*digi*WUFFUM

SDR Transmitter · PE1PzN · AD9363 · manual version 3.2 (19 October 2026)

1. What is digiWUFFUM
2. Connecting (I2S boards)
3. The first time: card, USB and IP address
4. Starting
5. Connecting to the 7020
6. The rack
7. The monitor bar
8. Modulatiemonitor (modulation monitor)
9. RDS
10. RF-spectrum (RF spectrum)
11. Afregeling (alignment)
12. Ontvanger (receiver)
13. Logboek (log)
14. Lights and alarms
15. Known limitations

1. What is digiWUFFUM

digiWUFFUM controls an FM broadcast transmitter (WBFM) on a Zynq 7020 SDR with AD9363 (a Pluto clone, 70 MHz to 6 GHz). At the same time it measures what goes on the air. The audio comes into the 7020 as I2S; the FPGA turns it into FM modulation.

Card 3.x. A cleaner signal: next to the channel it is quiet, and the channel filter is steep (a 200 kHz channel really is 200 kHz wide). The FPGA also has an **FM receiver**: with a DAC board added, you listen without delay to an FM station from 70 MHz to 6 GHz, while you transmit yourself.

New in 3.2.

- Noise reduction in the receiver in the 7020: slide `Ruis` (noise) from 0 to 10 (see chapter 12).
- No more clicks and pops in the DAC audio when you retune, change tabs, or switch mono or 75 µs.
- The alignment per frequency is now always applied, also after a restart of the 7020.

New in 3.1.

- Works on **every** 7020: on some boards no MPX came through with 3.0, or the DAC stayed silent. That is fixed.
- Button `Zoek` (search) next to the IP address: digiWUFFUM finds the 7020 by itself, via USB or via your network.
- Also works when your home network is 192.168.2.x (for example with KPN): the card solves that by itself.
- The transmitter no longer switches off by itself when the program is busy for a moment.

> **Only for radio amateurs and radio research, not for commercial use.** The program, the files for the SD card and this manual are given free of charge as they are, without any warranty. Use is at your own risk: PE1PzN is not liable for damage to equipment, interference to others or any other result of the use. You may only transmit within the terms of your own licence and on frequencies where you are allowed to; that is your own responsibility.

What the program does:

- switch the transmitter on and off, set frequency and power, presets;

- measure your own broadcast: deviation, pilot, RDS, MPX power, channel separation, RF spectrum;
- read RDS in full (PS, RadioText, RT+, clock, PTYN, Long PS, AF and more);
- control the MPX clipper and the channel filter in the FPGA;
- align the transmitter: tune out the image and carrier leakage, automatically, and saved per frequency;
- a second receiver with spectrum, waterfall and S-meter, to look at other stations;
- listen via the receiver in the 7020 and a DAC board (card 3.x), or via a second SDR (RSPdx).

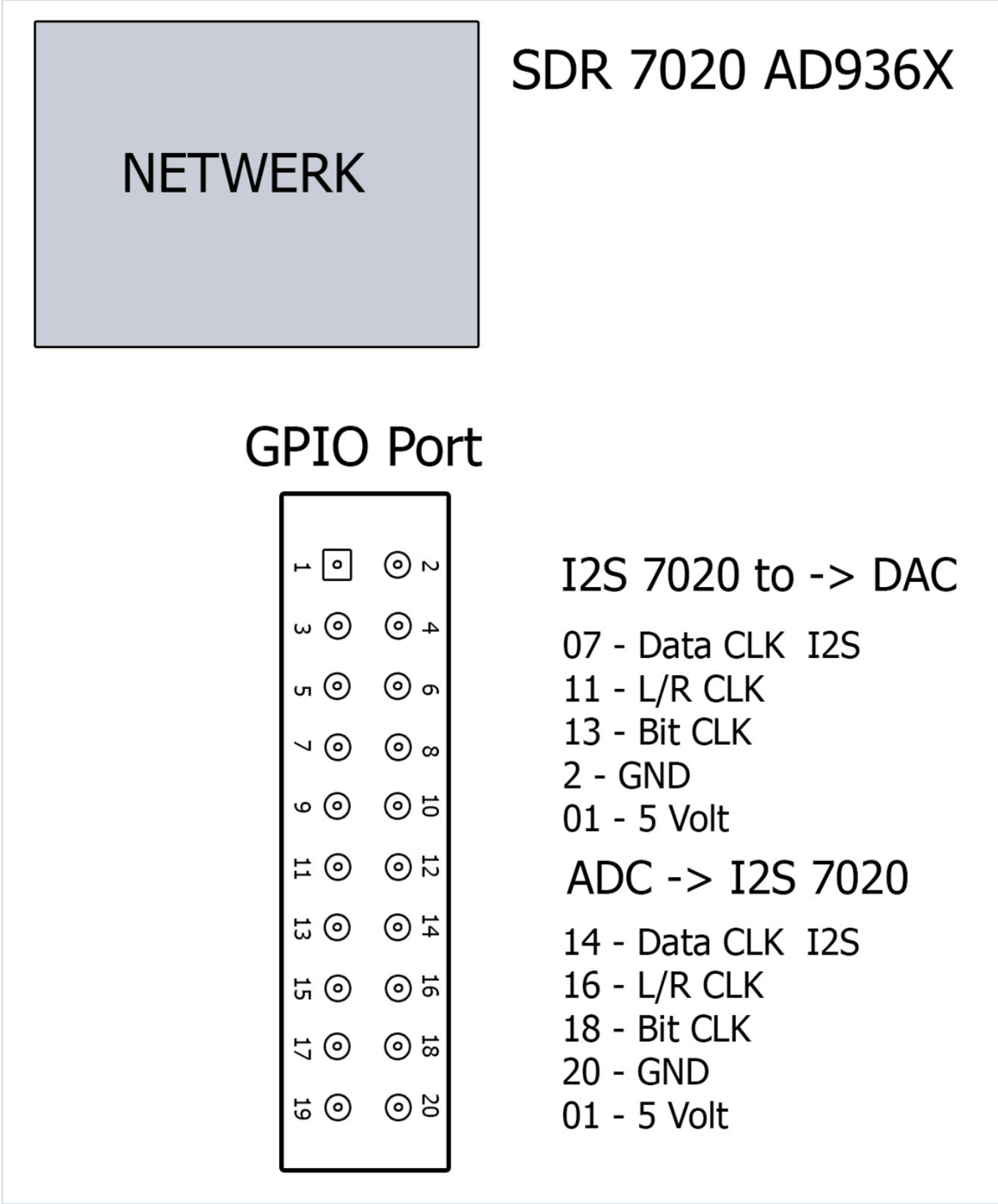
The FPGA design digiWUFFUM started from is by Willem, PE1PCF: github.com/Willem65/Vivado-Zynq-7020-Chinese-ADALM-Pluto-kloon.

2. Connecting (I2S boards)

The MPX signal comes in **analogue** on a small I2S ADC board close to the 7020. It converts it to I2S, and the FPGA turns it into FM modulation. Analogue up to the board, because I2S does not work over a long cable.

```
MPX source (stereo encoder)  --analogue-->  I2S ADC board  --I2S-->  JP5 of the 7020  -->
FPGA  -->  AD9363
AD9363 (RX1/RX2)  -->  FPGA (FM receiver)  -->  JP5  --I2S-->  DAC board  -->  speaker   (card
3.x)
```

The JP5 header of the 7020

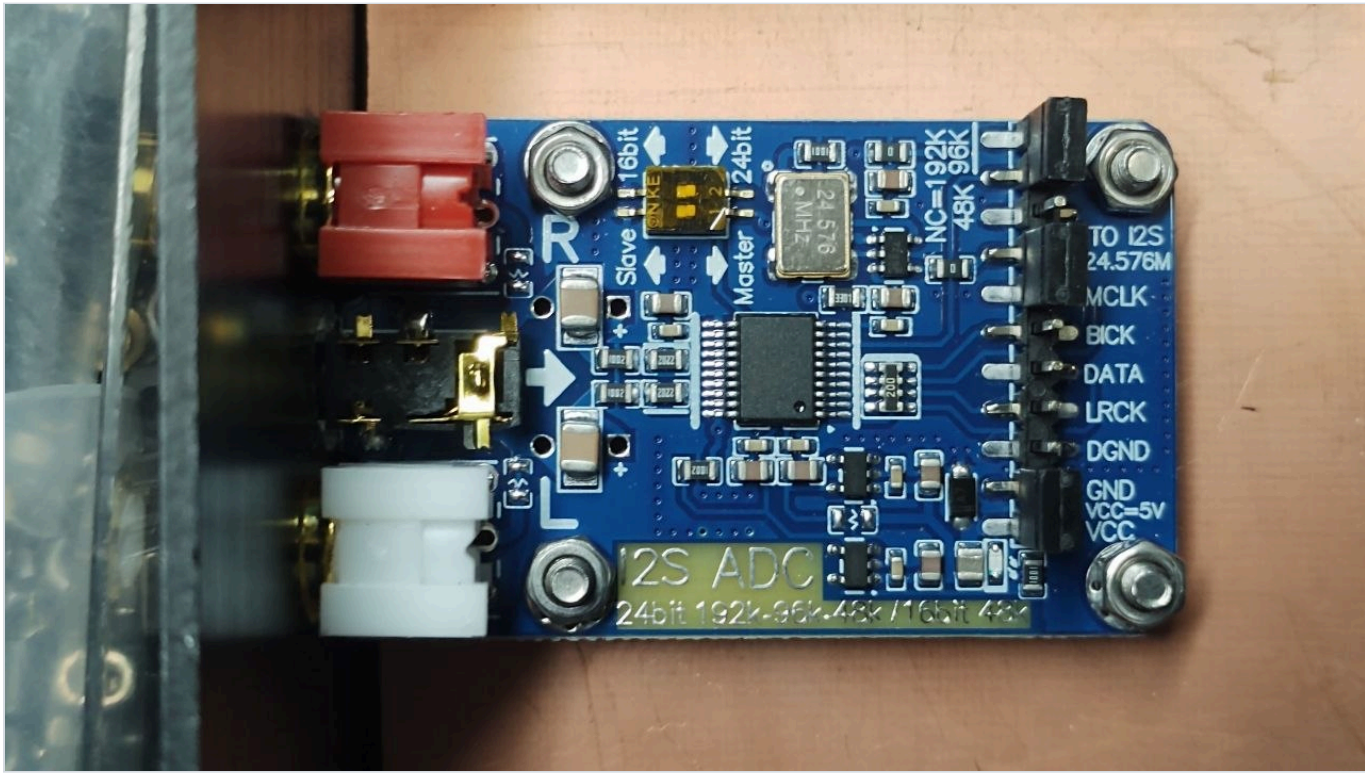


The GPIO port JP5: 2 x 10 pins. The even pins (right column) are on a 1.8 V bank, the odd pins 7 to 13 on 3.3 V.

Pin	What	Pin	What
1	5 V (supply)	2	GND
3	3.3 V (supply)	4	free (1.8 V)
5	1.8 V (supply)	6	free (1.8 V)
7	DAC: DATA (3.3 V)	8	free (1.8 V)
9	DAC: MCLK, not needed (3.3 V)	10	free (1.8 V)
11	DAC: LRCK (3.3 V)	12	free (1.8 V)
13	DAC: BCK (3.3 V)	14	**ADC: DATA** (1.8 V)
15	**do not use** (VCTCXO)	16	**ADC: LRCK** (1.8 V)
17	**do not use** (PTT)	18	**ADC: BCK** (1.8 V)
19	**do not use** (AD9363 SYNC)	20	GND

> Check the supply pins 1, 3 and 5 with a multimeter first, before you connect anything to them. The pinout comes from Willem (PE1PCF), who measured it on the board.

ADC board: MPX to the 7020 (digiWUFFUM)



The ADC board: switches on Master and 24 bit; sample rate jumper open (NC) = 192 kHz. Output BICK, DATA, LRCK, DGND; supply VCC 5 V. Do not connect MCLK and "TO I2S 24.576M".

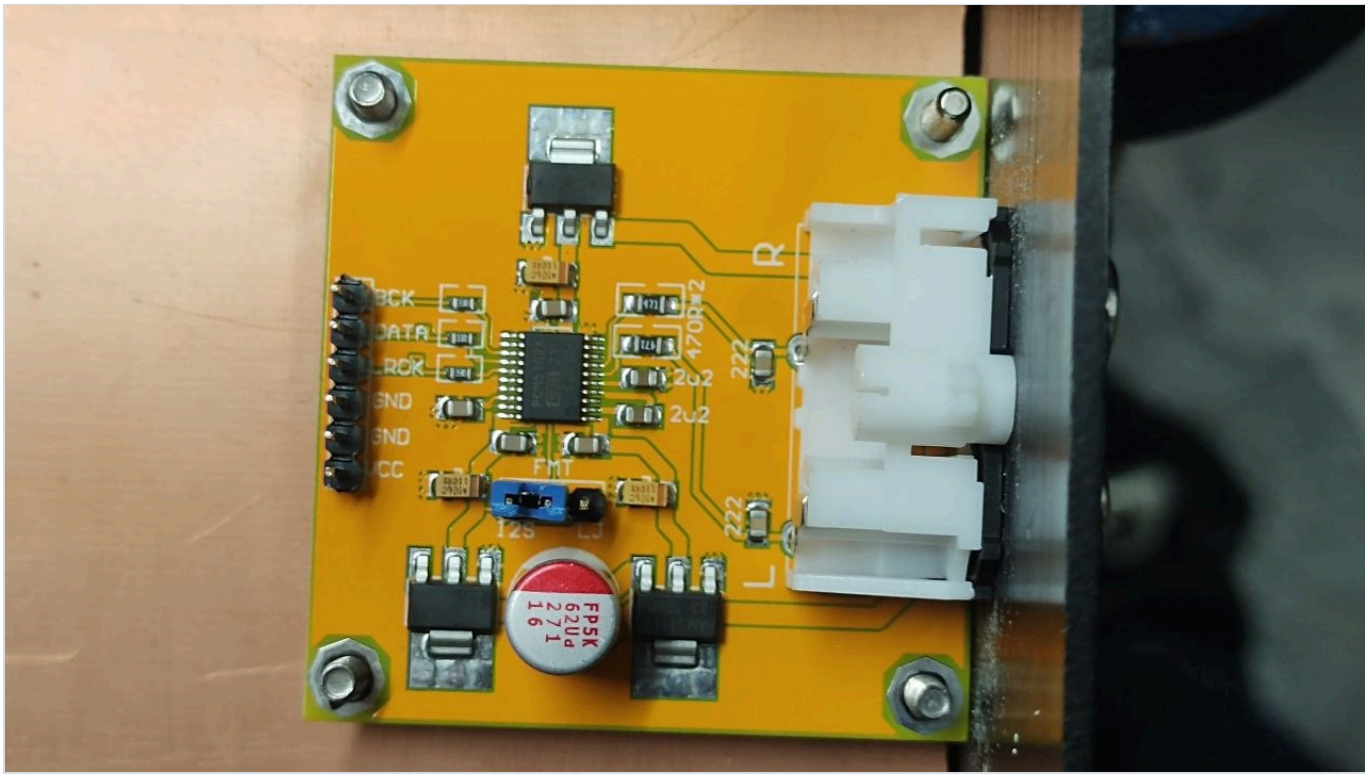
The board is called **I2S ADC Audio Capture Card Module** (24 bit, 192/96 kHz, Master/Slave). You can buy it for example at https://www.amazon.nl/dp/B0CQ5KK9XQ.

ADC board	JP5	Note
DATA (SDOUT)	14	via a divider: **1 kΩ in series, 1.2 kΩ to ground** (3.3 V becomes 1.8 V)
LRCK (WS)	16	
BCK	18	
GND	20 (and 2)	

- The ADC board is set to **Master** and **24 bit** (switches): it makes the clock itself, 192 kHz (BCK 12.288 MHz). The sample rate jumper must be **open** (NC = 192 kHz); on 96K or 48K it does not work.
- The MPX goes to the **left** input; the FPGA only uses the left channel.
- Pins 14, 16 and 18 are on a 1.8 V bank: without a divider the FPGA gets 3.3 V on those pins.
- The deviation is fixed in the FPGA. You set the MPX level with the potentiometer before the board: set it so the peak deviation (Modulatiemonitor) comes to about 75 kHz. The clipper catches the last peaks.

> **Without the ADC board (no BCK) the whole FPGA chain stands still**, the test tone too: then only a small carrier leak goes on the air. So always connect the board, also for alignment.

DAC board: audio out of the 7020 (card 3.x)



The DAC board with PCM5102A: input BCK, DATA, LRCK, GND, VCC 5 V; the FMT jumper on I2S.

From card 3.x there is an FM receiver in the FPGA. Its audio goes to a DAC board with a PCM5102A, exactly as with digiNICAM. There is no PC in between, so there is no delay either. The DAC board is not needed to transmit; only to listen.

DAC board (PCM5102A)	JP5	Note
DATA (DIN)	7	3.3 V, direct (no divider)
LRCK	11	
BCK	13	
GND	2 (and 20)	
VCC	1 (5 V)	the board has its own 3.3 V regulators

- **No divider to the DAC.** Pins 7, 11 and 13 are on the 3.3 V bank: the FPGA outputs 3.3 V, and the PCM5102A itself also runs on 3.3 V (high = from about 2.3 V). A divider is only needed where 3.3 V goes into a 1.8 V input, so only on the ADC side.
- 32 kHz, 24 bit, standard I2S; the 7020 is master. No MCLK needed: the PCM5102A makes it itself.
- The FMT jumper of the DAC board on I2S (low). If you only hear crackling, switch it over.
- If the ADC and DAC boards share pin 1 (5 V), split that pin (for example with a Y cable).

3. The first time: card, USB and IP address

You need two downloads: **digiWUFFUM SD-kaart 3.2** (for the 7020) and **digiWUFFUM 3.2** (for the PC). Always update them together.

Step 1. Making the SD card

1. Take an empty micro SD card, formatted as **FAT32**. Larger than 32 GB? Then make a small FAT32 partition in Windows (Disk Management), for example 1 GB.
2. Unzip `digiWUFFUM-SD-kaart-3.2.zip` and put the **loose files** directly on the card, so not in a folder. At the top level of the card you then see, among others, `BOOT.bin`, `devicetree.dtb`, `uImage` and `uEnv.txt`.
3. Eject the card properly and put it in the 7020 (power off).

Step 2. The program on the PC

1. Unzip `digiWUFFUM-V3-2.zip` to its own folder, for example `C:\digiWUFFUM`. Python is not needed.
2. Start `digiWUFFUM.exe`. If Windows shows a blue warning (SmartScreen), choose **More info** and **Run anyway**. You only need to do that once.
3. To look around first without a 7020, use `START DEMO.bat`.

Step 3. Starting via USB

The first time is easiest via USB: then you do not need to know anything about your network yet.

1. Connect the 7020 to the PC with a USB cable and switch the power on. Wait until it has started (a good half minute).
2. Windows sees the 7020, just like a normal Pluto, as a **network card** and as a **USB drive**.
3. In digiWUFFUM press `Zoek` next to the IP address. The program finds the 7020 and fills in the address by itself (via USB that is usually **192.168.2.1**). Then press `Verbinden` (connect). If it works, the card is good.

USB or network cable?

Via USB the 7020 also becomes a network: a small network between the PC and the 7020. That is handy for the first time, but it is much slower than a network cable. In practice USB reaches about 130 to 200 Mbit/s, and the small processor of the 7020 has to handle all that USB traffic itself. The network port of the 7020 is gigabit. For transmitting, USB is more than enough. But the RF spectrum and the receiver pull in samples, and then the picture can stutter via USB. So for regular use, use the network cable.

Why is USB slow here? Isn't it USB-C? USB-C only says something about the shape of the plug, not about the speed. Behind that plug the 7020 simply has USB 2.0, 480 Mbit/s on paper. But the 7020 does not send the data directly: it acts as a network card, and its small processor has to pack and unpack everything. Because of that, only 130 to 200 Mbit/s is left in practice. Via the network cable it is different: the network port of the 7020 is 1 gigabit (1000 Mbit/s). That is more than enough for everything. Note: your router or switch and the cable must then also be gigabit. With an old 100 Mbit/s switch you only get 100.

Step 4. Via the network cable

Want to use the 7020 via the network cable (handy if it is further away)? That is quickly done:

1. Plug in the network cable and (re)start the 7020. It gets an address from your router by itself.
2. In digiWUFFUM press `Zoek`. The address found is kept until you press Zoek again.
3. Press `Verbinden`.

Prefer a fixed address? On the USB drive of the 7020, open the file `config.txt` with Notepad. Under `[NETWORK]` fill in, for example, `ipaddr_eth = 192.168.0.173` and `netmask_eth = 255.255.255.0`, save, and **eject the drive** in Windows (only then does the 7020 take it over). Then restart the 7020.

> Turn on **automatisch** (automatic, chapter 5) once it works; then digiWUFFUM connects by itself from now on at start-up. After that the USB cable may stay in or come out; power can also come from a separate 5 V adapter.

4. Starting

File	What it does
`digiWUFFUM.exe`	The program, without the need for Python.
`START.bat`	Starts the program from the folder with its own Python environment (venv).
`START DEMO.bat`	Starts without a 7020, with a simulated stereo FM signal with RDS. Handy to look around.

> digiWUFFUM 3.2 belongs with the card **digiWUFFUM SD-kaart 3.2**. Always update program and card together. After connecting, the rack shows which card is inside, for example `FPGA 3.2 · AD9363`.

- The program can only run **once at a time**. If you start it a second time, you get a message. Two programs at the same time would get in each other's way at the 7020.
- At start-up the window adapts to the screen. For that it may restart briefly once; that is normal.

- At start-up the transmit power is always set to **−10 dB**, whatever was set last time.
- All settings are in `zender_instellingen.json` in the same folder and are saved by themselves.

5. Connecting to the 7020

At the top of the window are the connection details:

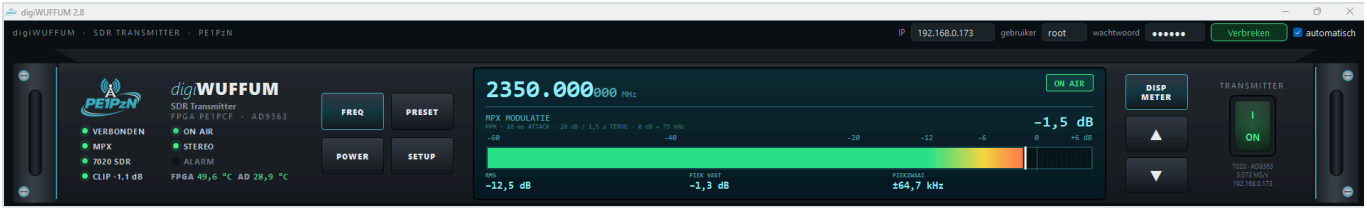
Field	Default	Explanation
IP	pluto.local	Network address of the 7020.
`Zoek`		Searches for the 7020 via USB and in your network, and fills in the address. It does not log in anywhere.
gebruiker (user)	root	Login of the 7020 (as with a normal Pluto).
wachtwoord (password)	analog	
`Verbinden`		Connects. Then becomes `Verbreken` (disconnect); disconnecting switches the transmitter off.
automatisch	off	Ticked: connect right away at start-up.

After connecting:

- the light **VERBONDEN** (connected) comes on;
- the program sets the frequency, the IQ correction and the alignment for this frequency on the 7020;
- it sets the clipper and the channel filter as set in the monitor bar (the FPGA starts with the clipper off);
- the test tone is always switched off;
- the monitor starts by itself, if it was not running yet;
- the light **7020 SDR** comes on: the 7020 is connected and answers.

> If the transmitter was already on when you connected (for example after a restart of the program), digiWUFFUM takes over the power the 7020 is using at that moment. The Logboek reports this.

6. The rack



At the top the connection details, below that the rack: brand and lights, buttons, display and the TRANSMITTER switch.

Lights

The same colour rule applies everywhere in the program: **green** = OK or on the air, **yellow** = off the air, **red** = something is wrong.

Light	Meaning
VERBONDEN	Connection with the 7020.
ON AIR / OFF AIR	**ON AIR**: transmitter on. **OFF AIR**: connected, but transmitter off.
MPX	Modulation is coming in.
STEREO	The pilot (19 kHz) is present.
7020 SDR	**Green**: the 7020 is connected and answers.
CLIPPER	**Green**: the MPX clipper in the FPGA is on. When it acts, it shows how many dB it takes off the highest peak (for example `CLIP -1,2 dB`, held for 1.5 s). `CLIPPER UIT` (clipper off): switched off. Grey `CLIPPER` without a connection or with a card without clipper.
ALARM	**Red** when there is a problem; the reason is shown below it. See chapter 14.

Next to the CLIPPER light are the temperatures of the FPGA and the AD9363: **green** below 70 °C, **yellow** up to 80 °C, **red** above that.

Buttons

Button	What it does
`FREQ`	Type in the transmit frequency in MHz (70 to 6000), for example *1285* or *435.5*.
`PRESET`	Presets, see below.
`POWER`	Transmit power (TX attenuation) from −89.75 dB to 0 dB (maximum), and the test tone on or off. Use **−4 dB** as the highest setting: at 0 dB the AD9363 distorts more and extra sidebands appear.
`SETUP`	Goes to the Afregeling tab.
`DISP`	Switches the display: SPECTRUM, METER (PPM) or STATUS.
`▲` `▼`	Frequency one step up or down. **Right mouse button** on an arrow: choose the step size, from 1 kHz to 1 GHz (default 100 kHz). This also works with PRESET.
TRANSMITTER	Toggle switch: transmitter on (I) or off (O). Only works when there is a connection.



Right mouse button on ▲ or ▼: choose the step size.

Tuning with the mouse wheel

Put the mouse on a **digit** of the transmit frequency and turn the wheel: only that digit changes. On the first digit that is 1 GHz, after the point 100, 10 or 1 kHz, and on the small digits 100, 10 or 1 Hz. There are always four digits before the point; below 1 GHz the first one is a dimmed 0, so the mouse stays on the same digit. Next to the digits the wheel turns with the step of the arrows.

> Avoid frequencies that are an exact multiple of 40 MHz, such as 1280.000 MHz: there the PLL of the AD9363 makes a small wobble. A few MHz off, for example 1283 MHz, is fine.

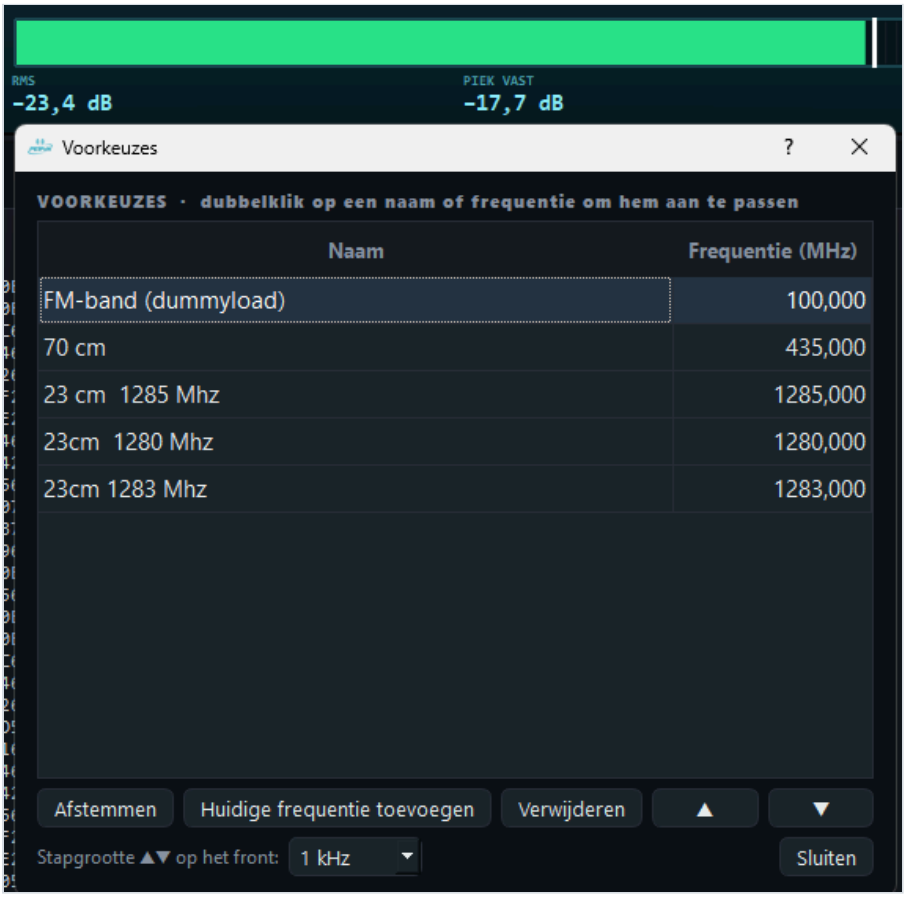
The display

The display always shows the transmit frequency and ON AIR or OFF AIR. With `DISP` you choose what is below it: a small spectrum, a PPM meter or the status (attenuation, peak deviation, pilot, MPX power, RDS name, which receiver is measuring, and the temperatures).



Display in METER mode: a PPM on the MPX (10 ms attack, 20 dB back in 1.5 s, 0 dB = 75 kHz deviation), with RMS, the held peak and the peak deviation below it.

Presets



The Voorkeuzes (presets) window (PRESET button).

Button	What it does
double-click	On a name or frequency: edit it.
`Afstemmen`	(Tune) Set the transmitter to the chosen preset.
`Huidige frequentie toevoegen`	(Add current frequency) The frequency the transmitter is on now, as a new preset.
`Verwijderen`	(Delete) Remove the chosen preset.
`▲` `▼`	Move the chosen preset up or down in the list.
Stapgrootte ▲▼ op het front	(Step size ▲▼ on the front) How big the step of the arrows on the rack is.

> With a new frequency, digiWUFFUM sets the alignment of the nearest saved frequency on the 7020 by itself (see Afregeling).

7. The monitor bar

Below the rack is the bar you use to control the **monitor**: the receiver that measures your own transmitter.

Part	Explanation
Bron (source)	What is used to measure: the 7020 itself (Pluto), an RSPdx, or Demo.
Ingang (input)	With the 7020: RX1 or RX2. RX2 is further from the transmitter and picks up less internal crosstalk. With the RSPdx: Ant A, Ant B (SMA) or Ant C (BNC, up to 200 MHz).
LNA	Gain of the receiver.
RDS	Untick with the 7020 as source: shorter measuring blocks, faster refresh, but then no RDS.
Meeluisteren (listen in)	The demodulated audio on the sound card of the PC. Works via a second SDR, the **RSPdx**. With the 7020 as source the tick box is grey. Listening with the 7020 works with card 3.x via the DAC board (see Ontvanger).
`Start monitor`	Starts or stops the measuring.
`Clipper aan` / `uit`	(Clipper on / off) The MPX clipper in the FPGA (card 2.3 and later). It sits after the upsampling, just before the FM modulator: whatever goes above the limit does not go out.
grens (kHz) (limit)	60 to 80 kHz, default 71. With the channel filter at 200 kHz the peak deviation comes out about 4 kHz higher than the limit: 71 kHz then gives about 75 kHz deviation.
knie (dB) (knee)	0 to 6 dB, default 3. This many dB below the limit the clipper starts to round off softly; above it, it clips hard. 0 = only hard clipping.
clipper light	How many dB the clipper takes off the highest peak (card 2.4 and later). `niet in FPGA` (not in FPGA): the card has no clipper.
Kanaal (channel)	The channel filter in the FPGA: **breed** (wide) = flat up to 150 kHz, best for stereo and RDS; **200 kHz** = flat up to 90 kHz, steeply down to 104 kHz (default); **190 kHz** = flat up to 85 kHz, steeply down to 98 kHz. On card 3.x, setting 200 kHz, everything from 104 kHz next to the carrier is more than 100 dB lower (calculated for the FPGA; the AD9363 itself adds a little, see Known limitations).

> Clipper, limit, knee and channel are saved and set on the 7020 again at every connect.

Clipper and channel filter: why and how

Why a clipper. The deviation is fixed in the FPGA: the higher the MPX signal, the more deviation. Music has short peaks that go far above the average. If you set the level so those peaks never go above 75 kHz, the transmitter sounds quiet. If you set it higher, the peaks go over 75 kHz: the signal gets wider and the neighbours suffer from it. The clipper solves that: it lets everything through up to the limit and only cuts off the peaks. So the level can go up without the deviation going too far.

The knee. Hard clipping gives sharp corners, and so distortion. With a knee of, for example, 3 dB the clipper already starts to round off softly 3 dB below the limit; only at the limit is it really closed. More knee sounds softer, but starts acting earlier.

Why a channel filter. An FM signal has sidebands that reach further than the channel, especially with stereo and RDS. The channel filter removes everything outside the channel, so nothing more is seen next to your

transmitter in the RF spectrum. A side effect: because the filter removes the outer sidebands, the measured peak deviation comes out a bit higher than the limit of the clipper, at 200 kHz about 4 kHz. That is why the limit is 71 kHz by default: that gives about 75 kHz deviation.

How to set it:

1. Choose `Kanaal 200 kHz`, switch the clipper on, limit 71 kHz, knee 3 dB.
2. Play music that is recorded loud and turn up the potentiometer before the ADC board until the clipper light acts now and then.
3. On the Modulatiemonitor, look at **Piekzwaai** (peak deviation): the highest peak should be around 75 kHz. If it goes above that, set the limit half a kHz lower.
4. Look at the number at the clipper light. A few dB in the loudest parts is fine. If it keeps acting with many dB, the level is too high: the sound gets flat and distorted. Then turn the potentiometer back.
5. After that, check the **MPX-vermogen (BS.412)** (MPX power): with a lot of clipping it goes up. Above 0 dBr is too much.

> Without a clipper you only set the level with the potentiometer: then set it so the highest peak does not go above 75 kHz.

8. Modulatiemonitor tab (modulation monitor)



The modulation monitor.

Piekzwaai

The largest deviation up and down, in kHz and in percent of 75 kHz, held for two seconds. Below that the highest peak according to ITU-R SM.1268 (over 5 minutes) and the MPX power according to ITU-R BS.412 (over 60 seconds, in dBr).

Meters

Each meter has an **orange line: the target**. The number above it changes colour with how far you are from it.

Meter	Target	Explanation
Zwaai (deviation)	max. 100 %	100 % = 75 kHz.
Pilot	9 %	The level of the 19 kHz pilot (9 % = 6.75 kHz deviation).
RDS	3.4 kHz	The deviation caused by the RDS signal.
MPX-vermogen	max. 0 dBr	Average power according to BS.412. Above it, it sounds fuller, but it is not allowed.

MPX spectrum

The spectrum of the MPX signal up to 60 kHz: the mono sum (L+R), the pilot at 19 kHz, the stereo sidebands around 38 kHz and the RDS bump at 57 kHz. Here too there are target lines for the pilot and the RDS bump. With the mouse the crosshair shows the frequency and the level. Above the spectrum are the MPX peak, the pilot, the RDS injection and the **38 kHz-rest** (38 kHz residue): what is left of the suppressed 38 kHz carrier (the lower, the better).

Panels next to the peak deviation

Panel	What it measures
MPX-vermogen (BS.412)	Average MPX power over 60 s, and over the last second.
Pilot	Pilot deviation in kHz and percent, and the exact pilot frequency (should be 19000 Hz).
38 kHz-fase (38 kHz phase)	The error of the stereo subcarrier relative to twice the pilot. Target 0°. This mainly decides how good the channel separation is.
RDS-injectie (RDS injection)	The RDS deviation in kHz and percent of 75 kHz.
Draaggolf (carrier)	How much the carrier differs from the set frequency, in Hz.
Ontvangst (reception)	How strong the measuring signal comes in: S-units, dBµV and dB above the noise (see Ontvanger, S-meter).

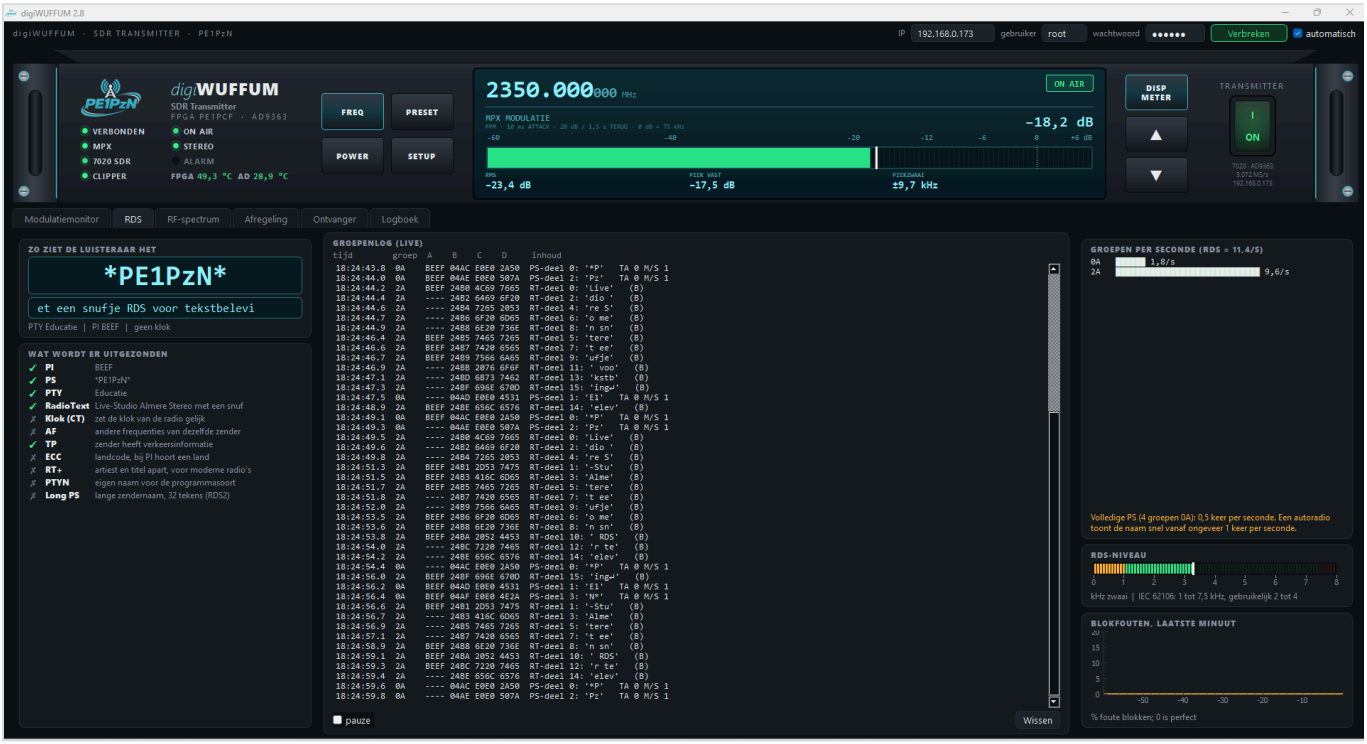
The monitor bar also shows short status labels: *stereo*, *pilot*, *RDS* with the station name, *BS.412 ok* and *overstuurd* (overloaded). They are green when all is well and grey when they do not apply.

Audio and channel separation

The levels of left and right after de-emphasis (50 µs). If you put music or a tone on **one channel**, the difference between L and R is the channel separation. Target: **40 dB or more**. With `30 dB dieper` (30 dB deeper) the meter looks further down, to −60 dB.

> The channel separation is almost entirely set by the phase of the 38 kHz subcarrier relative to the pilot. A few degrees of error already cost a lot of separation.

9. RDS tab



The RDS analyser.

Panel	What you see
Zo ziet de luisteraar het (what the listener sees)	The station name (PS) and the RadioText as a car radio shows them, also scrolling text.
Wat wordt er uitgezonden (what is broadcast)	Everything in the RDS: PI, PTY, PTYN, RadioText A/B, RT+ (artist and title), Long PS, clock (CT), ECC, AF list and the ODAs in use.
Groepenlog (live) (group log)	Every received group. `pauze` (pause) freezes the list, `Wissen` (clear) empties it.
Groepen per seconde (groups per second)	How often each group type comes by. RDS carries 11.4 groups per second in total.
RDS-niveau (RDS level)	The measured RDS deviation.
Blokfouten, laatste minuut (block errors, last minute)	How many RDS blocks came in with errors. With good reception this should be zero.

10. RF-spectrum tab



The RF spectrum around the carrier.

The spectrum of the broadcast around the carrier: green is the average, blue the held peak. The lines show a 200 kHz channel and the Carson bandwidth (264 kHz at 75 kHz deviation); the green area is the occupied bandwidth. `Piek vasthouden wissen` (clear peak hold) starts the peak again.

Panel	What it measures
Bezette bandbreedte 99 % (occupied bandwidth)	The width that holds 99 % of the power, and where it lies relative to the carrier.
Bandbreedte −26 dB (bandwidth)	The width between the points that are 26 dB below the top.
Draaggolf (Bessel-nul) (carrier, Bessel null)	How strong the carrier is relative to the total power.
Spiegelbeeld en LO-lek (image and LO leak)	Only with the test tone on; see Afregeling.

> **Check the deviation with the Bessel null:** put a 31.18 kHz tone on the MPX (without pilot and RDS) and turn up the level. At exactly ±75 kHz deviation the carrier disappears (deeper than −40 dBc). That is a measurement that does not depend on any calculation.

11. Afregeling tab (alignment)



Alignment with the alignment helper on the right.

Why align

An IQ transmitter always makes a little **image** (the signal mirrored on the other side of the carrier) and a little **carrier leakage** (LO leak). How well that is suppressed differs per frequency. Aligning means: setting the sliders so that image and leak are as low as possible.

How to connect for alignment

The 7020 measures its own transmitter. Connect **TX1 to RX2 via a 30 dB attenuator**.

- **Why RX2 and not RX1:** the PCB tracks of RX1 and TX1 run right next to each other. Because of that, RX1 picks up RF from the transmitter directly, bypassing the cable, and then you measure the board instead of the signal. RX2 is further away and suffers much less from this.
- On the Modulatiemonitor, choose `RX2` as input.
- Watch the **Ontvangst** panel: if it turns red (too much signal), use more attenuation or set the transmit power lower.

Transmitting and measuring at the same time is possible with a home-made splitter: three SMA sockets in a star, with in each branch a resistor of **16.5 Ω** (a third of 50 Ω, so each port is 50 Ω).

```
              TX1
               |
            [16.5 Ω]
               |
   +-----------+-----------+
   |                       |
[16.5 Ω]               [16.5 Ω]
   |                       |
antenna / power amplifier   20 dB attenuator -> RX2
```

- Each branch loses 6 dB. So 6 dB less goes to the antenna, and 6 + 20 = 26 dB less to RX2.

- Use small SMD resistors with short leads, as close to the sockets as possible (see the photo).



Home-made splitter: three SMA sockets with three 16.5 Ω SMD resistors in a star.

What is on the tab

Panel	Contents
IQ-correctie in de FPGA (IQ correction in the FPGA)	The registers of the FPGA design: *gain_i*, *gain_q*, *phase* (and *dc_i*, *dc_q* for the carrier leakage, see below). Plus the test tone buttons: `Testtoon 4,8 kHz`, `Testtoon 58,6 kHz` and `Testtoon uit` (test tone off).
AD9363 kalibratie (AD9363 calibration)	The built-in corrections of the AD9363 (*calib_scale_i/q*, *calib_phase_q*), and the buttons `tx_quad kalibratie` and `rf_dc_offs kalibratie`. `Alles neutraal` (all neutral) sets everything back to the starting position.
Afregeling per frequentie (alignment per frequency)	The list of saved frequencies. `Huidige waarden bewaren voor deze frequentie` (save current values for this frequency) and `Wis deze frequentie` (delete this frequency).
Afregelhulp: live gemeten (alignment helper: measured live)	The spectrum around the test tone, with image and LO leak in dBc, in colour. Plus `Automatisch afregelen` (automatic alignment), `Band afregelen` (align band) and `Beste waarden wissen` (clear best values).

Aligning by hand

1. Transmitter on, monitor on (source 7020 or RSPdx).
2. Switch the test tone on: 58.6 kHz is easiest, because then image and tone are far apart. With the test tone on, the program switches to a **fast alignment mode**: it then only measures RF and refreshes quickly.
3. Turn *gain_q* and *phase* (with the `-` and `+` next to them) until the image is as low as possible.
4. For the carrier leakage: turn *dc_i* and *dc_q*. On older cards do the `tx_quad kalibratie` first.
5. Click `Huidige waarden bewaren voor deze frequentie`.
6. Test tone off again.

Automatic alignment

Conditions: connected, monitor on (7020 or RSPdx), transmitter on. Click `Automatisch afregelen`. The program:

1. switches the test tone on and lowers the transmit power if needed, so the receiver is not overloaded (an overloaded receiver makes image itself and then gives a wrong result);
2. tunes the carrier leakage with *dc_i* and *dc_q* (card 3.x; on older cards a *tx_quad* calibration first);
3. finds the best position per slider: coarse first, then fine with steps of 1, just like you do by hand;
4. measures again at every step, so it does not steer on an old measurement;
5. saves the result for the current frequency and sets power and test tone back.

During alignment the button becomes `Stop`. Stopping keeps the last values found.

Band afregelen

`Band afregelen` aligns a whole series of frequencies one after another, every 5 MHz. In the list next to it you choose which band: 431 to 440 (431, 435 and 440), 1240 to 1300 or 2340 to 2400 MHz. That takes a while; after that the transmitter goes back to the old frequency. 430 MHz itself is skipped: there is a spurious line from the 7020 itself there (35 × 12.288 MHz = 430.077 MHz), and the *tx_quad* calibration gives a bad result.

> If you later choose a frequency that is not saved, the program uses the nearest saved frequency. The Logboek reports which one.

> **Saving by hand.** If you align by hand and then click `Huidige waarden bewaren voor deze frequentie`, the last measurement (image and leak) is added to the list. On card 3.x the *tx_quad* calibration usually makes it worse; use *dc_i* and *dc_q* there.

12. Ontvanger tab (receiver)



The receiver with the RSPdx in panorama (5 MHz) on the FM band.



The receiver on your own transmitter (2350 MHz). Bottom left the panel **Ontvanger in de 7020 (DAC)** (receiver in the 7020): audio from the DAC, squelch 9 dB, status with S/N and stereo.

A second receiver, separate from the monitor, to look at other stations. You can listen in two ways: with card 3.x via the receiver in the 7020 and a DAC board (see below), or via a second SDR, the RSPdx, on the sound card of the PC.

Part	Explanation
Bron and Ingang	Which receiver: the 7020 (RX1/RX2), an RSPdx (A/B/C) or Demo. RTL-SDR and HackRF are recognised, but cannot be used yet.
`Zoek apparaten`	(Search devices) Checks again which receivers are connected.
`Start ontvanger`	(Start receiver) Starts or stops the receiver.
volgt zender / eigen frequentie (follows transmitter / own frequency)	Listen on the transmit frequency, or on a frequency of your own. With `▲` `▼` and the step size (1 kHz to 1 GHz) you tune. Or put the mouse on a **digit** of the frequency and turn the wheel, just like with the transmitter.
Beeld (view)	*1,4 MHz*: with audio and RDS. *5 of 9 MHz panorama* (5 or 9 MHz): only spectrum and waterfall, to search for stations. *20 MHz* only works with the 7020. With card 3.x the 7020 makes the wide view by scanning along the band in pieces; that only works when the audio from the DAC is off. If you listen via the DAC, the view stays 2 MHz wide.
Bandbreedte (bandwidth)	*breed (meten)* (wide, measure): for an accurate deviation measurement. *200, 180, 150 of 120 kHz*: narrower, like a radio, against neighbouring stations. You can also drag the bandwidth lines in the spectrum with the mouse.
Versterking (gain)	Automatic or by hand; with the RSPdx also the LNA setting.
Meeluisteren	With volume and choice of sound card. Works via a second SDR, the RSPdx; not with the 7020 as source.
Spectrum	In **dBµV**, measured from the noise floor: the noise floor is always at −14 dBµV. Move the mouse over the spectrum: the cross shows the frequency, the height in dBµV and how many dB that is above the noise. If you put the cross on the carrier, you read the same as on the S-meter.
S-meter	How far the carrier comes above the noise floor. Noise floor = S1 = −14 dBµV, 6 dB per S-unit, so **S9 = 48 dB above the noise = 34 dBµV**. Calibration is not needed: it does not depend on the gain.
Programma (programme)	The station name, RadioText, peak deviation and bandwidth of the received station.
Spectrum en waterval (spectrum and waterfall)	**Click** to tune (snaps to 10 kHz). The **mouse wheel** also tunes. Green line = the receiver, red line = your own transmitter, orange lines = the bandwidth (drag to set).

Ontvanger in de 7020 (DAC), card 3.x

The FPGA of card 3.x has a complete FM receiver: stereo, de-emphasis and a squelch. The audio goes directly to the DAC board on JP5 (see Connecting). Transmitting and receiving work at the same time.

Part	Explanation
Geluid uit de DAC (audio from the DAC)	Switches the receiver in the 7020 on. It listens on the frequency of this tab and on the input chosen at the top (RX1 or RX2).
Squelch	From off to 20 dB. The squelch checks how steady the carrier is: an FM signal has a constant strength, noise does not. Below the threshold the audio closes softly. A few dB is usually enough.
Ruis	Noise reduction, from 0 (off) to 10. From 1 to 5 stereo goes gradually to mono: that is where most of the noise is, and the treble stays. From 6 to 10 the treble also goes down step by step (from 12 kHz to 5 kHz): less hiss. Card 3.2.
mono	Always mono, also when there is a pilot. Less noise with a weak signal.
75 µs (VS) (US)	De-emphasis of 75 µs, as in America. In Europe this is off (50 µs).
Status	Frequency, level in dBFS, signal-to-noise ratio (S/N), stereo with the pilot level, and whether the squelch is open.

> If you listen to your own transmitter, use **RX2**: RX1 is right next to TX1 on the board and picks up more directly.

> If you choose the same device for the monitor and the receiver, they share it. They are then always on the same frequency with the same settings.

13. Logboek tab (log)

Everything the program does, with the time: connecting, frequency changes, transmitter on and off, alignment results and error messages. Look here when something does not do what you expect.

14. Lights and alarms

The **ALARM** light comes on with:

Message	What is going on
zwaai … kHz (boven 75) (deviation above 75)	The peak deviation goes above 75 kHz. Switch the clipper on, set the limit lower, or the MPX level lower (monitor bar).
MPX-vermogen … dBr (boven 0) (MPX power above 0)	The average MPX power (BS.412) is too high.
temperatuur boven 75 °C (temperature above 75 °C)	The FPGA or the AD9363 is getting too hot.
ontvanger overstuurd (receiver overloaded)	The measuring signal is too strong: more attenuation between transmitter and receiver, or lower LNA.
geen nieuwe meetdata (no new measurement data)	The monitor no longer gets measurements. Check the connection or the receiver.

15. Known limitations

- **Listening with the 7020** works via the receiver in the FPGA and a DAC board (card 3.x). Listening on the sound card of the PC works via a second SDR, the RSPdx: the 7020 sends its signal to the PC in pieces, with gaps between them. That is fine for measuring, not for listening.
- **Wide panorama** (5, 9 or 20 MHz) with the 7020 only works when the audio from the DAC is off: the transmitter, the monitor and the receiver in the 7020 share one tuning.
- Do not set the power higher than **−4 dB**; at 0 dB the AD9363 distorts more.
- At an exact multiple of 40 MHz (for example 1280.000 MHz) the PLL of the AD9363 makes a small wobble. Choose a few MHz off.
- At about ±700 kHz from the carrier there are two small lines of about −56 dBc. That is within the limit for radio amateurs (−50 dBc).
- RTL-SDR and HackRF are recognised, but cannot be used as a receiver yet.