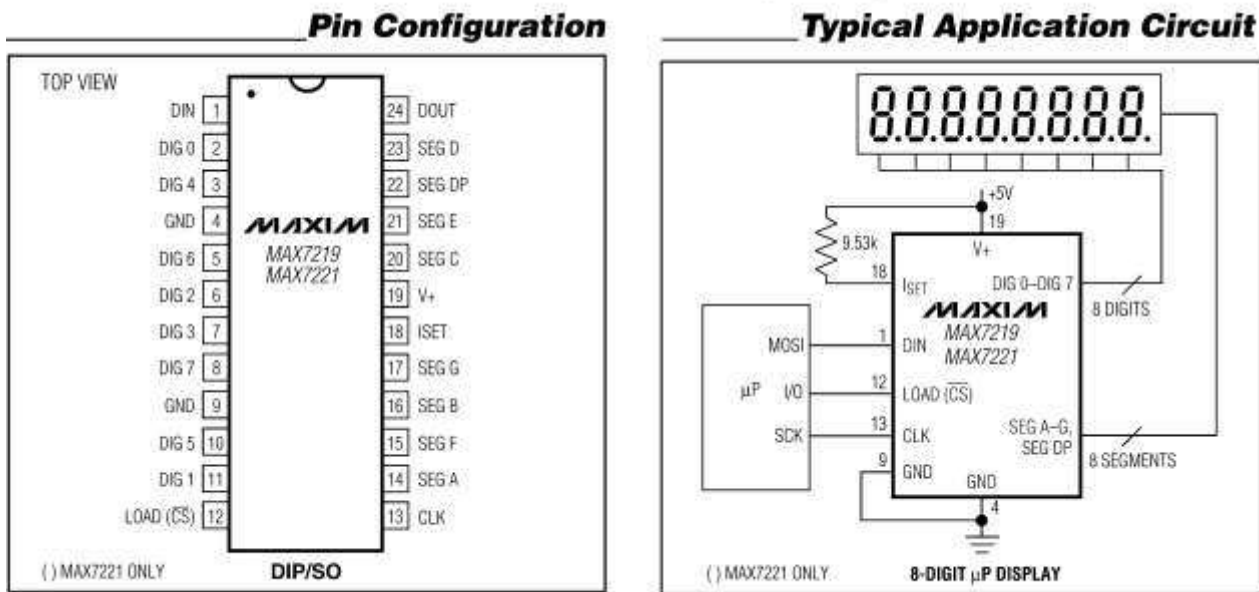
void loop ()
{
    display number (4, xcounter);
    if (xcounter> MaxCount)
    {
        xcounter = 0;
    }

    display number (8, ycounter);
    if (ycounter> MaxCount)
    {
        ycounter = 0;
    }
}

```

Connections / schedule

The connections are listed in the sketch above. O p itself is not so exciting. T interrupt inputs must not hang loose. Depending on the circuit in which the numerator is included must be up or pull-down resistor and be used so reliable counts t e getting pull .



SPI and QSPI are trademarks of Motorola Inc. MICROWIRE is a trademark of National Semiconductor Corp.

Figure 2 From the datasheet

Outlining a schedule is not necessary if you look at the picture above.

The sketch shows the pins 1, 12 and 13 already appointed. In addition, all the pins with "DIG" begin to Common Cathode (!!) of the 7-segment display. Everything starts with "SEG" goes to all the relevant segments of the displays. Remains GND and +5 V on pin 9 to pin 19 By DOUT pin 24, we do nothing. This pin is only used when you want to switch. Multiple MAX7219 series

Finally we have 18 pin, ISET. I with a resistance of 33K to put 5V. With this resistance, the maximum current in the segments arranged. According to the data sheet indicates a resistance of 9.53 KOhm a current of 40mA. Depending on the 7-segment displays this value should be adjusted.

I also have a 10uF electrolytic capacitor in the power line and even a 100nF capacitor included.

Concluding remarks

In this article, I am a bit deeper into the program code of the Arduino. Hopefully Tue a thing or clarified. I have no library used because to make those things. Ondoorzichtelijk In the sketch shown, you can instantly create a link to the datasheet, this is very handy if you want to customize yourself.



Figure 3 Module eBay that has yet to be felt to the test.

Soon I hope the readymade module to receive from China and I'm going to see if it is doing just as well as my own b n (r) ouwsel. If there are problems with it, I now know how it should work. The idea to design a PCB itself etchings, etc. I soon dropped, but given the price of this kit. There will be fixed in any case, no schedule to be delivered, so the effort for this article is not for nothing.

Peter has some testing with my scaler done. It seems like there is also entering false pulses. May lie in the wiring the cause. Maybe add a capacitor to suppress. Spikes To be continued

Links / references

[1] Tronixstuff

<http://tronixstuff.com/2013/05/16/the-MAX7219-LED-display-controller-real-or-fake/>

[2] Data sheet MAX7219

<http://www.maximintegrated.com/en/products/power/display-power-control/MAX7219.html>

[3] Arduino Cookbook, recipe 7:12 " Driving Multi Digit, 7-Segment LED Displays Using MAX7221 Shift Registers "

Divider 1000 - part 2

Summary

This article describes a 1000 s-divider. This divider can be used to measure frequencies up to 7 GHz. All which resulted in a signal s or 7 MHz, this can be measured with almost every counter. For higher accuracy the divider can be usefully configured as a 100-divider. This design uses the ADF4107 from Analog Devices and a PIC12F675.

Introduction

In December 2013 Kunstmaan [3] I wrote a story about a 1000 divider with a CPLD. The design described has a 128-divider followed by a 7.8125-divider in a CPLD. The nice thing about this design is the denominator in the CPLD. The design also has some drawbacks: it is a relatively large design with two prints and is "only" to 3 GHz. In addition, the power consumption is fairly high and has the 128-divider poorly available.

In this article I also mentioned an alternative based on the ADF4107, a PLL Analog Devices. This summer I started working on this and the result is described in this article.

As a starting point, I design BG1CEO taken [1] in which I have applied. Several improvements

The idea behind the design is pretty simple. The most important part is the ADF4107, this is a PLL IC from Analog Devices. This IC has an upper limit of 7 GHz, unfortunately I can not test this, and a sensitivity of more than -20 dB.

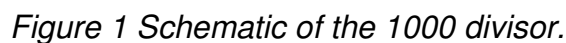
The PLL is composed among other things, of an N-divider, an R-divider and a phase detector. In particular, the N-divider is of interest because it can be taken outside. Both the N and R divider can be programmed by the user. Among the ADF4107 we only use N-divider. The PLL function, phase detector and the R-divisor, we do not. Use

A 1000-divisor is useful because of its easy reading: reading for MHz GHz and you have the measured value. The disadvantage is that you lose accuracy. If higher accuracy is desired, the divisor can also be set. Jumper with a 100-divisor

The schematic

In itself it is not all that exciting. The control of the ADF4107 is built according to the datasheet. Instead of the Atmel microcontroller, which also requires an external

We also see in the diagram a flip flop type 7474. This is required for the needle-shaped pulses from the ADF4107 to a square wave to turn. This is because some counters might have problems with counting needle-shaped pulses. The use of a flipflop stands for an additional two divider. The consequence is that the ADF4107 by 500 50, respectively, must divide.



In Kicad I made the schematic and PCB design. The 50 Ohm input is designed for a print thickness of 0.8mm. I do not know. Whether this is so critical

The PCB is good self-etching. For the fans, I have a movie available.

Software

The design of the software is similar to the software for the 1.7 GHz signal generator [2]. I literally follow the datasheet of the ADF4107, page 17, with the "Initialization Latch Method".

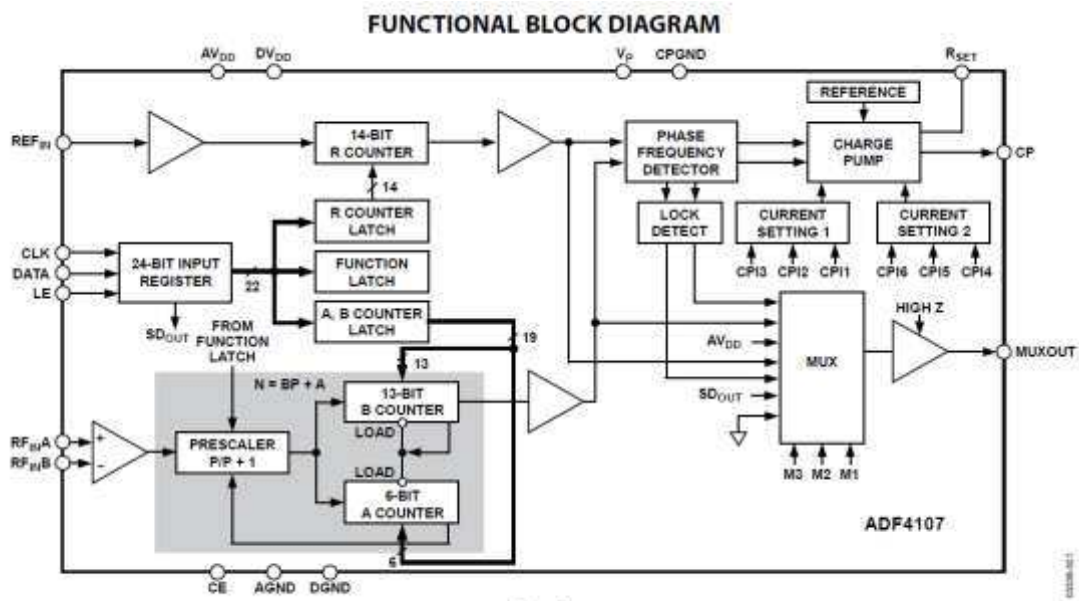


Figure 1.

Fig 2 Functional scheme of the ADF4107

In the Function register you can set the prescaler to 8/9, 16/17, 32/33 or 64/65. I set to 16/17.

Do you want a 500-denominator then the B-Counter at $(500/16) = 31$ stand. The A counter is then on $(500 - (16 * 31)) = 4$.

For the 50-divider, the B-counter $(50/16) = 3$, and the A-counter $(50 - (16 * 3)) = 2$.

The program covers a total of 211 lines of code. Below is a small part showing:

```
// 2 Program initialization latch (11 in two LSBs).
GPIO.F2 = 0;

GPIO.F1 = 0;
GPIO.F0 = 1; GPIO.F0 = 0; // Clock P2
GPIO.F1 = 1;
GPIO.F0 = 1; GPIO.F0 = 0; // Clock P1
```

What the PIC actually does is generate a serial bit stream for controlling the ADF4107. The serial bit stream is on GPIO.F1 and app with GPIO.F0 clocked.

When GPIO.F2 goes down, data can be clocked. Then on the data line (GPIO.F1) and put a zero with the line:

" GPIO.F0 = 1; GPIO.F0 = 0 ", a clock pulse designed to clocks. The zero inwards

Thereafter, the data line is put at 1 and then again a clock pulse to clock. The 1 inward, this way of communication is called bit-banging. Advantage is which is not a difficult libraries are needed why does this on almost all microcontrollers.

I can make the whole program on request. It is too long to print here.

Construction

Make the PCB first tailored for a tin box of 36 x 72 mm. Drill for the doo r n metaliseri gene holes of 0.7mm and solder wires on both sides of the print set.

The most difficult part to solder is the ADF4107, which has a distance between pins of 0,65mm. With a fine soldering iron and a small amount of flux it has been possible for me to soldering. This chip T 74VHC74 is in comparison to the ADF4107 ridiculously large. Soldering the remaining 0805 resistors and capacitors should be no problem.

The 12F675 should be placed because the socket at the top of the board must be soldered. In a socket with turned contacts Pin 8 must also be soldered in place. To the ground plane The remaining holes for the PIC must be removed so as to cause. Not short with a 3 mm drill bits Shoot to clean this up not by the board. This cleanup must occur at the jumper and the power connector.

Everything works well, then the PCB in the tin box can be soldered. At the entrance is a SMA connector and the output is a BNC male chassi s part. The idea is that the divider 1000 is connected. Directly to the frequency counter

Through a 50 Ohm cable with SMA connector, go to the object being measured. Few extra length is not a problem.

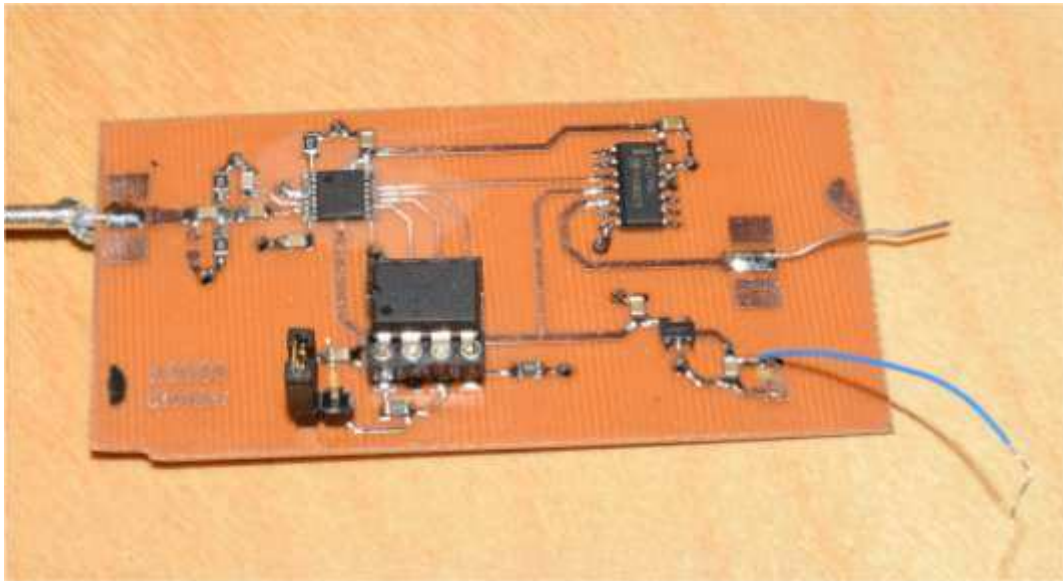


Figure 3 The actual print

Links

[1] http://www.qsl.net/bg1ceo/Tech_topic/rf/7G_prescaler/prescaler.html

[2] 1000 divisor with a CPLD

Kunstmaan december 2013, Page 149

[3] Measurement Transmitter for 1698 MHz

Kunstmaan September 2013, Page 110

Sketch of the display board of the WRX-1700

Summary

This article describes the Arduino sketch of the Display Print of the WRX-1700. The sketch Provides the I2C signals for the UV916 and controlling the LCD display.

Introduction

In all preparations to make a complete document for the manual, I learned that no description is made of the Arduino sketch for the display board. This is because the first version of the display board is made for the PIC 16F684. Later I because of the popularity of the Arduino decided to use it.

An important consideration was the ease with which you can get some soon. Done In contrast, some amateurs who regard it as a great challenge to see to write that occupies as little space as possible a program in assembler (as with large computers plays memory deficit in microcontrollers). Here I have no problems, in addition to programming a display board to resolve in the design of the WRX-1700. Enough problems So the easier it is the better.

As it turned out, the 16F684 still rather small for what it was all necessary. With the Arduino, this was a lot easier. A small disadvantage is that the chip is larger and requires an external crystal.

The display board has a number of functions, for which we need the Arduino:

- With a rotary switch to select a receive frequency and can display on a display.
- Generating the I2C signals to UV916 on the receiver board.
- Showing the signal strength.
- Stopping the oscillator of the HRPT demodulator when APT is received.

For the other functions: controlling LEDs, squelch, volume, we have the Arduino not required. In the future, it is also possible to go. Started with a rotary encoder, for example. Tuning, Here we have again called the Arduino needed.

For a description of the hardware of the display board, I refer to the Kunstmaan [1].

In this article I will describe the sketch and I have walked in the writing of this. Zoal against it When you write a sketch, it is strongly recommended to include. Many comments This writing the comment you doing in the sketch itself. Text beginning with `///
"///" is seen as a comment. Comments write not only for others to whom you want the`

sketch show or want to develop further, but also for yourself. When you take a look a few months later to your sketch then it's really useful if you can read why you did something.

As a rule, comment written in English. When the commentary is in English, you can also share your sketch internationally. I write this article because some members of the English language is a stumbling block to see what happens in a sketch.

Description of the sketch

```
#include <Wire.h>

#include <LiquidCrystal.h>

#define APT 13

#define LOGOUT 0

#define TUNING SWITCH 1
```

After some comments that I did not show begins here at line 28, the sketch with reading in the libraries. The Wire.h library is needed for I2C and LiquidCrystal.h for control (not via I2C) of the LCD display.

On the next line with the "#define" the value 13 attached to the word APT. Wherever the word APT prevents reads the Arduino 13 Nothing more than that, we here say nothing about inputs or outputs.

APT is an output on the display board to turn the receiver on APT received your state. HRPT oscillator at a fixed frequency LOGOUT is an input for signal level and TUNER SWITCH is the rotary switch for the alignment that works with a voltage divider.

```
aantalSamplesLogout const int = 10;

int i, posTuneSwitch0, posTuneSwitch1, totLogout, gemLogout0, gemLogout1;

int samplesLogout [aantalSamplesLogout];

unsigned int band, div_lsb, div_msb;

String freqmessage;
```

Next we need to identify all the variables with which we do something. A variable is actually a small piece of memory in the Arduino where we can store anything. Because the Arduino is a small processor variables, we try to use as little as possible of memory

a type. For example, the two int bytes. The int may take the "unsigned int" between - 32768 and +32,767, between 0 and 65,535 a value.

```
D4 D5 D6 // RS E D7

Liquid Crystal lcd0 (12, 11, 2, 3, 4, 5);

void setup ()
{
    Wire.begin ();

    Serial.begin (57600);

    pinMode (APT, OUTPUT);

    delay (1000);

    lcd0.begin (2.16);

    lcd0.setCursor (0, 1);

    lcd0.print ("S-meter");

    i = 0;

    for (int j = 0; j <aantalSamplesLogout j ++ )
    {
        samplesLogout [j] = 0;
    }
}
```

Then we will create the object lcd0 type Liquid Crystal. To an object we can assign functions and features.

Setup routine

With the line "void setup" we begin a subroutine. A subroutine is a piece of code that one or several times can be performed. In this way we can keep the sketch clear.

The setup routine is typically for the Arduino: it runs at startup the Arduino only once. After the execution of the subroutine, the Arduino further to the main program "loop" and remains therein. At this point the Arduino differs from the programming language C.

What happens in the setup? First, the Wire library turned on, then slide we indicate that we want to communicate. Using the serial port In this sketch we do not do this, but when troubleshooting it is useful to be able to send to see what's going on. Data through the serial port to the outside

We then indicate whether the input or output ports are. pinMode relates to the digital pins of the Arduino. The instruction delay means that we are waiting to give to round off. Initialization Arduino time

Showing the signal strength I do to pass on. The average of ten samples This will display quieter because the numbers are not constantly changing. The samples are in a set (array) put which we will later count. The last four rules ensure that the range of values to zero are put.

The for loop is a common structure and consists of three parts: the beginning, where variable j is put to 0, then the condition which must be met in order to perform the loop and finally the multiplication factor. j ++ means that the variable j is incremented by one. The for loop is thus aborted if j reaches the value 10.

Subroutine for UV916

Thereafter, one of the main subroutines: controlling the UV916 via I2C. In the comments of the subroutine is how to calculate the number of steps for the PLL is.

```
void sent2uv916 ()
{
    / *
    Down Converter is minus 1557 MHz
    Calculation of number of steps:
    N = (Freq_rec + 36.45 MHz) * 16
    2783 (.2) = (137.5 + 36.45) * 16
    * /

    delay (100);

    Wire.beginTransaction (97); // Address
    Wire.write (div_msb); // Delermsb
    Wire.write (div_lsb); // Delerlsb
    Wire.write (0x8E); // Charge Pump
    Wire.write (band); // Band
```

```
Wire.endTransmission ();
}
```

The control of the UV916 tuner goes through I2C. In Kunstmaan from June 2008 is all one and the other described.

A total of 5 bytes must be written to the UV916:

	MSB							LSB
Address	1	1	0	0	0	MA1	MA0	0
Part fact. MSByte (div_msb)	0	N14	N13	N12	N11	N10	N9	N8
Part fact. LSByte (div_lsb)	N7	N6	N5	N4	N3	N2	N1	N0
Charge-pump and test bits	1	0	0	0	1	1	1	OS
Output ports & control bits	P7	P6	P5	P4	P3	P2	P1	P0

The address for the UV916 0xC2 (hex C2), according to the datasheet this is always a valid address. MA1 and MA0 have the values 0 and 1 in the I2C addressing most rightmost determines whether it is a write or read operation. In our case, we write to the UV916 and there is a 0 in Arduino, this last bit is not included in the address, but in the beginning of the command transmission. The address is binary "1 1 0 0 0 0 1", which is decimal 97.

Subsequently, the part number is to be determined. The variables *div_msb* and *lsb div_* are determined in the main loop and the two bytes which determine the part number.

The intermediate frequency is from 33 to 39 MHz. For the calculation of the oscillator frequency and the number of steps the center frequency must be added. At the receive frequency

An example to clarify: the reception frequency is 137.5 MHz, the center frequency is at 36.45 MHz, the oscillator frequency is 173.95 MHz. The number of steps required: $173.95 \text{ MHz} / 62.5 \text{ KHz} = 2783.2$. Completed is 2783 steps. In hex: 0ADF. The first byte (variable *div_msb*): 0A, the second byte (variable *div_lsb*): DF.

You get little rounding. In this case, that step 0.2, this is 12.5 kHz ($62.5 \text{ kHz} * 0.2$). The center frequency is chosen so that the maximum variation in the receipt of weather satellites is 12.5 KHz.

The fourth byte is set to 0x8E. The last bit OS (op amp output) should be set to 0 for normal operation.

With the last byte, "Output ports" where the band is shown to be matched:

- | | | |
|----------|-------------|------|
| • Low | 45 -170 MHz | 0xA0 |
| • Middle | 170-450 MHz | 0x90 |
| • High | 450-860 MHz | 0x30 |

From the main routine, the variable *tape* is put.

For more information about programming the UV916 is found in the data sheets of the TSA5512 and UV916 or FQ916.

So to tune the tuner to 137.5 MHz off the following bytes sent via I2C:

Address: 0xC2
Dealer msb: 0x0A
Dealer lsb 0xDF
Control byte 1: 0x8E
Control byte 2: 0xA0

With the command endTransmission all bytes are sent to the UV916 at one time.

The main routine loop

As mentioned, the Arduino will remain constant in this routine. Come to the end jumps the Arduino back to the beginning and start again from scratch.

Two things need to be checked constantly and therefore are included in the course:

- Check the tuning knob is turned.
- The reading of signal strength and will display it on the display.

```
void loop ()  
{  
    // From the AD converter, we receive 12 values:
```

```
// 0, 91, 185, 278, 372, 464, 557, 651, 744, 837, 930, 1023.

posTuneSwitch1 = int (analog read (TUNING SWITCH) + 46) / 93 + 1;

// Check if there is a change in the position of the tuning switch.

if (posTuneSwitch0 != posTuneSwitch1)
```

On the fifth line with the command `analog read` the analog port 1 is read. The value that the Arduino returns between 0 and 1023, the complex calculation which makes coming back a value from 1 to 12 is in the variable *posTuneSwitch1*.

Because the Arduino is in a continuous loop, we want to check whether the dial is rotated. Otherwise, we would send a permanent I2C commands to the UV916 while this is not necessary. Or this may hurt I have not tested. The last line above is tested whether the dial is turned, we compare the old value with the new value.

If it is twisted then you come in the following piece of software.

```
{

    band = 96;

    switch (posTuneSwitch1)
    {

    case 1: div_msb = 0x0A;

            div_lsb = 0xA7;

            digitalWrite (APT, HIGH);

                    // 1234567890123456

            freqmessage = "134.00 MHz APT"; // 2727 steps

            break;

    case 2: div_msb = 0x0A;

            div_lsb = 0xD9;

            digitalWrite (APT, HIGH);

            freqmessage = "137.10 MHz APT"; // 2777 steps

            break;

    case 3: div_msb = 0x0A;
```



```

div_lsb = 0xDF;

digitalWrite (APT, HIGH);

freqmessage = "137.50 MHz APT"; // 2783 steps

break;

```

Not all case situations described here, it would be that long.

At the beginning we set the variable *band* at 96.

Then we come into the switch statement, literally a switch. Depending on the value in *posTuneSwitch1* the program section is executed after the corresponding case.

The variables *div_msb* and *div_lsb* have been appointed. The digital port 13 of the Arduino, we write the value *high* road, HRPT oscillator is stopped. The text to be shown on the display is put in the variable *freqmessage*.

After the switch statement, still within the part that is carried out because the tuning knob is rotated the following code is executed:

```

sent2uv916 ();

posTuneSwitch0 = posTuneSwitch1;

lcd0.setCursor (0, 0);

lcd0.print (freqmessage);

```

Our subroutine for UV916 is invoked. The variable for the values of the switch can be put right. We do not, then it will run the switch statement again. Finally, the cursor of the LCD display put in the upper left and the text is displayed.

We now arrive at the part in the part of the program that is always carried out, namely, the display of the signal strength. This is a complex piece of program because it is not possible to send a number to the LCD display, it must first be omgezet. Graag to a string I hear suggestions how this part of the program better or more elegant can be written. Maybe it neater possible with the *sprintf* function, which is described in the previous KM in the article on the GPS module.

```

samplesLogout [i] = analog read (LOGOUT);

delay (10);

i ++;

if (i >= aantalSamplesLogout)

```

```
{
    i = 0;
}
```

Every time the Arduino goes through the main loop, the analog port 0 read. The value is put in the array `samplesLogout`. If the maximum number of samples in the array is reached, the counter is put to zero. The result of this is that the array *`samplesLogout []`* contains the following ten signal strengths, eg.: [624, 615, 620, 630, 622, 624, 625, 621, 624, 623].

The following rules count all these values and calculate the average:

```
totLogout = 0;
for (int j = 0; j <aantalSamplesLogout j++)
{
    totLogout = totLogout samplesLogout + [j];
}
gemLogout0 = totLogout / (aantalSamplesLogout);
```

The calculation of this average occurs every time the Arduino goes through the main loop.

Showing the result is a different story. In the variable *`gemLogout`* is a value between 0 and 1024, this does not show off on the display. Rather have something between 0-100 and a position after the decimal point.

First we check if the result is changed, this is similar to what we do when checking the dial: if (`gemLogout1 = gemLogout0!`)

Is there a difference then run the code below:

```
lcd0.setCursor (8, 1);
lcd0.print ("");

StrLogout String = String (gemLogout1 / 10);
strLogout.concat (".");
strLogout.concat (gemLogout1 10%);
```

```
lcd0.setCursor (13 - strLogout.length (), 1);  
  
lcd0.print (strLogout);  
  
gemLogout1 = gemLogout0;
```

On the first two lines where the value is printed first emptied. This is necessary because for example the values. 102.3 is not the same length as 6.5. The value 6.5 are shown then 10 (from 102.3) still to remain.

On the following lines the variable *strLogout* filled with the measured value divided by 10 on the next line is a "." back and then put the rest of the division by 10.

The last two lines make sure the cursor is put. In the right place This is necessary because the length of the text may vary.

The last line of the two variables for the averages are aligned. This is to prevent the display is refreshed constantly and with this restless.

Closing

This brings us to the end of the article. This sketch is not as exciting in itself and can certainly be improved. I can send the sketch on request. In the extradited Arduino's for WRX1700 this sketch is programmed.

Future adjustments can be:

- Use of a rotary encoder so as to be able to use it. An unlimited list of satellites Now you have up to twelve positions.
- Control of a four line display, possibly via I2C. The Wire module is all for the UV916 present.
- New frequencies can enter without having to program. Arduino again
- Installing a scanner that searches for the strongest signal.
- Or control from the orbit calculation.

Let you know results!

LINKS

[1] Display Printing for the WRX-1700

Artificial satellite in September 2012

First Image: LRIT

Summary

This article Describes our efforts to receive LRIT signals with a 1.60 parabolic disk and an RTL SDR dongle with GNU Radio.

Introduction

In the previous Kunstmaan [1] Rob wrote that we want to review in preparation for the reception of QPSK of the Feng Yun / Metop satellites. Reception technique The idea is that the QPSK demodulator should also lock to the LRIT signal the BPSK signal LRIT. using the dish in Dwingeloo we were unable to receive the LRIT signal.

It's fun to go to Dwingeloo but practically it is not. Rather a dish closer to home and see what is possible.

Satellite

Harrie has a quick test with his setup for polar satellites with a dish of 80-90 cm. You get t a signal barely above the noise comes out. We tired s at so looking for a larger dish.

According to *"TD 08 - MSG LRIT Direct Dissemination Service"* a dish of 1.80 meters would suffice. On, March meeting log d e Wim that he had e and saucer of 1.60 meters, d it wa s well worth a try. In late April, the dish was mounted on the rack behind the bus and on to Harry!



Fig 1. Dish on bicycle carrier.

Then put the whole thing down and tested. Unfortunately, the signal was not better than what we had received. Dwingeloo We even sent a message to Eumetsat whether there are problems with the direct broadcasts, this was not the case.



Fig 2. Prepared dish at Harrie.

In the past contacts between Rob and Martin Blaho been on the direct reception LRIT, Martin has a working installation! So gone consulted with Martin vraa g what equipment he uses.

His receiving system consists of:

- E and saucer of 2.20 meters with a circularly polarized illuminator
- LNA with 0.7dB noise figure of Kuhne
- Downconverter
- TV dongle with the E4000 tuner.

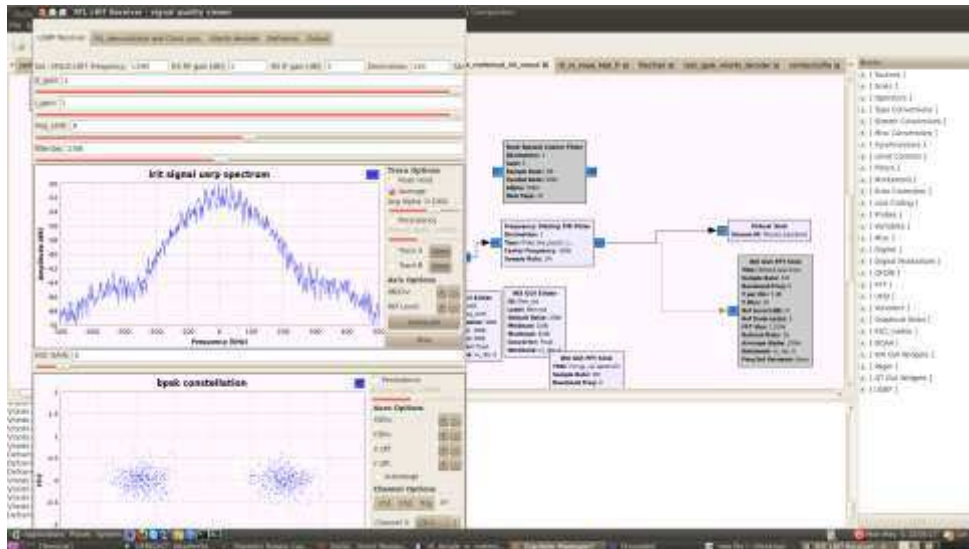


Fig 3. GNU Radio with Martin. Upper left you can see that the LRIT signal 15 dB above the noise floor comes out. Among the BPSK constellation diagram. The cloud of dots is well apart. Right in the background you can see a part of the receiver.

Optimizations

The 1.60 meter dish some optimizations can be made:

- Dish Wim was supplied as a lighting technician with a bus does. Probably a properly tuned helical still better because it is an irradiated resonant circuit.
- Optimal focal point of the imagesetter out.
- Testing and adjusting the Meteosat Wefax 7.

In addition, the QPSK demodulator can be optimized by using in the demodulator. Has a bandwidth less After optimizing the dish and narrowing the bandwidth of the receiver do not want to lock.

Signal

Time to start measuring. The Rigol spectrum analyzer we can not measure at 1691 MHz. Therefore first the LNC-1700 as down-converter is attached in order to measure. At 137.5 MHz

If you set the bandwidth of the window well, you can measure up to about -135 dBm.

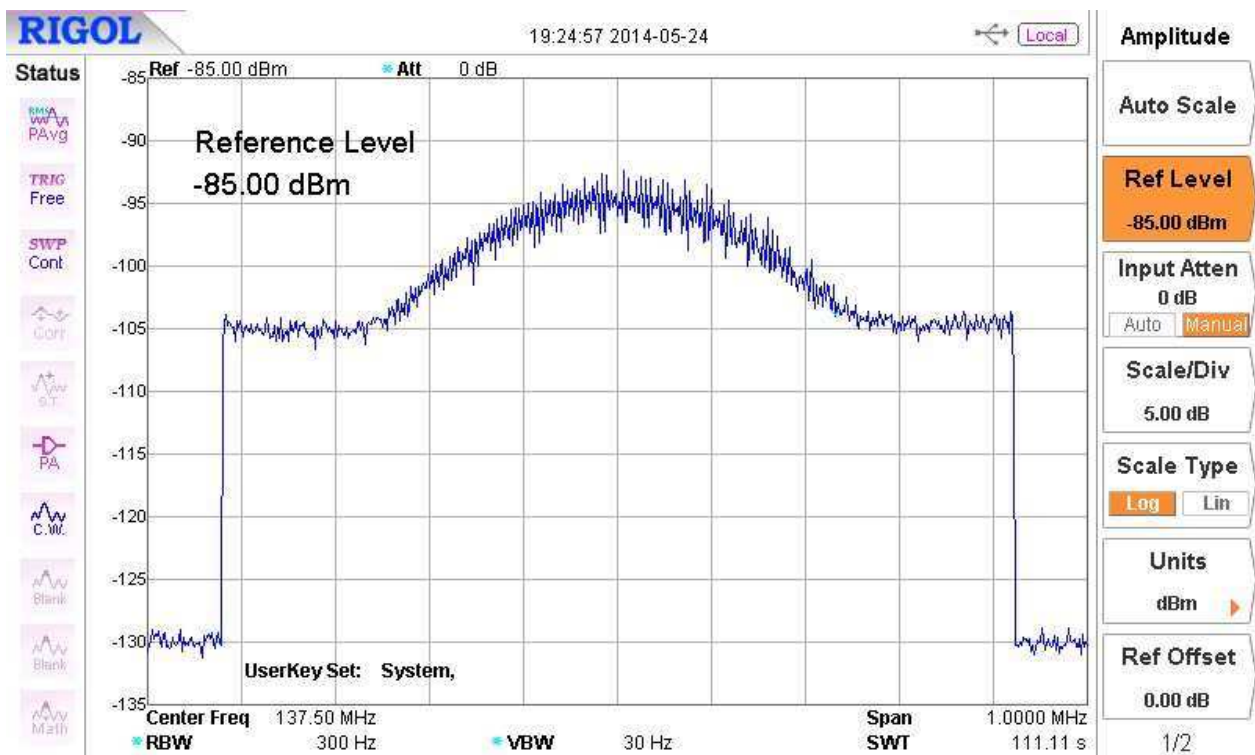


Fig 4. LRIT signal at 137.5 MHz

In the picture you can see the noise floor when switched converter at -130, the noise floor outside of the band at -105 and top that approximately 10 dB higher at -95 dBm. That's enough to note that the signal from MSG-3 above the noise floor is only 10 dB. The Costas loop is above the noise floor did not get working with 10 dB

When the original e Meteosat bus is fitted, the signal 3 dB sinks, long live the resonant helical!

GNU R adio

The hardware route was temporarily a dead end. What is software? Martin uses GNU Radio to receive. LRIT What if we try to get the optimized 1.60 meter satellite dish and GNU Radio LRIT going to receive?

Martin was kind enough to send. Receive his anger With GNU R adio receiver does this by email and not by DHL, but so easy.

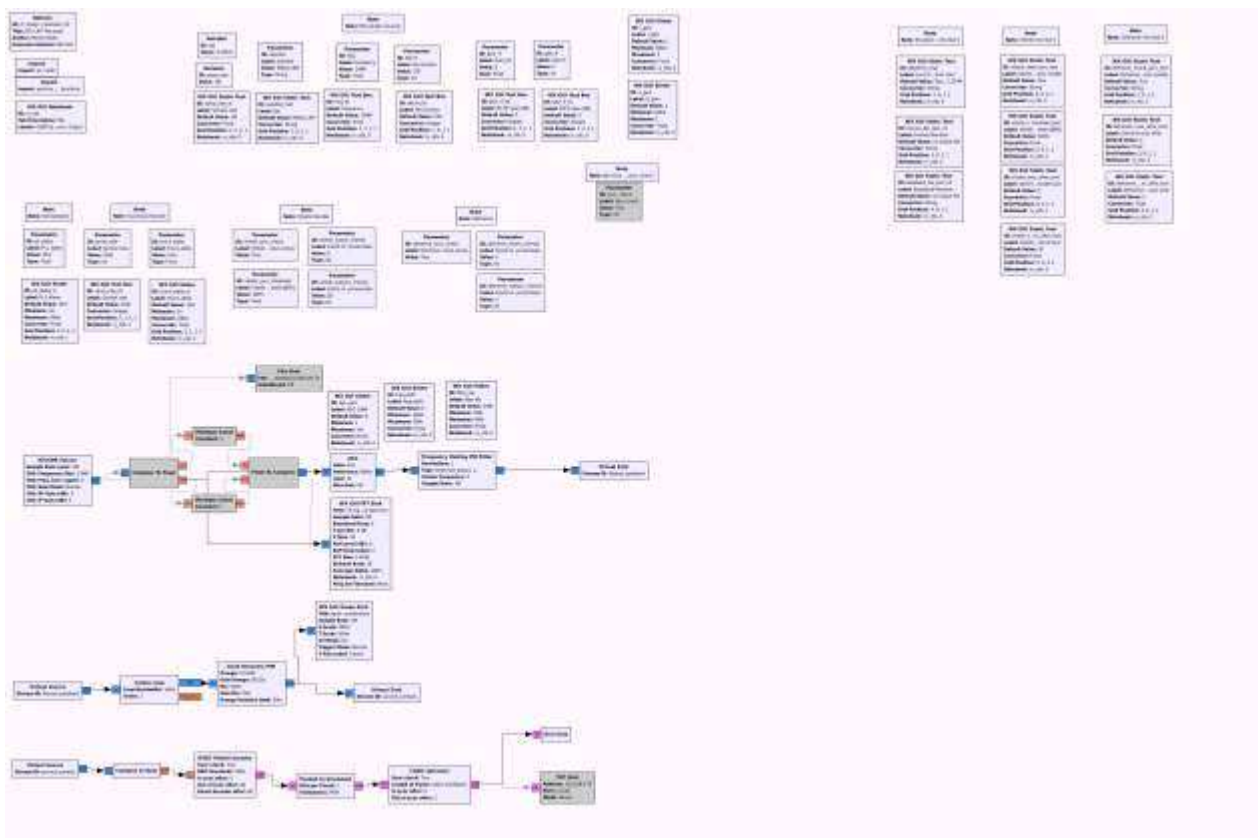


Fig 5. Receiver Martin.

Still, there had to be the one and the other to be tinkered with, not with a soldering iron, but with a text editor. He uses include self viterbi decoders developed.

Practice

On Thursday, May 28th at Harry with RTL dongle from China 7 Euro. This attached behind the LNC1700 to see if we can get. Well constellation diagram GNU Radio is on a separate USB stick that is in my laptop.

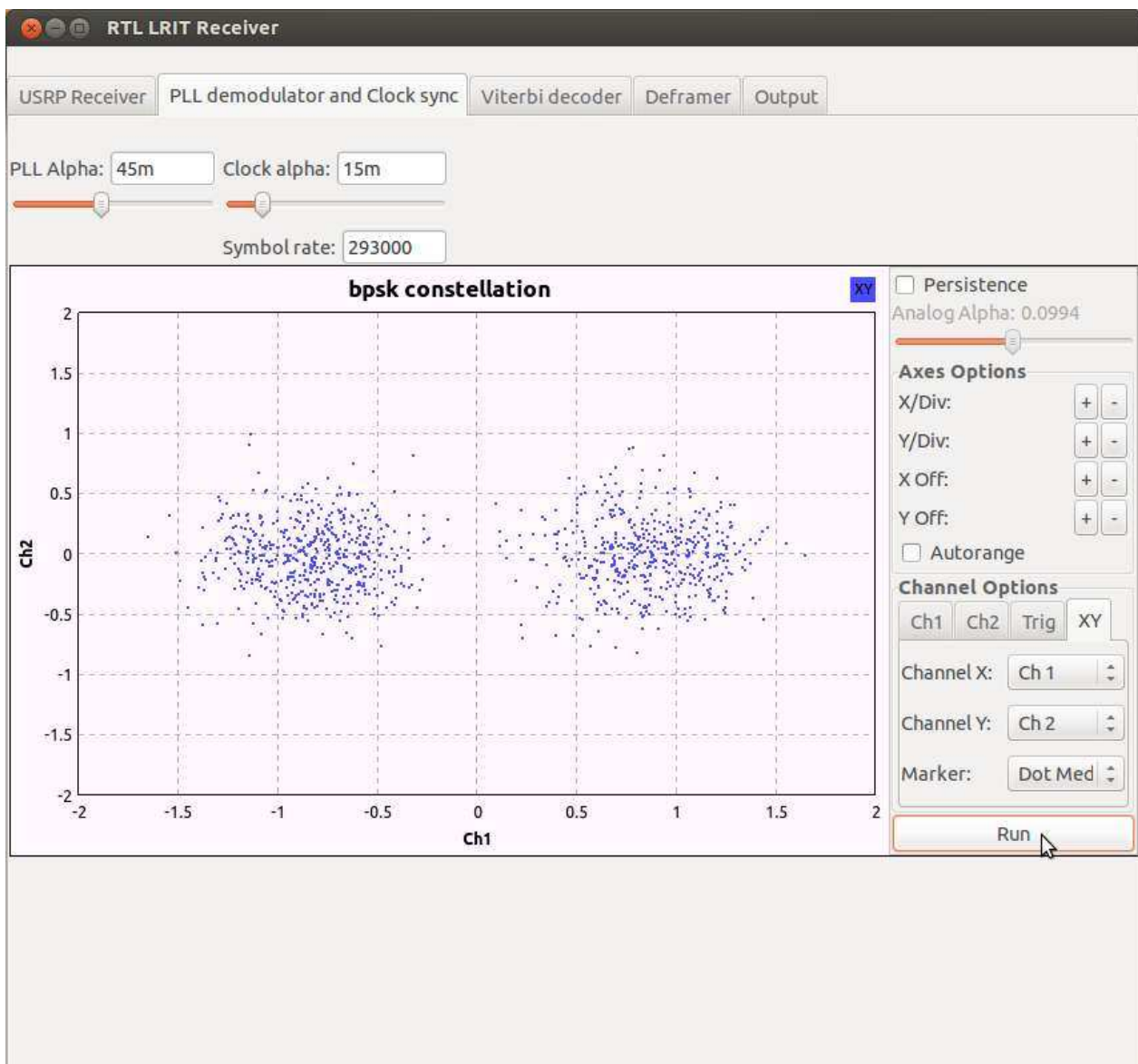


Fig 6. Constellation Diagram in GNU Radio

The constellation diagram looks pretty and Martin agrees. The signal is indeed 10 dB above the noise floor.

The bandwidth can be set by software on 238 KHz with a very steep FIR filter one fine group delay. This combination can not do in hardware.

On June 14 at Harry to record the signal and then to Martin to send it. Martin is the baseband signal decode and convert it to a picture. The amount of data that you include is unimaginable. A recording of one minute is a rate of 500 MB. I have included two files, approximately 1.5 GB of data! The signal is recorded at a rate of 1 Mbps. The dongle

has a maximum of around 2.4 Mbps, the data rate of Fengyun and Metop this is insufficient for HRPT enough.

When you receive an LRIT signal, it is questionable whether what you can do because it is in many cases encrypted with. The six-hour products are MSG unencrypted. In addition, the three-hour products from Meteosat 7 MTSAT2, GOES13 and GOES15 also available.

A few days later, the message of Martin that he has managed to put the data successfully to a picture A dish van1,60 meters is sufficient.

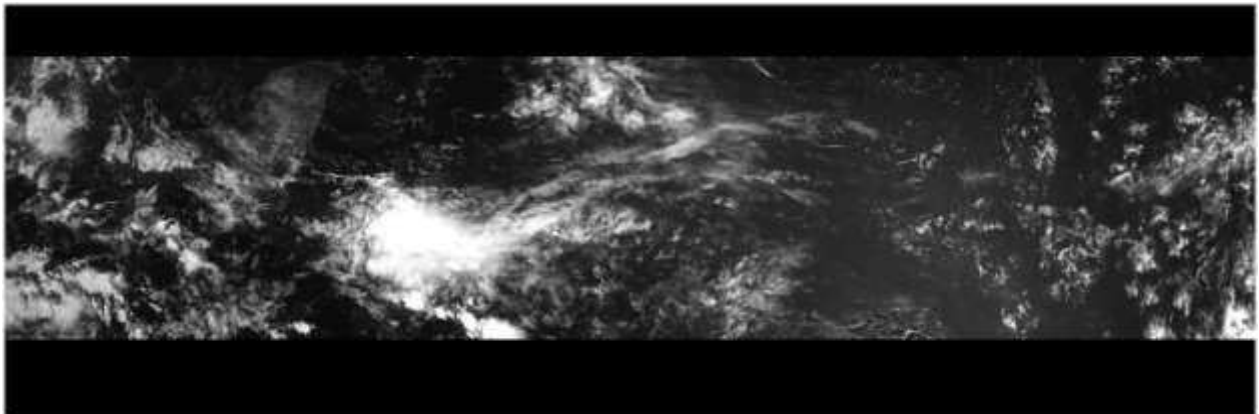


Fig 7. Received five segments of the GOES13 with a 1.60 meter dish.

Reception Installation

LRIT broadcast by the MSG-3 in the 1691 MHz. The receiving system consisted of the following components:

- 1.60 dish with helical setter.
- Two 14 dB LNA.
- LNC-1700 with 24 dB gain. The LRIT signal is at 134 MHz.
- RTL SDR DVB-T dongle with RTL2832 + R820T.
- GNU Radio on a USB3 stick.

Summarizing

Receiving this first image has nevertheless yielded some new insights. In particular, filtering can be better in software than in hardware. Furthermore, you can easily (international) working with software.

In this article I want to dwell on the software story. Too much This is quite complex and subject to change. Besides GNU Radio is also used to get conversion software. Xrit2pic visible in the images you prefer files from GNU Radio which can then be picked up. Directly xrit2pic

The RTL dongle itself is suitable to receive. To 2 GHz signal Maybe we can let expire? The downconverter

In the corner a hardware analog multiplier between I and Q as is usual with BPSK for Costa Loop would be worth trying. Continued ...

Links

[1] Even if your antenna so big

Artificial satellite in July 2014, Volume 41, page 87.

NOAA16 after 13 years of service and 70 655 revolutions deactivated on June 9.
NOAA 17 has been disabled for faulty scan engine on 10 April 2013.

Sporadically LRPT signals METEOR M N1 reported on 137.1 MHz. There is then a small risk of interference to NOAA 19. However, it is frequently signal at 1700 MHz catching these METEOR. This mode (referred to as MHRPT) is not compatible with HRPT but can be decoded with the new Rob Alblas decoder!

Older NOAA's (Tiros-N, NOAA 6 and 9) can give sporadic interference at 137.50 MHz if these satellites are illuminated by the sun.

Fengyun 3A, 3B and 3C AHRPT only send out, this is not received with standard HRPT receiver and decoder. Unfortunately, this is not entirely AHRPT according to standard so that even a Metop AHRPT receiver is not suitable for the FY-3 series!

Meteor M N2 is to receive 137,100 with RTL dongle!

NPP (NPOESS Preparatory Project) only sends out the X-band with 15 Mbit / s. It is recommended a tracking dish with a diameter of 2.4 meters!

EUMETCast is to receive EURO-BIRD-TM 9A at 9 ° E in DVB-S and DVB-S2 in 10A. The reception of EUMETCast data is for amateurs free of annual fees, you should, however, to register with EUMETSAT. One time you need to buy software (60 Euro) and key (40 Euros). To further increase the amount of data sent through EUMETCast possible in the future making is EUMETSAT in August 2014 passed from DVB-S to DVB-S2 VCM mode. The new transponder is on Eutelsat 10A, 11263 MHz H, which is at 10 degrees East. Unfortunately, existing DVB-S and "DVB-S2 VCM without" recipients after December 31, 2014 are not more useful because then stops the parallel broadcast.

Since August 14, the transponder on 11263 MHz Horizontal operational. The broadcast standard DVB-S2 VCM 8PSK 3/5 (Basic Service) or 16APSK 2/3 (High Volume Service) with a symbol rates to 33 Msps. In addition to the Ayecka SR1 and Novra S300E receiver are now also the TBS-6925 and 5925 working, but we are still waiting for an update to improve stability and compatibility. If only the Basic mode is enabled, it is already possible reception with 80cm dish. HVS after the tests but requires a dish of at least 120cm. The more advanced TBS-6983 PCIe card is expected soon. It has two tuners and an advanced dual demodulator, and will be 6925 to replace him in time. Therefore, I recommend users to have to wait to buy a new receiver.

In the EO Portal EUMETCast users can log in and their personal information and input propositions understand and modify it. Also registering as a new user and prolong the license is possible here. You can on-line indicate which products you want to receive your EKV. Existing EUMETCast users have received an email with an explanation and a login code and password. To receive MetOp and / or use a Ramdisk Modis is necessary, the following EPS data channels are recommended:

EPS 10 MetOp AVHRR
EPS 15 NOAA GAC
EPS-18 EPS News Service

Thanks to David Taylor and Douglas Deans for the info.