RPG-FMCW-94-SP/DP
94 GHz W-band Cloud Doppler Radar
Instrument Installation, Operation and Software Guide (Version 2.10-1)

Edited by©: Th. Rose, Radiometer Physics GmbH, Germany

Document Change Log

Date	Issue/Rev	Change
25.10.2015	00/01	Work
30.10.2015	01/00	Release
03.08.2016	01/01	Adding chapter 2.11 as a description of polarimetric observation data
16.11.2016	01/02	Update for software version 2.0 and adding installation chapter
10.01.2016	01/02	Update for software version 2.1, adding quicklooks

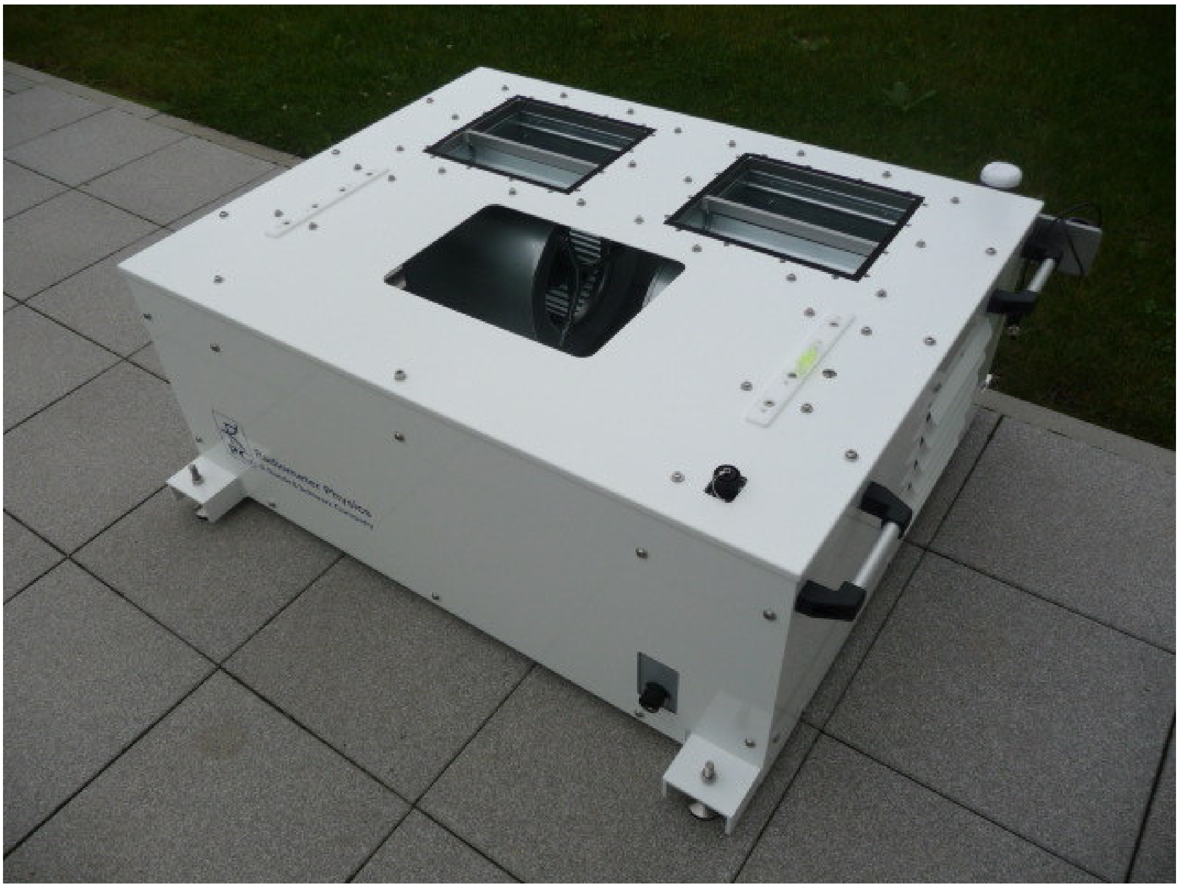
Table of Contents

1 Instrument Installation

The RPG-FMCW-94-XP radars are delivered with a complete set of accessories to operate the radar in an outdoor environment and to perform regular instrument calibrations. This includes (in addition to the radar itself) a radar stand with rain mitigation system, a set of power and data cables, a signal converter, an external calibration target and target frame, a weather station plus GPS receiver, a complete software package for the radar PC (R-PC) and external host PC (H-PC) and several tools to assemble the system.
As optional hardware an elevation / azimuth scanner can be added that replaces the radar stand (which is only configured for zenith observations).

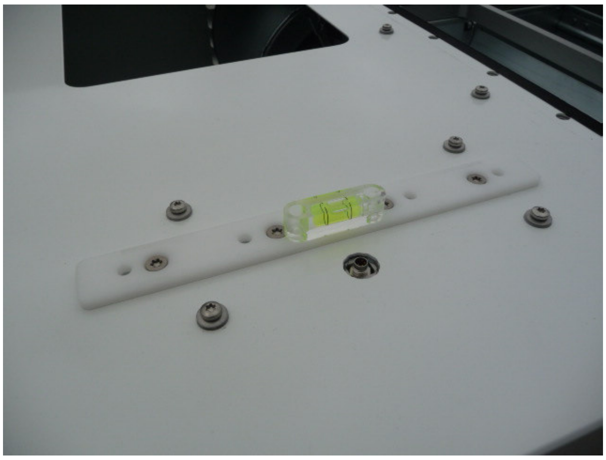
1.1 Setup of Zenith Observation System

The installation starts with the setup of the radar's stand. The stand contains the strong blower fans and the electronics for controlling the blower switching from the radar.



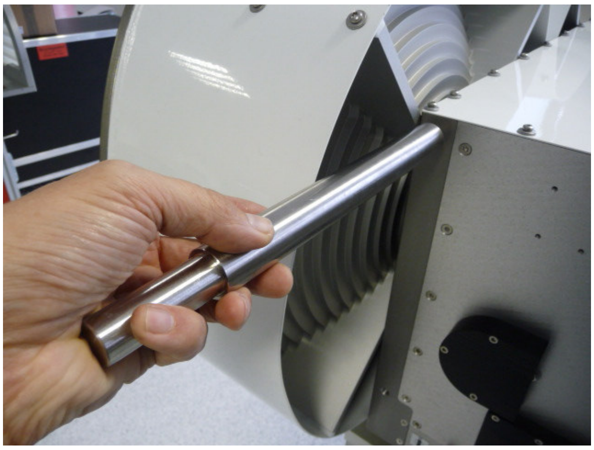
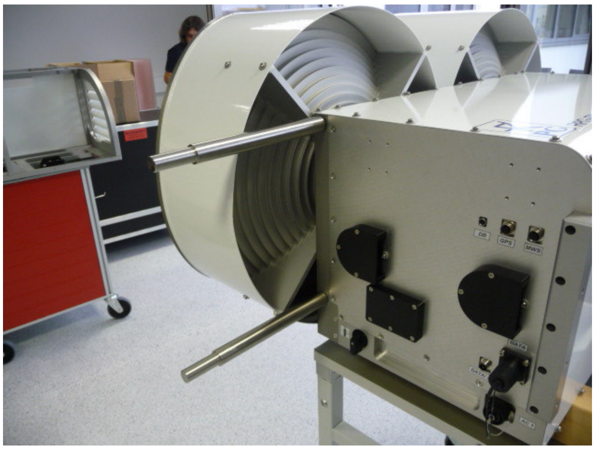
Zenith observation radar table with blowers and power switching interface.

The stand has four adjustable feet for horizontal alignment:

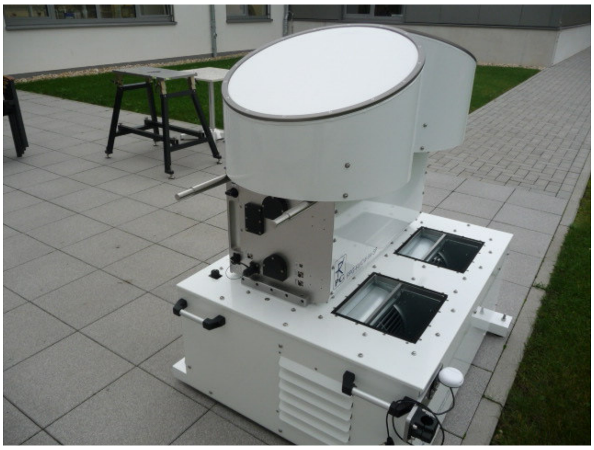
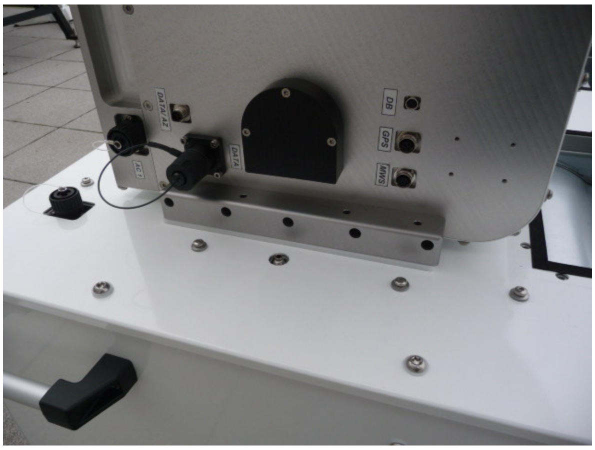



Any type of spirit levels may be used to measure the horizontal alignment in both directions of the table. A metric 18 mm wrench is needed to turn the adjustable feet and to finally lock the nuts.

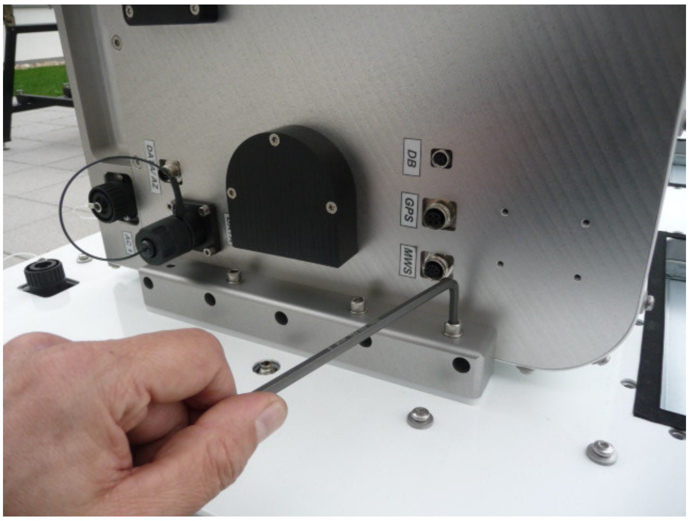
The next step is to unpack the radar body and mount the four grab poles to the four M10 threads:

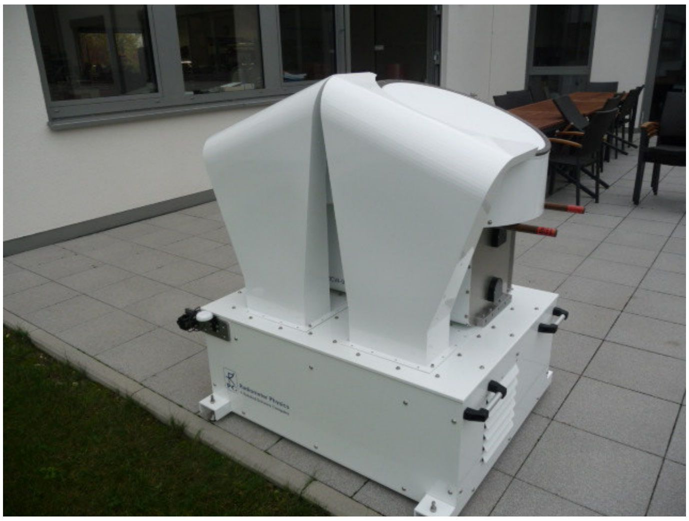
The radar body's weight is close to 90 kg so that four people are required for carrying it to the table and aligning it on the two white plastic slides:

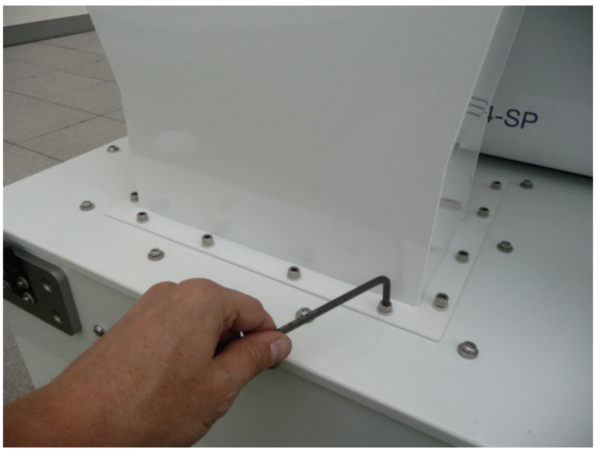
Fasten the radar body to the table with 8 M6 x 45 screws (four on each side):



After unpacking the two blower air baffles, position them on the table as following:

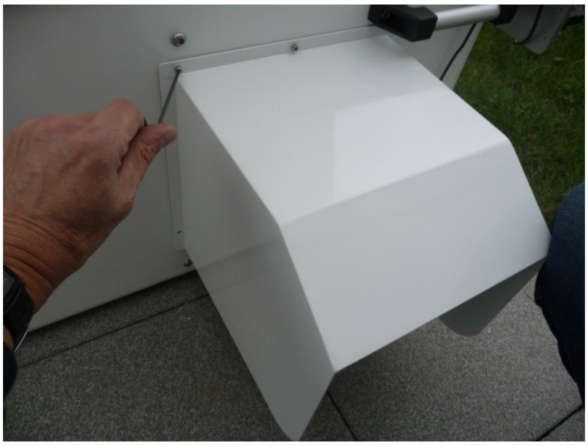


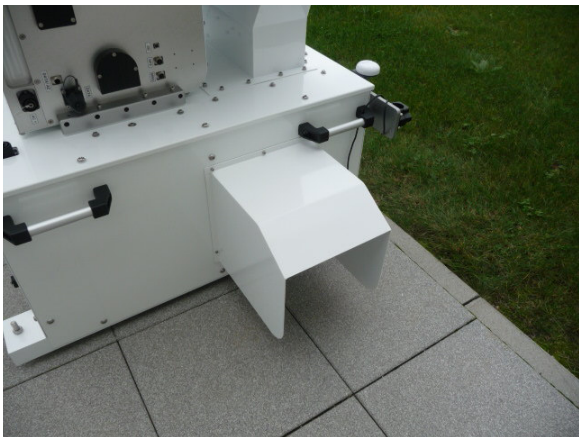
Fasten the air baffles to the window frames first and then to the table:



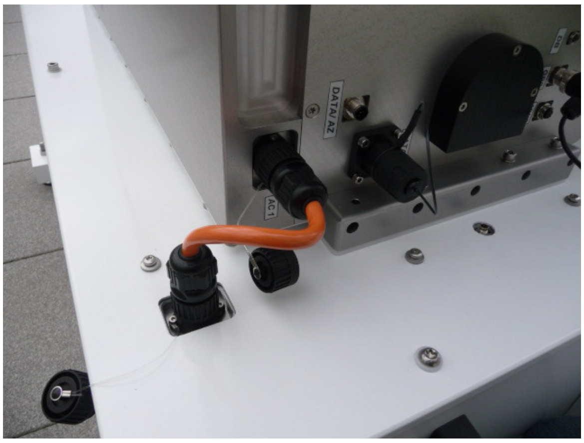
Use 6 M4 counter-sunk screws with plastic washers (3 on each window frame) to fix the air baffles to the radar antenna structure. Then fasten them to the table (16 M6 x 12 screws for each baffle).

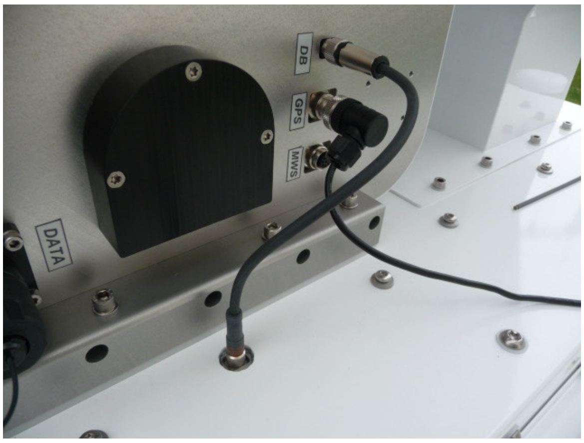
Then mount the two air inlet screens to the table's side walls:



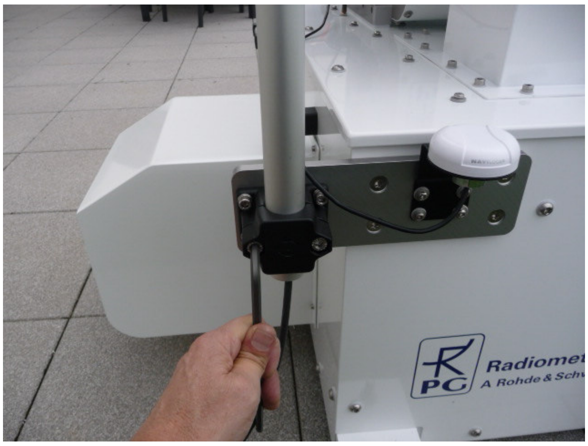


Connect the radar's power input socket to the table's supply socket using the short orange cable. Also connect the 'DB'-socket (Dew Blower) to the table. This connection allows the radar to switch the blower / heater on and off, depending on the current rain status and surface rel. humidity level.





Attach the weather station (Vaisala) to the side panel and connect the GPS receiver and weather station to the radar:



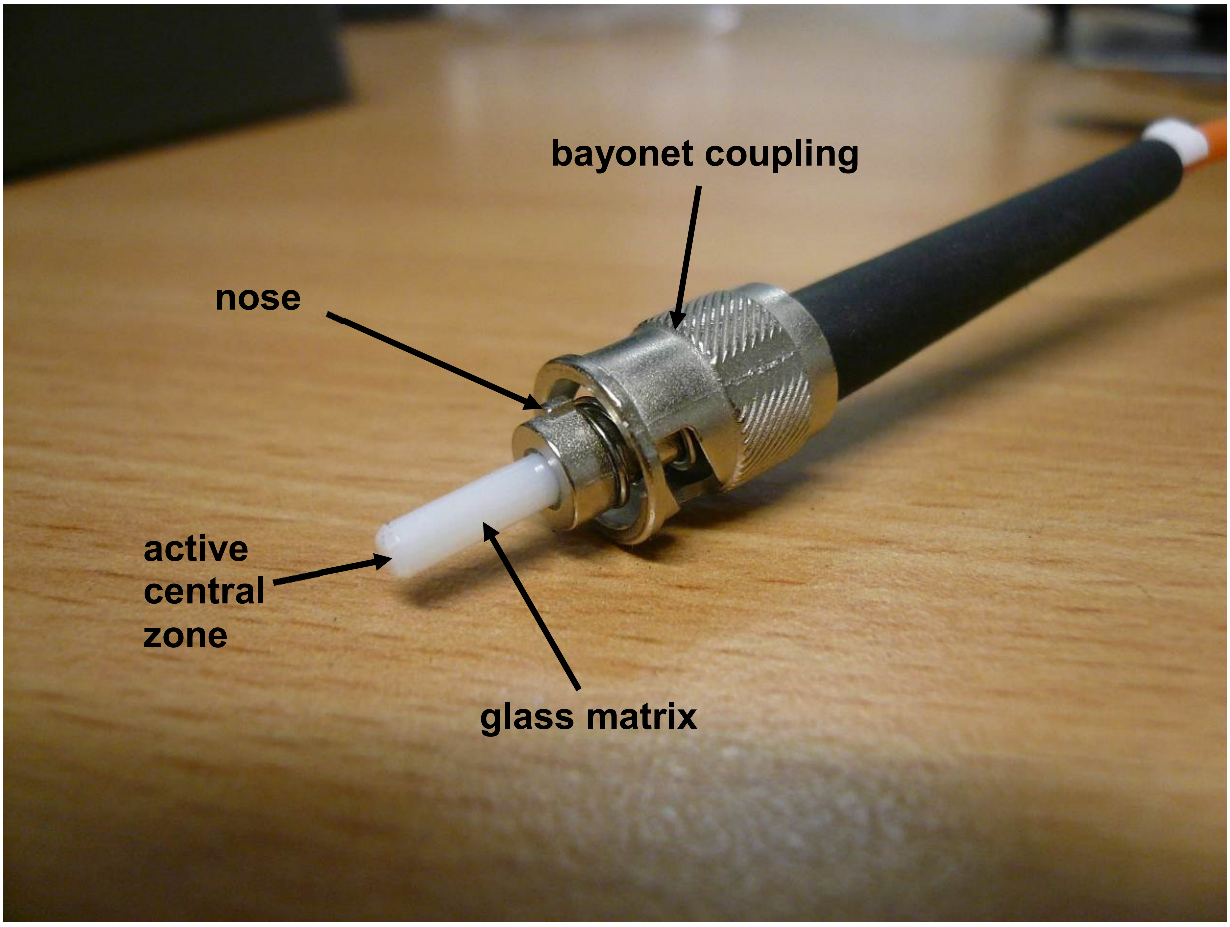
Now connect the fibre optics data cable to the radar:

Connecting the fiber optics data cable.

Make sure that the plug slides smoothly into the socket and finally turn the outer cylinder of the plug clockwise (by applying some force!) to seal the connector from water. Without sealing the connector, the connection will not be water proof!

The other end of the fiber cable is connected to a MOXA Fiber-to-LAN-TCP/IP converter as indicated below. Each of the two line ends has a nose which fits into the fiber socket. After the connector is sliding into the socket (the nose guided by the slit), the bayonet coupling has to be pushed against a spring inside the coupling and then turned clockwise.

MOXA Fiber-to-LAN-TCP/IP converter.



Details of the glass fiber connector.

The converter has an external power cable. When the power cable and the two fiber lines are connected, the power LED and FX LED are turned on. Make sure that the TX fiber line (orange) gets connected to the TX converter output and the RX fiber line (blue) to the RX converter input. Via LAN-TCP/IP connection the converter can be connected either directly to the Host PC or to a network. If the LAN cable is connected, the power and FX LEDs are on and the 100M and TP LEDs are flashing.

2

2 line fiber optics to LAN-TCP/IP converter.

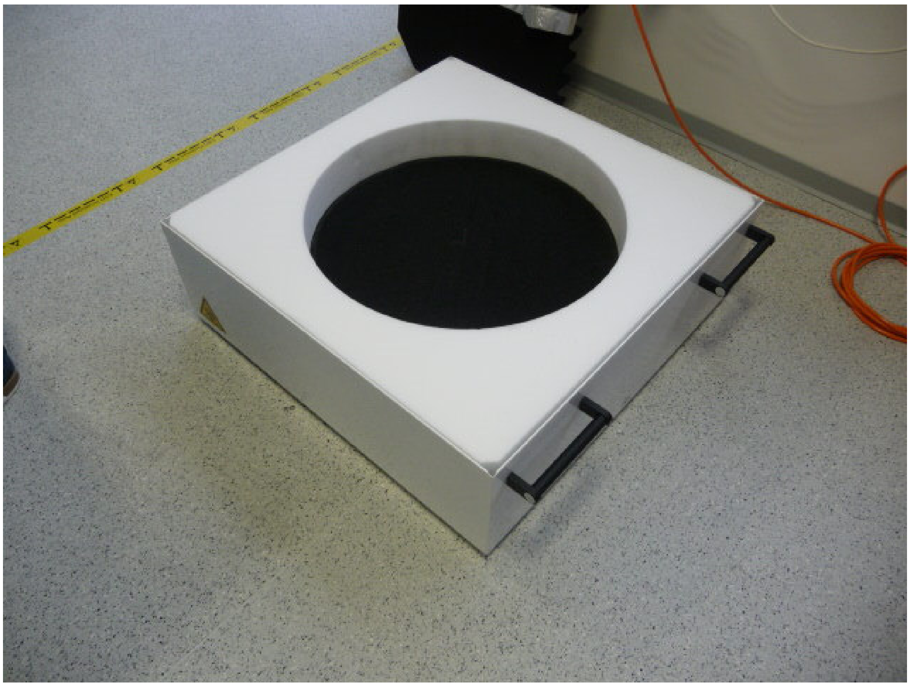
Finally, connect the power cable to the radar's power socket: This finishes the hardware installation procedure.

Connect the Ethernet cable of the converter to a host PC (with pre-installed radar software) and follow the procedure in chapter 3.2 (Getting Started) in order to establish a data connection to the radar.

1.2 Setup of a Scanning System

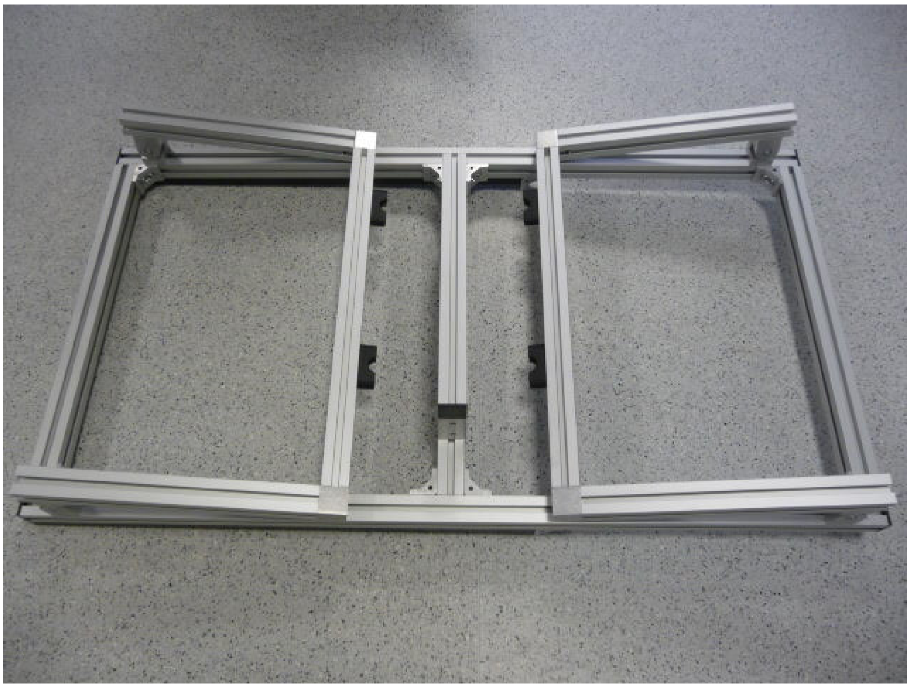
1.3 Absolute Calibrations

As described in section 2.5, the radar receiver and its direct detection channel need to be calibrated by two absolute temperature standards, namely a target at ambient temperature and another one at liquid nitrogen (LN2) temperature. The following calibration procedure assumes that the radar has been turned on and that a host PC is connected to it while its radar software is communicating with the radar. Let the radar warm up for 30 minutes before the calibration is started.
The cold calibration target can be either the cloud free sky with known sky temperature at the radar's frequency (measured by a 90 GHz radiometer, if available) or a LN2 cooled absorber stored in a thermally insulating container:

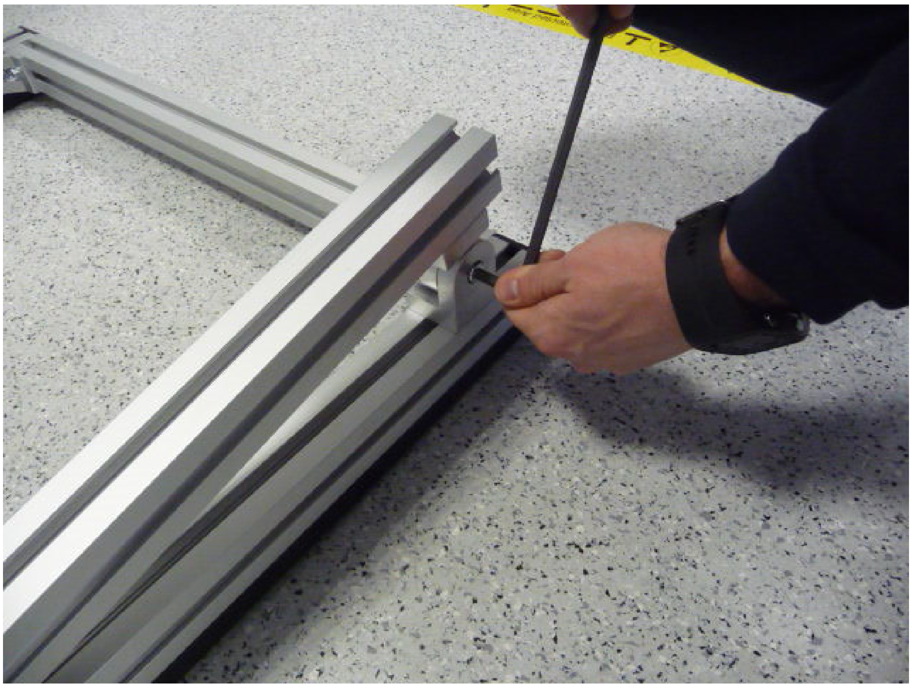


LN2 calibration target.

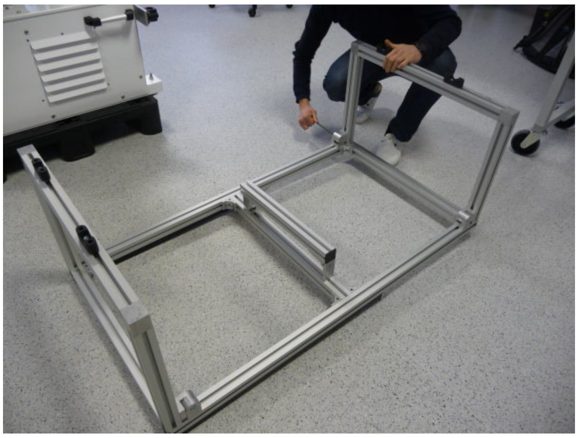
Before the LN2 target can be used for calibration, a mounting frame has to be installed. This frame is delivered together with the LN2 target in a folded configuration:



The next photo sequence describes how to setup the target frame.

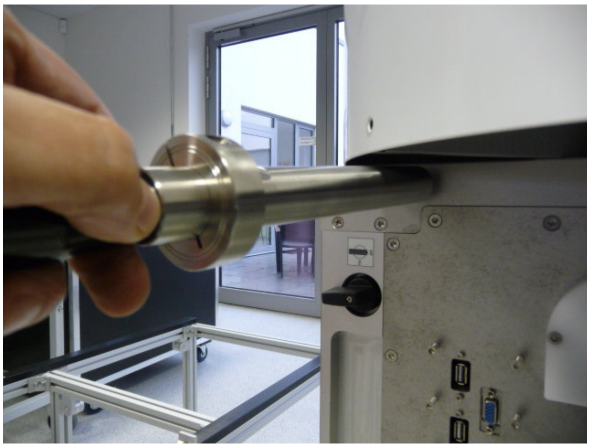


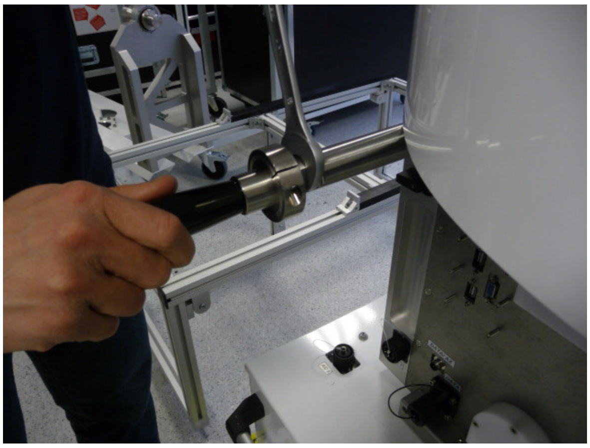
Unclamp the frame's hinges and unfold it:



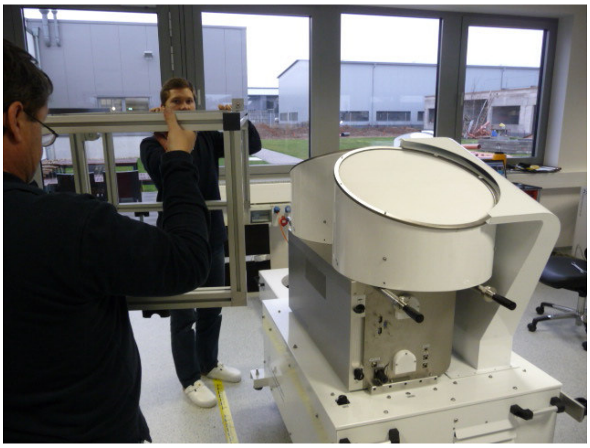

After unfolding the frame, tighten the hinge screws.

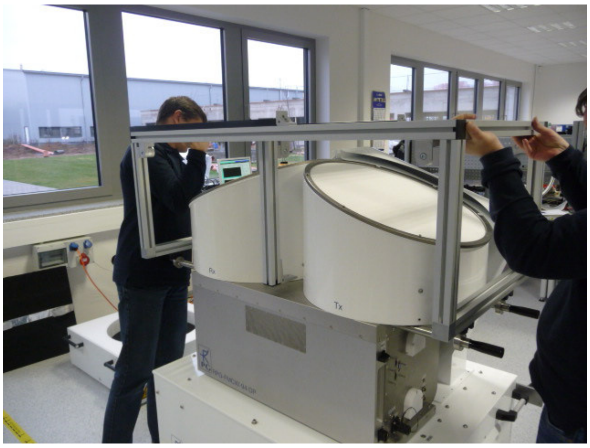
Mount the four rods with the black handles to the radars housing's M10 threads. Use a wrench (M19) to fix the rods firmly to the radar housing:

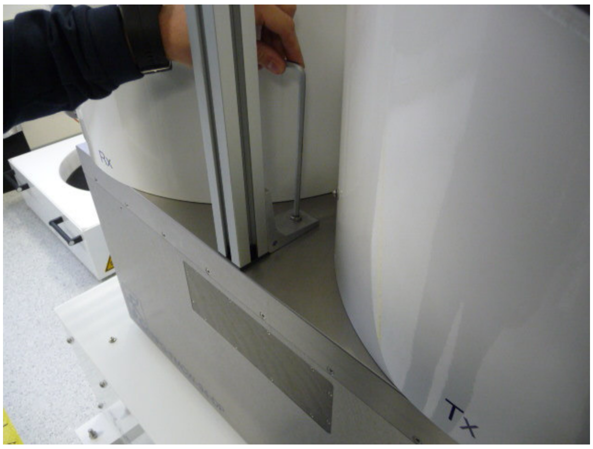




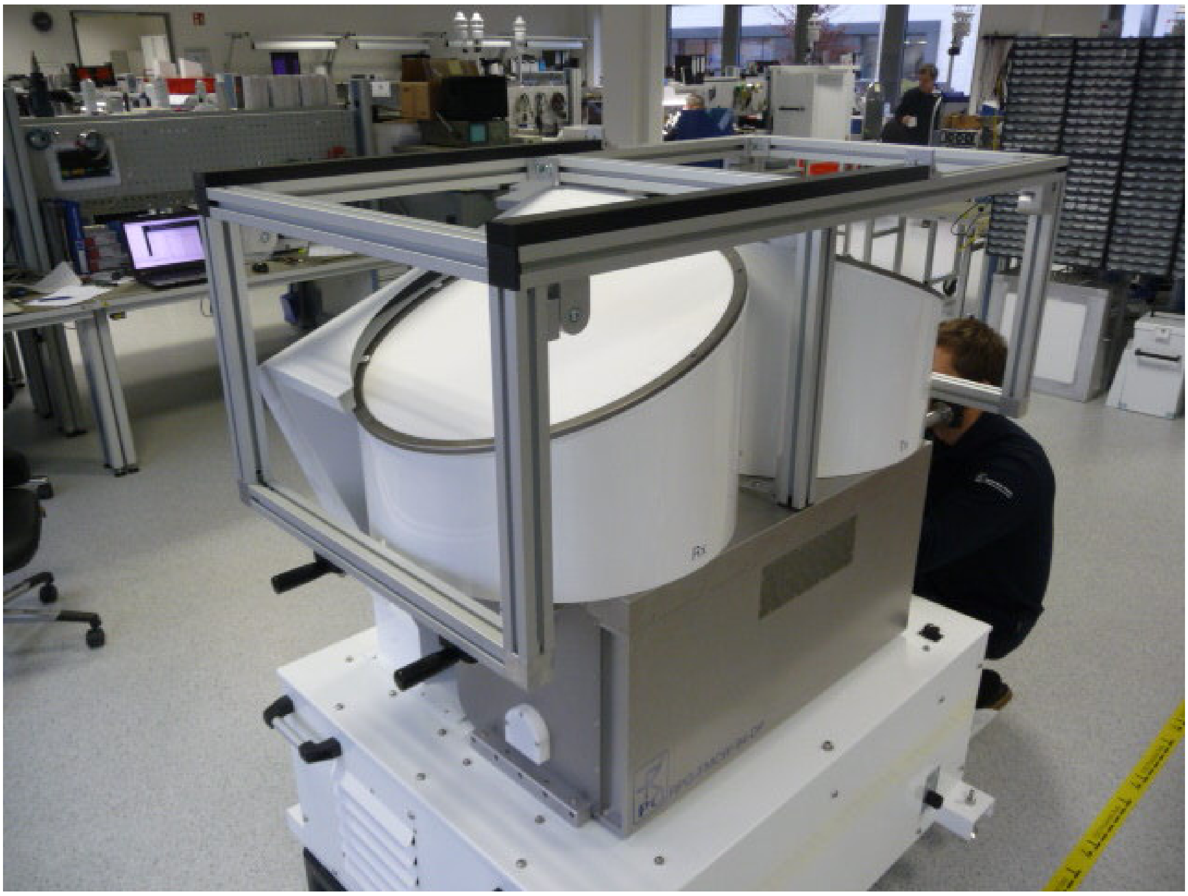
Move the frame to the radar (two persons are required) and rest it on the four rods:



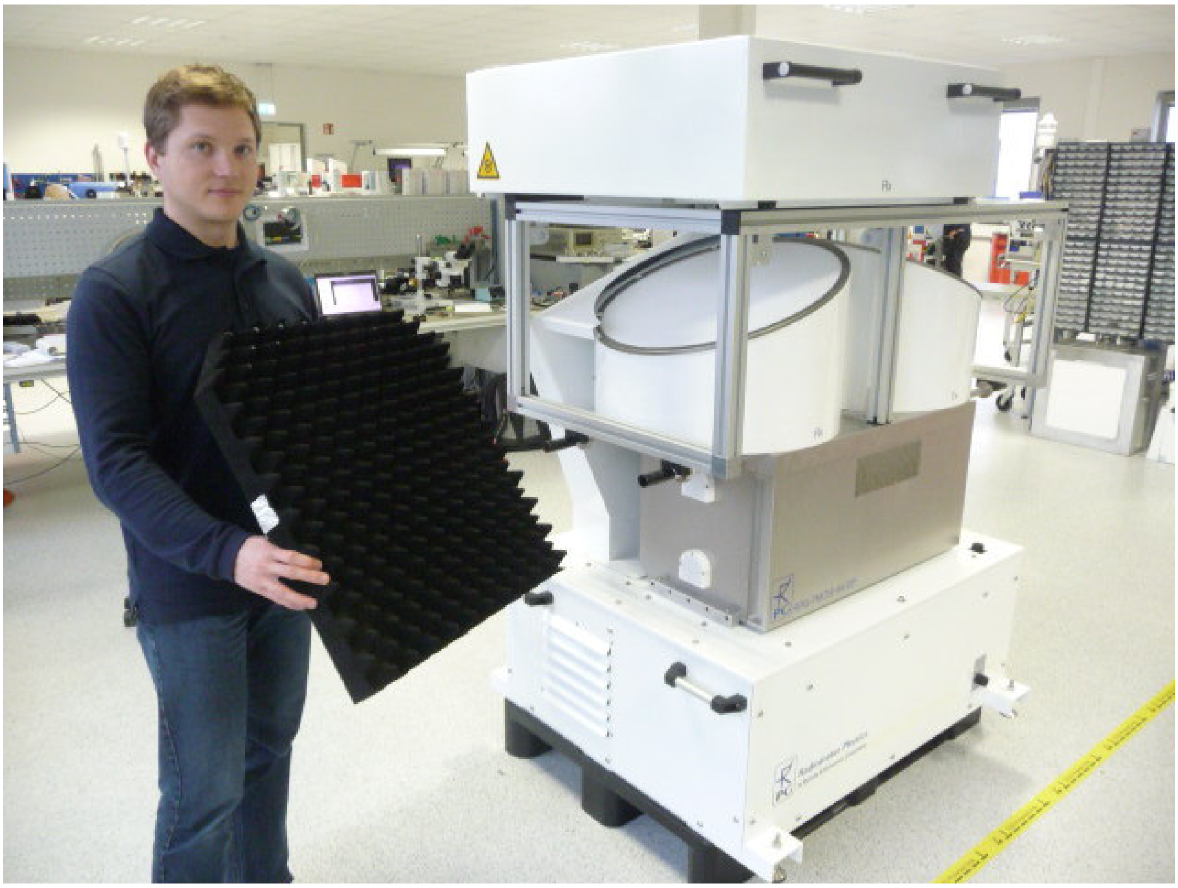


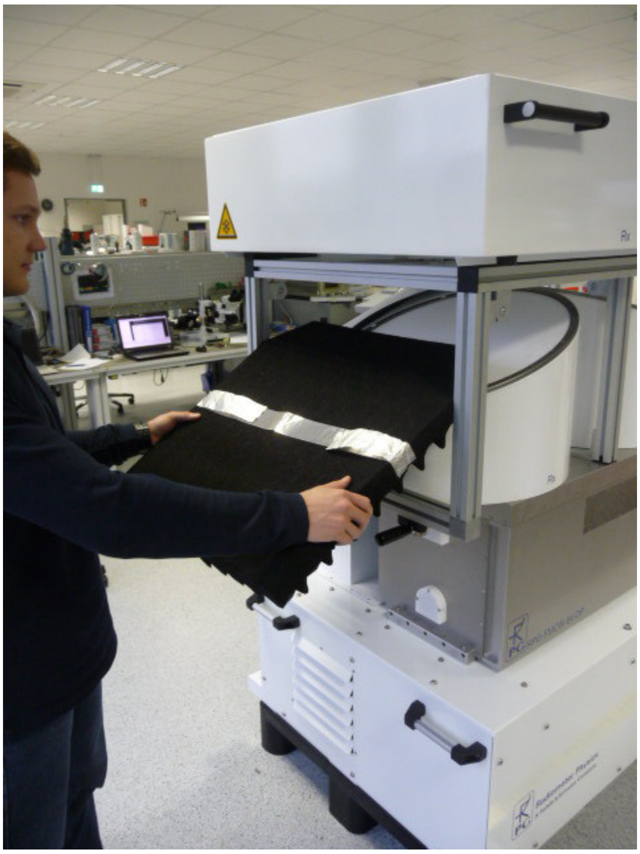



Finally fasten the target frame as indicated above.

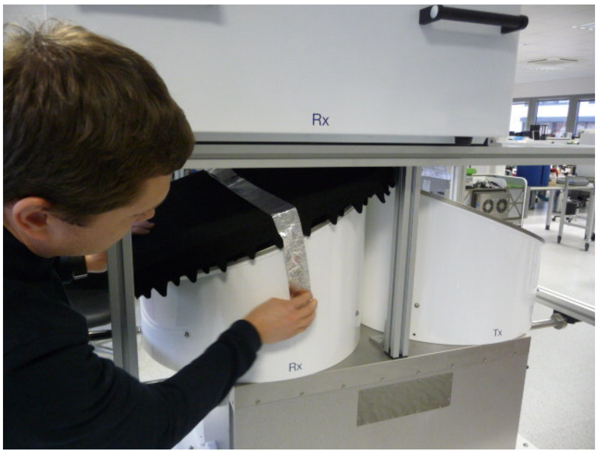


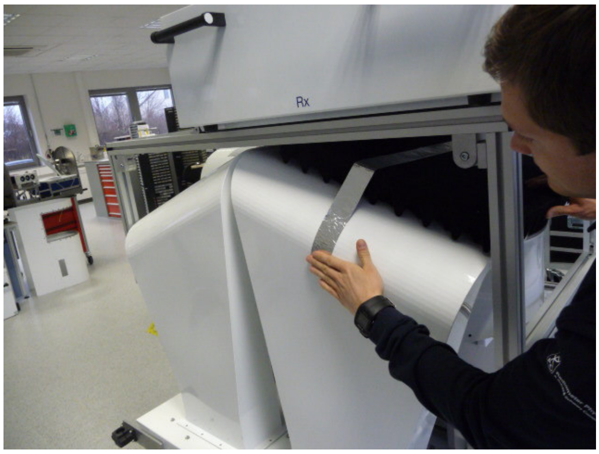
Now slide the empty LN2 target onto the black plastic sliding bars and take the ambient temperature target (a simple black foam absorber) in order to slip it over the receiver (RX) window (the pyramidal structure facing down towards the window):



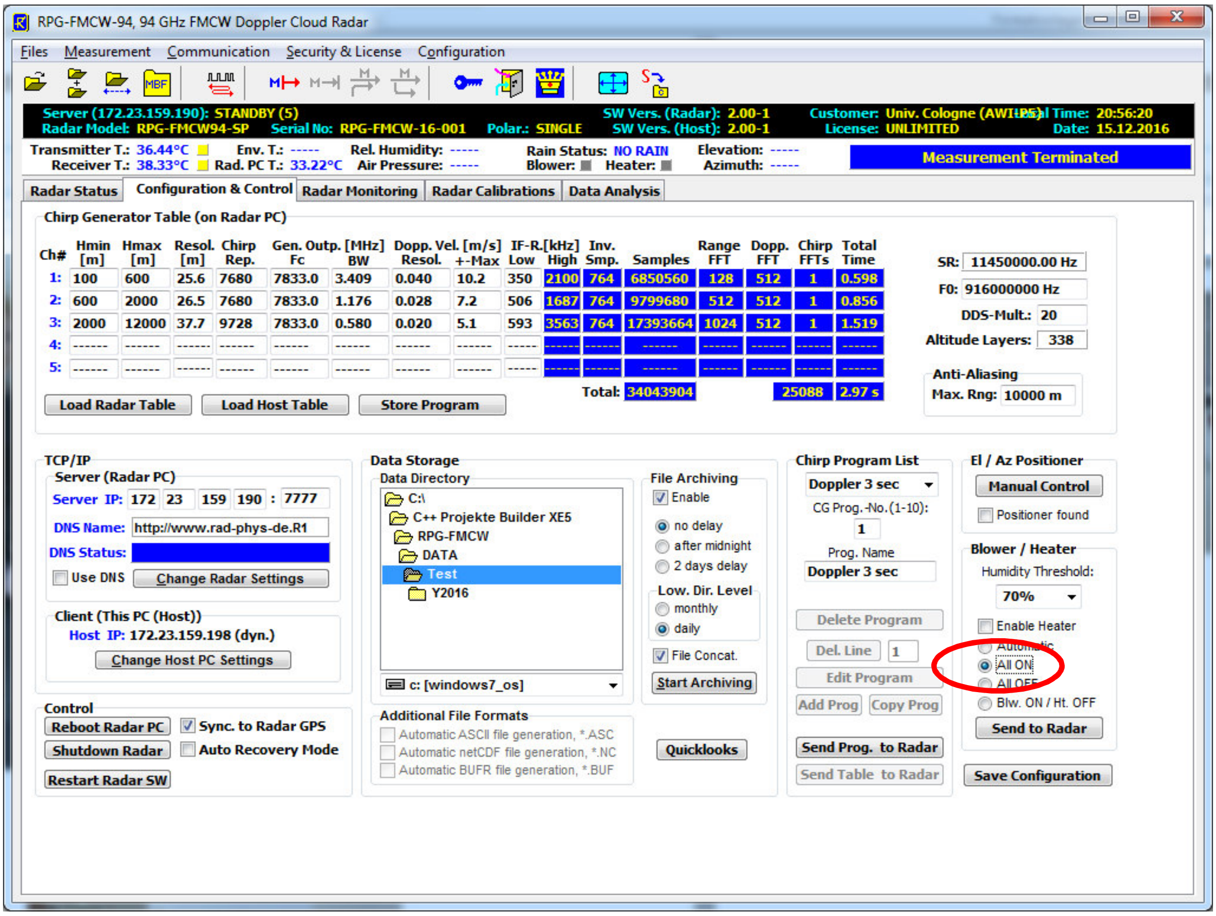


Fasten the ambient target with a tape on both sides of the RX radiation shield and baffle:



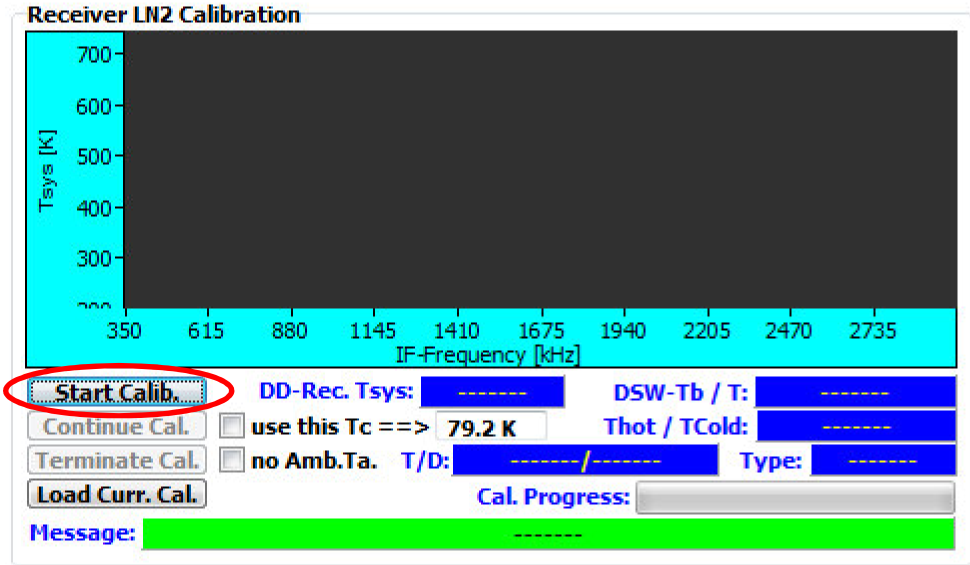


Now turn on the blower (if it is not already running):



The blower will cool the absorber to the environmental temperature within a short time (about 1 minute) due to the low thermal capacity of the absorber material.

After the ambient target is thermally stabilized, click on the ***Radar Calibrations*** tag:



Start the calibration with ***Start Calib.***. The radar will then integrate on the environmental temp. target for about one minute while measuring the environmental temperature through the weather station. When the integration period has passed, the user is requested to change to the LN2 calibration target. Remove the ambient temp. target first before filling the LN2 target with liquid nitrogen:

Use the small blue polyurethane box for filling the target. This is more convenient than handling a LN2-dewar. **Only fill the target to half full until the black absorber is covered with the liquid! If the target is completely filled, it is too dangerous to remove it from the target frame after calibration due to spilling liquid nitrogen. The target will also be too heavy in this case.**

Then go back to the radar calibration box on the host PC and click ***Continue Cal.*** to start the integration on the cold target. When this is finished the radar will automatically add another integration cycle on the Dicke Switch (turned on). This completes the absolute calibration and the results are grafically displayed
The cold target has 4 handles to be moved from the target frame after calibration. The black plastic bars simplify the sliding of the target from the frame.

Pay attention to general safety guidelines while using liquid nitrogen for calibration. RPG is not responsible for injuries caused by improper handling or insufficient clothing.
The minimum requirements regarding clothing are:

1. **Cryogenic hand gloves**
2. **Plastic apron covering chest, legs and arms**
3. **Face protection shield**
4. **Closed shoes**

2 Theory of Operation

2.1 The Radar Equation

The radar range equation represents the physical dependences of the transmit power and the wave propagation up to the receiving of the echo-signals. The power P_R returning to the receiving antenna is given by the radar equation, depending on the transmitted power P_T, slant range R and the reflecting characteristics of the target (described as the radar cross-section σ). At known sensitivity of the radar receiver the radar equation determines the maximum range theoretically achieved by a given radar setup. Furthermore one can assess the performance of the radar setup with the radar equation.

In this context we assume that electromagnetic waves propagate through the atmosphere under ideal conditions, i.e. without dispersion.

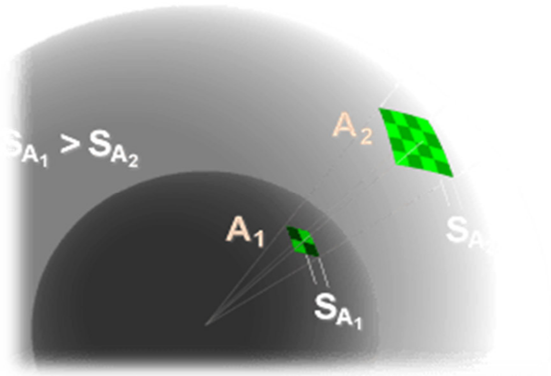


Figure 1: Non-directional power density diminishes as geometric spreading of the beam.

When microwave energy is emitted by an ideal isotropic radiator, it propagates uniformly in all directions. Therefore, areas with the same power density are forming spheres (A= 4 π R²) around the radiator. The power density on the surface of a sphere is inversely proportional to the square of the sphere's radius.

The non-directional power density is given by:

$$P_d = \frac{P_T}{4\pi R^2} \quad , P_T = transmitter\ power$$

The transmitter's antenna gain G_T leads to an amplification of the power density along the beam axis

$$\tilde{P}_d = P_d G_T$$

Typical antenna gains of high frequency radars are in the order of 10^5 or higher.
At the target location, the amount of reflected power is determined by the target's scattering cross section σ:

$$P_r = \frac{P_T G_T}{4\pi R^2}\sigma$$

On its way back to the receiver, the reflected power is diluted in the same way as the power travelling from the transmitter to the target with the effective antenna aperture A replacing σ in the formula above and assuming that the scattering target is acting as an isotropic radiator. The power P_R received by the radar is then given as:

$$P_R = \frac{P_T G_T}{(4\pi)^2 R^4}\sigma A$$

Using the well known relationship between antenna gain and effective aperture

$$A = \frac{G_R \lambda^2}{4\pi}$$

we end up with the general form of the radar equation (assuming $G_T = G_R$):

$$\frac{P_R}{P_T} = \frac{{G_T}^2 \lambda^2}{(4\pi)^3 R^4}\sigma \qquad (2.1.1)$$

The details of the scattering process are completely contained in the cross section σ. In the case of a point target and assuming the target size to be large compared to the radar's wavelength, σ is proportional to the geometrical cross section. For instance, a metal sphere of radius r has a cross section $\sigma = \pi r^2$.

If the target is a volumetric scattering medium (like a cloud) with the transmitter power density distribution close to a Gaussian distribution (the overlap of the RPG-FMCW radar's antenna beam pattern with a Gaussian pattern is 95%), the scattering volume is given by

$$V = \pi \left(\frac{\theta R}{2}\right)^2 \frac{\delta R}{2\ln(2)} \qquad (2.1.2)$$

δR is the range resolution and Θ is the antenna's half power beam width in radians. The reflectivity in cloud radar applications is often expressed in terms of an equivalent Rayleigh back scattering cross section:

$$\sigma = \frac{\pi^5}{\lambda^4} K^2 \sum_i^V {D_i}^6 \qquad (2.1.3)$$

The D_i are the various particle diameters in the scattering volume V and the sum is taken over this volume. K is given by the complex dielectric constant ε of the medium as:

$$K^2 = \left(\frac{\varepsilon - 1}{\varepsilon + 2}\right)^2$$

For pure water, |K| is close to 0.86 at 90 GHz.

It is common practice in cloud radar applications to define the equivalent reflectivity Z_e, which is wavelength independent and can be used to compare reflectivity data from radars operating at different frequencies:

$$Z_e = \frac{10^{18} \cdot \sum_i^V {D_i}^6}{V} \quad , \qquad [Z_e] = \frac{mm^6}{m^3} \qquad (2.1.4)$$

Combining equations (2.1.1), (2.1.2), (2.1.3) and (2.1.4) we get:

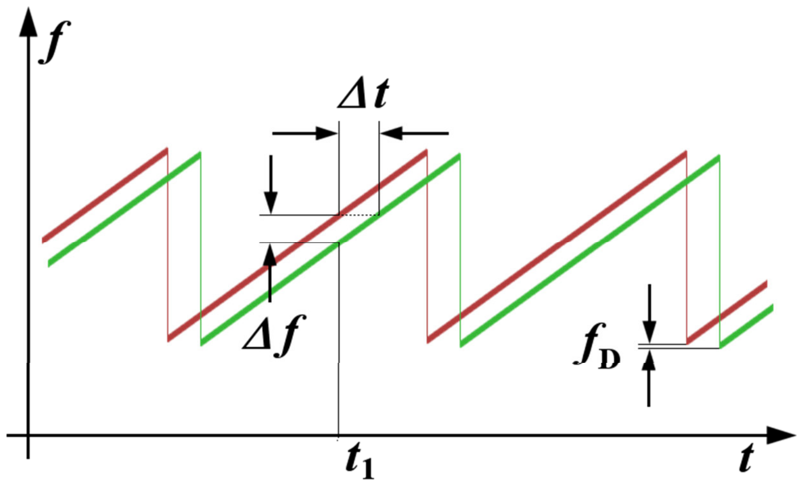
$$\frac{P_R}{P_T} = \frac{{G_T}^2 \lambda^2}{(4\pi)^3 R^4} \frac{V \pi^5 K^2}{10^{18} \lambda^4} Z_e = \frac{{G_T}^2 \theta^2 \pi^3 K^2}{5.12 \cdot 10^{20} \ln(2)\, \lambda^2} \frac{\delta R}{R^2} Z_e = C_R \frac{\delta R}{R^2} Z_e \qquad (2.1.5)$$

C_R is the radar constant summarizing the optical radar parameters (G_T, Θ, λ) and dielectric properties of the medium (K^2, assumed to be constant).

2.2 FMCW Radar Operation

In contradiction to traditional pulsed radars, FMCW radars (Frequency Modulated Continuous Wave) do not require a concentrated high power transmission of several kW peak power. Instead, the transmitter power is emitted continuously, but with linearly varying frequency. The advantages are:

- Low transmitter power in the order of a few Watts. Radar networks in urban areas are possible due to ease of deployment approval for low power transmitters.
- 100% integration duty cycle
- Transmitter can be realized in solid state technology, leading to long lifetime and low cost
- Radar calibration is more accurate compared to pulsed radars, because the transmitter power can be determined more accurately.
- No high power magnetron tubes and high voltage power supplies are required, reducing the overall radar power consumption.
- High flexibility of range/Doppler resolution variation along the ranging path
- Lower costs for the overall radar



The FMCW radar is continuously emitting saw tooth frequency chirps as indicated by the red curve in the diagram above:

Here the green curve represents the reflected echo of a point source as received by the radar. The time delay between the two curves is proportional to the target distance R:

$$\Delta t = \frac{2R}{c} \;, \qquad c = speed\ of\ light$$

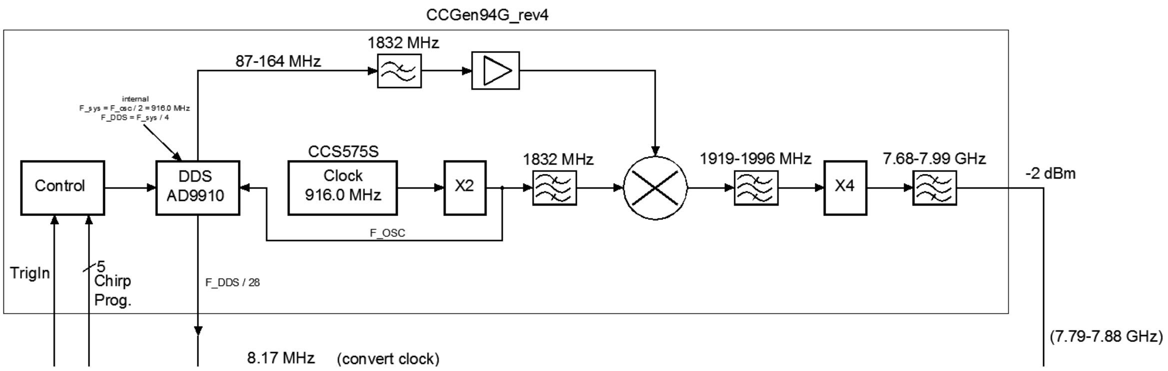
The frequency mixer inside the radar measures the frequency difference Δf (IF spectrum) which is given by the chirp's ramp slope S = Δf / Δt:

$$\Delta f = f_{IF} = \frac{2S}{c}\, R \,, \qquad S = \frac{B}{T_c} \qquad (2.2.1)$$

B = chirp bandwidth, T_c = chirp duration. The frequency resolution δf for measuring f_{IF} is 1 / T_c. Therefore

$$\frac{f_{IF}}{\delta f} = \frac{2B}{c}\, R = \frac{R}{\delta R} \,, \qquad \delta R = \frac{c}{2B} \quad (range\ resolution) \qquad (2.2.2)$$

In RPG's FMCW radars the chirp sequences are generated by a chirp generator module based on a DDS (Direct Digital Synthesizer):



The DDS is clocked by a fundamental oscillator of frequency f_0 = 916 MHz. The chirp generator generates an output signal f_{cg} in the range 7.79 GHz to 7.88 GHz which is then multiplied by 12 to become the transmitter signal at W-band (94 GHz). f_{cg} is derived from the DDS output frequency f_{DDS} by the following relation:

$$f_{cg} = 4 \cdot (2f_0 + f_{DDS})$$

as can be seen from the chirp generator layout above. The DDS's ramp generator uses an internal time base

$$\Delta T_{RG} = \frac{4}{f_0} \qquad (2.2.3)$$

and a frequency resolution (referenced to the radar's transmitter output at W-band) of

$$\Delta f_{RG} = \frac{48 \cdot f_0}{2^{32}}$$

In order to generate frequency ramps of slope S as in equation (2.2.1), the generator can be programmed with an integer step size number S_z:

$$S = S_z \frac{\Delta f_{RG}}{\Delta T_{RG}} = S_z \frac{3 f_0{}^2}{2^{30}}$$

S_z can be computed by using equation (2.2.1):

$$S_z = \frac{2^{29} c}{3 f_0{}^2} \frac{f_{IF}}{R}$$

The factor f_{IF} / R is mapping the IF frequency range to the distance range. When a range resolution δR shall be achieved, the number of ramp steps N can be derived from (2.2.2):

$$\delta R = \frac{c}{2 S_z \Delta f_{RG} N}$$

Because of (2.2.3) the chirp duration is then fixed to be

$$T_c = \frac{4N}{f_0} \qquad (2.2.4)$$

The mixer's IF voltage signal is sampled by a fast ADC board and directly converted to binary code. The sampling clock for the ADC board is generated by the chirp generator as well to ensure a perfect synchronization of ramp generation and ADC sampling. The generator's sampling frequency is given by

$$f_{SR} = \frac{f_0}{4M} \quad , \qquad M = 4 * (n + 1) \;\; , \;\; n = integer\ number \geq 1$$

This limits the sampling rate to a maximum of f_0 / 32 = 28.625 MHz.
The number of samples in a single chirp N_{FFT} is converted to a digital frequency spectrum by an FFT (Fast Fourier Transform):

$$T_c = \frac{N_{FFT}}{f_{SR}} \quad , \qquad N_{FFT} = \frac{N}{M} \qquad (using\ (2.2.4))$$

2.3 Power Spectrum

The mixer output voltage signal is digitized with a sampling rate f_{SR} and the output spectrum is determined by transforming the voltage time series via a Fourier transform for a limited period of N_{FFT} samples:

$$V(t) \quad ==> \quad S(f_{IF}) \qquad (2.3.1)$$

Due to the discrete form of Parceval's theorem the mean power of the time series and spectrum over the sampling interval is identical:

$$P_m = \frac{1}{N_{FFT}} \sum_{k=1}^{N_{FFT}} |V_k|^2 = \frac{1}{{N_{FFT}}^2} \sum_{n=1}^{N_{FFT}} |S_n|^2 \qquad (2.3.2)$$

Because V_k is a sequence of real numbers, the following symmetry applies to the complex Fourier components:

$$S(-f_{IF}) = [S(f_{IF})]^* \quad , \quad f_{IF} \leq \frac{f_{SR}}{2} \quad (Nyquist\ Limit) \qquad (2.3.3)$$

Therefore, the total power sum on the right side in (2.3.2) can be written as:

$$P_m = \frac{2}{{N_{FFT}}^2} \sum_{n=1}^{N_{FFT}/2} |S_n|^2 = \sum_{n=1}^{N_{FFT}/2} P_n \qquad (2.3.4)$$

Thus the power spectrum is computed from the Fourier transform in the following way:

$$P_n = \frac{2|S_n|^2}{{N_{FFT}}^2} \quad for \quad 1 \leq n \leq \frac{N_{FFT}}{2} \qquad (2.3.5)$$

This power spectrum (also called the periodogram estimate) is only defined at positive frequencies below the Nyquist frequency limit.
The power normalized ranging Fourier transform follows from (2.3.5):

$$\tilde{S}_n = \frac{S_n}{N_{FFT}} \quad for \quad 1 \leq n \leq N_{FFT} \qquad (2.3.6)$$

2.4 Doppler Spectrum

Sampling a chirp sequence leads to a power spectrum P_n (see equation (2.3.3)) with n corresponding to a certain IF frequency f_{IF}, which is linearly related to a range R via equation (2.2.1). For one final ranging sample, consisting of several hundred range bins (one for each altitude layer), a few thousand chirps (meaning power spectra) are integrated over time. Therefore, for each range bin n (corresponding to a certain altitude layer) there is a time series of normalized spectral Fourier components

$$\tilde{S}_n(t_i) \quad , \quad t_0 = 0\,s, \; t_1 = T_c, \; t_2 = 2T_c, \ldots, \; t_k = kT_c \qquad (2.4.1)$$

The time series sampling interval is T_c. Another Fourier transform (often called the Doppler or 2D transform) over this time series reveals Doppler frequency shifts caused by moving particles in the scattering volume at the altitude related to range bin n. These Doppler frequency shifts linearly depend on the particle velocities v:

$$\frac{f_d}{f_t} = \frac{2 \cdot v}{c} \quad , \quad f_t = transmitter\ frequency \qquad (2.4.2)$$

The Doppler transform is given by:

$$\tilde{S}_n(t) \quad ==> \quad DT_n(f_d) \qquad (2.4.3)$$

with maximum Doppler frequency $f_{dm} = 1/2T_c$. If D_{FFT} is the number of samples in (2.4.1), the Doppler frequency resolution is:

$$\delta f_d = \frac{2 \cdot f_{dm}}{D_{FFT}} \qquad (2.4.4)$$

In contradiction to the ranging FFT in (2.3.1), the input time series for the Doppler FFT is not an array of real, but complex numbers. Therefore, the symmetry relation in (2.3.3) does not hold. The transform in (2.4.3) consists of D_{FFT} independent complex numbers:

$$DT_n(f_d)\ , \quad -f_{dm} \le f_d \le f_{dm} \qquad (2.4.5)$$

The Doppler frequency spectrum is then computed as the normalized power spectrum of the transform in (2.4.3):

$$DS_n(f_d) = \frac{|DT_n(f_d)|^2}{{D_{FFT}}^2}\ , \quad -f_{dm} \le f_d \le f_{dm} \qquad (2.4.6)$$

It should be noticed, that the integral over the Doppler spectrum equals P_n in (2.3.5):

$$P_n = \sum_{-f_{dm}}^{f_{dm}} DS_n(f_d) \qquad (2.4.7)$$

There is a restriction for the maximum Doppler velocity $f_{dm} = 1/2T_c$, which is caused by the fact that T_c cannot be chosen to be arbitrarily short. This problem is often referred to as the 'Doppler dilemma' in radar technology and is relevant for both, FMCW and pulsed radars. After a frequency chirp has been sent into the atmosphere, its echo will take about 70 µs for a maximum range of 10 km to complete the round trip. During that period, the radar should not emit another chirp (or pulse in the case of pulsed radars), because otherwise the echoes of two adjacent chirps will overlap in time and cannot be separated anymore. After this 70 µs 'dead time' the chirp has to last for at least another 30 µs to give time for measuring an FFT sample. Therefore, the minimum chirp repetition time is about 100 µs corresponding to a maximum Doppler velocity f_{dm} of 5 kHz. By using equation (2.4.2) we end up with the following restriction for v_m:

$$v_m = \frac{c\, f_{dm}}{2\, f_t} \qquad (2.4.7)$$

This leads to a v_m of 75 m/s for an X-Band radar ($f_t = 10\ GHz$), 21.5 m/s for a K-Band radar ($f_t = 35\ GHz$) and 8 m/s for a W-Band radar ($f_t = 94\ GHz$). Beyond this velocity range the Doppler spectrum becomes prone to aliasing effects, meaning that spectral components exceeding the maximum velocity will be mirrored to the opposite side of the spectrum. This effect can be easily corrected when observing in zenith direction by starting at the cloud top, where the particle velocities are close to zero and processing the spectra from top to bottom while maintaining the continuity of moving directions. The real problem occurs if the radar is scanning in elevation without having a range bin with a zero reference spectrum to start with.

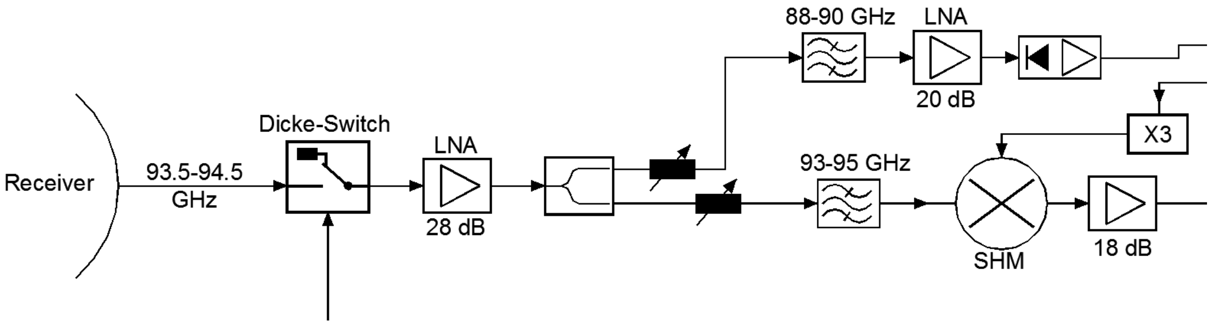
Another obvious problem occurs with very broad Doppler spectra exceeding the total velocity range of $2 \cdot v_m$.

2.5 Radar Receiver Calibration

P_n in (2.3.5) is the power spectrum at the input of the ADC board. Finally we are interested in the power spectrum P_{Rn} incident to the radar antenna (see equation (2.1.5)). The factor between P_n and P_{Rn} is called the receiver Gain:

$$P_n = G_{rn} \, P_{Rn} \qquad (2.5.1)$$

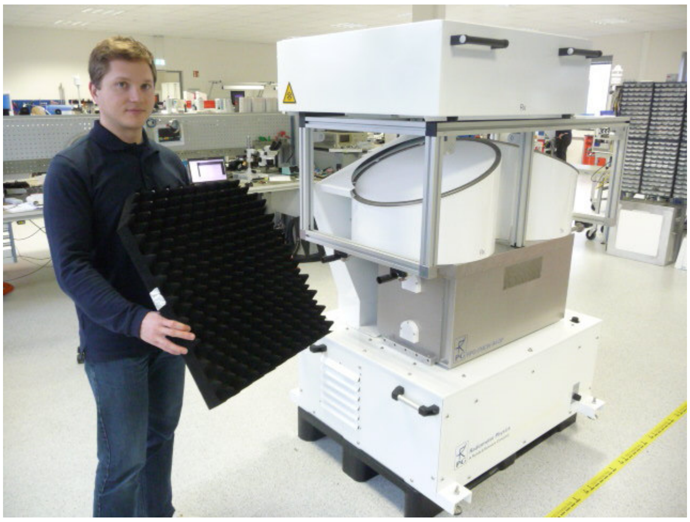
The receiver gain should also include all losses in the feed horn and antenna system.
The radar receiver is equipped with a Dicke switch calibration device located at the receiver input:



This device can switch the receiver input between the receiver antenna and an internal black body termination (attenuator), which is stabilized to a constant physical temperature T_{DS}.

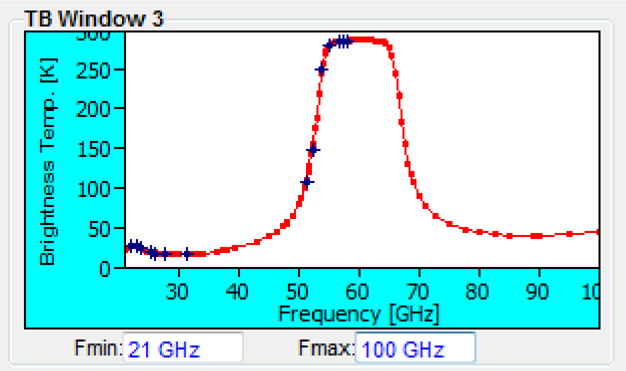
In an absolute radar calibration the receiver is sequentially terminated with two absolute calibration standards in front of the receiver's Cassegrain antenna system. This can be realized in two ways:

1. Using two external black body targets, one at ambient temperature T_H and one at liquid nitrogen temperature (LN2) T_C.



2. Using the (cloud free) sky as a cold target and get the sky brightness temperature from another measurement, for instance a RPG-HATPRO radiometer. This way the

calibration of the other instrument can be transferred to the radar. For the hot (ambient) target the radar can use the internal Dicke switch, which has been calibrated by a previous absolute calibration with an external ambient target (the Dicke switch is considered to be a secondary standard). In this scenario the relatively large external targets as used in 1. are not required at all.



Spectrum retrieval output (red curve) of a RPG-HATPRO radiometer for the frequency range [20 GHz, … , 100 GHz]. The 90 GHz brightness temperature can be used to calibrate the RPG-FMCW-94 cloud radar.

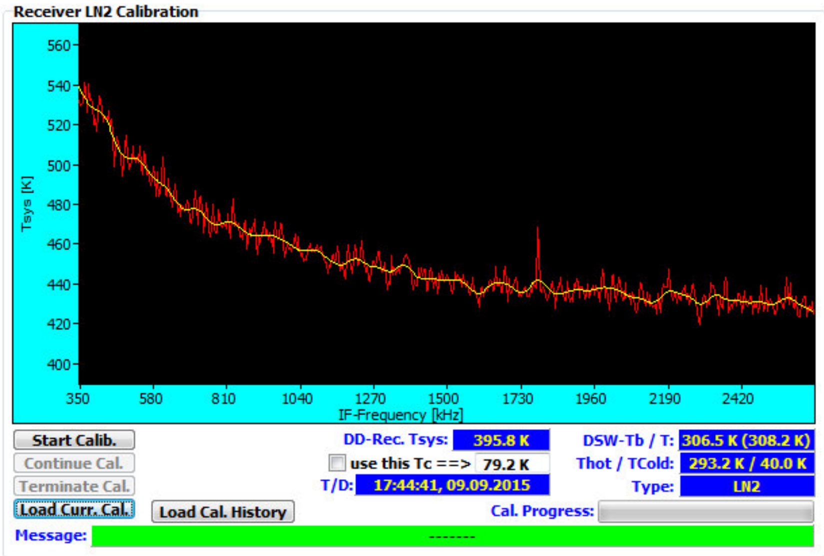
Whatever method is chosen, both of them result in a hot / cold calibration with two calibration standards at temperatures T_H and T_C.

A special feature of RPG-FMCW-94 cloud radars is that they include a passive broad band (2 GHz bandwidth) channel operated at a centre frequency of 89 GHz. This channel is very useful to provide information about the integrated liquid water path (LWP) in cloud observations. The frequency spacing from the radar channels around 94 GHz is wide enough to allow for an effective low pass filtering, protecting the 89 GHz radiometer from the strong signals of the radar operation at 94 GHz.
Therefore, an absolute calibration has to calibrate all radar channels plus the passive 89 GHz channel (in the case of an RPG-FMCW-94 cloud radar).

The first step is to determine the system noise temperature of each individual radar channel n (meaning for each bin in the IF spectrum) and the passive channel:

$$P_{nH} = G_n(T_{rn} + T_H) \quad , \quad P_{pH} = G_p\left(T_p + T_H\right) \qquad (2.5.2a)$$
$$P_{nC} = G_n(T_{rn} + T_C) \quad , \quad P_{pC} = G_p\left(T_p + T_C\right) \qquad (2.5.2b)$$

The index n represents all radar IF channels while the index p represents the passive 89 GHz channel. The T_{rn} are the radar receiver channel's system noise temperatures (the intrinsic noise contribution from the radar receiver channels) and T_p is the corresponding temperature for the passive broad band channel at 89 GHz. G_n and G_p are the receiver gains and P_{nH} , P_{nC} and P_{pH} , P_{pC} are the measured power spectrum numbers of equation (2.5.1) measured on the hot (ambient temperature) and cold targets respectively. From equations (2.5.2a) and (2.5.2b) all receiver gains and system noise temperatures can be computed.
We have now separated the radiometric contributions of the receiver from those of the external signals (sky, LN2 target, ambient target).



The yellow line represents a fit to each radar channel's system noise temperature.

In a second step the internal Dicke switch is turned on (it terminates the receiver input with the internal black body standard at temperature T_D). By using the receiver's system noise and gain parameters calculated in the first step, the Dicke switch's brightness temperature T_{DS} can be determined. This temperature should be very close to the physical Dicke switch temperature:

$$T_{DS} = \frac{P_{pD}}{G_p} - T_p \qquad (2.5.3)$$

Both temperatures T_D and T_{DS} are stored in the calibration file for later use.

The gain factors G_n and G_p are not the receiver Gain factors referred to in equation (2.5.1) because they link the power spectrum values to brightness temperatures but not to the power levels measured at the receiver's antenna input. The transformation of the brightness temperature gain factors to power related gain factors of equation (2.5.1) can be performed by applying the Rayleigh Jeans approximation. With $\tilde{P}_{nT}$ being the power received by the radar receiver channel n at the antenna input when the receiver is terminated by a target of brightness temperature T, this power is given as:

$$\tilde{P}_{nT} = k_B B_n T \quad , \quad B_n = \frac{f_{SR}}{N_{FFT}} = \frac{1}{T_c} \quad (radar\ channel\ bandwidth) \qquad (2.5.4)$$

with k_B being the Boltzmann constant. The Rayleigh Jeans approximation validity is justified for the high brightness temperatures observed by the radar ($k_B T \gg h\nu\ or\ T \gg 4.5\ K\ at\ 94\ GHz$) at frequencies below 100 GHz. The required gain factors G_{rn} of equation (2.5.1) can then be computed from the brightness temperature related gain factors G_n:

$$G_{rn} = \frac{P_{nT}}{\tilde{P}_{nT}} = \frac{G_n}{k_B B_n} \qquad (2.5.5)$$

It should be noticed that the G_n factors' unit is $[W/K]$ while the G_{rn} are dimensionless (as usual, in the order of 80 dB). The G_{rn} factors include ALL corrections due to antenna and feed horn losses, as well as the complete chain of receiver amplifiers and mixer conversion losses. No other corrections need to be applied to link the measured power spectrum levels of each radar bin to the real power received at the antenna input within the bin's bandwidth B_n. The radar equation in (2.1.5) can be written in the more detailed form for a single bin of the IF spectrum with P_n being the received power in this bin (S_{rn} is the radar scaling factor):

$$\frac{P_n}{P_T} = \frac{G_{rn} C_R}{F_R(R)} Z_e \quad \lll==\ggg \quad Z_e = \frac{P_n F_R(R)}{P_T G_{rn} C_R} \equiv S_{rn} P_n F_R(R) \;\; with \;\; F_R(R) \equiv \frac{R^2}{\delta R} L_{ovlp}(R) \quad (2.5.6)$$

L_{ovlp} is the beam overlap loss given in equation (2.8.1) and $F_R(R)$ is the range factor.
During radar operation, the receiver periodically (every 10 minutes) performs a 'zero calibration' cycle by turning the Dicke switch on and measuring one sample on the Dicke switch termination black body instead of the sky (the transmitter is turned off during this cycle). Because this calibration is a single point calibration, the radar software assumes the T_n to be constant over time. The G_{rn} are readjusted according to equations (2.5.2a, using T_{DS} instead of T_H) and (2.5.6). This way a long term stability of the radar's receiver accuracy can be maintained. **We recommend a repetition of the absolute calibration every 6 months in order to re-calibrate the radar's channel system noise temperatures T_n.**
The integration time on each of the calibration targets during the absolute calibration process is 60 seconds. According to the radiometer formula, the receiver noise measured in K for a single IF bin of typically 4 kHz bandwidth and about 500 K system noise temperature is:

$$\Delta T_n = \frac{T_n}{\sqrt{60\ s \cdot 4000\ Hz}} \approx 1\ K\ RMS \;\; =\gg \;\; \Delta T_{p-p} \approx 4\ K \qquad (2.5.7)$$

The peak-to-peak noise is in the order of 4 K only, showing that an absolute calibration with passive black body targets of brightness temperatures in the range [77K, 300K] is well suited to calibrate a radar receiver with high accuracy. It should be noticed, that the receiver sensitivity achievable for radars operated at K-band or X-band is significantly higher compared to W-band. Therefore, the proposed method is even better suited for these radars, assuming a careful receiver design is focusing on receiver sensitivity optimization.

2.6 Radar Sensitivity Limit

The radiometer formula also provides the radar's sensitivity limit when the temperature standard deviation on the left side of equation (2.5.7) is converted to power using the Rayleigh-Jeans approximation. The brightness temperature T_n in (2.5.7) for the sensitivity limit is composed of the radar receiver noise temperature T_{rn} of equations (2.5.2) and the passive atmospheric emission at operating frequency. The later contribution can be derived from the passive receiver channel sky brightness temperature T_{sky}:

$$\Delta T_n = \frac{T_{rn} + T_{sky}}{\sqrt{\tau \cdot B_n}} \qquad (2.6.1)$$

Here τ is the chirp sequence integration time and B_n is the channel bandwidth of (2.5.4). In order to derive the sensitivity limit SL_n in Z_e units we combine equations (2.5.4) and (2.5.6) to get

$$\frac{2\, k_B B_n \Delta T_n}{P_T} = \frac{C_R}{F_R(R)} SL_n \quad \lll=\ggg \quad SL_n = \frac{2\, k_B B_n \Delta T_n}{P_T C_R} F_R(R) \qquad (2.6.2)$$

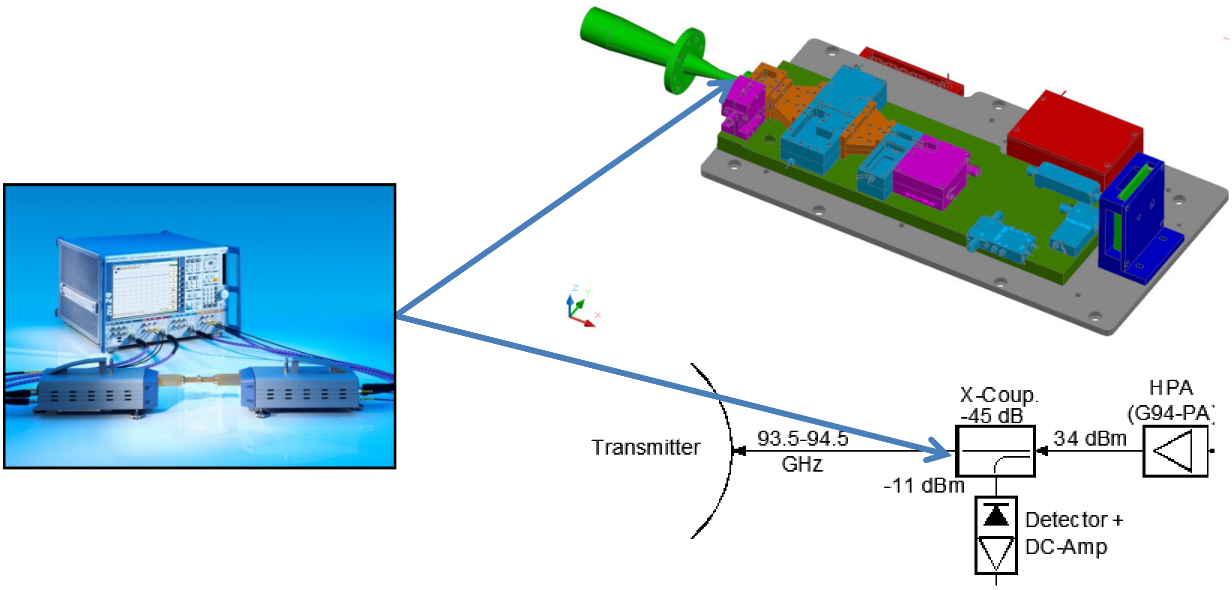
Here we have set the detection threshold to $2 \cdot \Delta T_n$.
Substituting (2.6.1) into (2.6.2) yields:

$$SL_n = \frac{2\, k_B \sqrt{B_n} F_R(R)}{P_T C_R \sqrt{\tau}} \left(T_{rn} + T_{sky}\right) \quad (2.6.2)$$

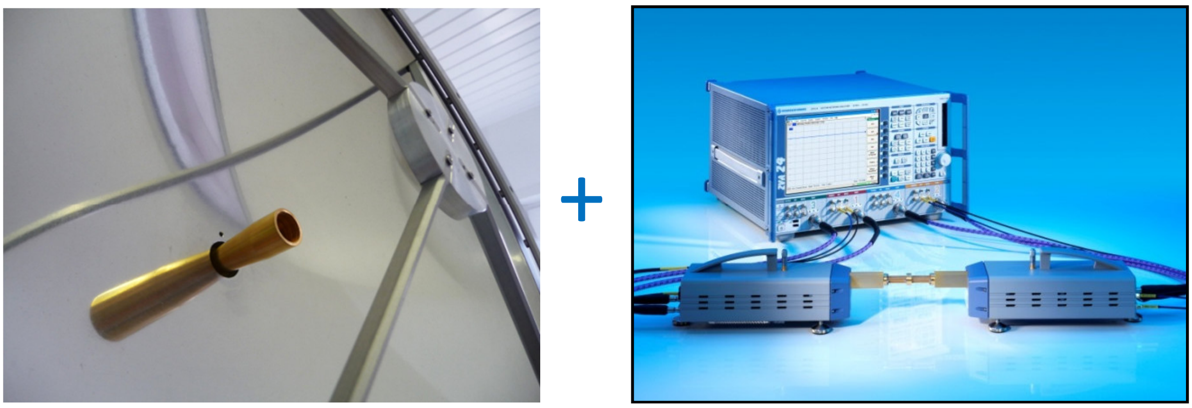
The SL_n array is stored to the level 0 and level 1 data files for each radar sample. The array is calculated independently for different polarisations because the T_{rn} are related to separate receivers in dual pol. radars.

2.7 Radar Transmitter Calibration

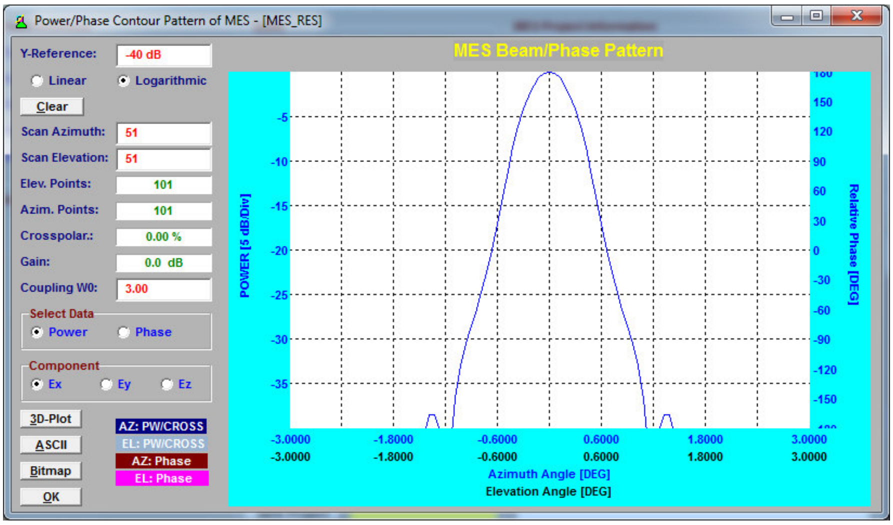
In order to complete the radar's internal calibration, also the transmitter power needs to be characterized as good as possible. In addition, the transmitter power should be continuously monitored over time to ensure that changes in output power are captured and taken into account in the radar calibration procedure.



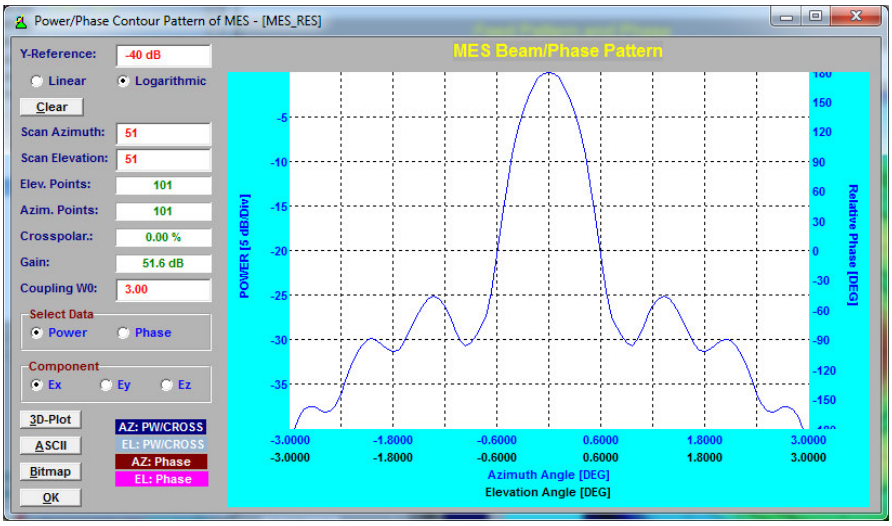
In RPG radars, the receiver and transmitter RF components are all thermally stabilized within a few hundred mK. This ensures a stable transmitter power generation and power measurement. Once thermally stabilized, the maximum output power is determined by a retraceable precision Rhode & Schwarz W-band power head (accuracy 0.1 dB). A small fraction of the transmitter power is coupled to an RF detector for power monitoring. Care was taken to ensure that this detector always operates in its linear detection regime. While the transmitter power is measured by the power head, the detector reading is monitored to determine its calibration factor. This measurement does not take into account the feed horn and antenna losses. These losses need to be determined separately.



The feedhorn insertion loss has been measured by a Rhode & Schwarz Network Analyzer to be 0.3 +/- 0.05 dB ($L_f = 1.072$).



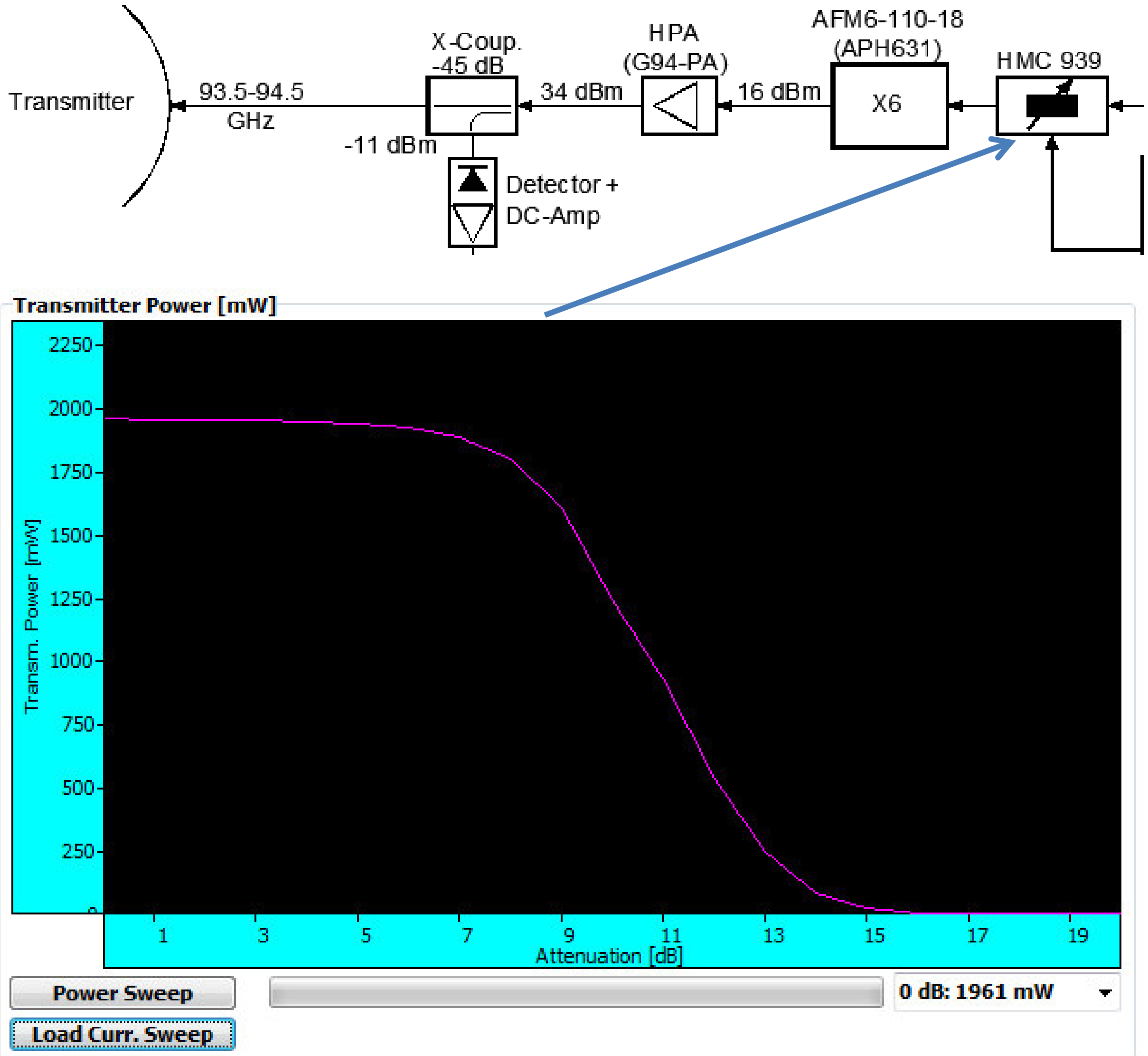
Antenna pattern without sub-reflector blockage



Antenna pattern including sub-reflector blockage

The Cassegrain antenna system losses are dominated by sub-reflector blocking. A detailed analysis reveals a power loss of 11% ($L_{sr} = 1.11$) by the sub-reflector shadowing the output beam.
In addition, the sub-reflector changes the beam pattern compared to the unblocked antenna. The Cassegrain antenna gain needs to be corrected accordingly.

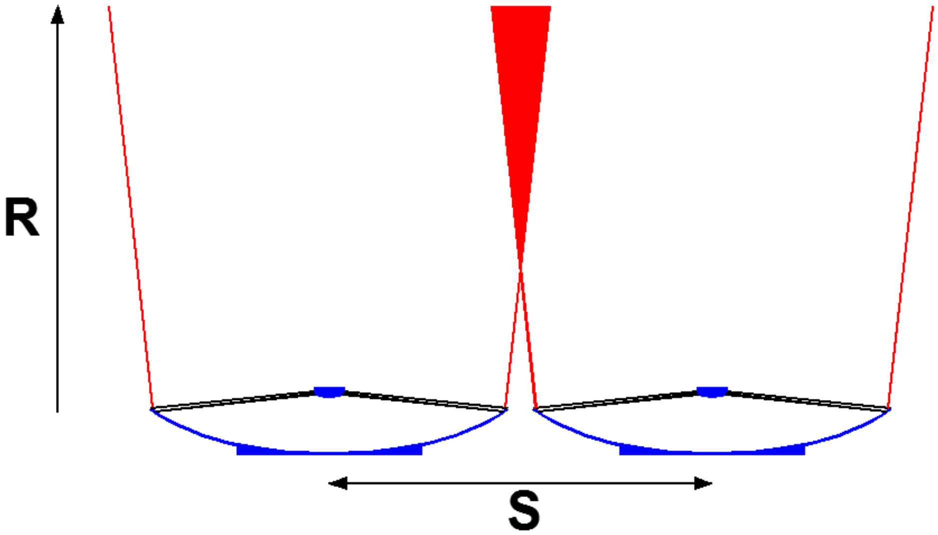
The radar receiver is operated under a high dynamic range of input signal levels. On one hand its sensitivity must be as high as possible to detect faint objects like liquid water clouds at high altitudes. On the other hand the radar should also respond linearly to precipitation events where large droplets located close to the radar reflect a high fraction of the transmitter power back to the receiver. In order to cope with the high dynamic range requirements, the radar's transmitter power can be reduced in digital steps by a maximum of 16 dB.



For this purpose a digital 5 bit attenuator is included in the transmitter's signal path. During radar operation the input signal level to the receiver is measured and the transmitter power is automatically adjusted to protect the receiver from saturation effects.
A power sweep is performed every hour to monitor changes in output power. These changes are then used to tune P_T in the radar equation (2.5.6).

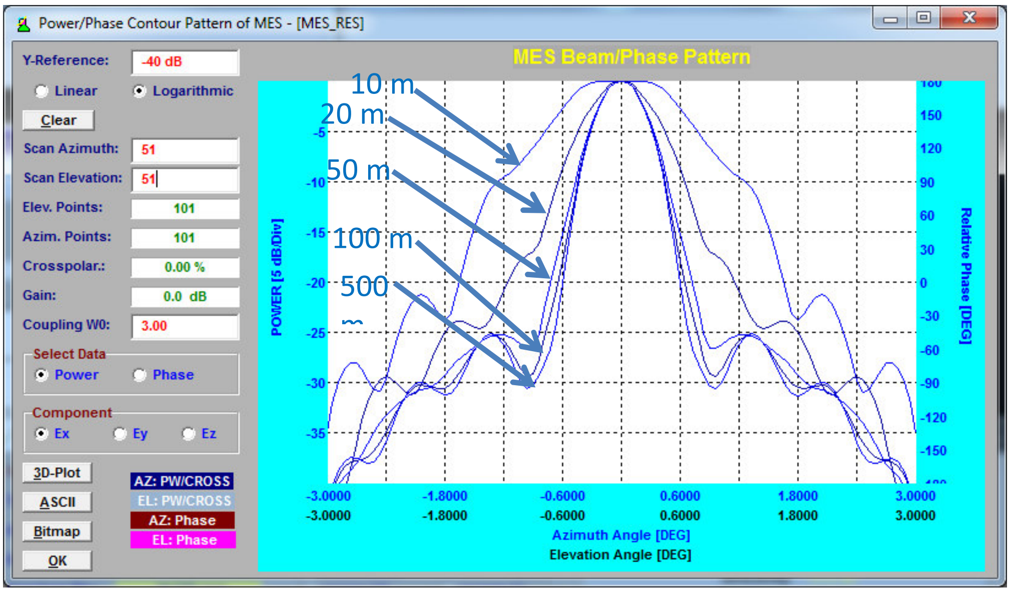
2.8 Beam Overlap Correction

All FMCW-radars with high dynamic range are realized as bi-static systems with a separate antenna for transmitter and receiver. This is mandatory because the transmitter is always active during the receiver operation and a power transfer from the transmitter to the receiver must be attenuated by at least 80 dB to protect the receiver from saturation. If the receiver would share the same antenna with the transmitter, it would be technically impossible to separate the transmitted power from the received power by a 80 dB suppression factor.



The obvious disadvantage of the bi-static configuration is the imperfect beam overlap close to the radar. If the beam pattern strongly overlaps with a Gaussian beam pattern (which is fulfilled by RPG's radar antennas to 95%), the loss caused by an incomplete overlap is:

$$L_{ovlp} = e^{\frac{2\ln(2)S^2}{\theta^2 R^2}} \qquad (2.8.1)$$



Beam pattern variation in the near field region.

For an antenna separation S of 568 mm and a HPBW of 0.48° (RPG-FMCW-94 parameters) this leads to a loss of 10% at 300 m distance from the radar. However, this loss is not critical because it only affects reflections from objects close to the radar which are relatively strong. The RPG-FMCW radars are capable of detecting reflectivity levels of -60 dBz down to 50 m altitude.

Another important issue for all radars (not only bi-static systems) is related to the antenna far field, because the antenna gain used in the radar equation (2.1.5) is usually calculated or measured in the far field region. In the antenna's near field region the antenna gain G_T deviates from the far field value. Therefore, in order to determine the radar's lowest ranging limit, the far field distance should be checked. This limit is about 50 m for the RPG-FMCW-94 radar.

2.9 Calibration Cross Check with Metal Sphere Reflector

There is a way to experimentally cross check the quality of the radar calibration procedures described above. The method is based on the radar equation (2.1.1) including all calibration parameters applied to a metal sphere with diameter D_{sp} in a distance R:

$$P_n = P_T \frac{G_{rn}}{L_{at}L_{ovlp}L_f L_{sr}} \frac{{G_T}^2\lambda^2}{(4\pi)^3 R^4} \frac{\pi {D_{sp}}^2}{4} \qquad (2.9.1)$$

The calibrated antenna gain loss factors $\frac{G_T}{L_f L_{sr}}$ are not squared in (2.9.1), because the receiver antenna losses are already included in G_{rn}. The loss factors $L_f L_{sr}$ in (2.9.1) only represent the transmitter antenna losses. L_{at} is the atmospheric loss which can be significant at W-band. We determine this factor through a radiative transfer model taking into account the water vapor and temperature profiles along the radar signal path as well as the surface pressure.
Equation (2.9.1) can be applied for testing the calibration parameters' quality, because all numbers on the right side of (2.9.1) are known, so that P_n can be calculated. The question is how well the computed P_n match the measured ones.

In order to determine P_n experimentally, RPG has established a measurement setup, which is used for the calibration validation of all radars produced by the company. A metal sphere (48 mm diameter) is launched with a helium balloon to an altitude of approx. 150 m in a distance of close to 1000 m from the radar. The radar itself is mounted to a positioner equipped with a telescope for pointing the radar precisely to the metal target. The metal sphere is held down to a fixed altitude by three thin strings which stabilize the sphere horizontally. The vertical distance between the sphere and the balloon is about 10 m to ensure that the measured reflected power is not affected by the balloon.

Here is an example of a typical result:

- Reflector diameter: 48 mm
- Reflector altitude: approx. 140 m
- Reflector distance from radar: 960 m
- Atmospheric attenuation: -0.6 dB (path length 1920 m)
- Expected power ratio P_r/P_T: -127.1 dB
- Measured power ratio P_r/P_T: -127.5 dB
- Deviation from expected value: **-0.4 dB**

The measurement should be performed under low wind conditions. However, the reflector will always swing a little back and forth or left and right, so that it can be difficult to get it right into the center of the radar beam. In order to cope with this problem, the radar ranging chirp sequence duration is set to 0.2 sec and the detection software features a peak-hold function to sample the maximum response from the reflector. When the radar is pointing to the target, the peak-hold function is initialized and measurements are taken for at least 60 seconds. Then another peak-hold initialization is performed and the procedure is repeated. The standard deviation of about 100 peak-hold values is typically in the order of 0.15 dB which is an indication for the measurement accuracy.

Experimental setup: Radar mounted to a scanner, launching the helium balloon and metal sphere reflector.

The deviation between expected and measured power ratios could be explained by additional transmitter antenna losses (e.g. caused by the sub-reflector struts) that are not taken into account or uncertainties in the G_T calculation. Finally, -0.4 dB error in the overall power budget is quite small and can be considered as another loss factor to be added to equation (2.9.1).

2.10 Reflectivity Corrections

There are two reasons for the radar beam power attenuation while it is travelling through the atmosphere:

- Reflectivity losses due to scattering power out of the beam axis
- Atmospheric absorption by gases (oxygen, water vapour, nitrogen) and hydrometeors (absorption by ice particles is negligible)

2.10.1 Scattering Losses

The reflectivity Z_e at higher altitude layers is underestimated because the power incident to a scattering volume in this altitude has already been reduced by passing through lower layers where scattering may have removed a part of the transmitter power out of the radar beam. For the discussion of how to correct for this effect, we neglect other sources of attenuation here.

When travelling upwards in the atmosphere, the power incident to the first scattering volume bin at distance R is:

$$P_i = \frac{P_T G_T}{4\pi R^2}\frac{V}{\delta R} = \frac{P_T G_T}{4\pi R^2}\frac{\pi(\theta R)^2}{8\ln(2)}$$

Here V is the scattering volume in (2.1.2). Because of scattering, this power is reduced by the amount (see (2.1.3) and (2.1.4)):

$$P_l = \frac{P_T G_T}{4\pi R^2}\sigma = \frac{P_T G_T}{4\pi R^2}\frac{\pi(\theta R)^2}{8\ln(2)}\frac{K^2 \delta R \pi^5}{10^{18}\lambda^4} Z_e$$

Behind the scattering volume, the remaining power is:

$$P_i - P_l = \frac{P_T G_T}{4\pi R^2}\frac{\pi(\theta R)^2}{8\ln(2)}\left(1 - \frac{K^2\pi^5}{10^{18}\lambda^4}\delta R Z_e\right) = \frac{P_T G_T}{4\pi R^2}\frac{\pi(\theta R)^2}{8\ln(2)}\frac{1}{L_{sc}} \qquad (2.10.1)$$

The loss factor $1/L_{sc}$ occurs twice for power scattered behind the scattering volume (both ways up and down). Therefore, all reflectivity values Ze behind the scattering volume (looking from ground direction) need to be increased:

$$\tilde{Z}_e = Z_e {L_{sc}}^2 \qquad (2.10.2)$$

For instance, if we consider a 94 GHz radar with a range resolution of 30 m and a 3000 m layer of 15 dBZe reflectivity, the Ze correction for scattering volumes behind this layer would be:

$$L_{sc} = 1.0021 \quad ==\gg \quad {L_{sc}}^{200} \approx 1.51 \cong 1.8\,dB \quad (2.10.3)$$

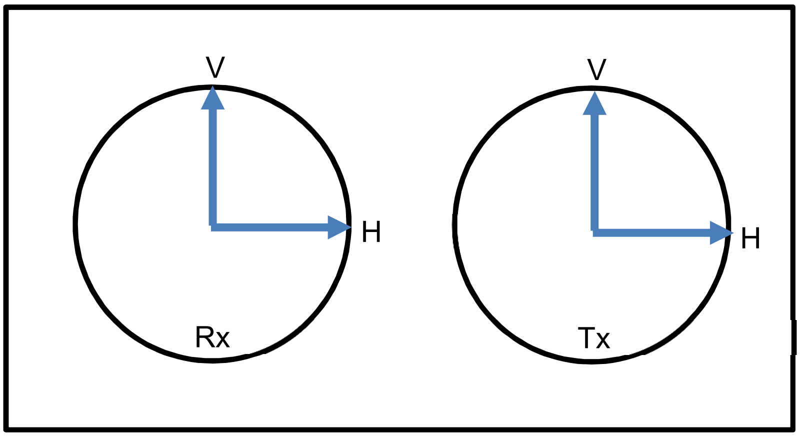
2.10.2 Gas Absorption Loss

Corrections of similar magnitude can be expected from losses introduced by gas absorption, e.g. by water vapor. When the radar is combined with a RPG-HAPTRO humidity and temperature profiler, the gas absorption correction is automatically performed by using an attenuation Neural Network retrieval based on the humidity and temperature profiles from this instrument.

2.11 Cloud radar polarimetry

2.11.1 Reference Polarisation Basis

For an analysis of polarization states and their transformation a reference basis has to be chosen. Usually the Cartesian reference basis formed by radar antenna feeds is used. The Cartesian reference bases used throughout this manual are shown in the figure below. When the radar is pointed to 0° elevation, i.e. the transmitted signal propagates parallel to the ground, the orientations of unit vectors, forming the basis, correspond to horizontal and vertical directions. Therefore, the unit vectors and corresponding polarization channels are denoted as “horizontal” (H) and “vertical” (V), respectively. The reference basis are related to the radar antennas and therefore at 90° elevation (radar pointing to zenith), the denotations H and V are kept the same, although the correspondence with respect to the ground is violated.



2.11.2 Representation of Polarisation States

It is known that polarization states and transformations of electromagnetic waves can be described with the Jones formalism. Taking into account that the coherency bandwidth of the radar hardware is wider than the transmitted signal bandwidth (narrowband approximation), the polarization state of a radar signal in the far field (plane wave approximation) can be represented in the Cartesian basis by the following Jones vector up to an absolute phase shift:

$$\mathbf{e} = \begin{bmatrix} \dot{E}_H \\ \dot{E}_V \end{bmatrix} = \begin{bmatrix} E_H \\ E_V e^{j\Delta\Phi} \end{bmatrix}. \qquad (2.11.1)$$

$\dot{E}_H$ and $\dot{E}_V$ are complex projections of the electric field to the unit vectors forming the reference basis, E_H and E_V are the magnitudes of $\dot{E}_H$ and $\dot{E}_V$, respectively, $\Delta\Phi = \Phi_H - \Phi_V$ is the phase difference between $\dot{E}_H$ and $\dot{E}_V$, Φ_H and Φ_V are the absolute phases of $\dot{E}_H$ and $\dot{E}_V$, respectively.

The wave $\mathbf{e}$ can be represented in an arbitrary orthogonal polarization basis $(\mathbf{e}_1;\mathbf{e}_2)$:

$$\mathbf{e}' = \mathbf{R}\mathbf{e}\,, \qquad (2.11.2)$$

where $\mathbf{R}$ is a 2 x 2 rotational operator. Columns of the matrix $\mathbf{R}$ represent unit vectors $\mathbf{e}_1$ and $\mathbf{e}_2$ in the original basis and in general has the following form:

$$\mathbf{R} = \begin{bmatrix} \dot{R}_1 & -\dot{R}_2^* \\ \dot{R}_2 & \dot{R}_1^* \end{bmatrix}, \qquad (2.11.3)$$

Where $\dot{R}_1$ and $\dot{R}_2$ are Cayley-Klein Parameters:

$$\dot{R}_1 = \cos\varepsilon\cos\theta - j\sin\varepsilon\sin\theta$$
$$\dot{R}_2 = \cos\varepsilon\sin\theta - j\sin\varepsilon\cos\theta$$

Where ε and θ are ellipticity and orientation angles of the new basis with respect to the original one. Note that the determinant of the matrix $\mathbf{R}$ is 1, i.e. the transformation Eq. (2.11.2) does not change the power of the wave $\mathbf{e}$.

2.11.3 Polarimetric Configurations of the Radar

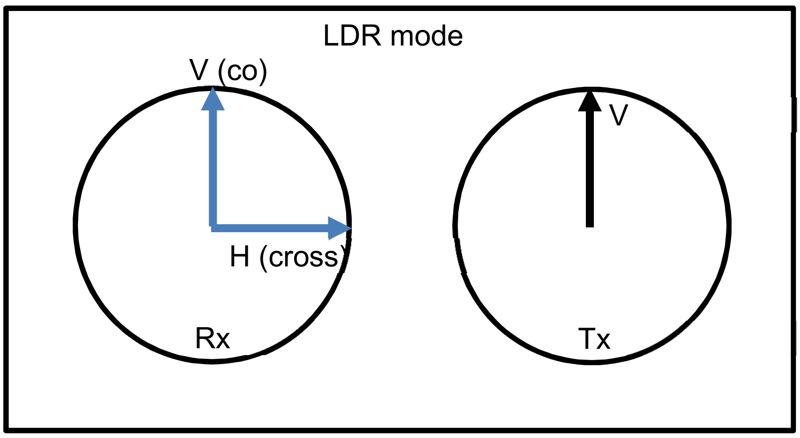
There are two polarimetric options available for RPG cloud radars: Linear depolarization ratio (LDR) mode and Simultaneous Transmission Simultaneous Reception (STSR) mode. In the LDR mode the transmitter signal $\dot{E}_\Sigma$ is radiated from the vertical channel:

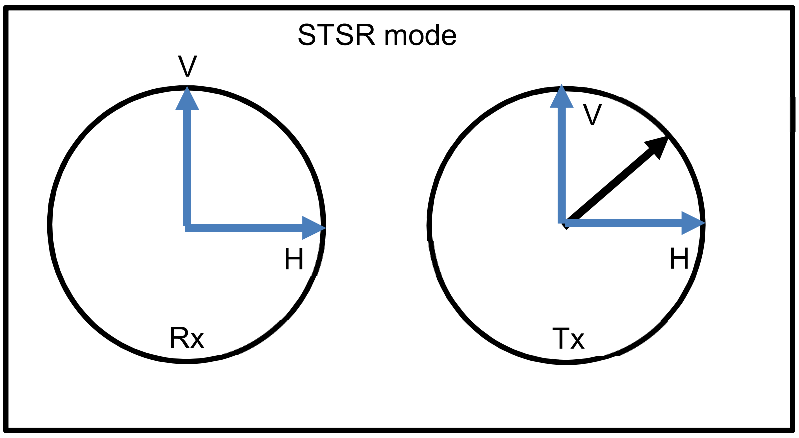
$$\mathbf{e}_t = \dot{E}_\Sigma \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad (2.11.4)$$

In the STSR mode the transmitter antenna is rotated by $\theta = 45°$ clockwise and therefore, the radar transmits horizontal and vertical components simultaneously:

$$\mathbf{e}_t = \begin{bmatrix} \cos 45° & \sin 45° \\ -\sin 45° & \cos 45° \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} \dot{E}_\Sigma = \frac{\dot{E}_\Sigma}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad (2.11.5)$$

As the phase difference $\Delta\Phi$ is 0 deg, the transmitted wave has linear polarization with 45° orientation.





2.11.4 Statistical Properties of the Received Signals

In both polarimetric configurations the radar has two coherent receivers and thus allows for simultaneous reception and analysis of the complex amplitudes $\dot{E}_H$ and $\dot{E}_V$ of received signals in the horizontal and vertical channels, respectively. Using a sequence of $\dot{E}_H$ and $\dot{E}_V$ for a certain range bin, complex spectra $\dot{S}_H(f_k)$ and $\dot{S}_V(f_k)$ are calculated as explained in chapters 2.3 and 2.4 of the manual. Here f_k is the Doppler frequency shift of the spectrum line k, the dot over a denotation stands for the complex number. In the following, only a single spectral component is considered for brevity.

In general, the orthogonal components $\dot{S}_H$ and $\dot{S}_V$ of the received electromagnetic wave $\mathbf{e}_r$ vary with time. Statistical properties of the signals in the horizontal and vertical channels can be described using the spectral form of the coherency matrix $\mathbf{B}$:

$$\mathbf{B} = \left\langle \mathbf{e}_r \mathbf{e}_r^{\dagger} \right\rangle = \left\langle \begin{bmatrix} \dot{S}_H \\ \dot{S}_V \end{bmatrix} \begin{bmatrix} \dot{S}_H^* & \dot{S}_V^* \end{bmatrix} \right\rangle = \begin{bmatrix} \langle \dot{S}_H \dot{S}_H^* \rangle & \langle \dot{S}_H \dot{S}_V^* \rangle \\ \langle \dot{S}_V \dot{S}_H^* \rangle & \langle \dot{S}_V \dot{S}_V^* \rangle \end{bmatrix} = \begin{bmatrix} B_{hh} & \dot{B}_{hv} \\ \dot{B}_{vh} & B_{vv} \end{bmatrix} \quad (2.11.6)$$

† is the Hermitian transpose sign, * is complex conjugation sign, <> is averaging over time. The main-diagonal components of the coherency matrix $B_{hh} = |\dot{S}_h|^2$ and $B_{vv} = |\dot{S}_v|^2$ correspond to power spectra in the horizontal and vertical receiving channels, respectively. The off-diagonal components $\dot{B}_{hv}$ and $\dot{B}_{vh}$ are covariancies of the signals in the vertical and horizontal receiving channels. In the single polarization configuration of the radar only the element B_{vv} is calculated and analyzed. A dual polarization configuration performs the computation of the elements B_{hh}, B_{vv}, and $\dot{B}_{hv}$. Taking into account that $\dot{B}_{hv} = \dot{B}_{vh}^*$, storing of the element $\dot{B}_{vh}$ is not required. <u>Note, even though the receiving polarization basis is the same for LDR and STSR modes, the polarization states of the transmitted signals in these modes are different and therefore the elements of the coherency matrix have different information content.</u>

A coherency matrix $\mathbf{B}$ can be decomposed to a sum of stable $\mathbf{K}$ and fluctuating $\mathbf{J}$ parts:

$$\mathbf{B} = \mathbf{J} + \mathbf{K} \quad (2.11.7)$$

$$\mathbf{K} = \langle \mathbf{e}_r \rangle \langle \mathbf{e}_r \rangle^{\dagger} = \begin{bmatrix} \langle \dot{S}_H \rangle \langle \dot{S}_H^* \rangle & \langle \dot{S}_H \rangle \langle \dot{S}_V^* \rangle \\ \langle \dot{S}_V \rangle \langle \dot{S}_H^* \rangle & \langle \dot{S}_V \rangle \langle \dot{S}_V^* \rangle \end{bmatrix}. \qquad (2.11.8)$$

In the case of non-coherent scattering, which is the case when cloud particles are observed, absolute phases of $\dot{S}_H$ and $\dot{S}_V$ are distributed uniformly from 0 to 2π. Therefore, their mean values and $\mathbf{K}$ are equal to 0. Note, that in the case of point scatterers or external coherent sources of radiation, $\mathbf{K}$ can differ from 0. A coherent polarization coupling produced by a radar antenna of poor quality can also bias the elements of the matrix $\mathbf{K}$.
As the coherency matrix $\mathbf{B}$ contains only the fluctuating part, the elements B_{ij} can be presented as follows:

$$B_{hh} = \sigma_h^2$$
$$B_{vv} = \sigma_v^2$$
$$\dot{B}_{hv} = \sigma_h \sigma_v \dot{R}_{hv}$$
$$\dot{B}_{vh} = \sigma_h \sigma_v \dot{R}_{hv}^*$$

Where σ_h and σ_v are standard deviations of the horizontal and vertical components, respectively, and $\dot{R}_{hv}$ is the complex correlation coefficient. Absolute value of $\dot{R}_{hv}$ shows statistical relation between the horizontal and vertical components, while its argument corresponds to the phase shift between horizontal and vertical components $\Delta\Phi$.

2.11.5 Non-Polarized and Fully-Polarized Coherency Matrix Components

The coherency matrix $\mathbf{B}$ can be decomposed as follows:

$$\mathbf{B} = A\mathbf{I} + \begin{bmatrix} C & \dot{D} \\ \dot{D}^* & F \end{bmatrix}, \qquad (2.11.9)$$

with the condition:

$$CF - \left|\dot{D}\right|^2 = 0. \qquad (2.11.10)$$

Here $\mathbf{I}$ is the 2 x 2 unit matrix. Parameters $A, \dot{D}, F$ can be found as follows:

$$A = \frac{1}{2}\left(\mathrm{Sp}\mathbf{B} - \left[\mathrm{Sp}^2\mathbf{B} - 4\det\mathbf{B}\right]^{1/2}\right),$$
$$C = \frac{1}{2}\left(B_{hh} - B_{vv} + \left[\mathrm{Sp}^2\mathbf{B} - 4\det\mathbf{B}\right]^{1/2}\right),$$
$$F = \frac{1}{2}\left(B_{vv} - B_{hh} + \left[\mathrm{Sp}^2\mathbf{B} - 4\det\mathbf{B}\right]^{1/2}\right).$$

The horizontal and vertical components of the first term of the Eq. (2.11.9) are not correlated ($\left|\dot{R}_{hv}\right|$=0) and therefore this term corresponds to the non-polarized part. The second term describes the fully-polarized part as the corresponding correlation $\left|\dot{R}_{hv}\right|$ is equal to 1. The degree of polarization can be found as a ratio of the non-polarized power to the total power of the electromagnetic wave:

$$\mu = \frac{2A}{\mathrm{Sp}\mathbf{B}} \qquad (2.10.11)$$

2.11.6 Coherency Matrix in Different Basis

Using Eqs. (2.11.2) and (2.11.6) the coherency matrix $\mathbf{B}$ can be represented in another polarization basis using the following unitary transformation:

$$\mathbf{B}' = \left\langle \mathbf{R}\mathbf{e}_r (\mathbf{R}\mathbf{e}_r)^\dagger \right\rangle = \left\langle \mathbf{R}\mathbf{e}_r \mathbf{e}_r{}^\dagger \mathbf{R}^\dagger \right\rangle = \mathbf{R}\mathbf{B}\mathbf{R}^\dagger \qquad (2.10.12)$$

Note, that transformation Eq. (2.10.12) does not change the non-polarized part as it is proportional to the unity matrix.

2.11.7 STSR sensitivity issue

In LDR mode most of the received power is concentrated in the vertical (co-polarized) receiving channel. The power in the horizontal (cross-polarized) channel is about 10 to 30 dB lower. Taking into account that the noise power is the same in both orthogonal channels, the total signal power is not calculated. Only the signal power in the vertical channel is used for scatter detection. In the case of the STSR configuration, the signal generated by the transmitter is “split” to two channels. Therefore, the amount of power received in the vertical channel is reduced by a factor of 2 in comparison to the single polarization and LDR configurations. This leads to a sensitivity loss of 3 dB if only the vertical channel is used for the reflectivity calculation.
The cloud radar sensitivity corresponds to a threshold above the mean noise power density:

$$T = Q\sigma_n,$$

Where σ_n is the standard deviation of the noise power density, and Q is a parameter related to a certain false alarm rate (typically 3 to 4).
In the STSR-configuration, non-coherent averaging of signals in the horizontal and vertical channels gains only 1.5 dB in sensitivity:

$$\sigma_{nav} = \frac{\sigma_n}{\sqrt{2}}.$$

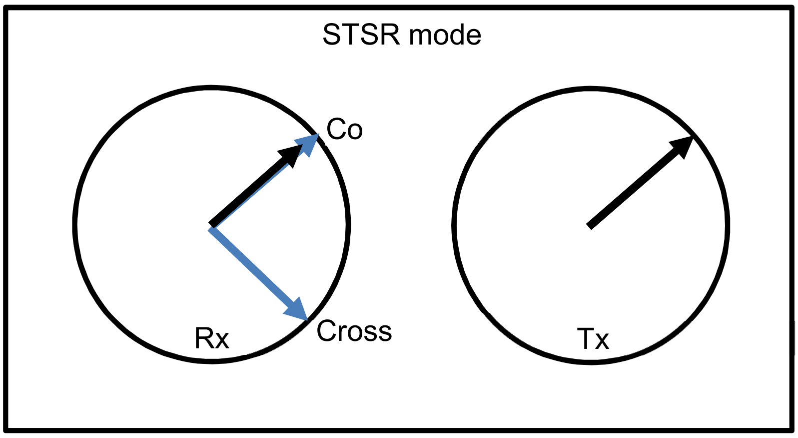
Taking into account that cloud particles do not significantly de-correlate cloud radar signals, the maximization of the power in one of the channels applying Eq. (2.10.12) can gain up to 3 dB in sensitivity. As the radar has the transmitting differential phase of 0°, the maximum power of the returned signal concentrated in the single component corresponds to the ±45° reference basis:

$$\mathbf{B}' = \begin{bmatrix} \cos 45^\circ & -\sin 45^\circ \\ \sin 45^\circ & \cos 45^\circ \end{bmatrix} \begin{bmatrix} B_{hh} & \dot{B}_{hv} \\ \dot{B}_{vh} & B_{vv} \end{bmatrix} \begin{bmatrix} \cos 45^\circ & \sin 45^\circ \\ -\sin 45^\circ & \cos 45^\circ \end{bmatrix} =$$
$$\frac{1}{2} \begin{bmatrix} B_{hh} + B_{vv} - 2\,\mathrm{Re}\,\dot{B}_{hv} & B_{hh} - B_{vv} + 2i\,\mathrm{Im}\,\dot{B}_{hv} \\ B_{hh} - B_{vv} - 2i\,\mathrm{Im}\,\dot{B}_{hv} & B_{hh} + B_{vv} + 2\,\mathrm{Re}\,\dot{B}_{hv} \end{bmatrix} = \begin{bmatrix} B_{xx} & \dot{B}_{xc} \\ \dot{B}_{cx} & B_{cc} \end{bmatrix}$$

For spherical cloud particle scattering the polarization state of the scattered wave is not changed, $B_{hh} = B_{vv}$.
Taking into account Eq. (2.11.10):

$$\mathbf{B}' = A\mathbf{I} + \begin{bmatrix} 0 & 0 \\ 0 & F \end{bmatrix}$$

The new basis is illustrated in the figure below.



It can be seen, that the element B_{cc} contains all the power of the fully polarized part while the power of the non-polarized part (contains noise and depolarized signal) is still equally splitted to B_{xx} and B_{cc}. This recovers the sensitivity loss up to 3 dB.

2.11.8 Polarimetric products

Using the coherency matrix the following spectral polarimetric parameters are calculated for Doppler components where the signal is detected:

Equation	LDR-mode parameter	STSR-mode parameter
1. $(B_{hh} - P_n)/(B_{vv} - P_n)$	Spectral linear depolarization ratio (sLDR)	Spectral differential reflectivity (sZ_{DR})
2. $\dfrac{\lvert\dot{B}_{hv}\rvert}{\sqrt{(B_{hh} - P_n)(B_{vv} - P_n)}}$	Spectral co-cross-channel correlation coefficient (ρ_{cx})	Spectral correlation coefficient (ρ_{hv})
3. $\arg[\dot{B}_{hv}]$	Spectral co-cross-channel differential phase (Φ_{cx})	Spectral differential phase (Φ_{DP})
4. $\dfrac{B_{hh} + B_{vv} - 2\,\mathrm{Re}[\dot{B}_{hv}] - 2P_n}{B_{hh} + B_{vv} + 2\,\mathrm{Re}[\dot{B}_{hv}] - 2P_n}$	Not applicable	Spectral slanted linear depolarization ratio (sSLDR)
5. $\dfrac{\lvert B_{hh} - B_{vv} + 2i\,\mathrm{Im}[\dot{B}_{hv}] - 2P_n\rvert}{\sqrt{(B_{hh} + B_{vv} - 2\,\mathrm{Re}[\dot{B}_{hv}] - 2P_n)(B_{hh} + B_{vv} + 2\,\mathrm{Re}[\dot{B}_{hv}] - 2P_n)}}$	Not applicable	Spectral co-cross-channel correlation coefficient in slanted basis

Table 2.11.1: Spectral polarimetric variables

P_n in the formulas above is the mean noise power density.

Summation of the coherency matrix elements over the spectral lines with the detected signal yields the following integrated polarimetric variables:

Equation	LDR-mode parameter	STSR-mode parameter
1. $\sum (B_{hh} - P_n) / \sum (B_{vv} - P_n)$	Linear depolarization ratio (LDR)	Differential reflectivity (Z_{DR})
2. $\frac{\left\| \sum \dot{B}_{hv} \right\|}{\sqrt{\sum (B_{hh} - P_n) \sum (B_{vv} - P_n)}}$	Co-cross-channel correlation coefficient (ρ_{cx})	Correlation coefficient (ρ_{hv})
3. $\arg\left[\sum \dot{B}_{hv}\right]$	Spectral co-cross-channel differential phase (Φ_{cx})	Spectral differential phase (Φ_{DP})
4. $\frac{\sum \left(B_{hh} + B_{vv} - 2\,\mathrm{Re}\left[\dot{B}_{hv}\right] - 2P_n\right)}{\sum \left(B_{hh} + B_{vv} + 2\,\mathrm{Re}\left[\dot{B}_{hv}\right] - 2P_n\right)}$	Not applicable	Spectral slanted linear depolarization ratio (SLDR)
5. $\frac{\sum \left\| B_{hh} - B_{vv} + 2i\,\mathrm{Im}\left[\dot{B}_{hv}\right] - 2P_n \right\|}{\sqrt{\sum \left(B_{hh} + B_{vv} - 2\,\mathrm{Re}\left[\dot{B}_{hv}\right] - 2P_n\right) \sum \left(B_{hh} + B_{vv} + 2\,\mathrm{Re}\left[\dot{B}_{hv}\right] - 2P_n\right)}}$	Not applicable	Spectral co-cross-channel correlation coefficient in slanted basis

Table 2.11.2: Integrated polarimetric variables

In the STSR mode two additional parameters are calculated. Using the spectral lines induced only by Rayleigh scattering (criterium has to be developed using observations) the specific differential phase shift and differential attenuation are calculated, respectively:

$$K_{DP} = \frac{d\left\langle \arg\left[\sum \dot{B}_{hv}\right]\right\rangle}{2dR} \tag{2.10.13}$$

$$A = \frac{d\left(\sum (B_{hh} - P_n) / \sum (B_{vv} - P_n)\right)}{2dR} \tag{2.10.14}$$

Here R is the range.

3 Radar Software

The following conventions are used in this software description:

- Messages generated by the program that have to be acknowledged are printed in red. Example: ***No connection to Radar PC!***
- Button labels are printed in green: ***Delete Program***
- Messages that have to be answered by ***Yes*** or ***No*** are printed in light blue: ***Overwrite the existing file?***
- Labels are printed in grey: ***UTC***
- Names of group boxes are printed in blue. Example: ***El / Az Positioner***
- Names of tags are printed in violet: ***Radar Monitoring***
- Names of menus are printed in black: ***File Transfer***
- Labels of Entry-Boxes are printed in light blue: ***Const. Elev. Angle***
- When a speed button shall be clicked, this is indicated by its symbol: 
- Hints to speed buttons are printed in brown: ***License Manager***
- Selections from list boxes are printed in magenta: ***DD-Rec. TBs***
- Selections from radio buttons or check boxes are printed in dark green: ***Ze***
- File names are printed in orange: ***MyFileName***
- Directory names are printed in dark blue: ***C:\Programs\RPG-FMCW***

3.1 Host Software Installation

3.1.1 Host PC Hardware Recommendations

The hardware recommendations for the host PC (H-PC) are:

- Intel I7 core, 8 parallel threads (speed requirement)
- 4 GB free RAM
- High resolution graphics screen (at least 768 vertical points)
- Ethernet interface (mandatory)

3.1.2 Directory Tree

In order to install the host software from a flash drive the following steps should be performed:

- Start the Windows® operating system (make sure you have Administrator rights)
- Start the Windows Explorer®
- Insert the Radar Software flash drive (included in the delivery package)
- In Windows Explorer® click on the flash drive
- Click on the ***RPG-FMCW***-folder (flash drive) and drag the whole folder to ***MY_DIRECTORY*** (user selectable). Notice: Do NOT drag the ***RPG-FMCW***-folder to the Windows Desktop. The host software will not work properly from the Desktop folder.
- Browse to the ***RPG-FMCW***-folder on the hard drive and localize the executable ***FMCW_H.EXE***. Create a shortcut of this program on the Windows® Desktop

- Right-click on the shortcut (Desktop) and select 'Properties->Compatibility'. Check the checkbox 'Run as Administrator' and exit with 'Apply'.

The directory tree contains a few files the user should notice.
Example: If '***MY_DIRECTORY***' is the directory ***D:\Programs*** the complete tree should look as:

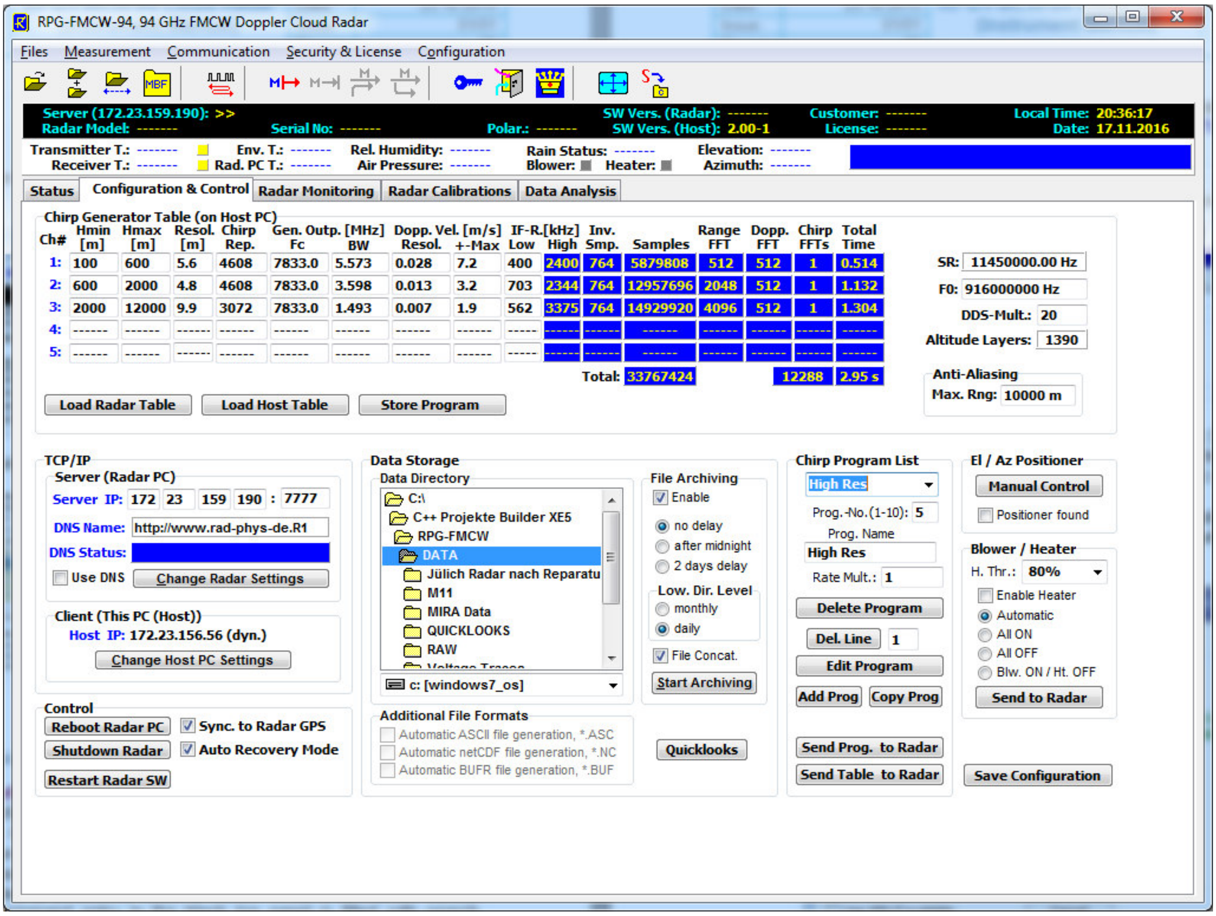
D:	
|---***Programs***	
|---***RPG-FMCW***	root directory, contains the executable ***FMCW_H.EXE***
|---***CONFIG***	configuration file ***FMCW_H.CFG***, chirp table ***CHIRP.TBL***
|---***DATA***	directory for data file archiving (can be changed)
|---***LICENSE***	license file ***LicID.DAT***.
|---***LOG***	TCP-IP log file ***Get_IP.LOG***, file access error log file
|	***FileAccessErrors.LOG***, pointing map log files
|---***RADAR PC***	Radar PC executable ***FMCW_R.EXE***, SW updates
|---***MDF_MBF***	directory for measurement definition files (MDF)
|---***TEMP***	temporary files, used during data file archiving

The Host PC (H-PC) is defined as the 'external' PC. It communicates with the Radar PC (R-PC) inside the instrument. The R-PC collects the radar data and evaluates it (FFTs, Doppler spectra, calculation of moments, reflectivity corrections, etc.). The data link between H-PC and R-PC is via Ethernet. The radar can be plugged into a network (fixed IP) and is then accessed by the H-PC from any network location.
The R-PC has a similar directory structure as the H-PC:

C:	
|---***RPG-FMCW-R***	root directory, contains the executable ***FMCW_R.EXE***,
|	previous executable version ***FMCW_R.OLD***
|	executable launcher ***Spawn_FMCW_R.EXE***
|	DLLs for data acquisition and digital boards
|---***CALIB***	calibration files ***ABSCAL.CLB***, ***ABSCAL.HIS***, zero cal.
|	files ***X_ZERO.CAL***, anti-alias filter correction
|	***RecLpfCorr.DAT***
|---***CHIRP_GEN***	chirp generator files for re-programming chirp table
|---***CONFIG***	configuration file ***FMCW_R.CFG***, chirp table ***CHIRP.TBL***
|	abs. calib. chirp definition file ***ABSCAL.CHP***
|	pointing chirp definition file ***FASTSCAN.CHP***
|	positioner baudrate and drive parameters ***POSI.BDR***,
|	***POSI.PAR***
|---***DATA***	directory for data file archiving
|---***LOG***	TCP-IP log file ***Get_IP.LOG***, FIFO reset log file
|	***FIFO_RES.LOG***, pointing map log files
|---***MDF***	directory for measurement definition files (MDF)
|---***RETRIEVAL***	retrieval files (neural network)
|---***ATN***	atmospheric attenuation retrieval file (ASCII)
|---***BINARY***	binary retrieval file versions
|---***LWP***	integrated liquid water path (LWP) retrieval
|---***SPC***	spectrum retrieval file
|---***TMR***	Tmr retrieval for sky tipping calibrations
|---***TEMP***	temporary files, used during data file archiving

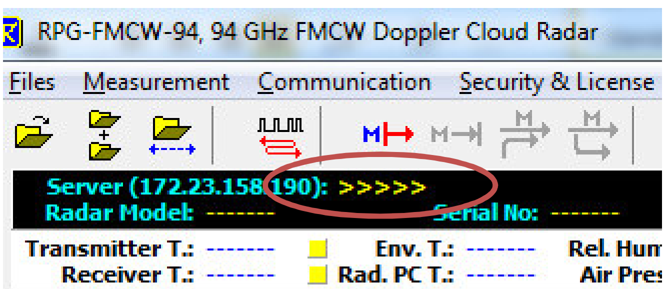
3.2 Getting Started

After the host software has been installed according to section 3.1, click  on the desktop to execute it:

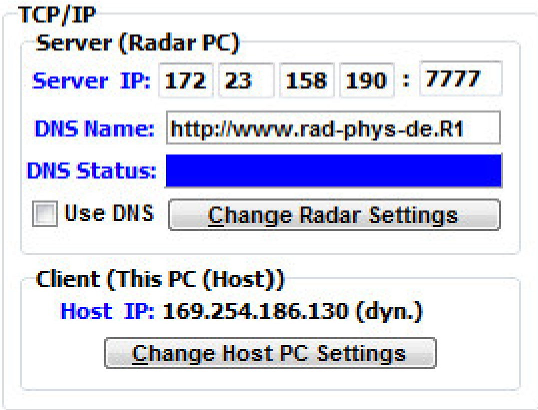


The screen is showing the ***Configuration & Control*** register tag. On top of the register tags, environmental parameters (surface sensor data), position and blower status are displayed. The black panel summarizes the radar ID information, as model number, polarisation, customer code, software version and license status.

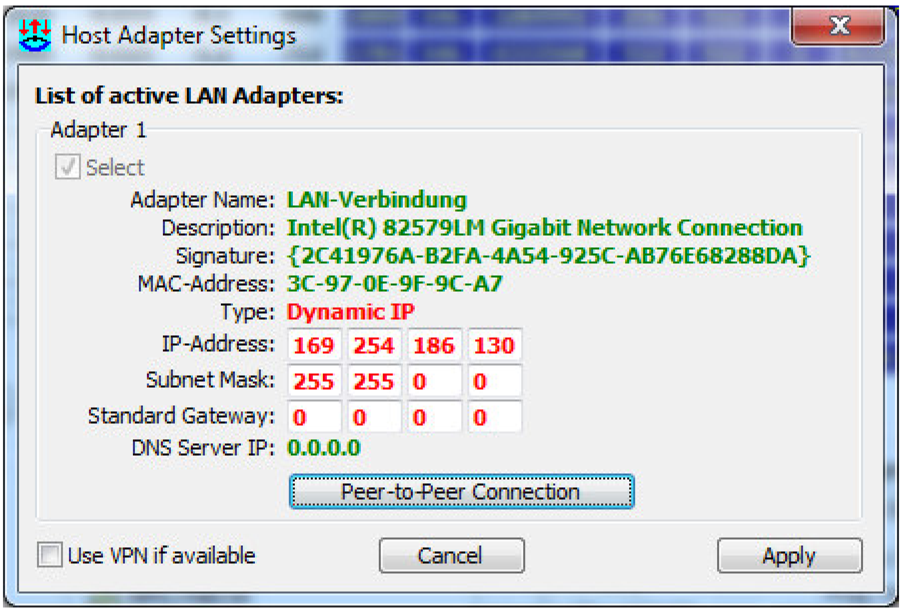
As soon as the H-PC application starts, it is looking for an Ethernet connection to a radar, assuming the H-PC is connected to a network, router or switch. When a connection cannot be established, the TCP-IP command entry in the black top panel is filled with search indicators:



The Host assumes a radar (Server) with a certain IP connected to the network or directly connected (peer-to-peer connection). This IP is defined in the ***TCP / IP*** box:



When a new radar is shipped, its IP setting is 192.168.0.1:7777 (default), subnet mask 255.255.255.0. This IP must be entered to the fields right of the label ***Server IP:***. Because the radar's subnet mask is 255.255.255.0, the host IP should be in the same subnet, e. g. 192.168.0.x (x can be any number except for 0 and 1).
The H-PC IP settings may be changed from within the radar application when clicking ***Change Host PC Settings*** (you must run the host software with Administrator rights for this command):



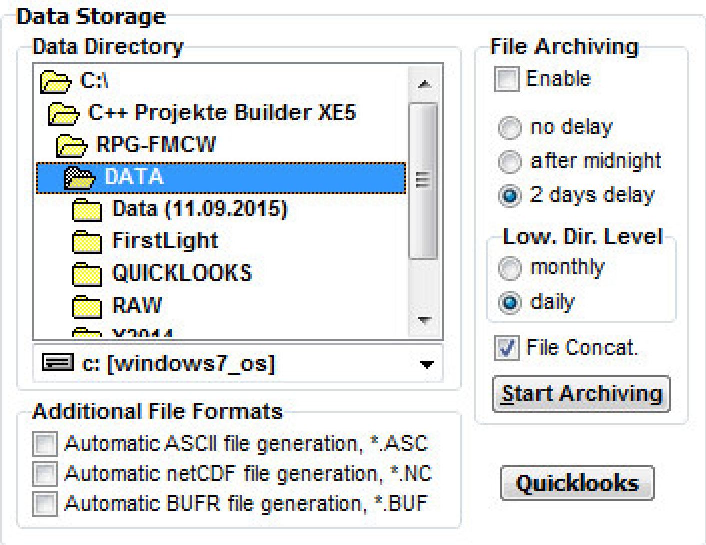
If the IP address you enter here is available within the network the host is connected to, the ***Apply*** command will automatically change the host IP accordingly. An alternative, of course, is the standard procedure using the Windows IP setting menu.

Once connected to the radar, its IP setting can be modified remotely with ***Change radar settings***.

The rest of the ***Configuration & Control*** panel is sub-divided into the following groups:

- Data Storage
- El / Az Positioner
- Blower / Heater
- Control
- Chirp Program List
- Chirp Generator Table

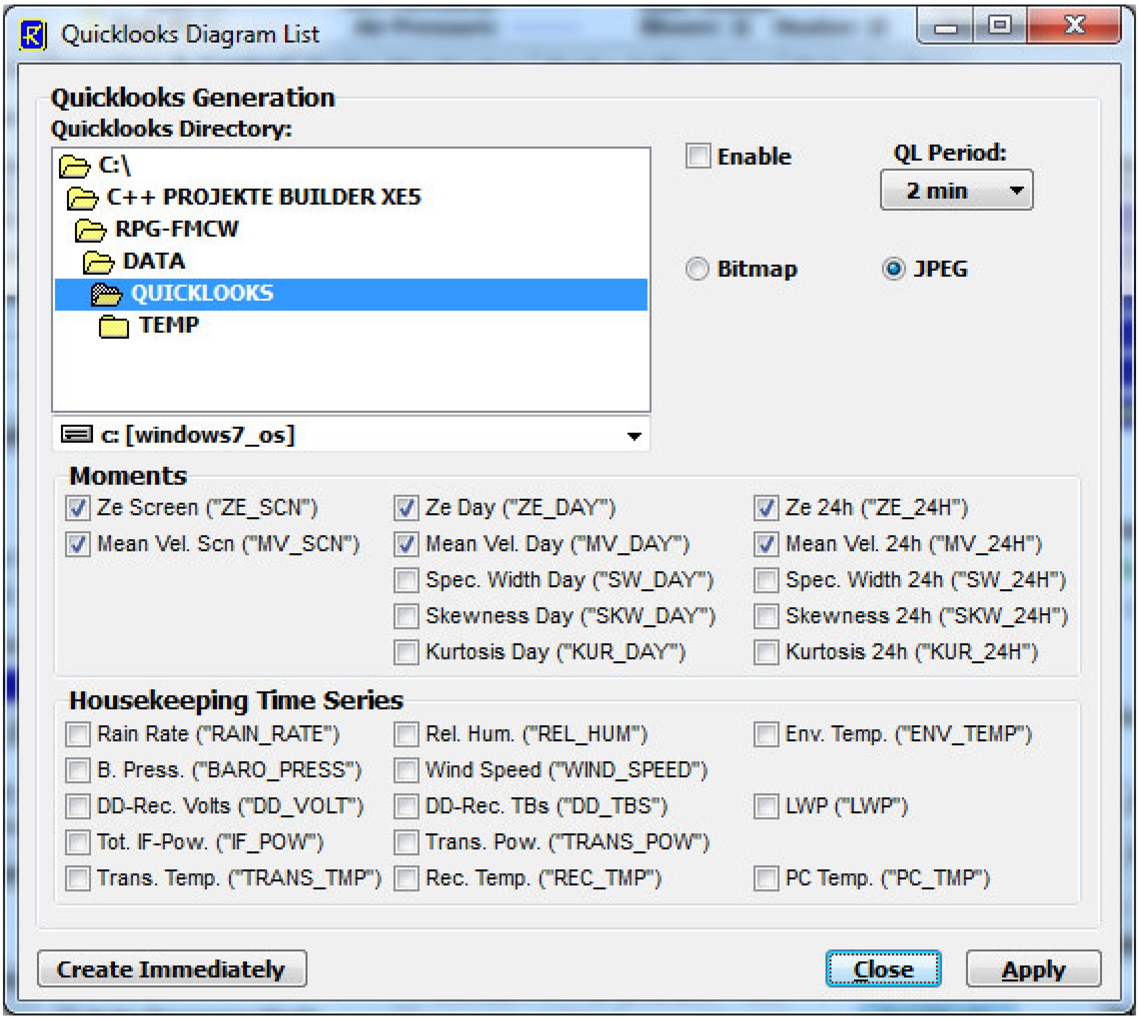
3.3 Data Storage



During measurements the recorded radar data is automatically stored to the directory selected in the ***Data Directory*** box (in binary format). If other formats are required, they may be selected in ***Additional File Formats***. These other formats are stored to the same data directory as the binary files.
Data archiving is a useful feature to prevent the data directory from being filled with ten thousands of files which may overload the operating system. MS operating systems cannot handle many (in the order of ten thousands) files in a single directory. If ***Enable*** is checked, the program automatically creates sub-directories in the data directory and stores the data files according to the year, month and day they are created. For example, a file ***16111623_P01.LV1*** would be stored in a directory ***…\RPG-XCH-DP\Data\Y2016\M11\D16***\ if ***daily*** is checked or in ***…\RPG-XCH-DP\Data\Y2016\M11*** if monthly is checked. Archiving, if enabled, is performed for data files immediately, after midnight or after 2 days, depending on the radio button selection. If the user wants to immediately archive data files, he may click the ***Start Archiving*** button.
Data files are created every hour. File concatenation to daily files (with deleting the hourly files) can be activated (checkbox ***File Concat.***).

3.3.1 Quick-looks Generation

In order to create quick-looks of radar reflectivity, mean velocity, etc., the user may open the related menu with ***Quicklooks***. The following menu opens:



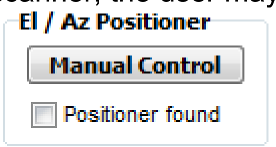
In the upper left corner the quick-looks file path is specified. Quick-looks generation is enabled / disabled by checking / unchecking ***Enable***. The updating period is set in the combo-box labelled **QL Period:** and ***Bitmap*** and ***JPEG*** file formats can be selected.
Within the **Moments** box the moments Ze, Mean Vel., Spec. Width, Skewness and Kurtosis can be selected as “Screen” (SCN), “Day” (DAY) and “24h” (24H) versions:

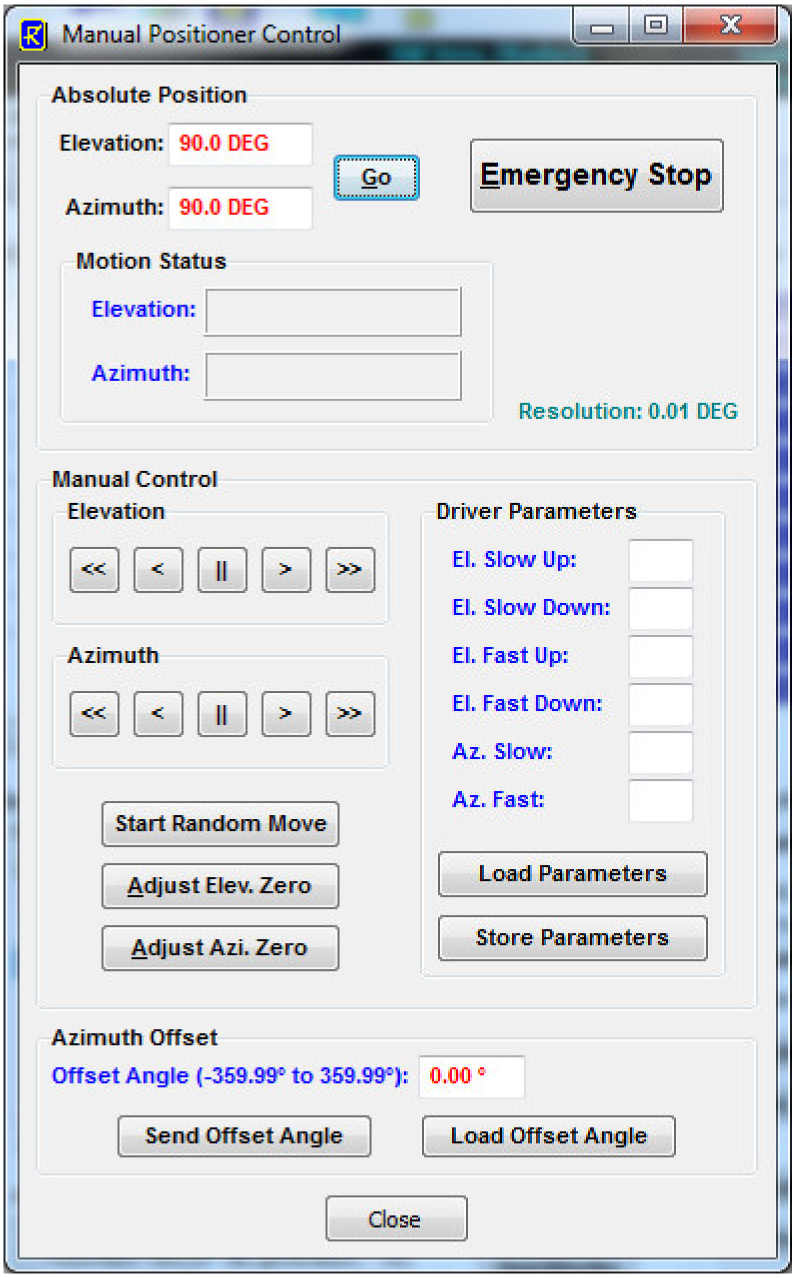
- The SCN-versions display the original current monitoring screens for Ze and mean Doppler velocity.
- The DAY-versions display all data collected during the current day.
- The 24H-versions display all collected data during the last 24 hours.

For DAY- and 24H-versions the data is taken from the archive. Therefore, in order to process these versions, data archiving must be enabled and “File Concatenation” must be disabled.
Additionally, all housekeeping data time series are available as quicklooks.
The names given in brackets are the associated filenames.

3.4 El / Az Positioner

When the radar is mounted to a scanner, the user may manually control its movements:



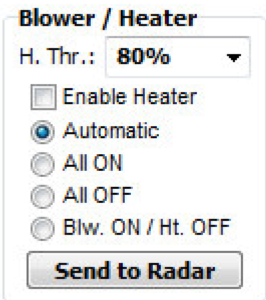


The angular stepper resolution is 0.02°.
Elevation movements are defines as follows:

- Zenith: +90°
- Horizontal: 0°
- To ground: -90°

The azimuth value range is set from 0° to 360° (clockwise from top view).
With the **<<**, **<**, **||**, **>**, and **>>** keys the positioner can be moved without specifying a target angle. Slow and fast motions can be selected and stopped by clicking the **||** button. In emergency cases any movement can be immediately terminated by clicking the ***Emergency Stop*** button.
The user should not modify the driver parameters in order to maintain a smooth positioner motion.

3.5 Blower / Heater Control

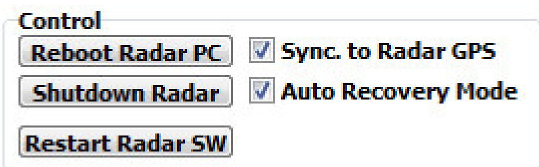


The radome blowers and air heater modules can be controlled in the following ways:

- Automatic: The blowers / heater are turned on during rain events or when the surface relative humidity exceeds the value listed in the humidity threshold combo box. A threshold of 70% is recommended.
- All ON: Enforces the blowers / heater to turn on immediately
- All OFF: Enforces the blowers / heater to turn off immediately
- Blw. ON / Ht. OFF: Only turns the blowers on and leaves the heaters off.

The heater modules may also be disabled completely when unchecking ***Enable Heater***.
The settings only become active if the parameters are ***Send to Radar*** while connected.

3.6 Other Controls

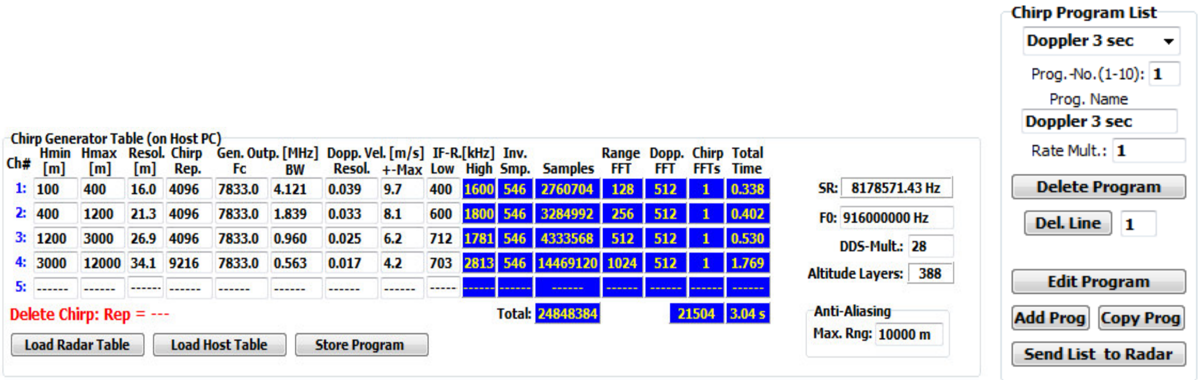


The radar PC can be remotely rebooted or shut down. The shutdown feature should be used whenever the radar shall be turned off (power off). Only the controlled shutdown procedure will ensure that all open files are closed and all interfaces are initialized.
Restarting only the radar software without resetting the radar PC is useful after software updates, when a new radar software version has been transferred and shall be started.
The H-PC system clock may be synchronized to the radar's GPS clock. This feature replaces a time server. The H-PC is synchronized every hour when a radar is connected and ***Sync. to Radar PC*** is checked.

When checking ***Auto Recovery Mode***, the user enables the radar to automatically restart a running measurement after it has been interrupted by a power failure. The host PC software permanently scans the Ethernet connection for the radar and will automatically reconnect to it to monitor and archive the next measurement samples.

3.7 The Chirp Table

As explained in chapter 2, the FMCW radar transmits a continuous sequence of saw-tooth frequency chirps. The chirp parameters, like duration and slope, determine the ranging resolution and range mapping to the IF band. With RPG's cloud radars it is possible to user define multiple chirp sequences for different altitude ranges. This is advantageous, because it allows for the optimization of range resolution, sensitivity, maximum Doppler velocity and Doppler resolution for different altitude layers. For instance, a higher range resolution of a few meters is often desirable within the planetary boundary layer (PBL), while a coarser resolution is acceptable (e.g. 30 m) above the PBL which increases sensitivity at longer distances from the radar (larger scattering volumes). Also the integration time for each altitude range can be individually adjusted by controlling the chirp repetition in each sequence.



A set of chirp sequences (a sequence is defined as a repetition of identical chirps) is called a program. Up to 10 programs may be defined by the user, for example to realize different sampling rates.

In the chirp table each line defines a chirp sequence and the white fields can be edited while the blue fields are parameters resulting from the edited settings. Each sequence has a minimum and maximum range and adjacent sequences must meet without overlaps or gaps between them (meaning the maximum range of a sequence must be the minimum range of the next sequence). The minimum range is 50 m, the maximum is 20.000 m.

The chirp's center frequency is set by the **Fc** parameter in the table. The numbers in the FC-column are the chirp generator's output frequencies in MHz. These are multiplied by a factor of 12 to be transmitted by the radar. The chirp's frequency bandwidth is listed in the BW-column (also referring to the chirp generator's output). Other columns are the **Resol.**-column (range resolution), the **Dopp. Vel. Resol**.-column (Doppler velocity resolution), the **+-Max**-column (maximum unambiguous Doppler velocity), the **Chirp Rep**.-column (chirp repetition, used to control the sequence duration) and the **IF-R Low**-column (the low IF limit in kHz, must be higher than 350 kHz).

According to equations (2.2.4), (2.3.3), (2.4.3) and (2.4.4) the different parameters depend on each other and are automatically adjusted when changes are made to editable fields.

With the ***Load Radar Table*** command the current chirp table is loaded from the radar (if connected to it). It is not possible to edit this table directly. When the host application is

started and a radar PC is found to connect, the radar's chirp table is automatically loaded, which is indicated by the table's box caption ***Chirp Generator Table (on Radar)***. The user may overwrite the table with the table stored locally on the H-PC (***Load Host Table***) in the ***…\CONFIG*** directory (*CHIRP.TBL*). Only the local table can be edited.
The ***Chirp Program List*** box summarizes commands for building and editing chirp tables. The uppermost combo box is listing the different table programs. Each program has a name and number (1-10). If a new program shall be added, a new name should be entered ('Prog. Name') and an unused program number ('Prog.-No. (1-10)') must be specified. Then a click on ***Add Prog*** will open a blank program table. With ***Copy Prog*** an existing program can be copied to a new program number and edited afterwards. After all changes to a program have been finished, ***Edit Program*** stores these changes to the *CHIRP.TBL* file.

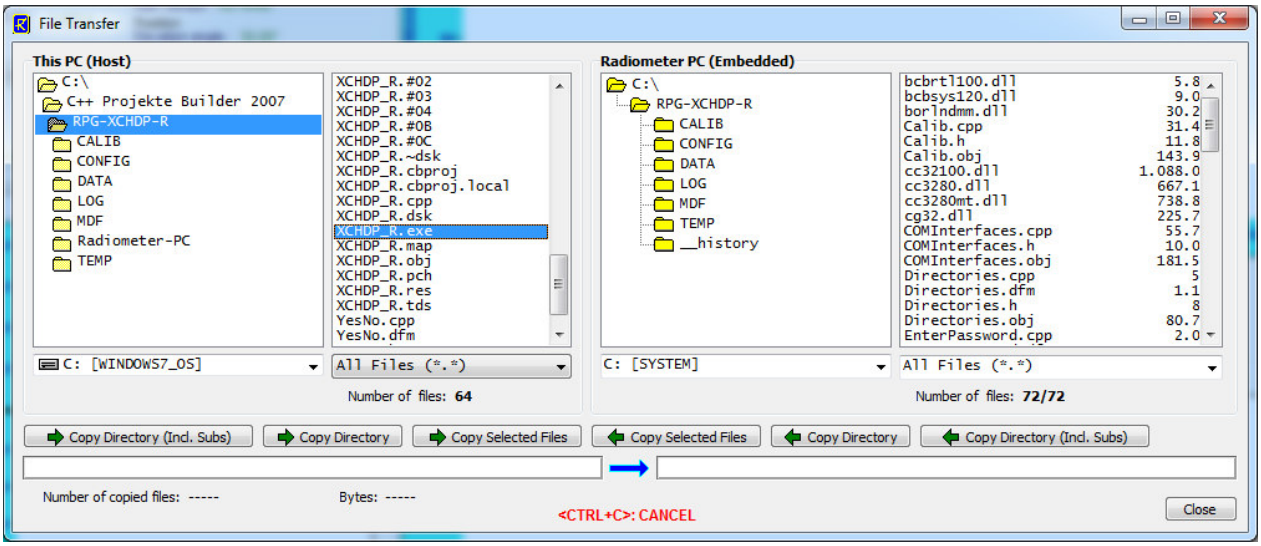
A modified table is sent to the radar with ***Send List to Radar***, overwriting the existing chirp programs on the chirp generator. The overwriting process may take a minute. After reprogramming the radar sends a log file of the successfully modified chirp sequences. This command can only be executed by the Administrator.

When a chirp table has been overwritten, the radar software should be restarted (section 3.6).

3.8 Exchanging Data Files

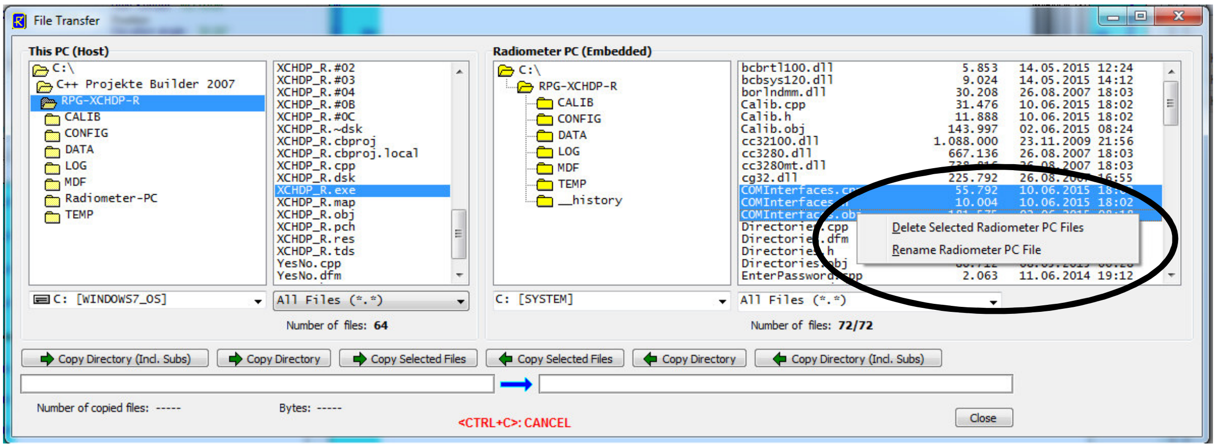
The radar PC is running a Windows®7 operating system. The radar software is stored in the directory ***C:\RPG-FMCW-R***. Write processes to this directory or its sub-directories is password protected. When upgrading the radiometer software, the new executable *FMCW_R.EXE* needs to be transferred to this root directory. This task should only be performed by the system administrator. **Overwriting *FMCW_R.EXE* with a non-operating software or deleting this executable file will disable all radar functions and requires to restore the executable directly on the radar PC without host access (using the direct access to the radar PC)!**

To get access to the radiometer directories click  (***File Transfer***).

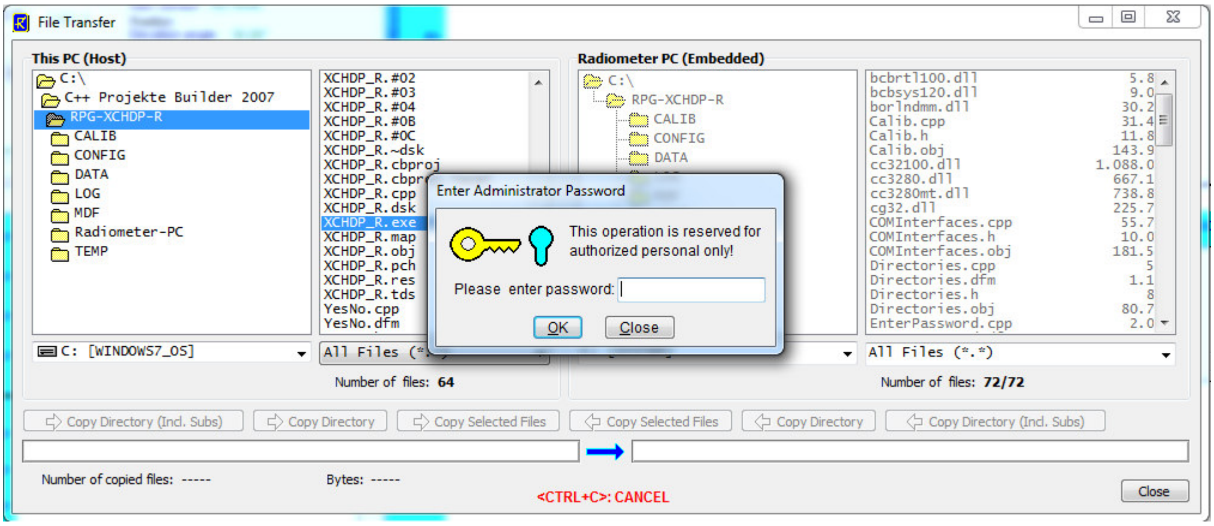


File transfer menu.

File transfer is necessary when backup data files need to be copied from the radar's hard disk to the host computer. If file backup is enabled for a measurement, the instrument stores all data files in its data directory ***C:\RPG-FMCW-R\DATA***. This data can then be downloaded after or during a measurement.
Files can be copied from one PC to the other by selecting them on the source directory, marking a directory on the destination PC and click ***Copy Selected Files***. Alternatively, a complete directory, with or without its sub-directories may be copied (***Copy Directory***, ***Copy Directory (Incl. Subs)***).
Files or directories can be deleted by marking them and clicking the right mouse button:



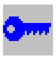
From the displayed drop-down list the user may select '***Delete Selected Radar PC Files***' or '***Rename Radar PC File***'. These functions are also available for directories.
If the user tries to send files to the ***C:\RPG-FMCW-R*** directory (the root directory), a password check is initiated:



The H-PC is asking for the entry of the Administrator password (APW) to complete the desired action. This mechanism prevents unauthorized users from overwriting important R-PC system files.

3.9 The License Manager

RPG's radars are delivered with a preliminary limited license of 30 days. Without activating an unlimited license, the radar terminates measurement execution when the limited license is expired. The common procedure to install a permanent license is the following:

Invoke the **License Manager** by clicking the  button.

License manager menu.

The user retrieves license status information with the ***Get License Status*** command. The license type (limited or unlimited) as well as the expiration date and time will be displayed. In order to obtain an unlimited license, the following steps have to be performed:

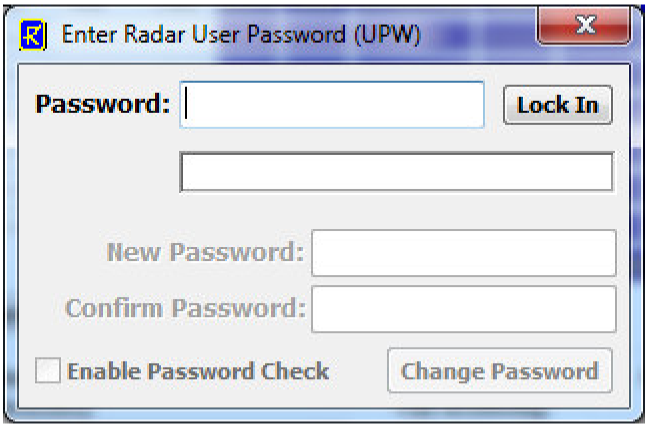
1. Connect to the radar and click the ***Get License ID from Radar PC*** button. The license ID code is then written to the file ***LicID.DAT*** stored in the H-PC ***…\LICENSE*** directory (see section 3.1).
2. Send the 'LicID.DAT' file to RPG (by e-mail to info@radiometer-physics.de). Then the 10 digit license code will be returned (also by e-mail).
3. Enter the 10 digit license code to the edit box in the license manager and click ***Send License Code***. The license manager will inform the user, if the unlimited license installation was successful or not. If not successful, please contact RPG again.

It is possible to extend a limited license. The procedure is as described above. Send the ***LicID.DAT*** file to RPG by e-mail and receive a license extension file ***LicCode.EXT***. Store this file to the H-PC ***…\LICENSE*** directory (see section 3.1) and click the ***Send License Extension File*** button in the license manager menu.

3.10 Administrator Password (APW)

System critical operations, for instance software updates or chirp table overwrites, are password protected. A new radar is shipped with the default Administrator password (APW) ***Administrator***. Note that the APW in this context is NOT the Administrator Password of the Windows® operating system. The APW referred to here is ONLY relevant for the radar application. The APW can be changed by clicking the button:

Before a new APW is defined, the old one must be entered first. The procedure is identical to the change of Windows® account passwords. The administrator password is stored on the radar PC (if connected) with ***Change Administrator Password***.
The host application will only ask for the APW when the host is connected to a radar. In stand-alone mode (without a connection) no password checking is applied.
When a connection to the R-PC is established, the radar will reserve this connection to the first client connecting to it. This first client is the real host. All other clients connecting after the host (secondary clients) will be informed that the R-PC is occupied. The locking to the host client remains for 5 minutes when the TCP-IP traffic between R-PC and H-PC has stopped. After this period, the R-PC is free for other clients to connect. If the R-PC is locked to a host client and a secondary client tries to connect to the radar, the host locking can be unlocked on the secondary client by the Administrator with ***Unlock Radar from Current Host***.
The radar is usually connected to a network. Therefore, theoretically everyone who knows the radar's IP address within this network can connect to it. In order to establish a protection mechanism, the Administrator may define a Radar User Password (UPW) ():



When the APW is entered as password, all disabled fields become enabled. The Administrator can then define a UPW, enable or disable the UPW checking and change the UPW, assuming the H-PC is connected to the R-PC.
The same menu is used for radar user lockin. If UPW checking is enabled on the radar, the UPW-window pops up for UPW entry the first time the user connects to the radar. When a valid UPW is entered (***Lock In***), the radar accepts the user for connection.

3.11 Software Updates

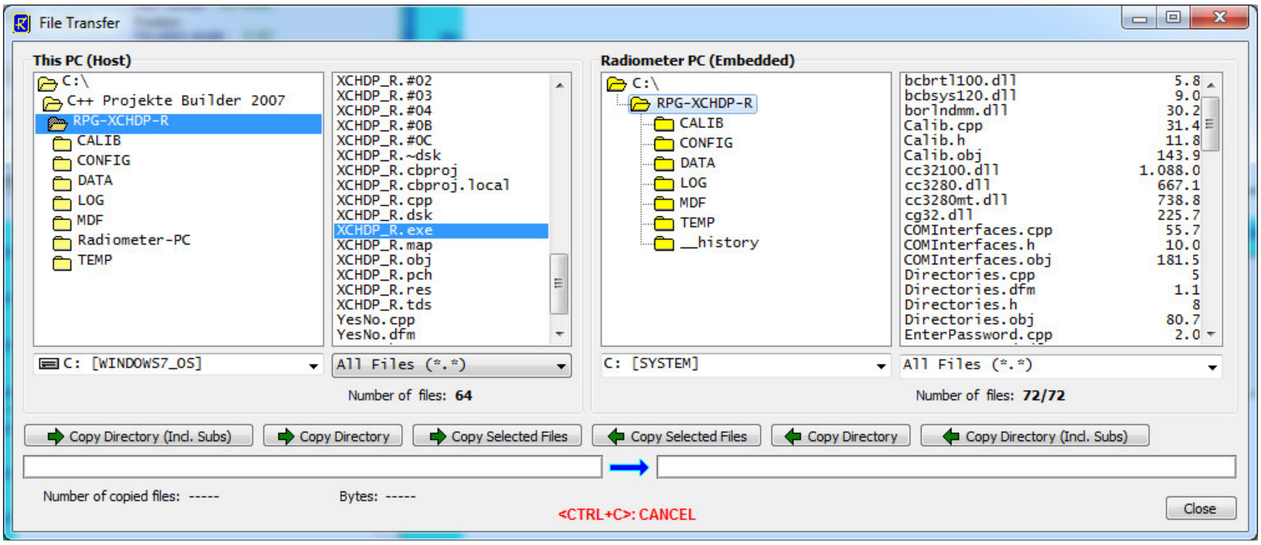
It may be desirable to update the H-PC and R-PC software version from time to time, in order to add advanced features to the data processing or to correct software bugs.
The radar SW is running on an embedded PC and is named ***FMCW_R.EXE***. This file is located in the radar's root directory ***C:\RPG-FMCW-R*** (see section 3.1).
The host SW name is ***FMCW_H.EXE*** and it is located in the application's root directory (…***RPG-FMCW***) on the H-PC.

For a SW update the following steps should be followed:

1. Step: Save the old software versions
a) Create a directory to save the old software versions (e.g. ***C:\MyPath\Save***).

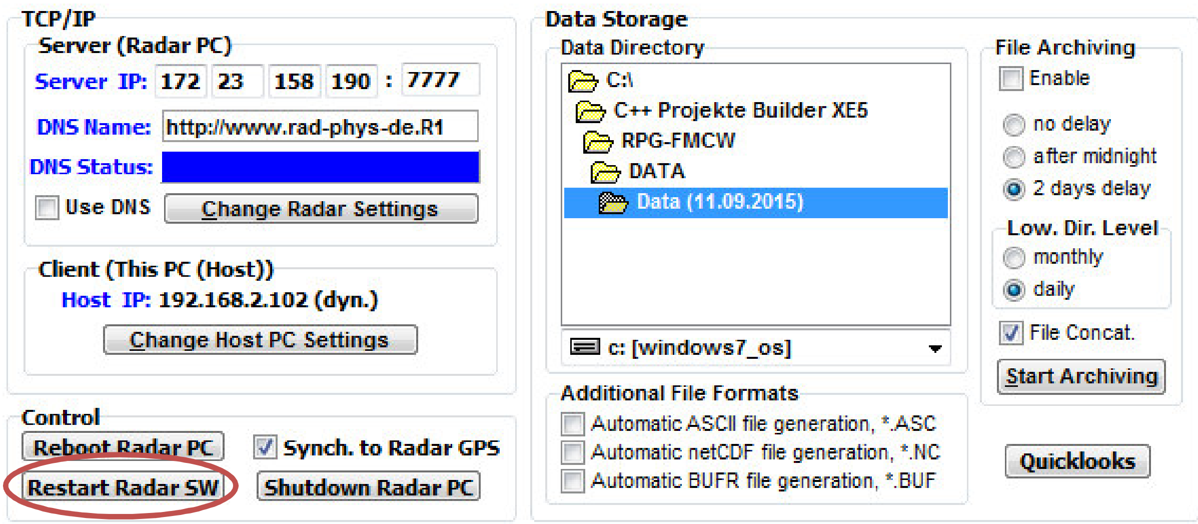
b) Connect the H-PC to the radar and enter the File Transfer Menu ( ). On the left side (H-PC, 'This PC (Host)') browse to the directory for saving the files (e.g. C:\MyPath\ SAVE) and on the right side (Radar) in the …***RPG-FMCW-R*** directory mark the ***FMCW_R.EXE*** file. Then click ***←Copy Selected Files***.
c) Locate the ***FMCW_H.EXE*** file in the …***RPG-FMCW*** directory on the H-PC and copy this file to the ***C:\MyPath\Save*** directory (by using the Operating System File Explorer).

2. Step: Overwrite the old versions by the new ones
a) Copy the new version of ***FMCW _R.EXE*** (the R-PC software) to an arbitrary directory on your host PC (e.g. …***RPG-FMCW\RADAR PC***). In the file transfer menu, browse to that directory. Mark the ***FMCW _R.EXE*** file in the file list within the ***This PC (Host)*** box and mark the …***RPG-FMCW-R*** directory in the ***Radar PC (Embedded)*** box. Click the ***Copy Selected Files→*** button. Because you are now going to overwrite a file in the radar's root directory, you must enter the Administrator password to proceed.



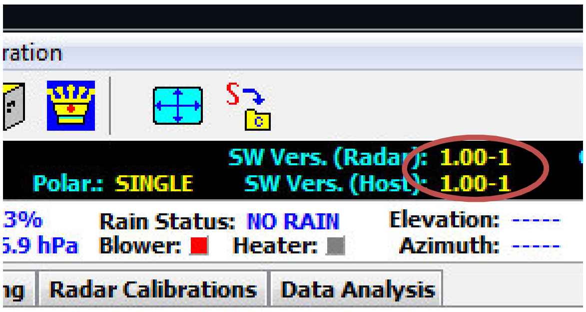
File Transfer Menu during software update procedure.

b) Reload the R-PC software (see below) to run the new ***FMCW_R.EXE*** version. Wait for approximately 10 seconds until the R-PC has restarted ***FMCW_R.EXE***. The H-PC will automatically re-connect to the radar.
c) Terminate ***FMCW_H.EXE*** on the host and overwrite it by the new version.
d) Execute ***FMCW_H.EXE*** and reconnect to the radar.



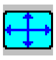
Reloading R_PC software (***Configuration & Control*** ***register page)***

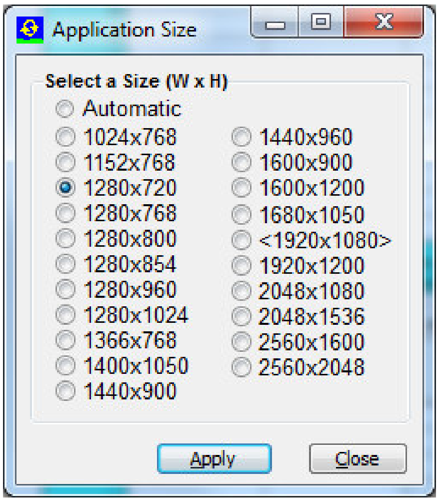
The software upgrade is finished. You can confirm the successful upgrade by reading the software version numbers of the embedded R-PC and the H-PC in the black status line on top of the application screen. Both version numbers should be identical.



3.12 Application Size on Screen

The host SW can scale the application to almost any common screen size. This may be useful if you want to run a beamer to display the host application in the beamer's screen resolution and size.

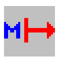
Click  to enter the **Application Size** menu:

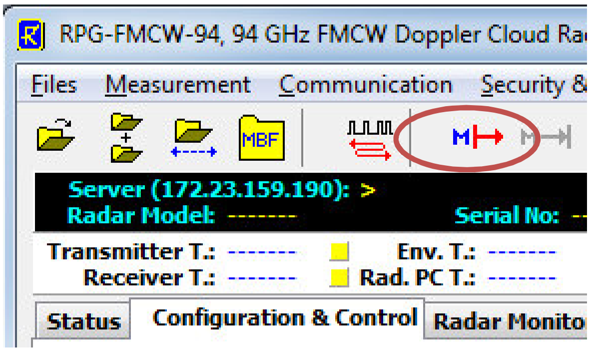


The screen size in brackets <…> is the host's maximum screen size but you can select any (smaller) sizes to fit the application on other screens (like a beamer). The application sizing can be done any time, even during a running measurement.

3.13 Starting Measurements

Before a measurement can be started on the radar PC, a measurement definition file (MDF) needs to be created first, containing all details of the measurement setup. This file is then sent to the radar for execution. Please refer to section 3.13.1 to learn how to create MDFs.

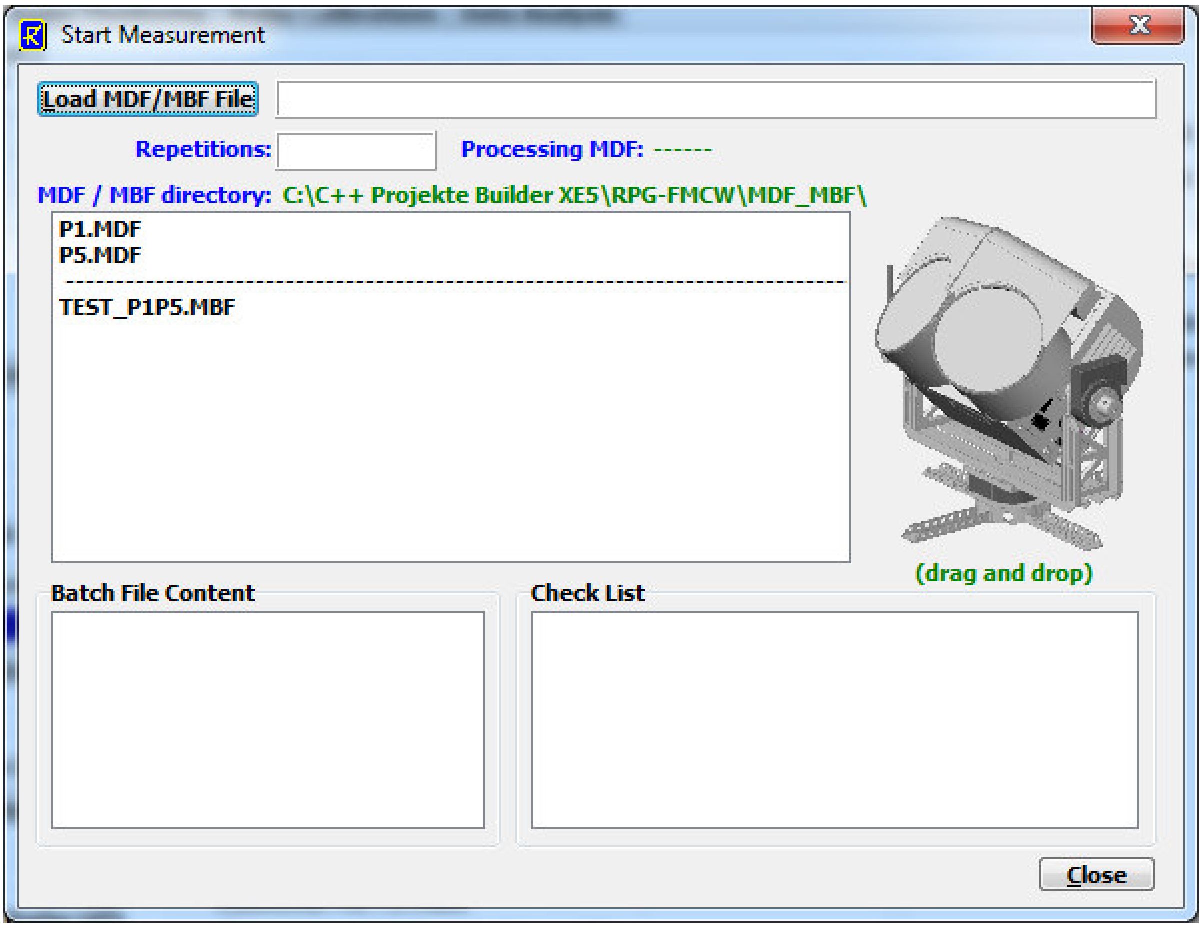
When a host successfully connects to the radar and the radar is in STANDBY mode, the radar is ready to start a measurement. This status is indicated by the enabled  button in the application's shortcut panel:



When an MDF or MBF is loaded (***Load MDF/MBF File***), its contents and repetition factor are displayed. In addition some pre-checks are performed, e.g. radar configuration, MDF version number, availability of chirp program number, etc. A variety of other checks ensure that no erroneous command data is sent.

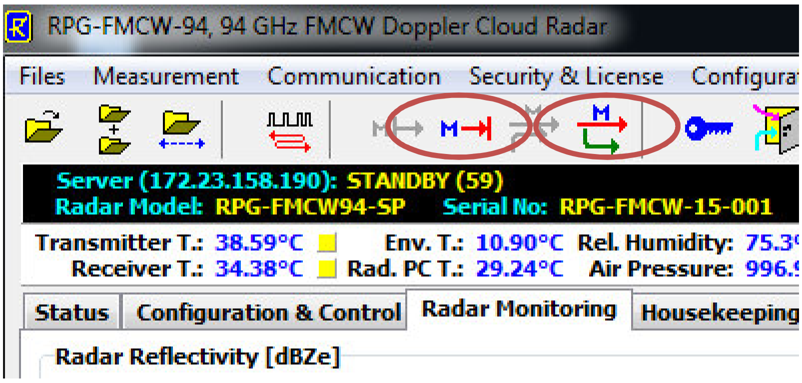
When the consistency check of a MDF is finished, the test result is displayed in the ***Check List***. The batch can only be sent to the radiometer if all consistency checks have finished with the status OK. Then the MBF is transmitted automatically.

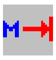
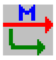
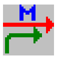
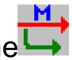
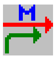
The H-PC 'remembers' the directory where MDFs and MBFs are stored from a previous ***Load MDF/MBF File*** command. This directory is listed in green. In the MDF / MBF list, MDFs are separated from MBFs by a dashed line. Dragging a file from the list and dropping it on the radar image on the right (or simply double clicking the file) is starting the measurement, if the consistency checks have been passed successfully. In this case the measurement launcher is closed automatically.



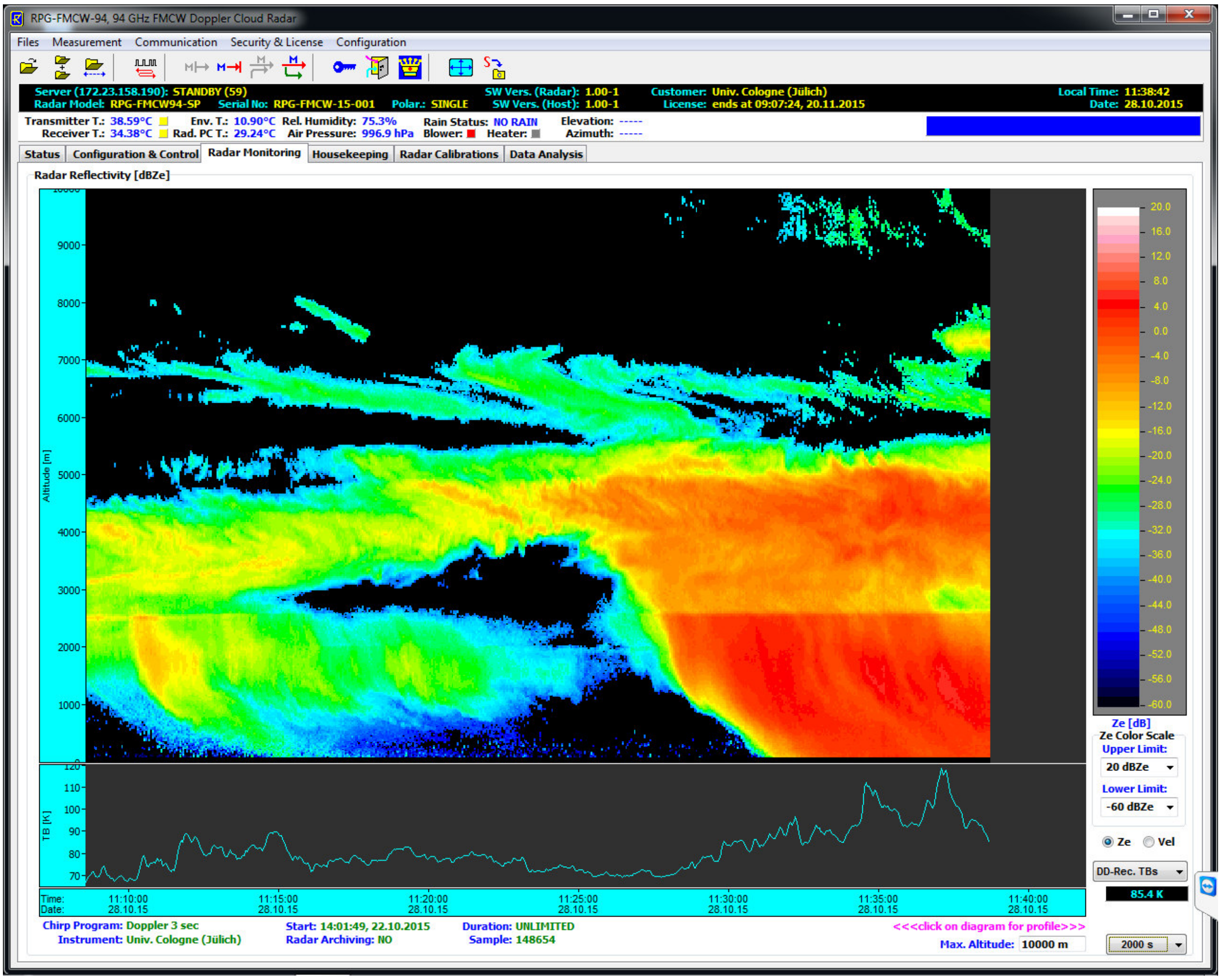
Measurement Launcher

Once a measurement has been launched, the control buttons in the shortcut panel change in the following way:



The  button is used to terminate the running measurement on both, the radar and the host, while the  button enables the host to drop off the measurement and leave the radar alone to continue. In both cases all monitored data samples are stored and the associated files are closed.
If the radar is running a measurement and the host connects to it, the H-PC realizes the active status and enables the  button for the host to jump on the measurement and start monitoring it. The  and  buttons do not affect the radar activities during a measurement, but act as host monitoring toggle switches.

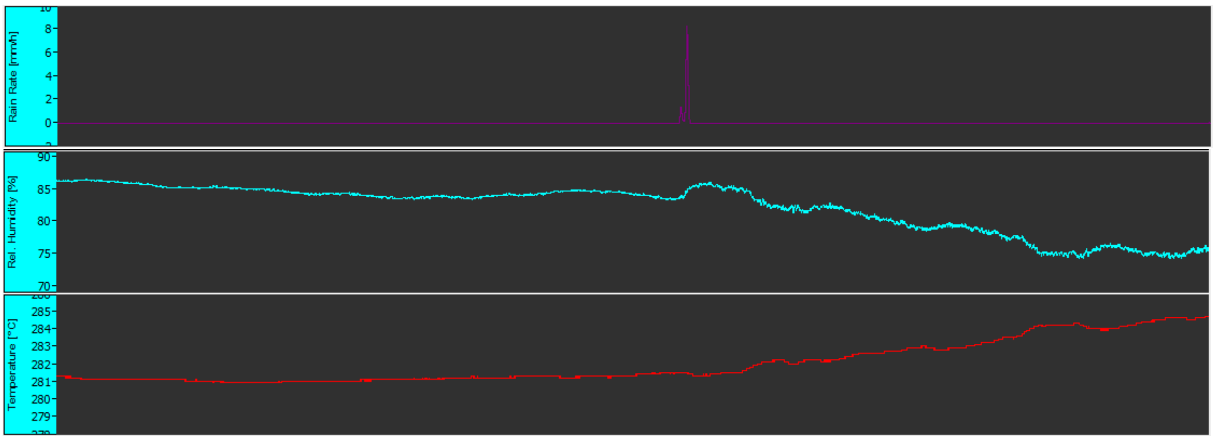
The radar profiles are displayed in the register page ***Radar Monitoring*** which acts as a real time display. In the main graphics area a color coded time series of reflectivity (in dBZe) or mean velocity (in m/s) is shown. Two radio buttons ***Ze*** and ***Vel*** switch between the two alternatives (higher moments can be displayed within the 'Open File' menu). The color coding limits are user adjustable.

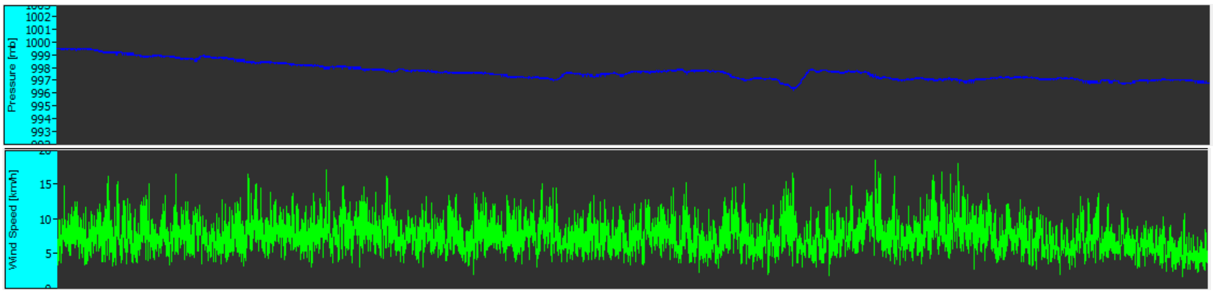


***Radar Monitoring* register page**

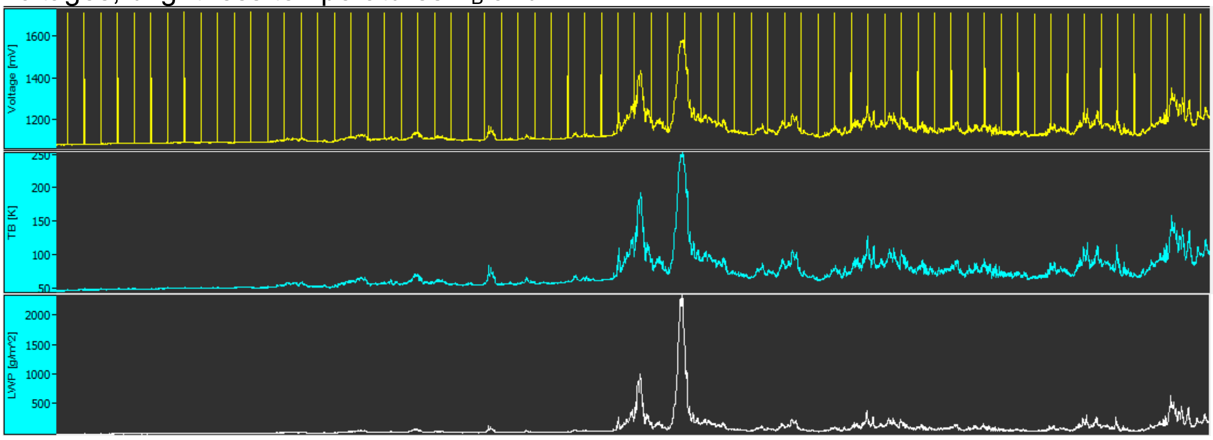
Underneath the main display area a switchable time series of different useful parameters is plotted. The parameter is selected from a combo box on the right side of the time series.

The radar is equipped with a weather station, providing information about environmental temperature, rel. humidity, barometric pressure, wind speed / direction and rain / snow rate:

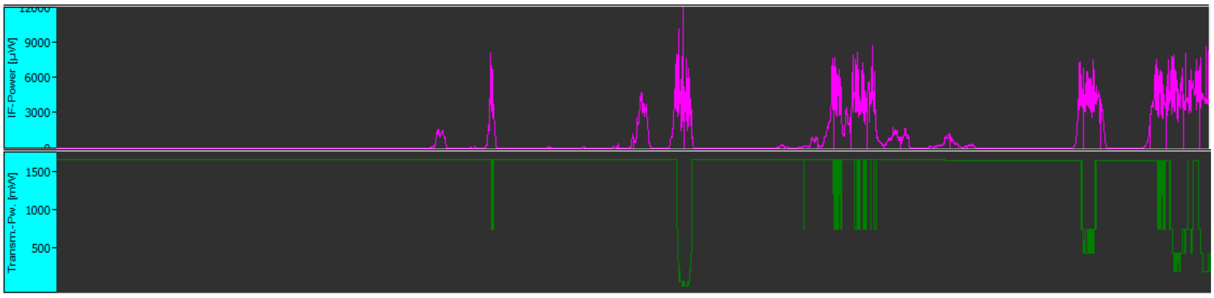




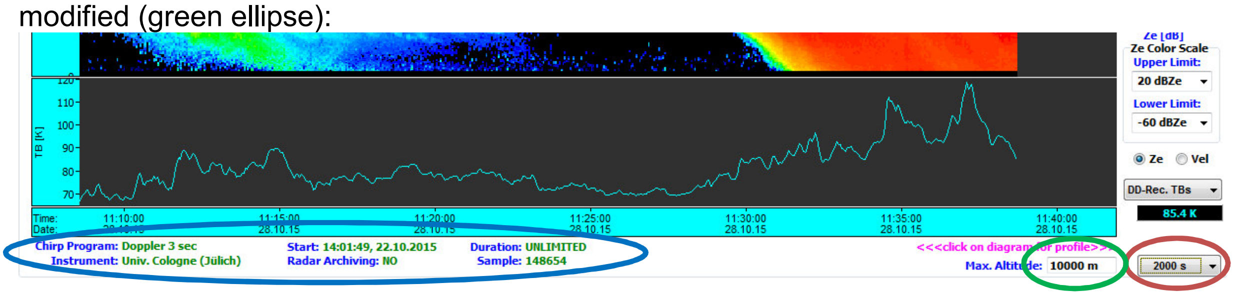
Another time series group is related to the direct detection passive channel at 89 GHz, which is intended for deriving LWP. Implemented are the DDR (Direct Detection Receiver) detector voltages, brightness temperatures T_B and LWP:



Additionally, information about the IF power level at the ADC board input (end of IF chain) as well as the transmitter power level are presented. The later one demonstrates the automatic power levelling during periods of strong reflections intended for preventing receiver saturation:



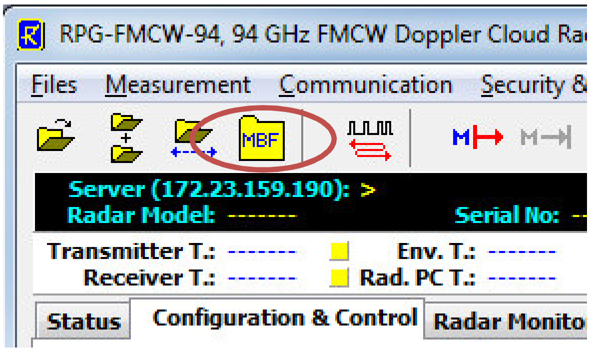
The time series time span is set in another combo box (red ellipse) at the bottom line of the screen. Also the maximum vertically displayed altitude in the main screen area can be modified (green ellipse):



Additional information about the chirp program in use, the measurement start and duration, the customer code and radar PC archiving status is plotted (blue ellipse).

3.13.1 Creating MDFs and Batch Files (MBFs)

In order to create a measurement definition file (MDF) to be sent to the radar PC, please click on the icon below:



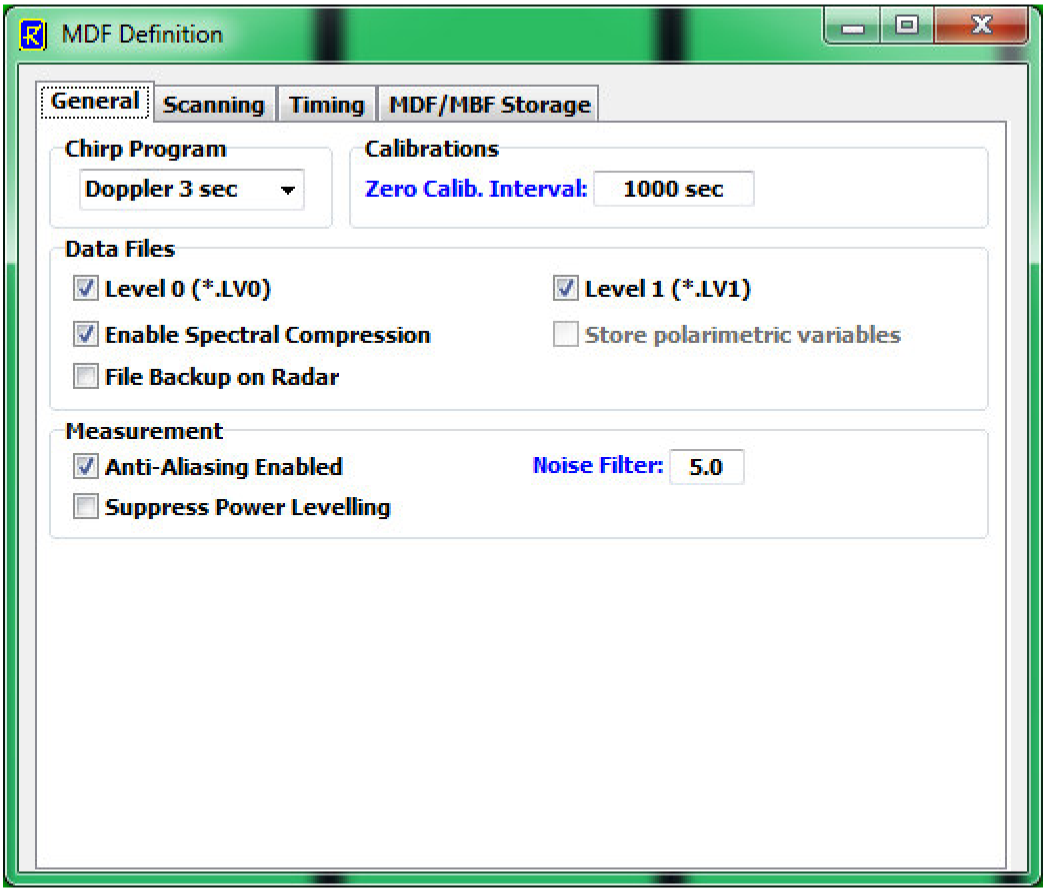
This launches the MDF window which consists of a register of four tags, summarizing different aspects of a measurement.

The ***General*** tag opens a setup menu with general measurement parameters:

Chirp Program: Selection of chirp program to be run on the radar. A single MDF can only define one of the available chirp programs.

Calibrations: Here the user specifies the period for zero calibrations. These calibrations are automatically performed inside the radar. As explained in the calibration chapter, the radar is using a Dicke Switch reference target for frequent calibrations of the radar receiver's gain which may drift in the time frame of hours. In a zero calibration, the radar's transmitter is turned off while the Dicke Switch is closed to terminate the receiver input with a well known radiometric temperature. Then the radar integrates on the Dicke Switch for one sample duration and recalibrates the gains of all radar IF channels as well as of the passive direct detection channel. We recommend setting the zero calibration period to about 1000 seconds. If the passive channel (used for integrated liquid water measurements) is not needed in the measurement, the zero calibration period can be set to be much longer, for instance 3600 seconds or more.

Data Files: This box contains parameters related to data file storage. The radar produces data files of different levels (LV0 = raw data, LV1 = pre-processed data, refer to Appendix A for more details). The storage of these data levels can be controlled by associated checkboxes.
Also the file backup on the radar PC is optional. If file backup is selected, the radar can operate as a stand-alone unit without an external Host PC monitoring it. The radar PC also performs a simple file archiving, similar as described in section 3.3, but without file concatenation.
Another interesting feature is the spectral compression which often reduces the file size by up to an order of magnitude. Details about spectral compression are given in Appendix A2.1.
If the instrument is a dual polarization radar, there are a lot of additional spectral variables to be evaluated which are described in detail in section 2.11. These variables are useful in a variety of applications or scan patterns, but are less useful when observing vertically. Additional storage of these variables increased the LV0 file size by a factor of 2 to 3 and therefore, the user may decide himself, if they shall be included in the LV0 data file, or not.

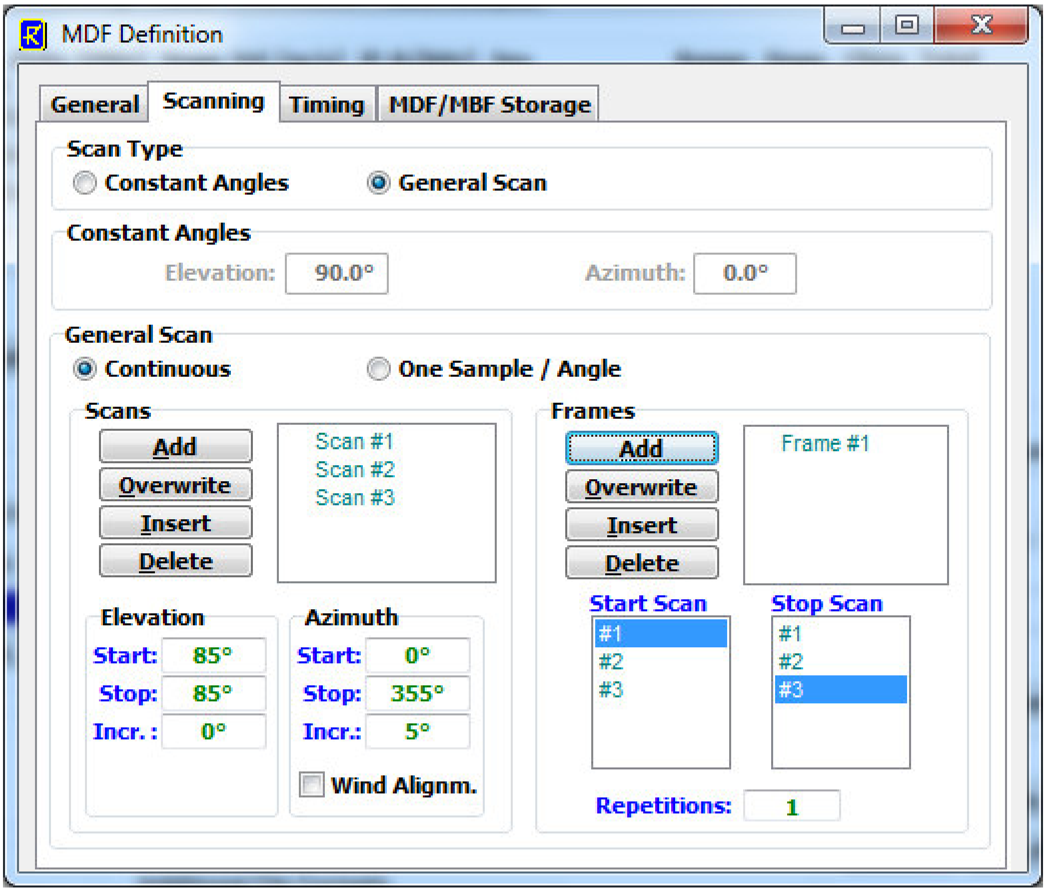


'General' tag of MDF definition window.

Measurement: In chapter 2 we explained the problem of aliasing in Doppler spectra, which is caused by insufficient chirp repetition rates (the 'Doppler dilemma'). The problem exists for pulsed radars as well. The radar software includes an algorithm to anti-alias the Doppler spectra but some users may want to apply their own methods. This is why anti-aliasing is optional and can be turned off, if desired.
Noise Filter: In spectral compression mode, the underground noise is removed from Doppler spectra and the noise floor is characterized by a standard deviation (STD). The noise filter factor is basically the number of STDs the radar used for deciding what is noise and what is valid data. Noise filter settings in the range of [3.0,…,5.0] are recommended for effective noise suppression and may depend on the chirp program in use. Please note that the higher the noise filter, the less sensitive the radar gets. **By setting the noise filter to 0.0, the noise cancelling is disabled and the radar transmits only raw spectra (no noise level removed, all spectra stored, no anti-aliasing performed).**
As mentioned before, the radar may automatically adjust the transmitter power if strong reflections (strong rain events) lead to a receiver saturation. This power levelling can be disabled by checking ***Suppress Power Levelling***.

The ***Scanning*** tag opens a setup menu summarizing all information required in scanning measurements:

Scan Type: Here the user selects between observations of constant elevation / azimuth angles and more complicated 'general' scans. Please note that if azimuth angles >0.0° or elevation angles ≠ 90° are selected, an elevation / azimuth positioner must be installed for moving the radar to the desired positions.



'Scanning' tag of MDF definition window.

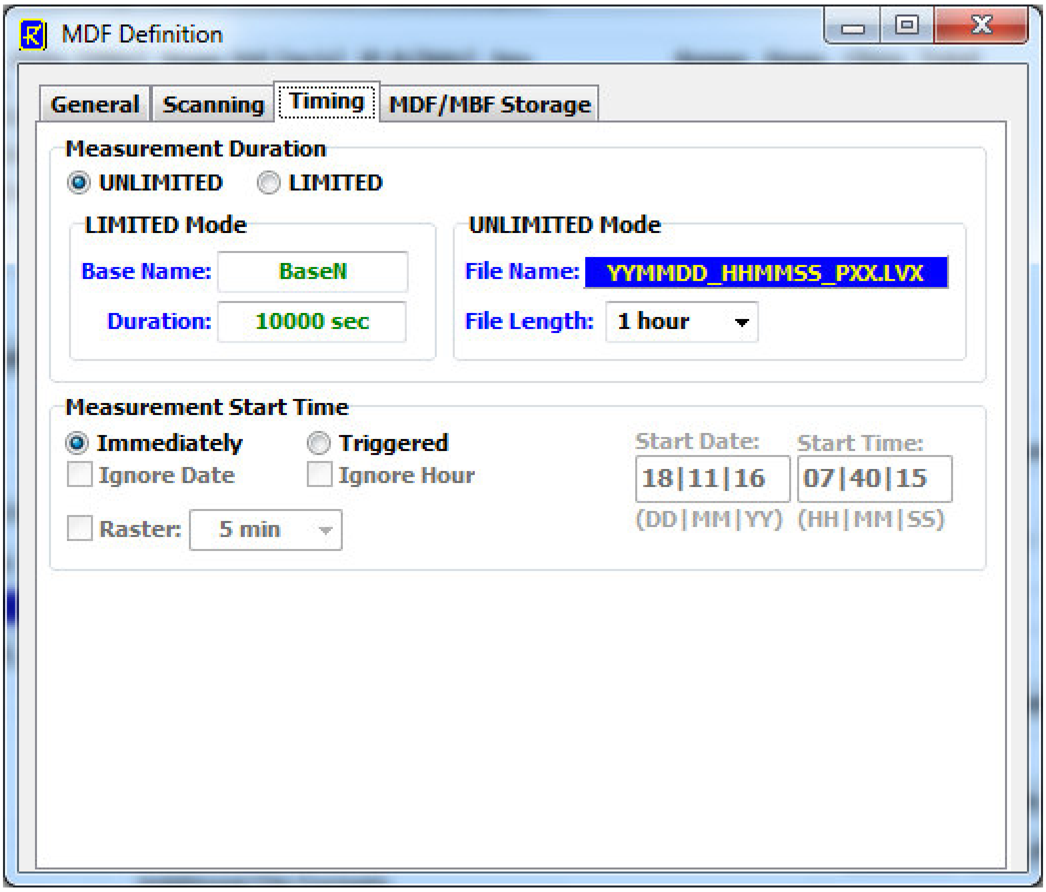
If ***General Scan*** is selected, arbitrary elevation and azimuth scan patters may be defined. When the optional elevation / azimuth scanner is not available, the radar ignores all angle definitions.
The radar movements are split into elementary scans from a start angle to a stop angle with a certain increment angle. These scans are numbered as Scan#1, Scan#2, ...

The radar does not execute single scans but only frames of scans. Each frame has a start scan and a stop scan (these can be identical) which form a 'loop' of scans that may be repeated arbitrarily. The concept of having two levels of movement definitions allows for the definition of complex scan procedures.
A frame is defined by selecting one of the scans in the start scan list and then clicking on one in the stop scan list. After entering the repetition number, the frame is added (or inserted) to the frame list (***Add*** or ***Insert***). It is possible to edit a frame definition using the ***Overwrite*** command or to delete it with ***Delete***. Three examples illustrate how a frame is executed:

1) Start: Scan#4, stop: Scan#6, repetitions: 3 ⇒
   Scan#4,Scan#5,Scan#6,Scan#4,Scan#5,Scan#6,Scan#4,Scan#5,Scan#6
2) Start: Scan#4, stop: Scan#2, repetitions: 2 ⇒
   Scan#4,Scan#3,Scan#2,Scan#4,Scan#3,Scan#2
3) Start: Scan#2, stop: Scan#2, repetitions: 1 ⇒
   Scan#2

The radar may perform sample measurements **continuously** while moving or in **One Sample / Angle** mode. When **continuous** scanning is selected, the increment elevation and azimuth angles of a scan definition are ignored, while in **One Sample / Angle** mode the radar does not measure while moving but only after reaching a discrete position in a scan. When ***Wind Alignment*** is checked within the azimuth scan definition, the positioner will ignore all azimuth settings but aligns the azimuth direction in parallel to the current wind direction.

The ***Timing*** tag opens a setup menu summarizing all information related to measurement start time, trigger modes and termination options:



'Timing' tag of MDF definition window.

If the measurement has a well-defined end time (automatic measurement termination, ***LIMITED*** mode) the radar needs a 'Base' filename in order to create an MDF specific filename. In ***UNLIMITED*** mode the radar creates file names automatically:

YYMMDD_HHmmSS_PXX_MCC.LV0 and

YYMMDD_HHMMSS_PXX_MCC.LV1

YY = Year, MM = Month, DD = Day, HH = Hour, mm = Minute, SS = Second, XX = chirp program number, MCC = measurement classification code (see Appendix A4). In ***LIMITED*** mode, the file names are slightly modified:

BaseN_YYMMDD_HHmmSS_PXX_MCC.LV0 and

BaseN_YYMMDD_HHMMSS_PXX_MCC.LV1

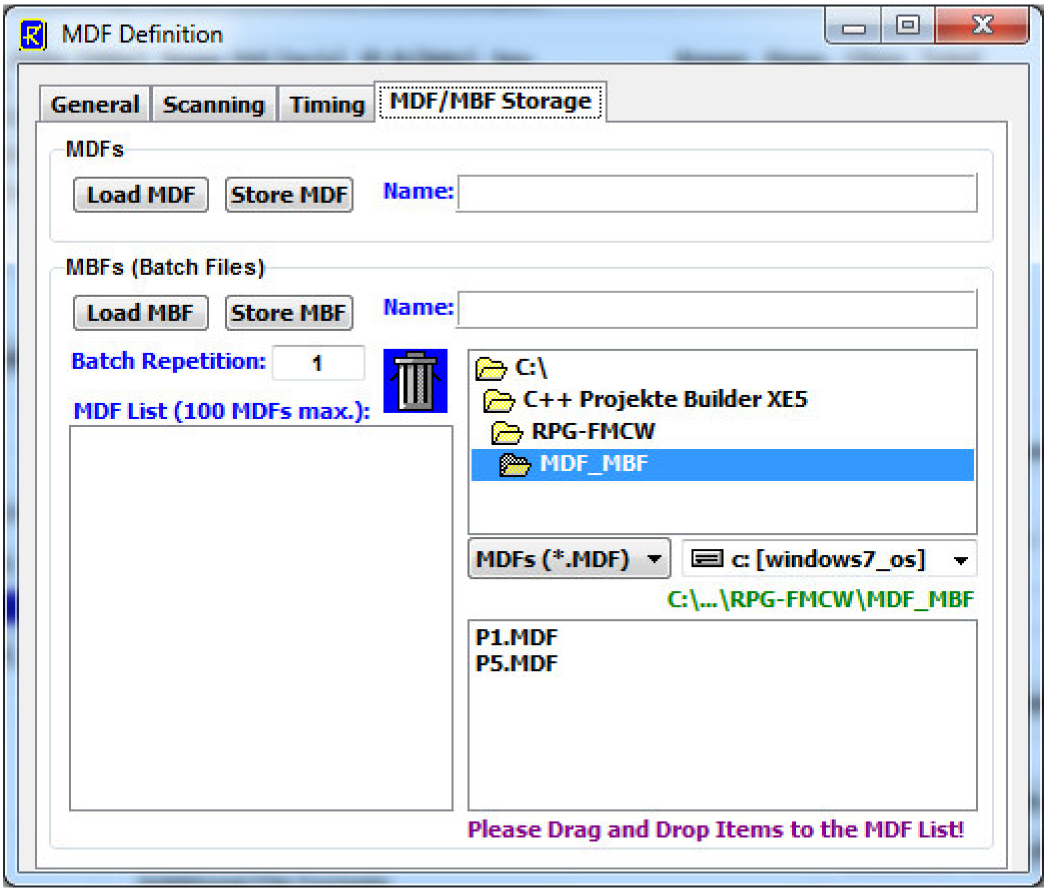
The different file naming convention is useful for archiving and automatic file concatenation of data. In batch operation, data files created by the same MDFs are automatically sorted for concatenation. This ensures that only data of the same kind (scan patterns or other) is linked together into the same daily file (if file concatenation is activated on the host PC, see section 3.3).

In ***LIMITED*** mode the measurement duration (starting from the trigger point) has to be specified, while in ***UNLIMITED*** mode the file length (in hours) can be selected. It is recommended to use a file length of one hour for unlimited measurements, because data files can reach an enormous size.

Start time and end time are important parameters for a measurement setup. There are two ways of triggering a measurement: Immediately after launching the MDF or at a certain time and date. Using a start time before the current time is equivalent to an immediate start. If the measurement start is triggered to a certain time, the check boxes ***Ignore Date*** and ***Ignore Hour*** allow for a date or hour independent triggering. This is particularly useful in a repeated multiple MDF batch measurement, where MDFs are repeated multiple times. A triggering to a certain date / time would trigger the MDF only once but not repeatedly. E.g. if ***Ignore Hour*** is checked (assuming ***Triggered*** mode is activated) and the 'Start Time' entry is set to 22|36|15, the measurement is triggered to 00:15:00, 01:15:00, 02:15:00, …, ignoring the current date and hour. If a more frequent trigger is required, one can use the ***Raster*** feature combined with a raster period. This mode assumes a start time of 00|00|00 and uses a raster period. For instance, if the period is 10 minutes and the current time is 10|17|32, the start trigger occurs at 10|20|00, 10|30|00, 10|40|00, etc.

The ***MDF / MBF Storage*** tag opens a setup menu for storing the MDF parameters to a certain filename, loading an existing MDF file or creating batch files (MDFs).
It is possible to send a *single* MDF **directly** to the radar. Multiple MDFs are packed into a MBF (measurement batch file). The concept is similar to the Scan/Frame relationship for scanning.
The MDFs in a batch file are executed sequentially in the order they are listed in the MDF list
The batch repetition number has the same meaning as the frame repetition factor for scanning: The MDF list forms a loop, which is repeated an arbitrary number of times. This offers the user a flexibility of combining different measurement tasks, which would otherwise not be compatible in a single MDF, for instance when combining different scan patterns (RHI, PPI, etc.) with zenith observations and multiple repetition cycles. The solution is to define different MDFs for each individual measurement task (e.g. a PPI scan) and combine them in a batch file with a certain repetition factor. The only restriction for MDF definitions in multi-MDF batches is that all MDFs in a batch list should be ***LIMITED*** mode MDFs.

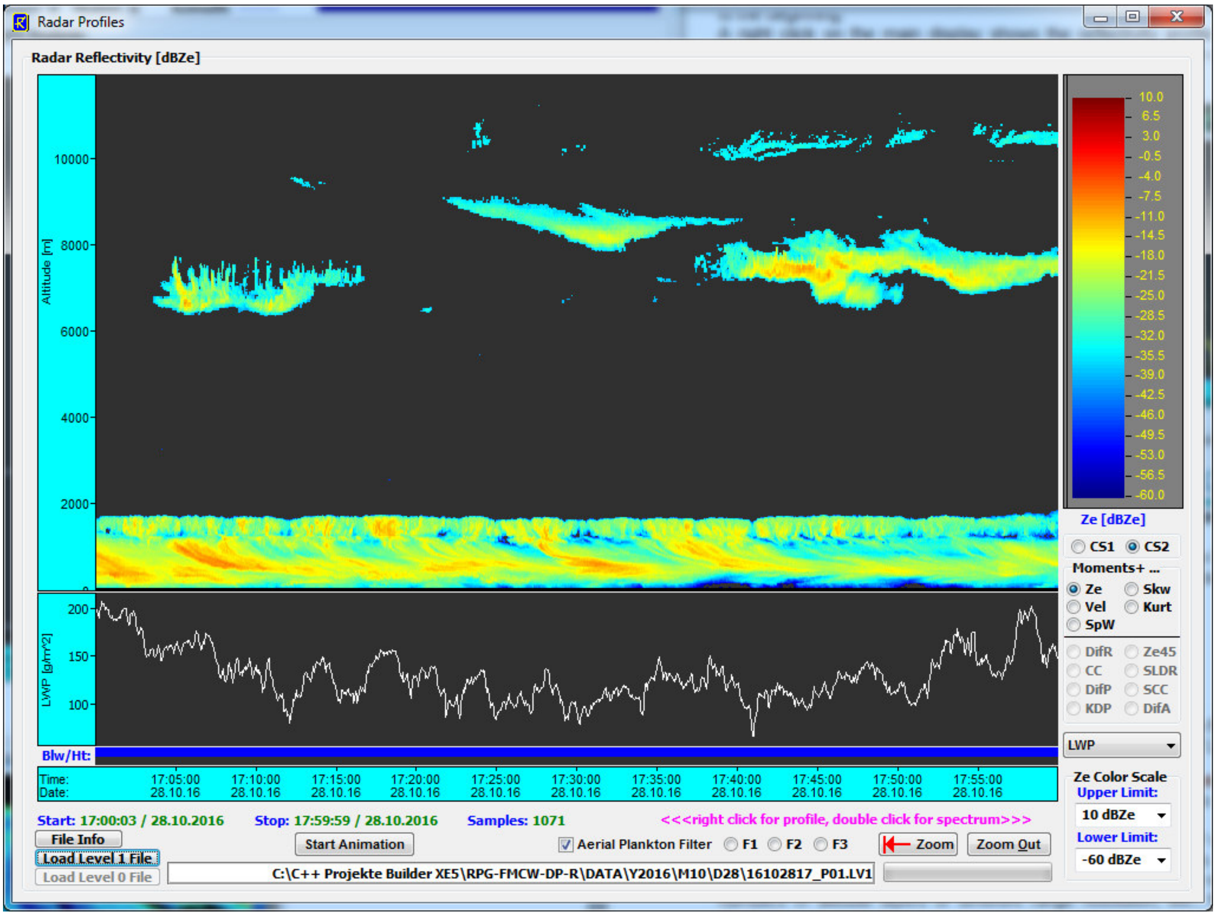
It is a good practice to store all MDFs in one directory (e.g. ***…\RPG-FMCW\MDF_MBF***). All MDFs in the selected directory are listed in the box in the lower right corner. From this list, the user may select each MDF he wants to add or insert to the MDF batch list by dragging the desired MDFs to the MDF batch list box. MDFs may also be deleted from the MDF batch list by dragging it to the waste bin. Store your measurement batch files (MBFs) in a single directory (like ***…\RPG-FMCW\MDF_MBF***).



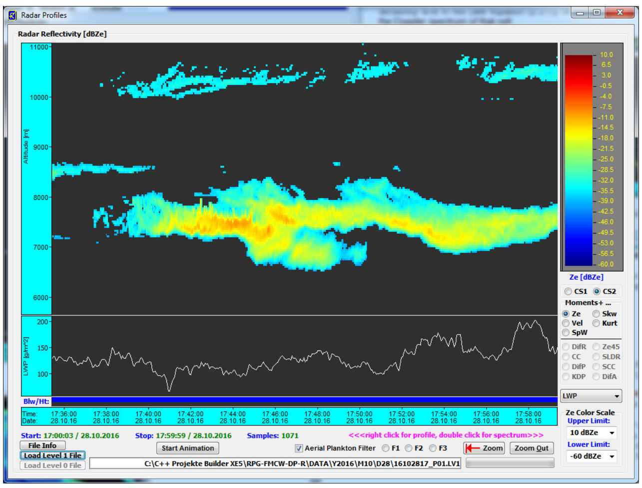
'MDF/MBF Storage' tag of MDF definition window.

3.14 Open Data Files

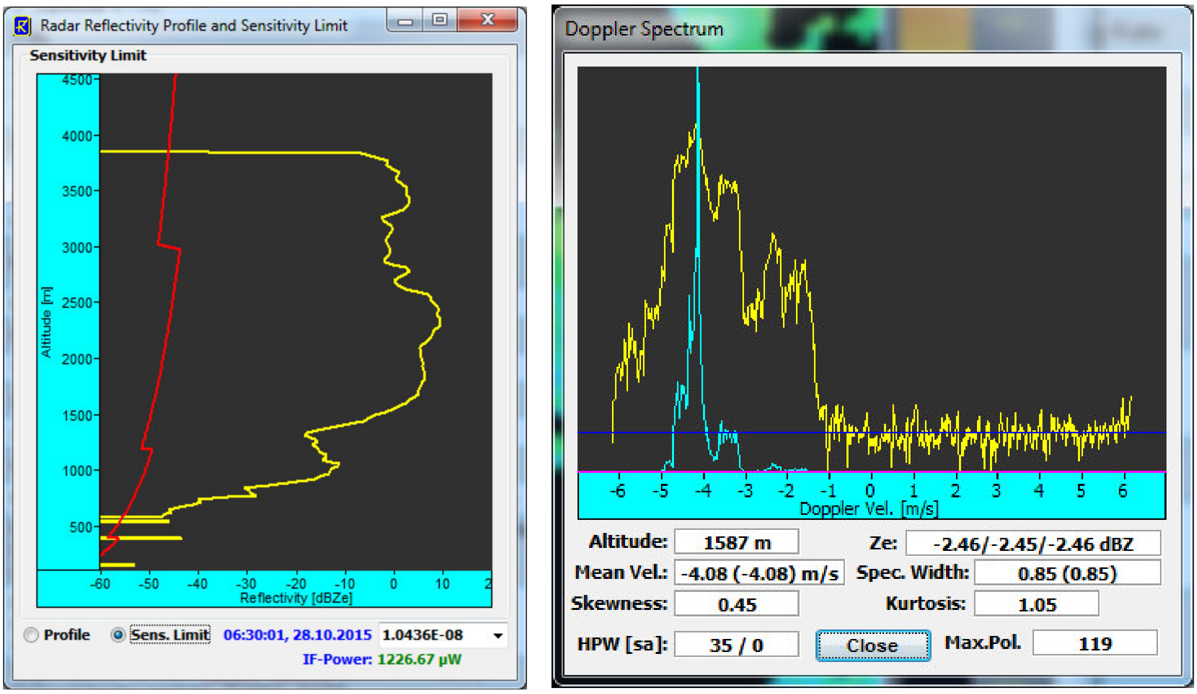
Existing data files can be loaded and inspected by clicking the  (Open Radar Data File) button. With ***Load Level 1 File*** a *.LV1 file is loaded:



Like during measurements, the main display may be switched between reflectivity ***Ze*** and mean velocity ***Vel***. But in addition the higher moments are also available: spectral width ***SpW***, skewness ***Skw*** and kurtosis ***Kurt***. The user may zoom into and out of the displayed data:



Clicking the button ***Zoom Out*** zooms back one step while the |← ***Zoom*** function zooms out to the beginning.
A right click on the main display shows the reflectivity profile in yellow together with the sensitivity limit in red (see equation (2.5.7)). A double click on a non-empty range cell plots the Doppler spectrum of that cell:

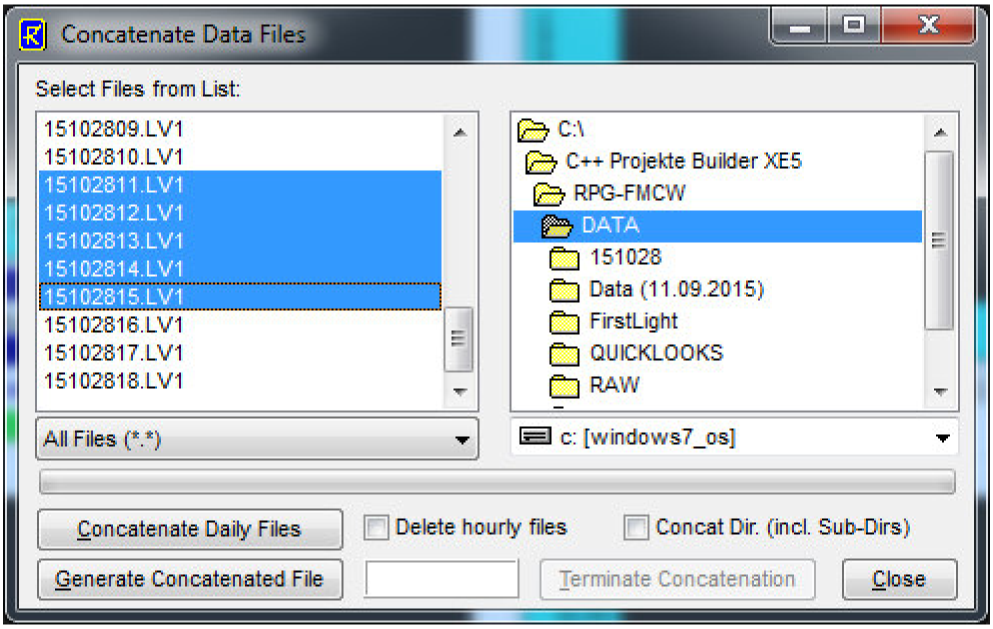


Vertical Ze-profile and Doppler spectrum

The blue curve in the Doppler spectrum display represents the linear Ze-spectrum while the yellow curve is the logarithmically scaled spectrum. The horizontal blue line marks the zero line of the noise floor (in log scale).
The structure of a level 1 data file is listed in appendix A.

3.15 Concatenating Data Files

In UNLIMITED mode the radar periodically generates new data filenames (e.g. every hour). It is often desirable to concatenate data files of the same type (*.LV1, etc.) to form bigger files (e.g. 24 hour files). This is possible by clicking  (***Concatenate Data Files***). A set of filenames is selected from the list and then concatenated to a single file with ***Generate Concatenated File***.
By using the ***Concatenate Daily Files*** command, the files of complete directories, with or without its sub-directories, may be concatenated to 24 hour files.
If the header information of two data files is not compatible, for example because of different numbers of altitude layers or different range resolution, etc., the concatenation process is aborted.

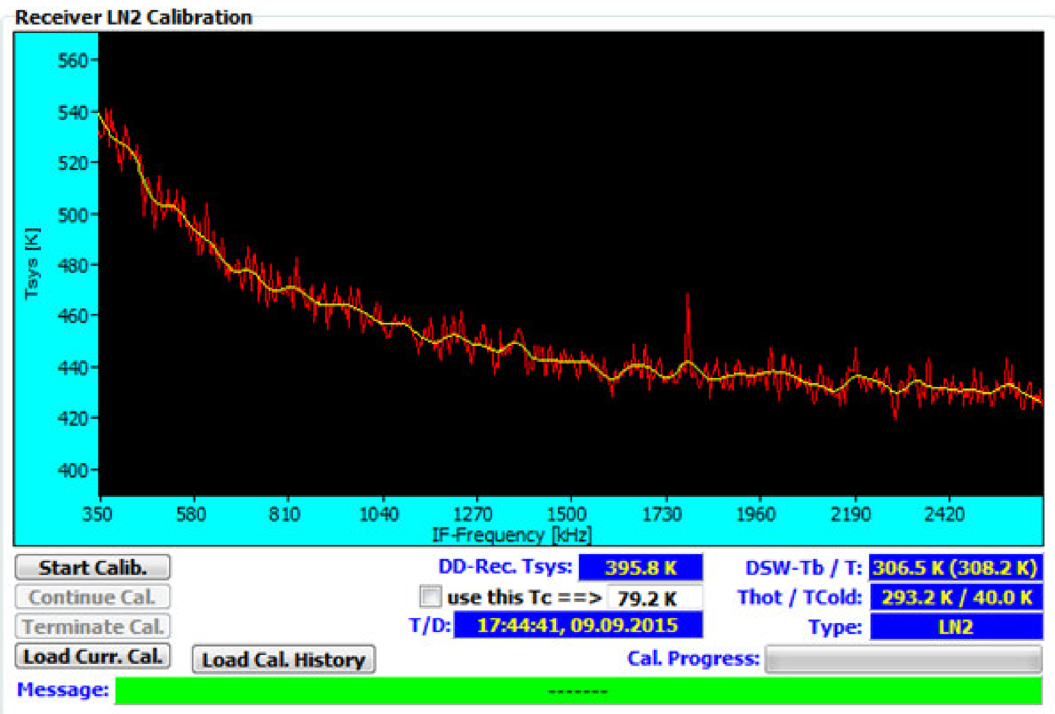


File concatenation menu.

The user may concatenate multiple files from a file list or perform the concatenation process in complete directories (all types in a row), with or without sub-directories. This way the post processing can be realized in a few mouse clicks for a complete data set, e.g. by selecting a 'year' directory (for instance Y2015).

3.16 Calibration Menus

The ***Radar Calibrations*** register page summarizes all radar calibrations described in chapters 2.5, 2.6 and 2.8.



Absolute receiver calibration box

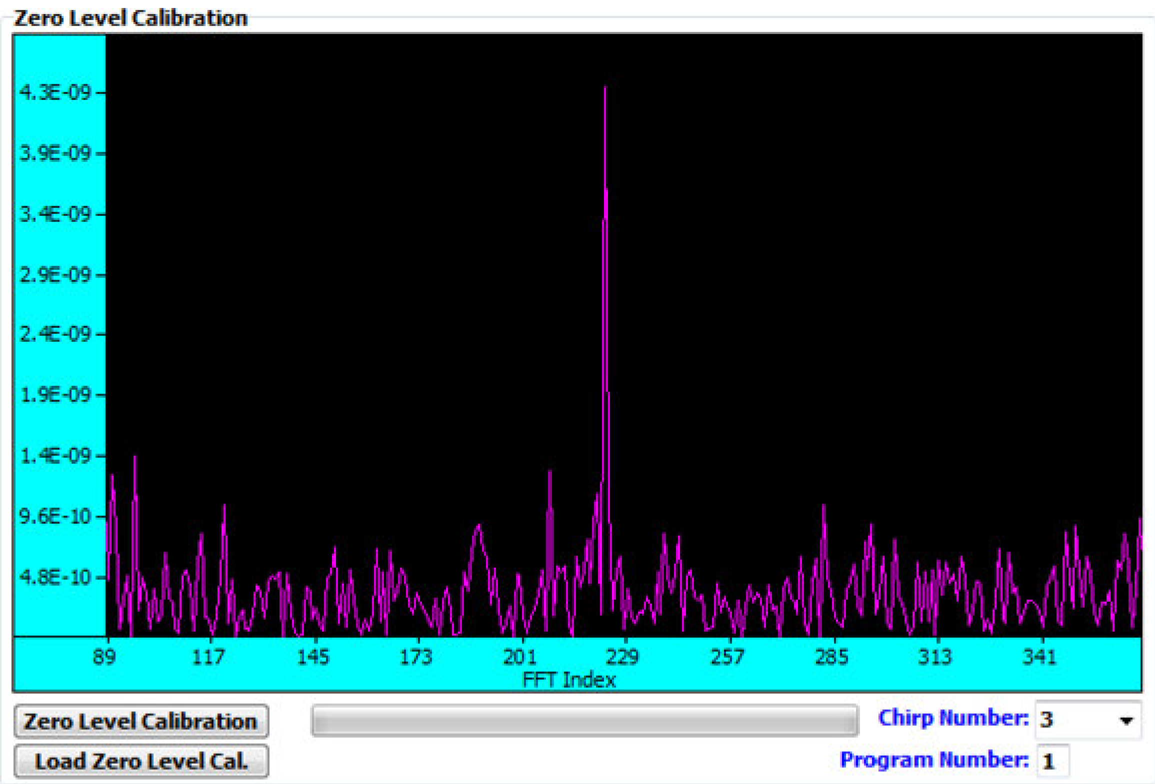
The receiver absolute calibration has been described in detail in section 2.5. It can be executed remotely from the host PC, if it is connected to the radar and no measurement is running. The radar is assuming an LN2 cooled cold calibration target as long as the ***use this Tc*** box is not checked. If checked, the temperature entry (in K) right to the ➔ arrow is used for the cold target temperature. This applies if the cloud free sky shall be scanned instead of the LN2 target, but it requires the knowledge of the sky temperature from a measurement of a different instrument, for instance a profiler like the RPG-HATPRO-G4.
The absolute calibration is started by clicking the ***Start Calib.*** button. The calibration status is continuously monitored in the message line. When a new target needs to be placed in front of the receiver, the calibration pauses and waits for the user to click ***Continue Cal.***.
The calibration is divided in three steps:

- Integration on ambient target (1 minute)
- Integration on cold target (1 minute)
- Integration on Dicke switch (1 minute)

After all steps have passed, the calibration is finished and its results are displayed graphically for each radar IF bin as well as for the passive 89 GHz channel.

Transmitter power sweeps, as described in section 2.6, are performed automatically during measurements (every hour) and the user may load the current sweep results. The initiation of a new sweep is only possible if no measurement is running on the radar.

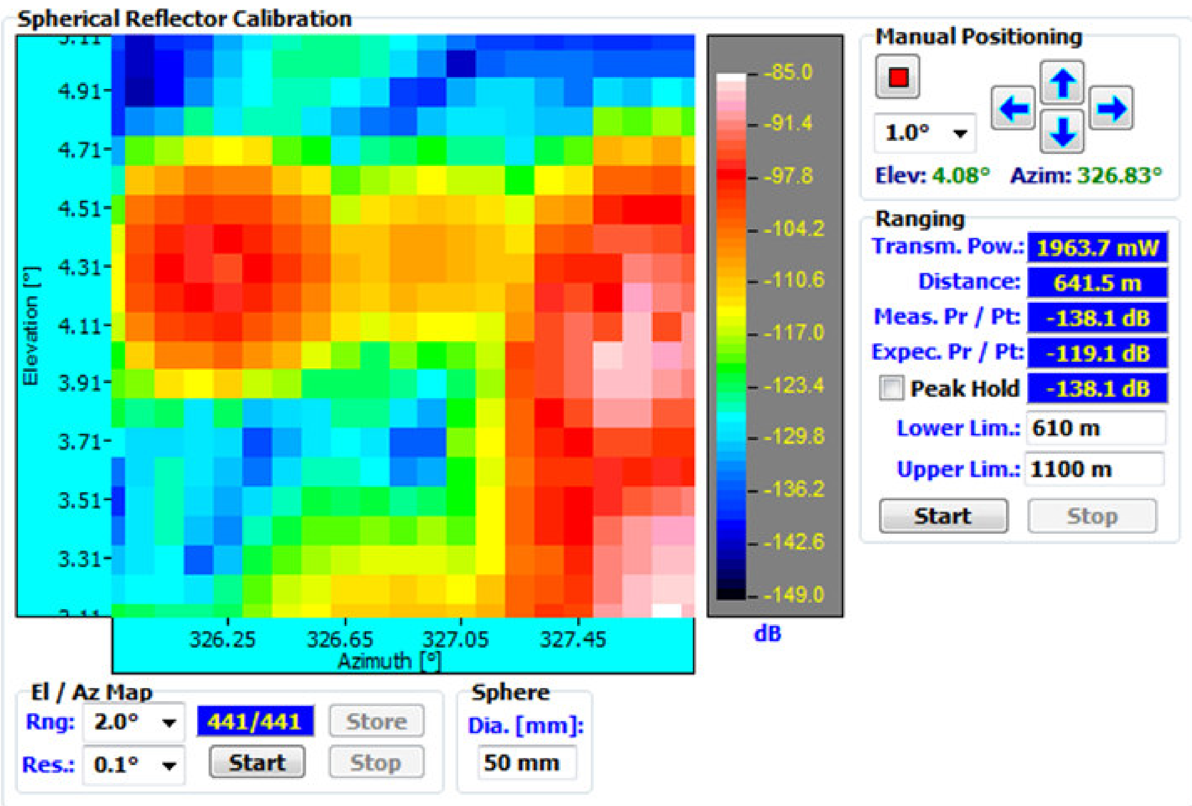
For each chirp program, a zero level integration is required. This measurement determines the intrinsic noise underground in each mapped IF bin. The information is used by the radar PC to subtract unwanted spectral features generated by the ADC board electronics.



ADC board spectrum without mixer contributions

Also this calibration is automatically executed before the first use of a program and takes about a few minutes. A long integration time is required to extract the spike spectrum from the receiver noise floor.

The calibration cross check measurement of section 2.8 is handled by the **Spectral Reflector Calibration** sub-menu commands. It requires the radar to be mounted on an El / Az scanner and a metal sphere to be launched with a helium balloon in a distance of approx.. 1 km from the radar. The procedure is performed at RPG facilities and is not required to be repeated later on by the user.

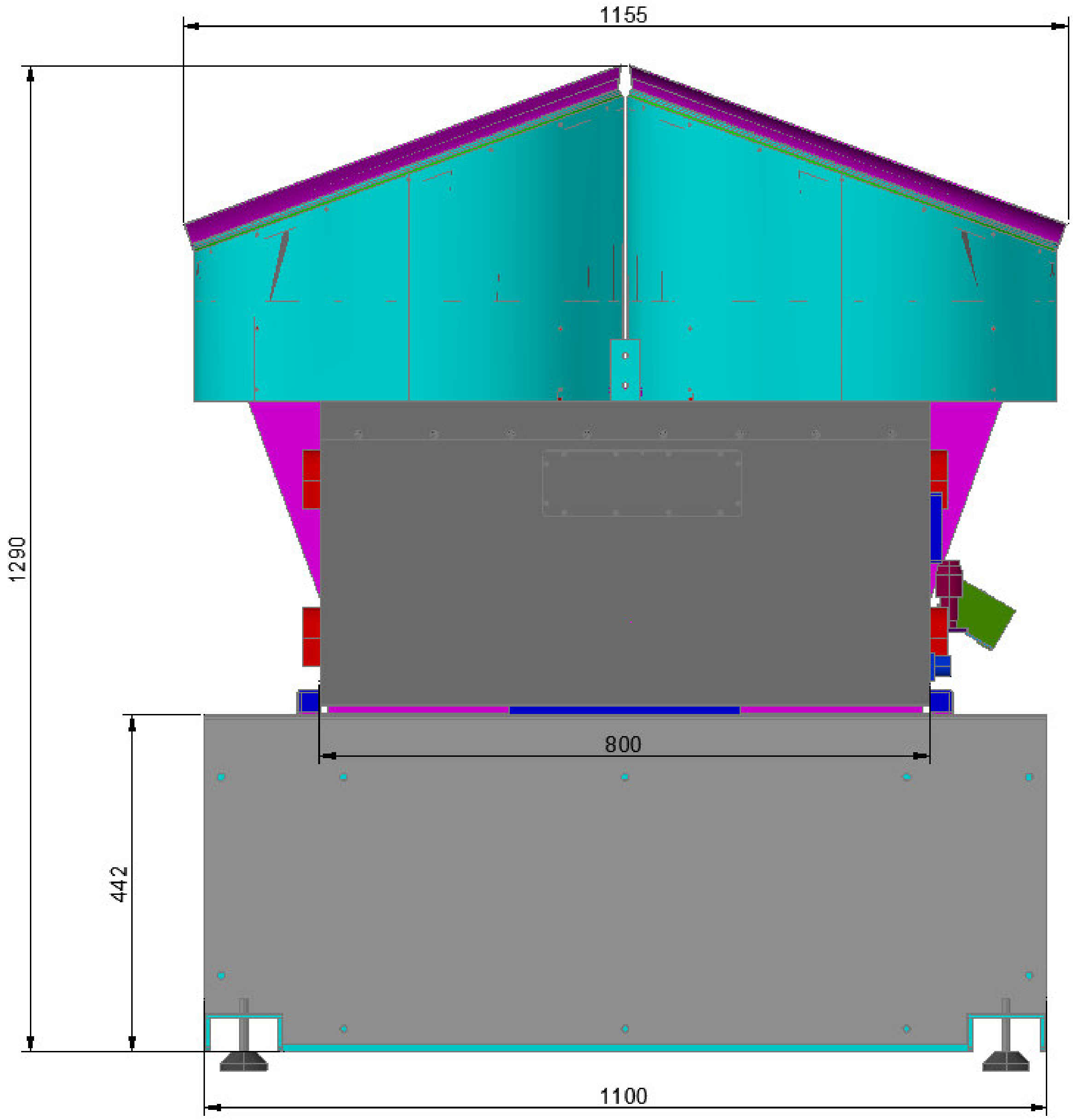


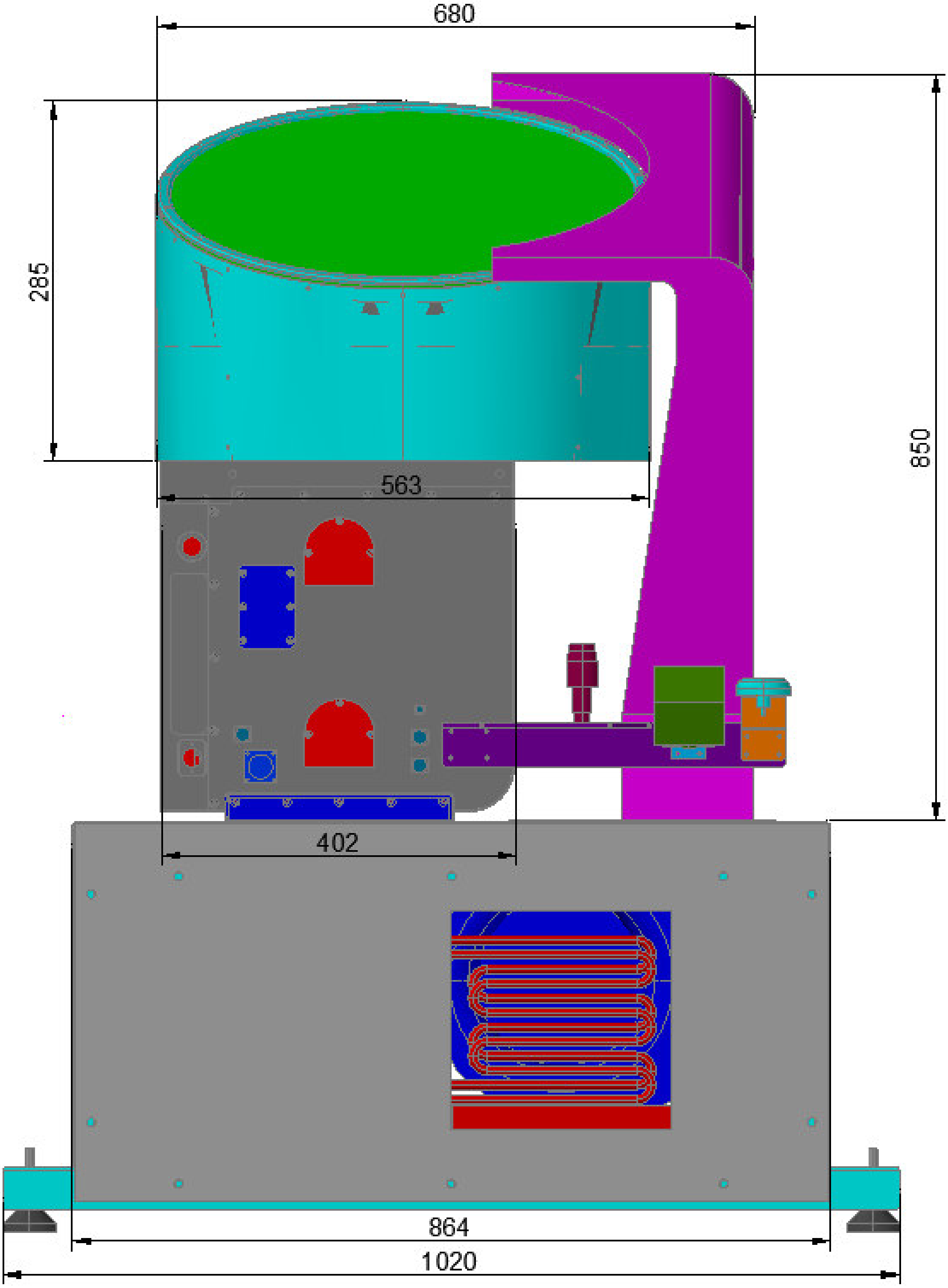
4 Instrument Specifications

Parameter	Specification
Frequency	94 GHz (λ=3.19 mm) ± 100 MHz typical, (adjustable by software between 92.3 and 95.7 GHz)
IF Range	350 kHz to 3 MHz
Transmitter Power	2 W typical (solid state amplifier)
Antenna Type	Bi-static Cassegrain with 500 mm aperture
Antenna Gain	51.6 dB
Beam Width	0.48° FWHM
Polarisation	V (optional V / H)
System Noise Figure	3 dB
Dynamic Range (Sensitivity)	-54 dBZe to +20 dBZe at 1 km height -45 dBZe to +20 dBZe at 3 km height -36 dBZe to +20 dBZe at 10 km height
Ranging	50 m to 18 km
Calibration	Transmitter power monitoring Receiver Dicke switch (for radar and DD channels) Hot / Cold absolute receiver calibration
Overall Radar Calibration Accuracy	Better than 0.4 dB
A/D Sampling Rate	8.2 MHz
Profile Sampling Rate	0.2 s to 30 s
Vertical Resolution	1 m to 100 m (user selectable)
Doppler Resolution	± 1.5 cm/s
Doppler Range	± 18 m/s max.
Chirp Variations	4 typical, 10 possible, re-programmable
Passive Channels	89 GHz for integral liquid water (LWP) detection (2 GHz BW)
Data Products	Reflectivity, Doppler-velocity, Spectral Width, Higher Moments Doppler Spectra LWC profiles
Data Formats	proprietary binary netCDF (conformity with CF convention) ASCII (only moment profiles)
Rain / Snow Mitigation System	Super blower for rec. / transm. radome (2000 m^3/h each) Optional heater modules (2 kW to 4 kW)
Weight	Radar main body: 100 kg Table: 130 kg

	Air conductors: 25 kg
Power Consumption	220 V AC, 50-60 Hz Radar: 400 W Blowers: 1000 W

5 Instrument Dimensions




680
285
850
563
402
864
1020

6 Safety Instructions

6.1 Operation Safety Issues

The RPG FMCW radars are huge and heavy instruments. The installation and operation of these systems requires the application of a few safety rules which are listed below together with their international signs:

1) The radar is made for outdoor use only. Operating the optional elevation / azimuth drive inside a building requires the permanent attention of trained personal. A remote operation inside a building is forbidden!

2) 

Turn off the radar power, while working on it.

3) 

The radar, when equipped with an optional elevation / azimuth drive, rotates 360° about **its** azimuth axis and +/- 90° about its elevation axis. The positioner applies strong forces to lift the more than 100 kg heavy equipment. During operation, keep a safety distance of at least 2 m from the radar.

4) 

While the azimuth drive is moving, keep away your hands from the instrument. Otherwise there is the risk of crushing hands, arms or legs. The strong forces of the positioner can easily brake bones!

5) 

For safety reasons, install a fence around the radar for warning people to enter the danger zone (a circle of 1 m radius circle around the centre of the radar).

6) 

When installing the radar, make sure the power cord is plugged into a power socket with proper grounding pin (PE = protection earth). Otherwise, the radar is electrically floating and the instrument may get more easily hit by lightning strokes. The user may also be exposed to high voltage strokes when touching the instrument, if the PE pin is not connected.

7) 

Use at least 4 people to carry the radar box and when installing it on the mounting table. The radar box weights about 100 kg.

8) 

Pay attention to the general safety guidelines while using liquid nitrogen for calibration.

Appendix A (Binary File Formats)

A1: LV1-Files (.LV1), Level 1 Data File (including Doppler moments and Doppler spectra), Version 1.0*

(this file structure was only used for the RPG-FMCW-94-SP prototype radar)

Variable Name	Type	# Bytes	Description
File Code	int	4	LV1-File ID (=789345), Version 1.0
Header starts here			
HeaderLen	int	4	header length in bytes (not including HeaderLen)
ModelNo	int	4	=0: 94 GHz single pol. =1: 94 GHz dual pol.
ProgName	char	Len(ProgName)+1	null terminated string of chirp program name
CustName	char	Len(CustName)+1	null terminated string of customer name
AltCount	int	4	number of altitude layers
Alts[ ]	float	4 x AltCount	altitude layer heights
SequCount	int	4	number of chirp sequences
RangeOffs[ ]	int	4 x SequCount	chirp sequences start index array in altitude layer array
dR[ ]	float	4 x SequCount	range resolution array for chirp sequences
DoppLen[ ]	int	4 x SequCount	number of samples in Doppler spectra of each chirp sequence
DoppRes[ ]	float	4 x SequCount	Doppler resolution [m/s] for each chirp sequence
DoppMax[ ]	float	4 x SequCount	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
CalInt	int	4	sample interval for automatic zero calibrations
AntSep	float	4	separation of both antenna axis (bistatic configuration), [m]
HPBW	float	4	cassegrain antenna HPBW [°]
SampDur	float	4	sample duration [sec]
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampLen_1	int	4	length of sample 1 [bytes], not including SampLen_1
SampTsec_1(1)	unsigned int	4	time of sample 1 [sec]
SampTms_1	int	4	milliseconds of sample 1 [msec]

RR_1	float	4	rain rate of sample 1 [mm/h]
RelHum_1	float	4	rel. humidity of sample 1 [%]
EnvTemp_1	float	4	environm. Temp. of sample 1 [K]
BaroP_1	float	4	barometric press. of sample 1 [hPa]
WS_1	float	4	wind speed of sample 1 [km/h]
WD_1	float	4	wind direction of sample 1 [°]
DDVolt_1	float	4	direct detection channel voltage of sample 1 [V]
DDTb_1	Float	4	direct detection brightness temp. of sample 1 [K]
LWP_1	Float	4	liquid water path of sample 1 [g/m^2]
PowIF_1	Float	4	IF power at ADC of sample 1 [µW]
Elev_1	Float	4	elevation angle of sample 1 [°]
Azi_1	Float	4	azimuth angle of sample 1 [°]
Status_1	Float	4	status flags of sample 1, 0/1: heater switch (ON/OFF) 0/10: blower switch (ON/OFF)
TransPow_1	Float	4	transmitter power of sample 1 [W]
TransT_1	Float	4	transmitter temp. of sample 1 [K]
RecT_1	Float	4	receiver temp. of sample 1 [K]
PCT_1	Float	4	PC temp. of sample 1 [K]
Res_1[ ]	Float	3 x 4	reserved, sample 1
RadC_1[ ](2)	Float	4	radar constant of sample 1
SeqStd_1[ ]	Float	4 x SequCount	standard dev. of noise power level in chirp sequences of sample 1
The loop over all range bins of sample 1 (loop index n) starts here			
ProfMsk_1[ ]	Char	AltCount (=AC)	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The following data is only stored if ProfMsk _1[n]=1			
Ze_1(n)	float	4	equiv. refl. of range bin n of sample 1, [dBZe]
MeVel_1(n)	float	4	mean velocity of range bin n of sample 1, [m/s]
SpW_1(n)	float	4	spectral width of range bin n of sample 1, [m/s]
Skew_1(n)	float	4	skewness of range bin n of sample 1
Kurt_1(n)	float	4	kurtosis of range bin n of sample 1
DoppSp_1(n)[ ]	float	4 x DoppLen	Doppler spectrum of bin n of sample 1
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
…	…	…	…
Sample TotSamp starts here			

…	…	…	…

(1) The time is expressed in number of seconds since 1.1.2001, 00:00:00
(2) RadC = 1/(50 Ohm G_{IF} P_t C), G_{IF} = IF gain, P_t = transmitter power [W], C = radar constant

A2: LV0-Files (.LV0), Level 0 Raw Data File, Version 2.0*

(this file structure is used by RPG-FMCW-94-SX radars since Nov. 2016)

Variable Name	Type	# Bytes	Description
File Code	**int**	**4**	**LV0-File ID (=789346), Version 2.0**
Header starts here			
HeaderLen	**int**	**4**	**header length in bytes (not including HeaderLen)**
ProgNo	**int**	**4**	**chirp program number in chirp table**
ModelNo	**int**	**4**	**=0: 94 GHz single pol. radar =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.**
ProgName	**char**	**Len(ProgName)+1**	**null terminated char string of chirp program name**
CustName	**char**	**Len(CustName)+1**	**null terminated char string of customer name**
Freq	**float**	**4**	**radar frequency [GHz]**
AntSep	**float**	**4**	**antenna diameter, [m]**
AntDia	**float**	**4**	**separation of both antenna axis (bistatic configuration), [m]**
AntG	**float**	**4**	**linear antenna gain**
HPBW	**float**	**4**	**antenna half power beam width [°]**
Cr	**float**	**4**	**radar constant, defined by equ. (2.1.5)**
DualPol	**char**	**1**	**=0: single pol. radar =1: dual pol. radar, LDR conf. =2: dual pol. radar, STSR mode (see section 2.11.3)**
CompEna	**char**	**1**	**spectral compression flag: 0: not compressed 1: spectra compressed 2: spectra compressed and spectral polarimetric variables are stored in the file**
AntiAlias	**char**	**1**	**0: Doppler spectra are not anti-aliased 1: Doppler spectra have been anti-aliased**
SampDur	**float**	**4**	**sample duration [sec]**
GPSLat	**float**	**4**	**GPS latitude**

GPSLong	float	4	GPS longitude
CalInt	int	4	period for automatic zero calibrations in number of samples
RAltN	int	4	number of radar ranging layers
TAltN	int	4	number of temperature profile layers
HAltN	int	4	number of humidity profile layers layers
SequN	int	4	number of chirp sequences
RAlts[ ]	float	4 x RAltN	ranging altitude layers
TAlts[ ]	float	4 x TAltN	temp. profile altitude layers (only if TAltN>0)
HAlts[ ]	float	4 x HAltN	hum. profile altitude layers (only if HAltN>0)
Fr[ ]	int	4 x RAltN	range factors (see equ. (2.5.6))
SpecN[ ]	int	4 x SequN	number of samples in Doppler spectra of each chirp sequence
RngOffs[ ]	int	4 x SequN	chirp sequence start index in altitude layer array
SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	chirp sequence range resolution [m]
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1[(1)]	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
RR_1	float	4	rain rate of sample 1 [mm/h]
RelHum_1	float	4	rel. humidity of sample 1 [%]
EnvTemp_1	float	4	environm. Temp. of sample 1 [K]
BaroP_1	float	4	barometric press. of sample 1 [hPa]
WS_1	float	4	wind speed of sample 1 [km/h]
WD_1	float	4	wind direction of sample 1 [°]
DDVolt_1	float	4	direct detection channel voltage

			of sample 1 [V]
DDTb_1	float	4	direct detection brightness temp. of sample 1 [K]
LWP_1	float	4	liquid water path of sample 1 [g/m^2]
PowIF_1	float	4	IF power at ADC of sample 1 [µW]
Elev_1	float	4	elevation angle of sample 1 [°]
Azi_1	float	4	aszimuth angle of sample 1 [°]
Status_1	float	4	mitigation status flags of sample 1, 0/1: heater switch (ON/OFF) 0/10: blower switch (ON/OFF)
TransPow_1	float	4	transmitter power of sample 1 [W]
TransT_1	float	4	transmitter temp. of sample 1 [K]
RecT_1	float	4	receiver temp. of sample 1 [K]
PCT_1	float	4	PC temp. of sample 1 [K]
Res_1[ ]	float	3 x 4	reserved, sample 1
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
PNv_1[ ]	float	RAltN x 4	total IF power in v-pol. measured at ADC input
PNh_1[ ]	float	RAltN x 4	total IF power in h-pol. measured at ADC input (only if DualPol>0)
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
SLh_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for horizontal polarisation (only if DualPol>0)
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
SpecBytes_1[n]	int	4	number of bytes of following spectral block
The following data is only stored if CompEna=0 (spectra contain noise floor)			
VSpec_1[n]	float	4 x SpecN	full Doppler spectrum (incl. noise), vertical pol., linear Ze
The following data is only stored if DualPol >0 (dual pol. radar)			
HSpec_1[n]	float	4 x SpecN	full Doppler spectrum (incl. noise), horizontal pol., linear Ze
ReVHSpec _1[n]	float	4 x SpecN	full covariance spectrum , real part (see equ. 2.11.6), linear Ze

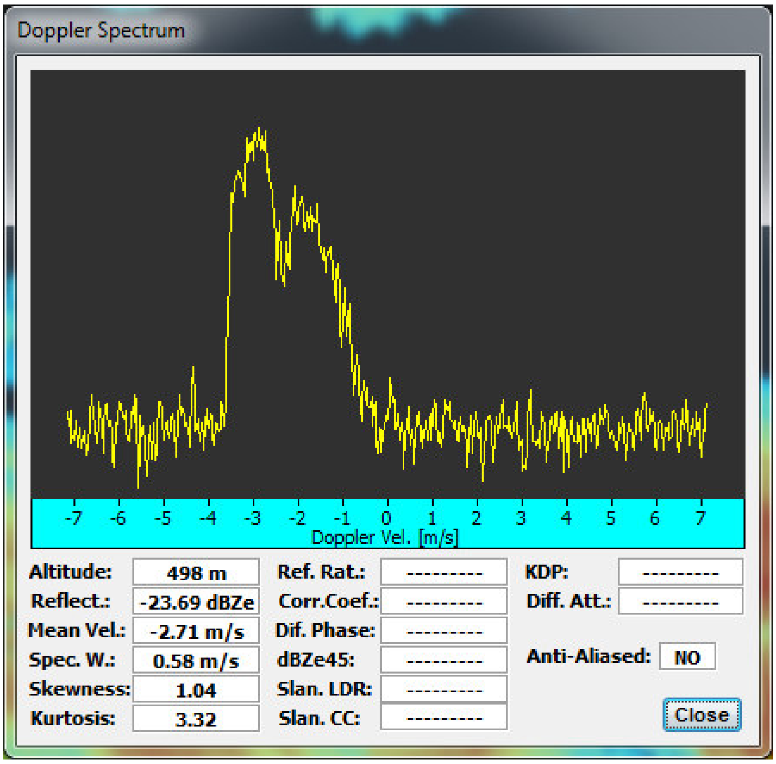
ImVHSpec _1[n]	float	4 x SpecN	full covariance spectrum , imaginary part (see equ. 2.11.6) linear Ze
The following data is only stored if CompEna>0 (spectral compression enabled)			
BlockN_1[n]	char	1	number of blocks in spectra
MinBkIdx_1[n]	short int	2 x BlockN_1[n]	minimum index of blocks in spectra
MaxBkIdx_1[n]	short int	2 x BlockN_1[n]	maximum index of blocks in spectra
VSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed Doppler spectrum, vertical pol., linear Ze
The following data is only stored if DualPol >0 (dual pol. radar)			
HSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed Doppler spectrum, horizontal pol., linear Ze
ReVHSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed covariance spectrum , real part (see equ. 2.11.6), linear Ze
ImVHSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed covariance spectrum , imaginary part (see equ. 2.11.6), linear Ze
The following data is only stored if CompEna=2 (include polar. spectral variables)			
RefRat_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed spectral differential reflectivity, see table 2.11.1 (product 1.), [dB]
CorrCoeff_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed spectral correlation coefficient, see table 2.11.1 (product 2.), [0,…,1]
DiffPh_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed spectral differential phase, see table 2.11.1 (product 3.), [rad]
The following data is only stored if DualPol =2 (dual pol. radar in STSR mode)			
SLDR_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed spectral slanted LDR, see table 2.11.1 (product 4.), [dB]
SCorrCoeff_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed spectral slanted correlation coefficient, see table 2.11.1 (product 5.), [0,…,1]
KDP_1[n]	float	4	specific differential phase shift, see equ. (2.10.13), [rad / km]
DiffAtt_1[n]	float	4	differential attenuation, see equ. (2.10.14), [dB / km]
The following data is only stored if CompEna>0 (spectral compression enabled)			
VNoisePow_1[n]	float	4	integrated Doppler spectrum noise power in v-pol., [Ze]
The following data is only stored if DualPol >0 (dual pol. radar)			
HNoisePow_1[n]	float	4	integrated Doppler spectrum

			noise power in h-pol., [Ze]
The following data is only stored if AntiAlias =1 and CompEna>0 (spectra are anti-aliased and compressed)			
AliasMsk_1[n]	char	1	mask indicating, if anti-aliasing has been applied (=1) or not (=0)
MinVel_1[n]	float	4	minimum velocity in Doppler spectrum [m/s]
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
…	…	…	…
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
…	…	…	…

(1) The time is expressed in number of seconds since 1.1.2001, 00:00:00

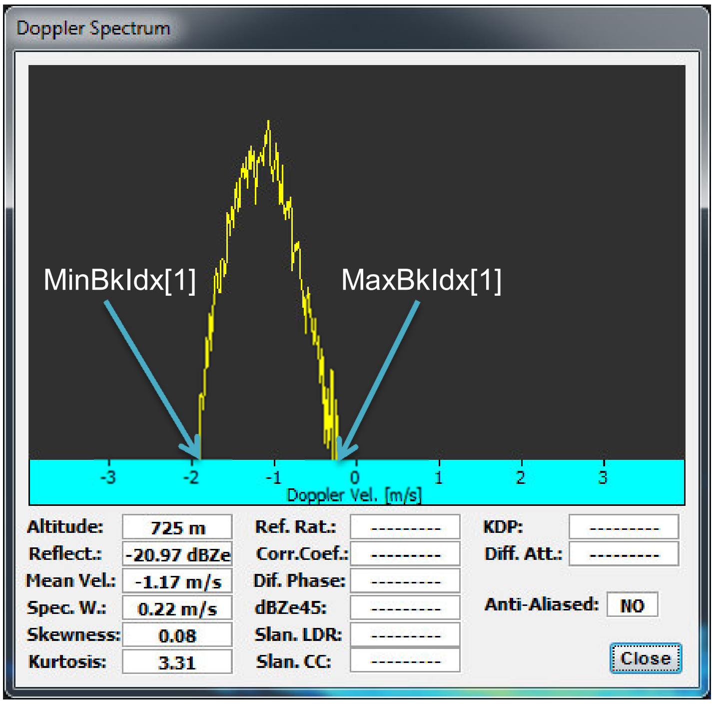
A2.1 Spectral Data Compression

In LV0 files the data compression of spectra can be enabled (CompEna>0). If CompEna=0, the spectral information is not noise stripped and not compressed (number of spectral samples = SpecN):



Uncompressed Doppler spectrum with unremoved noise level, shown in log scale

If the spectra are compressed (CompEna>0), the noise floor is removed and only the spectral information exceeding the noise underground is stored to the file. This reduces the file size extremely (by a factor of 5 – 10) and is also recommended, if the radar data has to be distributed over heavily loaded or slow networks. A typical compressed spectrum looks like this:



Compressed Doppler spectrum with removed noise underground, shown in log scale

The example spectrum above contains only a single block (BlockN =1). The block is specified by its minimum sample index (MinBkIdx[1]) and its maximum sample index (MaxBkIdx[1]) in the spectrum. With the spectral resolution

$$R_S = \frac{2\,MaxVel}{SpecN}$$

the velocity at an index k in the range [MinBkIdx[1],…, MaxBkIdx[1]] can be computed as:

$$V(k) = MinVel + k\,R_S$$

Of course a spectrum may have more than a single block (BlockN > 1). In this case MinBkIdx[] and MaxBkIdx[] are arrays of length BlockN and the minimum and maximum indices of block number n is given by MinBkIdx[n] and MaxBkIdx[n]. Each block contains exactly (MaxBkIdx[n] - MinBkIdx[n] + 1) spectral samples.

A3: LV1-Files (.LV1), Level 1 Data File (including Doppler moments), Version 2.0*

(this file structure is used by RPG-FMCW-94-SX radars since Nov. 2016)

Variable Name	Type	# Bytes	Description

File Code	**int**	**4**	**LV1-File ID (=789347), Version 2.0**
Header starts here			
HeaderLen	**int**	**4**	**header length in bytes (not including HeaderLen)**
ProgNo	**int**	**4**	**chirp program number in chirp table**
ModelNo	**int**	**4**	**=0: 94 GHz single pol. =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.**
ProgName	**char**	**Len(ProgName)+1**	**null terminated char string of chirp program name**
CustName	**char**	**Len(CustName)+1**	**null terminated char string of customer name**
Freq	**float**	**4**	**radar frequency [GHz]**
AntSep	**float**	**4**	**antenna diameter, [m]**
AntDia	**float**	**4**	**separation of both antenna axis (bistatic configuration), [m]**
AntG	**float**	**4**	**linear antenna gain**
HPBW	**float**	**4**	**antenna half power beam width [°]**
DualPol	**char**	**1**	**=0: single pol. radar =1: dual pol. radar, LDR conf. =2: dual pol. radar, STSR mode (see section 2.11.3)**
SampDur	**float**	**4**	**sample duration [sec]**
GPSLat	**float**	**4**	**GPS latitude**
GPSLong	**float**	**4**	**GPS longitude**
CalInt	**int**	**4**	**period for automatic zero calibrations in number of samples**
RAltN	**int**	**4**	**number of radar ranging layers**
TAltN	**int**	**4**	**number of temperature profile layers**
HAltN	**int**	**4**	**number of humidity profile layers layers**
SequN	**int**	**4**	**number of chirp sequences**
RAlts[ ]	**float**	**4 x RAltN**	**ranging altitude layers**
TAlts[ ]	**float**	**4 x TAltN**	**temp. profile altitude layers (only if TAltN>0)**
HAlts[ ]	**float**	**4 x HAltN**	**hum. profile altitude layers (only if HAltN>0)**
SpecN[ ]	**int**	**4 x SequN**	**number of samples in Doppler spectra of each chirp sequence**
RngOffs[ ]	**int**	**4 x SequN**	**chirp sequences start index array in altitude layer array**
SeqAvg[ ]	**int**	**4 x SequN**	**number of averaged chirps within a sequence**

SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	range resolution array for chirp sequences
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1(1)	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
RR_1	float	4	rain rate of sample 1 [mm/h]
RelHum_1	float	4	rel. humidity of sample 1 [%]
EnvTemp_1	float	4	environm. Temp. of sample 1 [K]
BaroP_1	float	4	barometric press. of sample 1 [hPa]
WS_1	float	4	wind speed of sample 1 [km/h]
WD_1	float	4	wind direction of sample 1 [°]
DDVolt_1	float	4	direct detection channel voltage of sample 1 [V]
DDTb_1	float	4	direct detection brightness temp. of sample 1 [K]
LWP_1	float	4	liquid water path of sample 1 [g/m^2]
PowIF_1	float	4	IF power at ADC of sample 1 [µW]
Elev_1	float	4	elevation angle of sample 1 [°]
Azi_1	float	4	aszimuth angle of sample 1 [°]
Status_1	float	4	mitigation status flags of sample 1, 0/1: heater switch (ON/OFF) 0/10: blower switch (ON/OFF)
TransPow_1	float	4	transmitter power of sample 1 [W]
TransT_1	float	4	transmitter temp. of sample 1 [K]
RecT_1	float	4	receiver temp. of sample 1 [K]
PCT_1	float	4	PC temp. of sample 1 [K]
Res_1[ ]	float	3 x 4	reserved, sample 1
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1

RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
SLh_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for horizontal polarisation (only if DualPol>0)
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
Ze_1[n]	float	4	linear reflectivity in Ze units for vert. pol. in range bin n of sample 1
MeanVel_1[n]	float	4	mean velocity [m/s] for vert. pol. in range bin n of sample 1
SpecWidth_1[n]	float	4	spectral width [m/s] for vert. pol. in range bin n of sample 1
Skewn_1[n]	float	4	spectral skewness for vert. pol. in range bin n of sample 1
Kurt_1[n]	float	4	spectral kurtosis for vert. pol. in range bin n of sample 1
The following data is only stored if DualPol >0 (dual pol. radar)			
RefRat_1[n]	float	4	Differential reflectivity in range bin n of sample 1, table 2.11.2 (product 1.), [dB]
CorrC_1[n]	float	4	Correlation coefficient in range bin n of sample 1, table 2.11.2 (product 2.), [0,…,1]
DiffPh_1[n]	float	4	differential phase in range bin n of sample 1, table 2.11.2 (product 3.), [rad]
The following data is only stored if DualPol =2 (dual pol. radar in STSR mode)			
Ze45_1[n]	float	4	slanted reflectivity in range bin n of sample 1, linear Ze
SLDR_1[n]	float	4	slanted LDR in range bin n of sample 1, table 2.11.2 (product 4.), [dB]
SCorrC_1[n]	float	4	slanted correlation coefficient in range bin n of sample 1, table 2.11.2 (product 5.), [0,…,1]
KDP_1[n]	float	4	specific differential phase shift in range bin n of sample 1, see equ. (2.10.13), [rad / km]
DiffAtt_1[n]	float	4	differential attenuation in range bin n of sample 1, see equ. (2.10.14), [dB / km]

The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
...	...	...	...
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
...	...	...	...

(1) The time is expressed in number of seconds since 1.1.2001, 00:00:00

A.4 File Format Examples

The file formats described in section A2 / A3 are complete and cover all possible settings of the variables:

1. **DualPol:** 0 = single polarisation radar
   1 = dual pol. radar with LDR mode configuration (see section 2.11.3)
   2 = dual pol. radar with STSR mode configuration (see section 2.11.3)
2. **CompEna:** 0 = spectra in file are not compressed (noise floor is not removed) and the spectra are not anti-aliased. Only the spectra of the coherency matrix are stored (see equ. (2.11.16)) but none of the higher products like spectral differential reflectivity, spectral correlation coefficient, spectral differential phase, spectral slanted LDR, spectral slanted correlation coefficient, differential phase and differential attenuation. This mode provides raw information to be further evaluated by the user.
   1 = spectra in file are compressed. Only the spectra of the coherency matrix are stored (see equ. (2.11.16)) but none of the higher products like spectral differential reflectivity, spectral correlation coefficient, spectral differential phase, spectral slanted LDR, spectral slanted correlation coefficient, differential phase and differential attenuation.
   2 = spectra in file are compressed and all spectral polarimetric variables are stored (full data evaluation mode).
3. **AntiAlias:** 0 = spectra are not anti-aliased.
   1 = spectra have been anti-aliased (only if CompEna > 0).

However, the file structure can be quite complex so that the file format tables in A2 / A3 become difficult to read. Therefore, the following examples provide individual file formats for special cases so that the user may select one of them representing his specific settings.

A.4.1 Example 1: DualPol = 0, CompEna = 0, AntiAlias = 0

LV0-File

Variable Name	Type	# Bytes	Description
File Code	int	4	LV0-File ID (=789346), Version 2.0
Header starts here			
HeaderLen	int	4	header length in bytes (not including HeaderLen)

ProgNo	**int**	**4**	**chirp program number in chirp table**
ModelNo	**int**	**4**	**=0: 94 GHz single pol. radar =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.**
ProgName	**char**	**Len(ProgName)+1**	**null terminated char string of chirp program name**
CustName	**char**	**Len(CustName)+1**	**null terminated char string of customer name**
Freq	**float**	**4**	**radar frequency [GHz]**
AntSep	**float**	**4**	**antenna diameter, [m]**
AntDia	**float**	**4**	**separation of both antenna axis (bistatic configuration), [m]**
AntG	**float**	**4**	**linear antenna gain**
HPBW	**float**	**4**	**antenna half power beam width [°]**
Cr	**float**	**4**	**radar constant, defined by equ. (2.1.5)**
DualPol	**char**	**1**	**=0: single pol. radar**
CompEna	**char**	**1**	**0: not compressed**
AntiAlias	**char**	**1**	**0: Doppler spectra are not anti-aliased**
SampDur	**float**	**4**	**sample duration [sec]**
GPSLat	**float**	**4**	**GPS latitude**
GPSLong	**float**	**4**	**GPS longitude**
CalInt	**int**	**4**	**period for automatic zero calibrations in number of samples**
RAltN	**int**	**4**	**number of radar ranging layers**
TAltN	**int**	**4**	**number of temperature profile layers**
HAltN	**int**	**4**	**number of humidity profile layers layers**
SequN	**int**	**4**	**number of chirp sequences**
RAlts[ ]	**float**	**4 x RAltN**	**ranging altitude layers**
TAlts[ ]	**float**	**4 x TAltN**	**temp. profile altitude layers (only if TAltN>0)**
HAlts[ ]	**float**	**4 x HAltN**	**hum. profile altitude layers (only if HAltN>0)**
Fr[ ]	**int**	**4 x RAltN**	**range factors (see equ. (2.5.6))**
SpecN[ ]	**int**	**4 x SequN**	**number of samples in Doppler spectra of each chirp sequence**
RngOffs[ ]	**int**	**4 x SequN**	**chirp sequence start index in altitude layer array**
SeqAvg[ ]	**int**	**4 x SequN**	**number of averaged chirps within a sequence**
SeqIntTime[ ]	**float**	**4 x SequN**	**effective sequence integration time [sec]**
dR[ ]	**float**	**4 x SequN**	**chirp sequence range resolution**

			[m]
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1[(1)]	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
RR_1	float	4	rain rate of sample 1 [mm/h]
RelHum_1	float	4	rel. humidity of sample 1 [%]
EnvTemp_1	float	4	environm. Temp. of sample 1 [K]
BaroP_1	float	4	barometric press. of sample 1 [hPa]
WS_1	float	4	wind speed of sample 1 [km/h]
WD_1	float	4	wind direction of sample 1 [°]
DDVolt_1	float	4	direct detection channel voltage of sample 1 [V]
DDTb_1	float	4	direct detection brightness temp. of sample 1 [K]
LWP_1	float	4	liquid water path of sample 1 [g/m^2]
PowIF_1	float	4	IF power at ADC of sample 1 [µW]
Elev_1	float	4	elevation angle of sample 1 [°]
Azi_1	float	4	aszimuth angle of sample 1 [°]
Status_1	float	4	mitigation status flags of sample 1, 0/1: heater switch (ON/OFF) 0/10: blower switch (ON/OFF)
TransPow_1	float	4	transmitter power of sample 1 [W]
TransT_1	float	4	transmitter temp. of sample 1 [K]
RecT_1	float	4	receiver temp. of sample 1 [K]
PCT_1	float	4	PC temp. of sample 1 [K]
Res_1[ ]	float	3 x 4	reserved, sample 1
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
PNv_1[ ]	float	RAltN x 4	total IF power in v-pol. measured at ADC input

SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
SpecBytes_1[n]	int	4	number of bytes of following spectral block
VSpec_1[n]	float	4 x SpecN	full Doppler spectrum (incl. noise), vertical pol., linear Ze
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
…	…	…	…
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
…	…	…	…

LV1 File (<u>note that the variables of the previous LV0 table marked in light blue are summarized in the following tables as Mon[20], the housekeeping monitoring array</u>)

Variable Name	Type	# Bytes	Description
File Code	int	4	LV1-File ID (=789347), Version 2.0
Header starts here			
HeaderLen	int	4	header length in bytes (not including HeaderLen)
ProgNo	int	4	chirp program number in chirp table
ModelNo	int	4	=0: 94 GHz single pol. =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.
ProgName	char	Len(ProgName)+1	null terminated char string of chirp program name
CustName	char	Len(CustName)+1	null terminated char string of customer name
Freq	float	4	radar frequency [GHz]
AntSep	float	4	antenna diameter, [m]
AntDia	float	4	separation of both antenna axis (bistatic configuration), [m]
AntG	float	4	linear antenna gain
HPBW	float	4	antenna half power beam width [°]
DualPol	char	1	=0: single pol. radar
SampDur	float	4	sample duration [sec]

GPSLat	float	4	GPS latitude
GPSLong	float	4	GPS longitude
CalInt	int	4	period for automatic zero calibrations in number of samples
RAltN	int	4	number of radar ranging layers
TAltN	int	4	number of temperature profile layers
HAltN	int	4	number of humidity profile layers layers
SequN	int	4	number of chirp sequences
RAlts[ ]	float	4 x RAltN	ranging altitude layers
TAlts[ ]	float	4 x TAltN	temp. profile altitude layers (only if TAltN>0)
HAlts[ ]	float	4 x HAltN	hum. profile altitude layers (only if HAltN>0)
SpecN[ ]	int	4 x SequN	number of samples in Doppler spectra of each chirp sequence
RngOffs[ ]	int	4 x SequN	chirp sequences start index array in altitude layer array
SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	range resolution array for chirp sequences
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1(1)	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
Mon[ ]	float	20 x 4	housekeeping monitoring array (see A.4.1 LV0 format)
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1

			0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
Ze_1[n]	float	4	linear reflectivity in Ze units for vert. pol. in range bin n of sample 1
MeanVel_1[n]	float	4	mean velocity [m/s] for vert. pol. in range bin n of sample 1
SpecWidth_1[n]	float	4	spectral width [m/s] for vert. pol. in range bin n of sample 1
Skewn_1[n]	float	4	spectral skewness for vert. pol. in range bin n of sample 1
Kurt_1[n]	float	4	spectral kurtosis for vert. pol. in range bin n of sample 1
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
...	...	...	...
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
...	...	...	...

A.4.2 Example 2: DualPol = 0, CompEna =1, AntiAlias = 0

LV0-File

Variable Name	Type	# Bytes	Description
File Code	int	4	LV0-File ID (=789346), Version 2.0
Header starts here			
HeaderLen	int	4	header length in bytes (not including HeaderLen)
ProgNo	int	4	chirp program number in chirp table
ModelNo	int	4	=0: 94 GHz single pol. radar =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.
ProgName	char	Len(ProgName)+1	null terminated char string of chirp program name
CustName	char	Len(CustName)+1	null terminated char string of customer name
Freq	float	4	radar frequency [GHz]
AntSep	float	4	antenna diameter, [m]
AntDia	float	4	separation of both antenna axis

			(bistatic configuration), [m]
AntG	float	4	linear antenna gain
HPBW	float	4	antenna half power beam width [°]
Cr	float	4	radar constant, defined by equ. (2.1.5)
DualPol	char	1	=0: single pol. radar
CompEna	char	1	1: spectra compressed
AntiAlias	char	1	0: Doppler spectra are not anti-aliased
SampDur	float	4	sample duration [sec]
GPSLat	float	4	GPS latitude
GPSLong	float	4	GPS longitude
CalInt	int	4	period for automatic zero calibrations in number of samples
RAltN	int	4	number of radar ranging layers
TAltN	int	4	number of temperature profile layers
HAltN	int	4	number of humidity profile layers layers
SequN	int	4	number of chirp sequences
RAlts[ ]	float	4 x RAltN	ranging altitude layers
TAlts[ ]	float	4 x TAltN	temp. profile altitude layers (only if TAltN>0)
HAlts[ ]	float	4 x HAltN	hum. profile altitude layers (only if HAltN>0)
Fr[ ]	int	4 x RAltN	range factors (see equ. (2.5.6))
SpecN[ ]	int	4 x SequN	number of samples in Doppler spectra of each chirp sequence
RngOffs[ ]	int	4 x SequN	chirp sequence start index in altitude layer array
SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	chirp sequence range resolution [m]
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1(1)	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]

QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
Mon[ ]	float	20 x 4	housekeeping monitoring array (see A.4.1 LV0 format)
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
PNv_1[ ]	float	RAltN x 4	total IF power in v-pol. measured at ADC input
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
SpecBytes_1[n]	int	4	number of bytes of following spectral block
BlockN_1[n]	char	1	number of blocks in spectra
MinBkIdx_1[n]	short int	2 x BlockN_1[n]	minimum index of blocks in spectra
MaxBkIdx_1[n]	short int	2 x BlockN_1[n]	maximum index of blocks in spectra
VSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed Doppler spectrum, vertical pol., linear Ze
VNoisePow_1[n]	float	4	integrated Doppler spectrum noise power in v-pol., [Ze]
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
...	...	...	...
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
...	...	...	...

LV1

Variable Name	Type	# Bytes	Description
File Code	int	4	LV1-File ID (=789347), Version 2.0
Header starts here			
HeaderLen	int	4	header length in bytes (not

			including HeaderLen)
ProgNo	int	4	chirp program number in chirp table
ModelNo	int	4	=0: 94 GHz single pol. =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.
ProgName	char	Len(ProgName)+1	null terminated char string of chirp program name
CustName	char	Len(CustName)+1	null terminated char string of customer name
Freq	float	4	radar frequency [GHz]
AntSep	float	4	antenna diameter, [m]
AntDia	float	4	separation of both antenna axis (bistatic configuration), [m]
AntG	float	4	linear antenna gain
HPBW	float	4	antenna half power beam width [°]
DualPol	char	1	=0: single pol. radar
SampDur	float	4	sample duration [sec]
GPSLat	float	4	GPS latitude
GPSLong	float	4	GPS longitude
CalInt	int	4	period for automatic zero calibrations in number of samples
RAltN	int	4	number of radar ranging layers
TAltN	int	4	number of temperature profile layers
HAltN	int	4	number of humidity profile layers layers
SequN	int	4	number of chirp sequences
RAlts[ ]	float	4 x RAltN	ranging altitude layers
TAlts[ ]	float	4 x TAltN	temp. profile altitude layers (only if TAltN>0)
HAlts[ ]	float	4 x HAltN	hum. profile altitude layers (only if HAltN>0)
SpecN[ ]	int	4 x SequN	number of samples in Doppler spectra of each chirp sequence
RngOffs[ ]	int	4 x SequN	chirp sequences start index array in altitude layer array
SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	range resolution array for chirp sequences
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			

TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1 (1)	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
Mon[ ]	float	20 x 4	housekeeping monitoring array (see A.4.1 LV0 format)
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
Ze_1[n]	float	4	linear reflectivity in Ze units for vert. pol. in range bin n of sample 1
MeanVel_1[n]	float	4	mean velocity [m/s] for vert. pol. in range bin n of sample 1
SpecWidth_1[n]	float	4	spectral width [m/s] for vert. pol. in range bin n of sample 1
Skewn_1[n]	float	4	spectral skewness for vert. pol. in range bin n of sample 1
Kurt_1[n]	float	4	spectral kurtosis for vert. pol. in range bin n of sample 1
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
...	...	...	...
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
...	...	...	...

A.4.3 Example 3: DualPol = 0, CompEna =1, AntiAlias = 1

LV0-File

Variable Name	Type	# Bytes	Description
File Code	int	4	LV0-File ID (=789346), Version 2.0
Header starts here			
HeaderLen	int	4	header length in bytes (not including HeaderLen)
ProgNo	int	4	chirp program number in chirp table
ModelNo	int	4	=0: 94 GHz single pol. radar =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.
ProgName	char	Len(ProgName)+1	null terminated char string of chirp program name
CustName	char	Len(CustName)+1	null terminated char string of customer name
Freq	float	4	radar frequency [GHz]
AntSep	float	4	antenna diameter, [m]
AntDia	float	4	separation of both antenna axis (bistatic configuration), [m]
AntG	float	4	linear antenna gain
HPBW	float	4	antenna half power beam width [°]
Cr	float	4	radar constant, defined by equ. (2.1.5)
DualPol	char	1	=0: single pol. radar
CompEna	char	1	1: spectra compressed
AntiAlias	char	1	1: Doppler spectra have been anti-aliased
SampDur	float	4	sample duration [sec]
GPSLat	float	4	GPS latitude
GPSLong	float	4	GPS longitude
CalInt	int	4	period for automatic zero calibrations in number of samples
RAltN	int	4	number of radar ranging layers
TAltN	int	4	number of temperature profile layers
HAltN	int	4	number of humidity profile layers layers
SequN	int	4	number of chirp sequences
RAlts[ ]	float	4 x RAltN	ranging altitude layers
TAlts[ ]	float	4 x TAltN	temp. profile altitude layers (only if TAltN>0)
HAlts[ ]	float	4 x HAltN	hum. profile altitude layers (only if HAltN>0)
Fr[ ]	int	4 x RAltN	range factors (see equ. (2.5.6))
SpecN[ ]	int	4 x SequN	number of samples in Doppler spectra of each chirp sequence

RngOffs[ ]	int	4 x SequN	chirp sequence start index in altitude layer array
SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	chirp sequence range resolution [m]
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1(1)	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
Mon[ ]	float	20 x 4	housekeeping monitoring array (see A.4.1 LV0 format)
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
PNv_1[ ]	float	RAltN x 4	total IF power in v-pol. measured at ADC input
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
SpecBytes_1[n]	int	4	number of bytes of following spectral block
BlockN_1[n]	char	1	number of blocks in spectra
MinBkIdx_1[n]	short int	2 x BlockN_1[n]	minimum index of blocks in spectra
MaxBkIdx_1[n]	short int	2 x BlockN_1[n]	maximum index of blocks in spectra
VSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed Doppler spectrum, vertical pol., linear Ze
VNoisePow_1[n]	float	4	integrated Doppler spectrum

			noise power in v-pol., [Ze]
AliasMsk_1[n]	char	1	mask indicating, if anti-aliasing has been applied (=1) or not (=0)
MinVel_1[n]	float	4	minimum velocity in Doppler spectrum [m/s]
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
...	...	...	...
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
...	...	...	...

LV1

Variable Name	Type	# Bytes	Description
File Code	int	4	LV1-File ID (=789347), Version 2.0
Header starts here			
HeaderLen	int	4	header length in bytes (not including HeaderLen)
ProgNo	int	4	chirp program number in chirp table
ModelNO	int	4	=0: 94 GHz single pol. =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.
ProgName	char	Len(ProgName)+1	null terminated char string of chirp program name
CustName	char	Len(CustName)+1	null terminated char string of customer name
Freq	float	4	radar frequency [GHz]
AntSep	float	4	antenna diameter, [m]
AntDia	float	4	separation of both antenna axis (bistatic configuration), [m]
AntG	float	4	linear antenna gain
HPBW	float	4	antenna half power beam width [°]
DualPol	char	1	=0: single pol. radar
SampDur	float	4	sample duration [sec]
GPSLat	float	4	GPS latitude
GPSLong	float	4	GPS longitude
CalInt	int	4	period for automatic zero calibrations in number of samples
RAltN	int	4	number of radar ranging layers
TAltN	int	4	number of temperature profile

			layers
HAltN	int	4	number of humidity profile layers layers
SequN	int	4	number of chirp sequences
RAlts[ ]	float	4 x RAltN	ranging altitude layers
TAlts[ ]	float	4 x TAltN	temp. profile altitude layers (only if TAltN>0)
HAlts[ ]	float	4 x HAltN	hum. profile altitude layers (only if HAltN>0)
SpecN[ ]	int	4 x SequN	number of samples in Doppler spectra of each chirp sequence
RngOffs[ ]	int	4 x SequN	chirp sequences start index array in altitude layer array
SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	range resolution array for chirp sequences
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1 (1)	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
Mon[ ]	float	20 x 4	housekeeping monitoring array (see A.4.1 LV0 format)
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
Ze_1[n]	float	4	linear reflectivity in Ze units for vert. pol. in range bin n of sample 1

MeanVel_1[n]	float	4	mean velocity [m/s] for vert. pol. in range bin n of sample 1
SpecWidth_1[n]	float	4	spectral width [m/s] for vert. pol. in range bin n of sample 1
Skewn_1[n]	float	4	spectral skewness for vert. pol. in range bin n of sample 1
Kurt_1[n]	float	4	spectral kurtosis for vert. pol. in range bin n of sample 1
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
…	…	…	…
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
…	…	…	…

A.4.4 Example 4: DualPol = 1 / 2, CompEna = 0, AntiAlias = 0

LV0-File

Variable Name	Type	# Bytes	Description
File Code	int	4	LV0-File ID (=789346), Version 2.0
Header starts here			
HeaderLen	int	4	header length in bytes (not including HeaderLen)
ProgNo	int	4	chirp program number in chirp table
ModelNo	int	4	=0: 94 GHz single pol. radar =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.
ProgName	char	Len(ProgName)+1	null terminated char string of chirp program name
CustName	char	Len(CustName)+1	null terminated char string of customer name
Freq	float	4	radar frequency [GHz]
AntSep	float	4	antenna diameter, [m]
AntDia	float	4	separation of both antenna axis (bistatic configuration), [m]
AntG	float	4	linear antenna gain
HPBW	float	4	antenna half power beam width [°]
Cr	float	4	radar constant, defined by equ. (2.1.5)
DualPol	char	1	=1: dual pol. radar, LDR conf.

			=2: dual pol. radar, STSR mode (see section 2.11.3)
CompEna	char	1	0: not compressed
AntiAlias	char	1	0: Doppler spectra are not anti-aliased
SampDur	float	4	sample duration [sec]
GPSLat	float	4	GPS latitude
GPSLong	float	4	GPS longitude
CalInt	int	4	period for automatic zero calibrations in number of samples
RAltN	int	4	number of radar ranging layers
TAltN	int	4	number of temperature profile layers
HAltN	int	4	number of humidity profile layers layers
SequN	int	4	number of chirp sequences
RAlts[ ]	float	4 x RAltN	ranging altitude layers
TAlts[ ]	float	4 x TAltN	temp. profile altitude layers (only if TAltN>0)
HAlts[ ]	float	4 x HAltN	hum. profile altitude layers (only if HAltN>0)
Fr[ ]	int	4 x RAltN	range factors (see equ. (2.5.6))
SpecN[ ]	int	4 x SequN	number of samples in Doppler spectra of each chirp sequence
RngOffs[ ]	int	4 x SequN	chirp sequence start index in altitude layer array
SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	chirp sequence range resolution [m]
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1[(1)]	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
Mon[ ]	float	20 x 4	housekeeping monitoring array (see A.4.1 LV0 format)

TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
PNv_1[ ]	float	RAltN x 4	total IF power in v-pol. measured at ADC input
PNh_1[ ]	float	RAltN x 4	total IF power in h-pol. measured at ADC input
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
SLh_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for horizontal pol.
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
SpecBytes_1[n]	int	4	number of bytes of following spectral block
VSpec_1[n]	float	4 x SpecN	full Doppler spectrum (incl. noise), vertical pol., linear Ze
HSpec_1[n]	float	4 x SpecN	full Doppler spectrum (incl. noise), horizontal pol., linear Ze
ReVHSpec _1[n]	float	4 x SpecN	full covariance spectrum , real part (see equ. 2.11.6), linear Ze
ImVHSpec _1[n]	float	4 x SpecN	full covariance spectrum , imaginary part (see equ. 2.11.6) linear Ze
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
…	…	…	…
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
…	…	…	…

LV1-File

Variable Name	Type	# Bytes	Description
File Code	int	4	LV1-File ID (=789347), Version 2.0
Header starts here			
HeaderLen	int	4	header length in bytes (not including HeaderLen)
ProgNo	int	4	chirp program number in chirp table
ModelNo	int	4	=0: 94 GHz single pol. =1: 94 GHz dual pol. STSR

			config. =2: 94 GHz dual pol. LDR config.
ProgName	char	Len(ProgName)+1	null terminated char string of chirp program name
CustName	char	Len(CustName)+1	null terminated char string of customer name
Freq	float	4	radar frequency [GHz]
AntSep	float	4	antenna diameter, [m]
AntDia	float	4	separation of both antenna axis (bistatic configuration), [m]
AntG	float	4	linear antenna gain
HPBW	float	4	antenna half power beam width [°]
DualPol	char	1	=1: dual pol. radar, LDR conf. =2: dual pol. radar, STSR mode (see section 2.11.3)
SampDur	float	4	sample duration [sec]
GPSLat	float	4	GPS latitude
GPSLong	float	4	GPS longitude
CalInt	int	4	period for automatic zero calibrations in number of samples
RAltN	int	4	number of radar ranging layers
TAltN	int	4	number of temperature profile layers
HAltN	int	4	number of humidity profile layers layers
SequN	int	4	number of chirp sequences
RAlts[ ]	float	4 x RAltN	ranging altitude layers
TAlts[ ]	float	4 x TAltN	temp. profile altitude layers (only if TAltN>0)
HAlts[ ]	float	4 x HAltN	hum. profile altitude layers (only if HAltN>0)
SpecN[ ]	int	4 x SequN	number of samples in Doppler spectra of each chirp sequence
RngOffs[ ]	int	4 x SequN	chirp sequences start index array in altitude layer array
SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	range resolution array for chirp sequences
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not

			including SampBytes _1
Time_1[(1)]	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
Mon[ ]	float	20 x 4	housekeeping monitoring array (see A.4.1 LV0 format)
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
SLh_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for horizontal polarisation (only if DualPol>0)
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
Ze_1[n]	float	4	linear reflectivity in Ze units for vert. pol. in range bin n of sample 1
MeanVel_1[n]	float	4	mean velocity [m/s] for vert. pol. in range bin n of sample 1
SpecWidth_1[n]	float	4	spectral width [m/s] for vert. pol. in range bin n of sample 1
Skewn_1[n]	float	4	spectral skewness for vert. pol. in range bin n of sample 1
Kurt_1[n]	float	4	spectral kurtosis for vert. pol. in range bin n of sample 1
RefRat_1[n]	float	4	Differential reflectivity in range bin n of sample 1, table 2.11.2 (product 1.), [dB]
CorrC_1[n]	float	4	Correlation coefficient in range bin n of sample 1, table 2.11.2 (product 2.), [0,…,1]
DiffPh_1[n]	float	4	differential phase in range bin n of sample 1, table 2.11.2 (product 3.), [rad]
The following data is only stored if DualPol =2 (dual pol. radar in STSR mode)			
Ze45_1[n]	float	4	slanted reflectivity in range bin n of sample 1, linear Ze
SLDR_1[n]	float	4	slanted LDR in range bin n of sample 1, table 2.11.2 (product 4.), [dB]

SCorrC_1[n]	float	4	slanted correlation coefficient in range bin n of sample 1, table 2.11.2 (product 5.), [0,…,1]
KDP_1[n]	float	4	specific differential phase shift in range bin n of sample 1, see equ. (2.10.13), [rad / km]
DiffAtt_1[n]	float	4	differential attenuation in range bin n of sample 1, see equ. (2.10.14), [dB / km]
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
…	…	…	…
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
…	…	…	…

A.4.5 Example 5: DualPol = 1 / 2, CompEna = 1, AntiAlias = 0 / 1

LV0-File

Variable Name	Type	# Bytes	Description
File Code	int	4	LV0-File ID (=789346), Version 2.0
Header starts here			
HeaderLen	int	4	header length in bytes (not including HeaderLen)
ProgNo	int	4	chirp program number in chirp table
ModelNo	int	4	=0: 94 GHz single pol. radar =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.
ProgName	char	Len(ProgName)+1	null terminated char string of chirp program name
CustName	char	Len(CustName)+1	null terminated char string of customer name
Freq	float	4	radar frequency [GHz]
AntSep	float	4	antenna diameter, [m]
AntDia	float	4	separation of both antenna axis (bistatic configuration), [m]
AntG	float	4	linear antenna gain
HPBW	float	4	antenna half power beam width [°]
Cr	float	4	radar constant, defined by equ. (2.1.5)

DualPol	char	1	=1: dual pol. radar, LDR conf. =2: dual pol. radar, STSR mode (see section 2.11.3)
CompEna	char	1	1: spectra compressed
AntiAlias	char	1	0: Doppler spectra are not anti-aliased 1: Doppler spectra have been anti-aliased
SampDur	float	4	sample duration [sec]
GPSLat	float	4	GPS latitude
GPSLong	float	4	GPS longitude
CalInt	int	4	period for automatic zero calibrations in number of samples
RAltN	int	4	number of radar ranging layers
TAltN	int	4	number of temperature profile layers
HAltN	int	4	number of humidity profile layers layers
SequN	int	4	number of chirp sequences
RAlts[ ]	float	4 x RAltN	ranging altitude layers
TAlts[ ]	float	4 x TAltN	temp. profile altitude layers (only if TAltN>0)
HAlts[ ]	float	4 x HAltN	hum. profile altitude layers (only if HAltN>0)
Fr[ ]	int	4 x RAltN	range factors (see equ. (2.5.6))
SpecN[ ]	int	4 x SequN	number of samples in Doppler spectra of each chirp sequence
RngOffs[ ]	int	4 x SequN	chirp sequence start index in altitude layer array
SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	chirp sequence range resolution [m]
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1(1)	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high

			Bit 3: no transm. power leveling
Mon[ ]	float	20 x 4	housekeeping monitoring array (see A.4.1 LV0 format)
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
PNv_1[ ]	float	RAltN x 4	total IF power in v-pol. measured at ADC input
PNh_1[ ]	float	RAltN x 4	total IF power in h-pol. measured at ADC input
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
SLh_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for horizontal polarisation
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
SpecBytes_1[n]	int	4	number of bytes of following spectral block
BlockN_1[n]	char	1	number of blocks in spectra
MinBkIdx_1[n]	short int	2 x BlockN_1[n]	minimum index of blocks in spectra
MaxBkIdx_1[n]	short int	2 x BlockN_1[n]	maximum index of blocks in spectra
VSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed Doppler spectrum, vertical pol., linear Ze
HSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed Doppler spectrum, horizontal pol., linear Ze
ReVHSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed covariance spectrum , real part (see equ. 2.11.6), linear Ze
ImVHSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed covariance spectrum , imaginary part (see equ. 2.11.6), linear Ze
VNoisePow_1[n]	float	4	integrated Doppler spectrum noise power in v-pol., [Ze]
HNoisePow_1[n]	float	4	integrated Doppler spectrum noise power in h-pol., [Ze]
The following data is only stored if AntiAlias =1			
AliasMsk_1[n]	char	1	mask indicating, if anti-aliasing has been applied (=1) or not (=0)
MinVel_1[n]	float	4	minimum velocity in Doppler spectrum [m/s]

The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
...	...	...	...
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
...	...	...	...

LV1-File

Variable Name	Type	# Bytes	Description
File Code	int	4	LV1-File ID (=789347), Version 2.0
Header starts here			
HeaderLen	int	4	header length in bytes (not including HeaderLen)
ProgNo	int	4	chirp program number in chirp table
ModelNo	int	4	=0: 94 GHz single pol. =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.
ProgName	char	Len(ProgName)+1	null terminated char string of chirp program name
CustName	char	Len(CustName)+1	null terminated char string of customer name
Freq	float	4	radar frequency [GHz]
AntSep	float	4	antenna diameter, [m]
AntDia	float	4	separation of both antenna axis (bistatic configuration), [m]
AntG	float	4	linear antenna gain
HPBW	float	4	antenna half power beam width [°]
DualPol	char	1	=1: dual pol. radar, LDR conf. =2: dual pol. radar, STSR mode (see section 2.11.3)
SampDur	float	4	sample duration [sec]
GPSLat	float	4	GPS latitude
GPSLong	float	4	GPS longitude
CalInt	int	4	period for automatic zero calibrations in number of samples
RAltN	int	4	number of radar ranging layers
TAltN	int	4	number of temperature profile layers
HAltN	int	4	number of humidity profile layers layers
SequN	int	4	number of chirp sequences

RAlts[ ]	float	4 x RAltN	ranging altitude layers
TAlts[ ]	float	4 x TAltN	temp. profile altitude layers (only if TAltN>0)
HAlts[ ]	float	4 x HAltN	hum. profile altitude layers (only if HAltN>0)
SpecN[ ]	int	4 x SequN	number of samples in Doppler spectra of each chirp sequence
RngOffs[ ]	int	4 x SequN	chirp sequences start index array in altitude layer array
SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	range resolution array for chirp sequences
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1(1)	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
Mon[ ]	float	20 x 4	housekeeping monitoring array (see A.4.1 LV0 format)
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
SLh_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for horizontal polarisation
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
Ze_1[n]	float	4	linear reflectivity in Ze units for vert. pol. in range bin n of sample 1
MeanVel_1[n]	float	4	mean velocity [m/s] for vert. pol. in range bin n of sample 1

SpecWidth_1[n]	float	4	spectral width [m/s] for vert. pol. in range bin n of sample 1
Skewn_1[n]	float	4	spectral skewness for vert. pol. in range bin n of sample 1
Kurt_1[n]	float	4	spectral kurtosis for vert. pol. in range bin n of sample 1
RefRat_1[n]	float	4	Differential reflectivity in range bin n of sample 1, table 2.11.2 (product 1.), [dB]
CorrC_1[n]	float	4	Correlation coefficient in range bin n of sample 1, table 2.11.2 (product 2.), [0,…,1]
DiffPh_1[n]	float	4	differential phase in range bin n of sample 1, table 2.11.2 (product 3.), [rad]
The following data is only stored if DualPol =2 (dual pol. radar in STSR mode)			
Ze45_1[n]	float	4	slanted reflectivity in range bin n of sample 1, linear Ze
SLDR_1[n]	float	4	slanted LDR in range bin n of sample 1, table 2.11.2 (product 4.), [dB]
SCorrC_1[n]	float	4	slanted correlation coefficient in range bin n of sample 1, table 2.11.2 (product 5.), [0,…,1]
KDP_1[n]	float	4	specific differential phase shift in range bin n of sample 1, see equ. (2.10.13), [rad / km]
DiffAtt_1[n]	float	4	differential attenuation in range bin n of sample 1, see equ. (2.10.14), [dB / km]
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
…	…	…	…
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
…	…	…	…

A.4.6 Example 6: DualPol = 1 / 2, CompEna = 2, AntiAlias = 0 / 1

LV0-File

Variable Name	Type	# Bytes	Description
File Code	int	4	LV0-File ID (=789346), Version 2.0
Header starts here			

HeaderLen	**int**	**4**	**header length in bytes (not including HeaderLen)**
ProgNo	**int**	**4**	**chirp program number in chirp table**
ModelNo	**int**	**4**	**=0: 94 GHz single pol. radar =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.**
ProgName	**char**	**Len(ProgName)+1**	**null terminated char string of chirp program name**
CustName	**char**	**Len(CustName)+1**	**null terminated char string of customer name**
Freq	**float**	**4**	**radar frequency [GHz]**
AntSep	**float**	**4**	**antenna diameter, [m]**
AntDia	**float**	**4**	**separation of both antenna axis (bistatic configuration), [m]**
AntG	**float**	**4**	**linear antenna gain**
HPBW	**float**	**4**	**antenna half power beam width [°]**
Cr	**float**	**4**	**radar constant, defined by equ. (2.1.5)**
DualPol	**char**	**1**	**=1: dual pol. radar, LDR conf. =2: dual pol. radar, STSR mode (see section 2.11.3)**
CompEna	**char**	**1**	**2: spectra compressed and spectral polarimetric variables are stored in the file**
AntiAlias	**char**	**1**	**0: Doppler spectra are not anti-aliased 1: Doppler spectra have been anti-aliased**
SampDur	**float**	**4**	**sample duration [sec]**
GPSLat	**float**	**4**	**GPS latitude**
GPSLong	**float**	**4**	**GPS longitude**
CalInt	**int**	**4**	**period for automatic zero calibrations in number of samples**
RAltN	**int**	**4**	**number of radar ranging layers**
TAltN	**int**	**4**	**number of temperature profile layers**
HAltN	**int**	**4**	**number of humidity profile layers layers**
SequN	**int**	**4**	**number of chirp sequences**
RAlts[ ]	**float**	**4 x RAltN**	**ranging altitude layers**
TAlts[ ]	**float**	**4 x TAltN**	**temp. profile altitude layers (only if TAltN>0)**
HAlts[ ]	**float**	**4 x HAltN**	**hum. profile altitude layers (only if HAltN>0)**
Fr[ ]	**int**	**4 x RAltN**	**range factors (see equ. (2.5.6))**
SpecN[ ]	**int**	**4 x SequN**	**number of samples in Doppler**

			spectra of each chirp sequence
RngOffs[ ]	int	4 x SequN	chirp sequence start index in altitude layer array
SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	chirp sequence range resolution [m]
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1(1)	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
Mon[ ]	float	20 x 4	housekeeping monitoring array (see A.4.1 LV0 format)
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
PNv_1[ ]	float	RAltN x 4	total IF power in v-pol. measured at ADC input
PNh_1[ ]	float	RAltN x 4	total IF power in h-pol. measured at ADC input
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
SLh_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for horizontal polarisation
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
SpecBytes_1[n]	int	4	number of bytes of following spectral block
BlockN_1[n]	char	1	number of blocks in spectra
MinBkIdx_1[n]	short int	2 x BlockN_1[n]	minimum index of blocks in spectra
MaxBkIdx_1[n]	short int	2 x BlockN_1[n]	maximum index of blocks in

			spectra
VSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed Doppler spectrum, vertical pol., linear Ze
HSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed Doppler spectrum, horizontal pol., linear Ze
ReVHSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed covariance spectrum , real part (see equ. 2.11.6), linear Ze
ImVHSpec_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed covariance spectrum , imaginary part (see equ. 2.11.6), linear Ze
RefRat_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed spectral differential reflectivity, see table 2.11.1 (product 1.), [dB]
CorrCoeff_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed spectral correlation coefficient, see table 2.11.1 (product 2.), [0,…,1]
DiffPh_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed spectral differential phase, see table 2.11.1 (product 3.), [rad]
SLDR_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed spectral slanted LDR, see table 2.11.1 (product 4.), [dB]
SCorrCoeff_1[n]	float	4 x BlockN_1[n] x (MaxBkIdx_1[n] - MinBkIdx_1[n] +1)	compressed spectral slanted correlation coefficient, see table 2.11.1 (product 5.), [0,…,1]
KDP_1[n]	float	4	specific differential phase shift, see equ. (2.10.13), [rad / km]
DiffAtt_1[n]	float	4	differential attenuation, see equ. (2.10.14), [dB / km]
VNoisePow_1[n]	float	4	integrated Doppler spectrum noise power in v-pol., [Ze]
HNoisePow_1[n]	float	4	integrated Doppler spectrum noise power in h-pol., [Ze]
The following data is only stored if AntiAlias =1			
AliasMsk_1[n]	char	1	mask indicating, if anti-aliasing has been applied (=1) or not (=0)
MinVel_1[n]	float	4	minimum velocity in Doppler spectrum [m/s]
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
…	…	…	…
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
…	…	…	…

LV1-File

Variable Name	Type	# Bytes	Description
File Code	int	4	LV1-File ID (=789347), Version 2.0
Header starts here			
HeaderLen	int	4	header length in bytes (not including HeaderLen)
ProgNo	int	4	chirp program number in chirp table
ModelNo	int	4	=0: 94 GHz single pol. =1: 94 GHz dual pol. STSR config. =2: 94 GHz dual pol. LDR config.
ProgName	char	Len(ProgName)+1	null terminated char string of chirp program name
CustName	char	Len(CustName)+1	null terminated char string of customer name
Freq	float	4	radar frequency [GHz]
AntSep	float	4	antenna diameter, [m]
AntDia	float	4	separation of both antenna axis (bistatic configuration), [m]
AntG	float	4	linear antenna gain
HPBW	float	4	antenna half power beam width [°]
DualPol	char	1	=1: dual pol. radar, LDR conf. =2: dual pol. radar, STSR mode (see section 2.11.3)
SampDur	float	4	sample duration [sec]
GPSLat	float	4	GPS latitude
GPSLong	float	4	GPS longitude
CalInt	int	4	period for automatic zero calibrations in number of samples
RAltN	int	4	number of radar ranging layers
TAltN	int	4	number of temperature profile layers
HAltN	int	4	number of humidity profile layers layers
SequN	int	4	number of chirp sequences
RAlts[ ]	float	4 x RAltN	ranging altitude layers
TAlts[ ]	float	4 x TAltN	temp. profile altitude layers (only if TAltN>0)
HAlts[ ]	float	4 x HAltN	hum. profile altitude layers (only if HAltN>0)
SpecN[ ]	int	4 x SequN	number of samples in Doppler spectra of each chirp sequence
RngOffs[ ]	int	4 x SequN	chirp sequences start index array in altitude layer array

SeqAvg[ ]	int	4 x SequN	number of averaged chirps within a sequence
SeqIntTime[ ]	float	4 x SequN	effective sequence integration time [sec]
dR[ ]	float	4 x SequN	range resolution array for chirp sequences
MaxVel[ ]	float	4 x SequN	max. Doppler velocity [m/s] for each chirp sequence (unambiguous)
Header ends here			
TotSamp	int	4	total number of samples
Sample 1 starts here			
SampBytes _1	int	4	length of sample 1 [bytes], not including SampBytes _1
Time_1 (1)	unsigned int	4	time of sample 1 [sec]
MSec_1	int	4	milliseconds of sample 1 [msec]
QF_1	char	1	quality flag of sample 1: Bit 1: ADC saturation Bit 2: spectral width too high Bit 3: no transm. power leveling
Mon[ ]	float	20 x 4	housekeeping monitoring array (see A.4.1 LV0 format)
TPr_1[ ]	float	TAltN x 4	temp. profile, sample 1
AHPr_1[ ]	float	HAltN x 4	abs. hum. profile, sample 1
RHPr_1[ ]	float	HAltN x 4	rel. hum. profile, sample 1
SLv_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for vertical polarisation
SLh_1[ ]	float	RAltN x 4	linear sensitivity limit in Ze units for horizontal polarisation
PrMsk_1[ ]	char	RAltN	mask array of occupied range cells of sample 1 0: range cell not occupied 1: range cell occupied
The loop over all range bins of sample 1 (loop index n) starts here			
The following data is only stored if PrMsk_1[n]=1			
Ze_1[n]	float	4	linear reflectivity in Ze units for vert. pol. in range bin n of sample 1
MeanVel_1[n]	float	4	mean velocity [m/s] for vert. pol. in range bin n of sample 1
SpecWidth_1[n]	float	4	spectral width [m/s] for vert. pol. in range bin n of sample 1
Skewn_1[n]	float	4	spectral skewness for vert. pol. in range bin n of sample 1
Kurt_1[n]	float	4	spectral kurtosis for vert. pol. in range bin n of sample 1
RefRat_1[n]	float	4	Differential reflectivity in range bin n of sample 1, table 2.11.2 (product 1.), [dB]

CorrC_1[n]	float	4	Correlation coefficient in range bin n of sample 1, table 2.11.2 (product 2.), [0,…,1]
DiffPh_1[n]	float	4	differential phase in range bin n of sample 1, table 2.11.2 (product 3.), [rad]
The following data is only stored if DualPol =2 (dual pol. radar in STSR mode)			
Ze45_1[n]	float	4	slanted reflectivity in range bin n of sample 1, linear Ze
SLDR_1[n]	float	4	slanted LDR in range bin n of sample 1, table 2.11.2 (product 4.), [dB]
SCorrC_1[n]	float	4	slanted correlation coefficient in range bin n of sample 1, table 2.11.2 (product 5.), [0,…,1]
KDP_1[n]	float	4	specific differential phase shift in range bin n of sample 1, see equ. (2.10.13), [rad / km]
DiffAtt_1[n]	float	4	differential attenuation in range bin n of sample 1, see equ. (2.10.14), [dB / km]
The loop over all range bins of sample 1 ends here			
Sample 1 ends here			
…	…	…	…
Sample TotSamp starts here			
SampBytes _ TotSamp	int	4	length of sample TotSamp [bytes], not including SampBytes _ TotSamp
…	…	…	…

A5. Filename Conventions

With software version 2.0 or later the filenames are fulfilling the following naming conventions:

Un-concatenated filenames contain the exact date and time of the first data sample in the format

YYMMDD_HHmmSS

Where YY is the year -2000, MM is the month, DD is the day, HH is the hour, mm is the minute and SS is the second.

All data files are starting with header sections depending on the chirp program that is used during the MDF measurement. When data files shall be concatenated, the program number of the two files to be concatenated needs to be the same. Therefore, the program number is always a part of the filename:

YYMMDD_HHmmSS_PXX

XX is the chirp program number.

All measurements are classified by the scan type. The following classification codes are defined:

ZEN : zenith observation
CEL : constant elevation and azimuth, but elevation is not equal to 90°
RHI : RHI scan (elevation scan with constant azimuth)
RHW : RHI scan with azimuth angle defined by wind direction
PPI : PPI scan (azimuth scan with constant elevation)
MIX : scan with mixed elevation and azimuth variation, neither PPI nor RHI

The measurement classification code is also part of all filenames:

YYMMDD_HHmmSS_PXX_MCC

MCC is one of the measurement classification codes defined above.

Level 0 and level 1 data files are distinguished by their file extensions:

YYMMDD_HHmmSS_PXX_MCC.LV0 for level 0 data files and
YYMMDD_HHmmSS_PXX_MCC.LV1 for level 1 data files.

The last filename convention is related to ***UNLIMITED*** and ***LIMITED*** mode measurements (see section 3.13.1). In ***UNLIMITED*** mode, the filename remains unchanged and a new file is created every hour. In ***LIMITED*** mode, the standard filename is preceded by the base name BasName defined in the measurement's MDF (see section 3.13.1) and the file is only closed when the associated MDF has finished:

BasName _YYMMDD_HHmmSS_PXX_MCC.LV0 for level 0 data files and
BasName _YYMMDD_HHmmSS_PXX_MCC.LV1 for level 1 data files.

This helps to distinguish between different MDFs within a batch (MBF) execution where only ***LIMITED*** mode MDFs are allowed.

When data files are concatenated to daily files, the resulting daily filename is reduced:

BasName _YYMMDD _PXX_MCC.LV0 for level 0 data files (***LIMITED*** mode) and
BasName _YYMMDD _PXX_MCC.LV1 for level 1 data files (***LIMITED*** mode).

or

YYMMDD _PXX_MCC.LV0 for level 0 data files (***UNLIMITED*** mode) and
YYMMDD _PXX_MCC.LV1 for level 1 data files (***UNLIMITED*** mode).