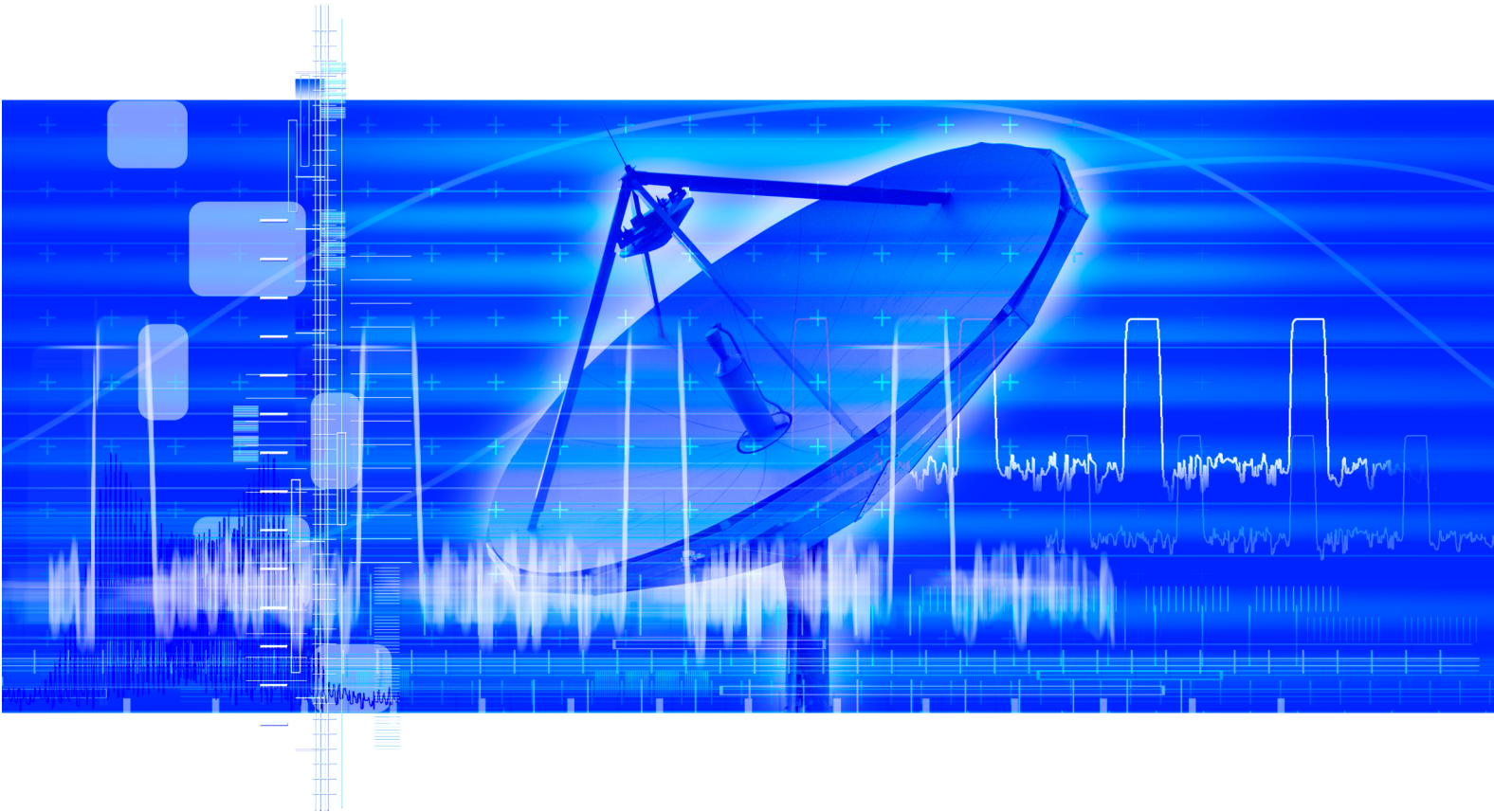


Software Manual



Pulse Sequencer Software

R&S[®] SMU-K6	1408.7662.02
R&S[®] SMJ-K6	1409.2558.02
R&S[®] SMATE-K6	1404.8006.02
R&S[®] AFQ-K6	1401.5606.00
R&S[®] AMU-K6	1402.9805.02

Printed in Germany



Dear Customer,

R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG. Throughout this manual, the R&S® SMU-K6, R&S® SMJ-K6, R&S® SMATE-K6, R&S® AFQ-K6, R&S® AMU-K6 is abbreviated as R&S Pulse Sequencer.

Trade names are trademarks of the owners.

Table of Content

Introduction	5
Installation	6
Minimum Hardware Requirements:	6
Software Requirements and Installation	6
Starting the Pulse Sequencer Software	7
The Project Tree	8
First Steps	9
Interfacing With The Instrument	10
Creating New Pulses	12
Timing Parameters	13
Delay Time	13
Rise Time	13
On Time.....	13
Fall Time.....	13
Off Time.....	13
PRI / PRF	14
Arbitrary Pulse Envelope	14
Level and Other Parameters	15
Jitter Settings	16
Uniform Distribution	16
Normal Distribution	17
Value List (Uniform).....	17
Value List (Ordered)	17
Shape (Interpolated).....	17
Using Tables as Source for Jitter Values	18
Modulation Settings	19
Built In Modulation Types	20
Marker Settings	23
Creating Sequences	24
The Sequence Editor	25
Other Sequence Settings	27

The Base Band Filter 29

Data Logging 31

Overlaying Multiple Entries 32

 Overlay Application Examples..... 33

 Radar Antenna TX, RX Simulation..... 33

 Blanking of Radar Receive Signal..... 33

Viewing the Sequence 33

 I/Q versus Time 33

 Cursor Functions 35

 I/Q Plane..... 35

 FFT 36

Multi Segment Waveforms 37

 Operating Multi Segment Waveforms 39

RF Lists..... 40

The Log Panel 41

Plug-ins..... 43

 The Plug-in Mechanism..... 43

 The Programming API..... 43

 Requesting the Plug-in Type 43

 Query Plug-in Version 43

 Define Plug-in Name 44

 Query Plug-in Comment / Explanation 44

 Query Plug-in Author 44

 Query Plug-in Error 44

 Plug-in Initialization 45

 Setup Plug-in Parameters 45

 Plug-in Modulation Engine 46

 Query Plug-in Configuration Parameters 46

 Setting Plug-in Configuration Parameters..... 47

Clock Rate Considerations 48

 Minimum Pulse Width..... 48

 Timing Error..... 49

 Dynamic Range 50

Introduction

The R&S Pulse Sequencer allows the flexible generation of complex pulses and pulse patterns. It is intended for use with the Rohde & Schwarz vector signal generators R&S SMU200A, R&S SMJ100A, R&S SMATE200A, R&S AMU200A and R&S AFQ100A.

This software provides an easy to use interface to build custom pulse envelopes, apply modulation or jitter as well as markers. It is also possible to build sophisticated test patterns for radar receiver tests. In addition, proprietary modulation schemes or envelopes can be applied by using the Pulse Sequencers plug-in mechanism.

Features:

- Easily generate complex pulse shapes and pulse patterns
- Create and manage a library of pulses as source for building pulse sequences
- Apply intra pulse modulation such as AM, ASK, FM, FSK, BPSK (Barker), QPSK, FM Chirps
- Extend built in modulation schemes with custom plug-ins
- Simulate technical systems by applying up to three jitter types to any pulse parameter and define the distribution
- Create multi segment waveforms for fast hopping between pulse patterns
- Create RF lists for fast hopping of frequencies and levels
- Organize your work in projects, pulse libraries and sequence libraries
- Compatible with R&S SMU200A, R&S SMJ100A, R&S SMATE200A, R&S AMU200A and R&S AFQ100A
- Automatic transfer of generated waveforms to vector signal generator using VISA Interface (GPIB, LAN, USB)
- Additional instrument options can be used to apply noise (AWGN), impairments or fading profiles to any pulse sequence. Two path instruments allow the combination and synchronization of two independent signals.

Installation

R&S Pulse Sequencer is intended for installation on a stand alone desktop PC running Microsoft Windows® XP Professional. The following is a list of prerequisites that should be met before installing the application:

Minimum Hardware Requirements:

- Pentium® III, Pentium® 4 at 750 MHz or faster
- 512 MB RAM
- 20 MB free HD space¹
- IDE/SATA or Ultra2/Ultra3 SCSI drive²
- LAN 100/1000 MBit for interfacing with instrument

¹The space is required for program installation. During waveform creation R&S Pulse Sequencer requires large temporary files. As a rule of thumb 9 Bytes per sample need to be considered for temporary file space. Example: 125 MSamples of waveform data call for about 1 GByte of temporary HD space.

² The HD is not only required to install the Pulse Sequencer software but also holds temporary data. Access should be as fast as possible to speed up waveform calculation.

Software Requirements and Installation

- Microsoft Windows® XP Professional
- Rohde & Schwarz VISA IO Libraries for Instrument Control, Rev. M.01.01 or other VISA Library, such as National Instruments VISA
- Instrument firmware release 02.04.xxx or higher

The installation of the R&S Pulse Sequencer is straight forward and simple. Execute the self extracting installer and follow the installation dialogs. After completion your application directory should contain the following structure:

```
... \Rohde-Schwarz \K6 Pulse Sequencer
    RS_PulseSequencer.exe      (application executable)
    settings.ini               (program settings file)
    \cvirte                    (run time environment files)
    \Plugins                   (plug-ins that are bundled with the software)
    \Projects                   (example project files)
    \Source Code               (code examples for custom plug-ins)
    \manual                    (user manuals)
    \help                       (online help files)
```

The R&S Pulse Sequencer software is started by executing 'RS_PulseSequencer.exe'. During start up the sub directory \Plugins is scanned for available user extensions. Plug-ins are automatically loaded and become available to the user in the main project tree.

The sub directory \Projects contains example project files.

Starting the Pulse Sequencer Software

After a fresh installation the R&S Pulse Sequencer starts with a default configuration.

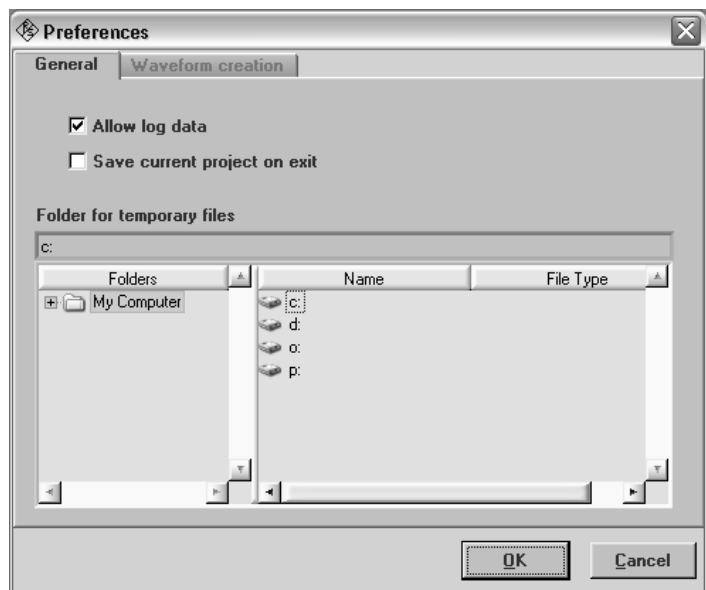
- All plug-ins from the sub directory /Plug-ins are loaded
- The project 'Examples.prj' is loaded
- All temporary files are located on C:\
- Program messages are written to the log panel
- Some example VISA connections are listed on the instrument panel

The first thing to do after a fresh installation should be to verify the general settings under *Options -> Preferences* from the menu bar.

Allow log data writes all program messages to a log panel. This may slow down some operations but provides useful information about what tasks are performed or errors.

Save current project on exit always saves the current project when R&S Pulse Sequencer terminates.

The **folder for temporary files** specifies the location where the R&S Pulse Sequencer keeps temporary data during waveform calculation. Read and write access to this drive should be fast. Thus, it is suggested to use a large local hard drive instead of network storage space. This setting is affective after the next program start since the software creates temporary files during start up.



The required file size depends on the waveforms created. As a rule of thumb 9 bytes per sample is required during waveform calculation. For example, if a pulse train requires 10 Msamples the temporary files rise to about 90 Mbytes. Using a baseband filter further increases the memory requirement by a factor of two.

All changes are accepted by pushing the **OK** button and saved on the next program shutdown. It is therefore suggested to exit and restart the Pulse Sequencer software if changes were made on the general settings tab.

The Project Tree

All user data, such as pulses, pulse sequences, jitter settings and multi segment waveforms are organized in projects. As a visual representation of a project information is structured in different libraries. That is the pulse and sequence library, multi segment waveforms as well as RF lists. The main tree on the left side of the program panel shows the content of the currently loaded project. Each book from the project tree indicates a library.

Newly created R&S Pulse Sequencer projects contain no data at all. Thus, setting up a new project always requires to define pulses first, then sequences and finally multi segment waveforms.

The following section describes the project tree content in more detail:

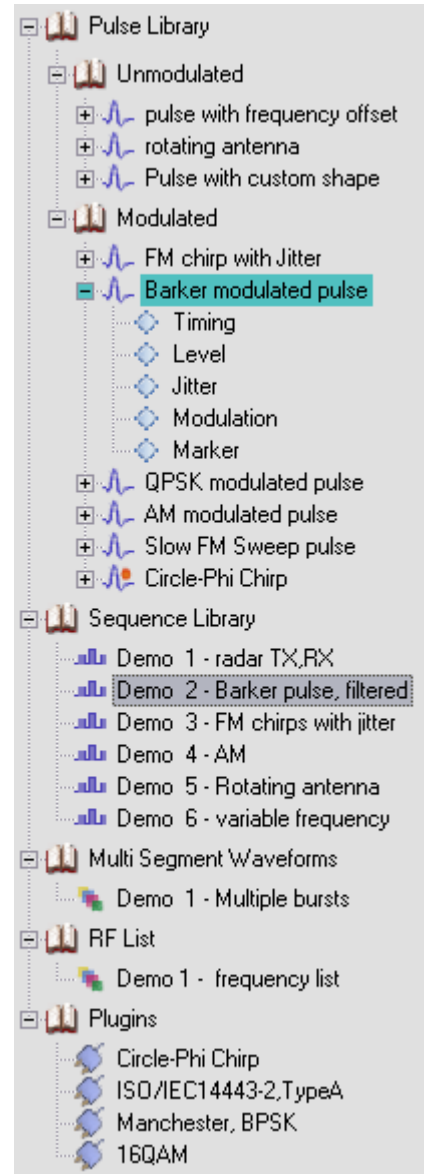
The **pulse library** is structured into unmodulated and modulated pulses. Pulses are the fundamental building blocks of any signal and thus need to be created first. Each pulse entry can be further expanded to unveil detailed settings, such as timing, modulation, jitter and marker data. Pulses that use custom plug-ins are indicated with a small red dot next to the pulse icon. Clicking on a pulse entry or one of its sub items brings up the associated editor window.

The **sequence library** contains pulse trains that are assembled from entries of the pulse library. The sequence can be compiled into a waveform which in turn can be uploaded to the vector signal generator. In addition to pulse entries a sequence definition may also contain additional information such as ARB clock rate, baseband filter settings and marker information.

Multiple sequences can be concatenated to form **multi segment waveforms**. One of the benefits of multi segment waveforms is that fast, arbitrary jumps between segments are possible. In addition, it is possible to batch compile and upload multiple sequences with a simple key press.

The Pulse Sequencer software can also be used to generate **RF Lists**. These lists contain frequency and level pairs which may be applied to any baseband signal. It is possible to use RF Lists in order to hop across a wide frequency or level range. RF list settings directly affect the up conversion stage and may thus be utilized with any baseband signal or multi segment waveform.

The **plug-ins** library shows all plug-in modules that were discovered during program start up. Plug-ins are Microsoft Windows DLLs that contain the math for any intra pulse modulation and may be used to extend the Pulse Sequencer software built in functionality. The software comes with example plug-ins that may be used as a starting point for own applications.



First Steps

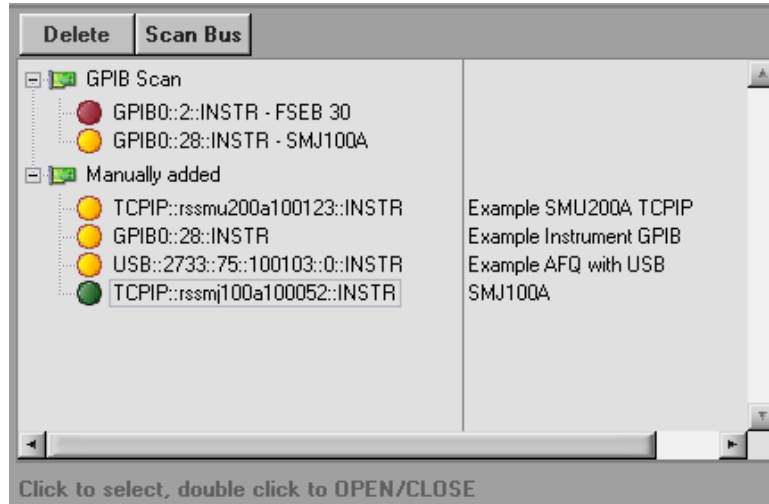
The following list outlines a typical workflow that is required to generate a sequence and compile it into a waveform.

- Create a new project (*File -> New Project*)
- Create a new pulse entry (*Create -> New Pulse*), assign it a name
 - Specify the pulse timing, e.g. rise time, on time, fall time and the edge shapes
 - Optionally set levels, frequency offset, phase if required
 - Note the default marker settings on the marker panel (Marker 1 active during rise, on and fall time)
- Create a new sequence (*Create -> New Sequence*), assign it a name
 - Add the new pulse by clicking the 'Add new Sequence Entry' button just above the sequence table
 - Set the pulse object to the newly created pulse
 - Increase the number of repetitions, if required and set the Marker 1 field to all by clicking into the field multiple times
 - Set the local waveform file name, e.g. C:\MyPulse.wv
- Hit the 'Create Waveform' button to build the sequence
- Hit the 'Transfer' button to switch to the instrument panel
- Add your instrument to the known instruments list or simply scan the local GPIB bus (if directly connected via GPIB). Use VISA resource strings for manually added instruments.
- Double click on the instrument. The yellow LED will turn green to indicate a good connection.
- Enter the remote file name, such as D:\MyPulse.wv
- Hit the 'Start Transfer' button

Once a sequence was calculated it is possible to change to the view panel. This panel allows the inspection of raw I/Q data as it is sent to the instrument.

Interfacing With The Instrument

The Pulse Sequencer software can interface with your instrument to upload and activate your waveforms, multi segment waveforms or RF lists. The software can keep a list of instruments and a connection is possible to one instrument at a time.



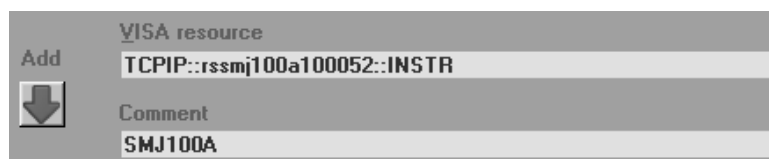
The instrument tree is divided into two sections. The first section lists all devices that were discovered during a scan of the local GPIB bus. The scan affects only board number zero (the first installed board). Supported instruments are marked with a yellow LED whereas unsupported devices are indicated by a red LED.

The second section contains instrument links that can be setup by the user. The Pulse Sequencer software uses VISA to connect to instruments. Thus, an instrument is always identified by a VISA resource string and physical connections such as GPIB, LAN or USB are supported through the software. Please see the manual of your instrument for supported interfaces and additional information.

A double click of any line marked with a yellow LED opens the instrument connection. If the connection was established the LED turns green.

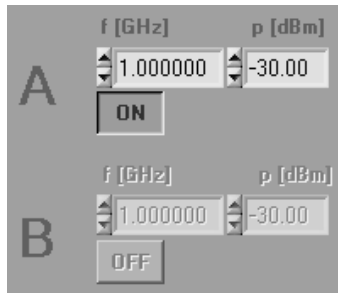
The **delete** button removes a marked entry from the list.

New instruments may be added at any time by using the controls shown below. The first line is used to enter the VISA resource string for the desired instrument whereas the second row may be used for an additional comment. The comment has no function but is displayed in the second column of the instrument tree.

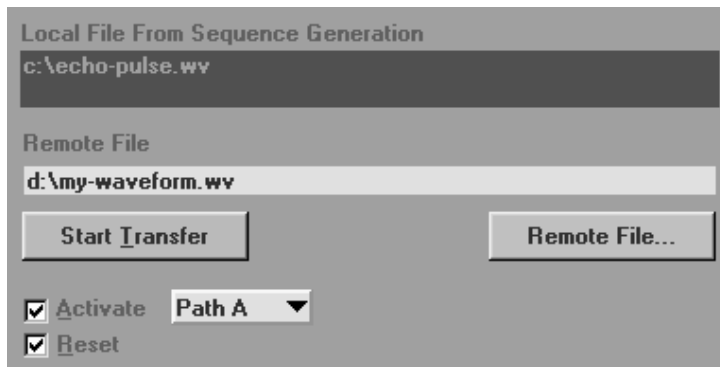


Clicking the **add** button appends the new entry at the end of the second tree section.

Once an instrument connection is established the software allows to remote control basic RF settings such as power, level and on / off state. These controls are only available if the current instrument supported these features. An AFQ100A for example does not contain any RF unit and thus the RF controls are greyed out. One path instruments do only show path A RF controls. All changes to the controls take effect immediately.



The lower part of the instrument panel can be used to transfer wave files to the instrument. The start transfer button is only active if an instrument connection is established. The **remote file** buttons opens a file browser for the remote file system on the instrument. Use the **activate** checkbox if waveform playback shall start automatically.



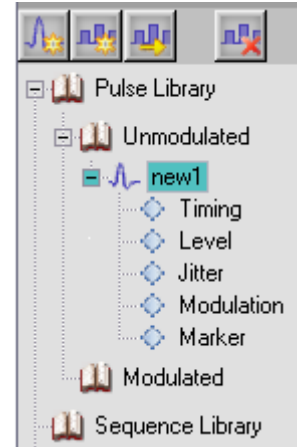
Creating New Pulses

Pulses are the building blocks of any sequence and thus need to be created as a very first step in a new project.

New pulses are created by either selecting 'Create -> New Pulse' from the menu bar or by clicking on the very left icon on top of the project tree.

In both cases a new unmodulated pulse with default settings is created and automatically added to the project tree. New pulse entries are always named 'new<n>' where n is a number starting at one and increasing for all newly created entries.

Next, pulse settings can be edited by directly selecting the pulse entry or one of its sub items. This brings up one of the pulse edit panels which are structured as listed below.



- **Timing**
This panel modifies all pulse envelope and timing related parameters, such as delay, rise, on, fall and off time. In addition, pulse repetition frequency or time may be specified as well as the shape of the rising and falling edges. It is also possible to use arbitrary data for the pulse envelope. The dialog offers an import mode where ASCII values can be imported and visualized.
- **Level**
The level panel is used to modify the pulse power during the on and off period. It also controls other parameters such as phase behaviour and frequency offset.
- **Jitter**
Up to three independent jitter values may be applied simultaneously to simulate real world scenarios. This panel controls all jitter related settings.
- **Modulation**
The modulation panel is used to assign a modulation to the pulse. The Pulse Sequencer software provides a range of most commonly used modulation schemes. In addition user plug-ins may be utilized to add custom pulse content. This dialog also sets the data source for modulation data.
- **Marker**
The marker panel allows assigns marker traces to any section of the pulse. Markers may be used to trigger external equipment such as spectrum analyzers or other signal generators. They can also trigger RF lists or multi segment waveforms.

The next paragraphs discuss all of the above listed panels in more detail.

Timing Parameters

Timing parameters affect the pulse shape and are usually the first and most important parameters to setup. The timing parameter panel can control all phases of the pulse. It is possible to specify time in nanoseconds (ns), microseconds (us), milliseconds (ms) or seconds (s). The total duration is automatically calculated and shown as sum below all settings.

The following section discusses the pulse phases in more detail.

The screenshot shows a control panel for pulse timing parameters. It includes the following fields and controls:

- Delay:** 0.000 us
- Rise:** 5.000 us, with a dropdown menu set to "linear"
- On:** 10.000 us
- Fall:** 5.000 us, with a dropdown menu set to "root cos"
- Off:** 30.000 us, with an unchecked checkbox
- PRI:** 50.000 us, with a checked checkbox

At the bottom, a horizontal bar represents the total duration, which is 50.000 us.

Delay Time

This is the time before the rising edge of the pulse. During this time the RF power is attenuated or totally suppressed. There is no modulation or data content present in this phase of the pulse. This setting may be used to shift the pulse location in time within a given window.

Rise Time

This parameter controls the total time of the rising pulse edge. The value is understood as the overall time from the very beginning to the very end of the slope. The RF level changes within this interval from the off-level to the on-level. In case of the off-level is more attenuated than the on-level a slope with rising RF power is generated. Typical values are 100 dB suppression for the off time and no suppression during the on-time.

Modulation may be preset during this phase of the pulse.

The type of the rising slope can be selected from linear, cosine and root cosine. Other shapes are possible using plug-ins or arbitrary envelope data.

On Time

During this time of the pulse the RF power is set to the on-level. Modulation may be present. This time period is understood as the total time from the very end of the rising edge to the very beginning of the falling edge.

Fall Time

For the falling time period the same applies as for the rising edge section. In contrast to the rising edge power is changed from the on-level to the off-level.

Off Time

The off period follows the falling edge of the pulse. The RF power is suppressed to the off-level value. No modulation is applied.

PRI / PRF

PRI stands for Pulse Repetition Interval whereas PRF stands for Pulse Repetition Frequency. Both numbers are used to control the overall time of a pulse including the leading delay and off time. This setting can be used alternatively to setting an off time.

The Pulse Sequencer software supports two methods of settings the total pulse time. One method is to simply add all times, such as delay, rising edge, on, falling edge and off. The summary is the total duration of the pulse. The second method uses PRF or PRI to control the total time. In this case the off time is automatically determined by adding the times for delay, rising edge, on period and falling edge. The remainder to the PRI is then used for the off time.

PRF or PRI settings are very useful if intra pulse timing parameters change (e.g. by jitter) but the total duration of the pulse cycle must remain constant.

Arbitrary Pulse Envelope

Instead of defining a pulse by its rise, fall and on time it is also possible to load custom envelope data.

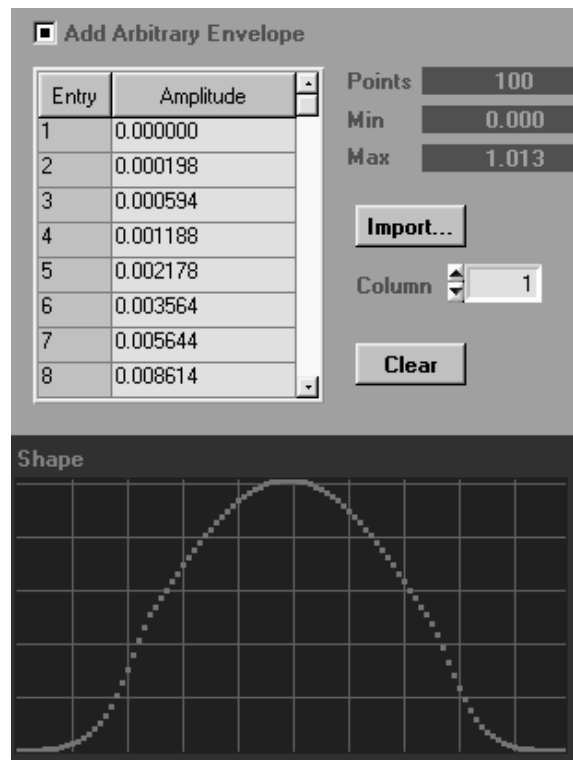
Custom envelope data only affects the level values versus time and thus can be used with any kind of modulation or other pulse setting. The basic idea behind custom envelopes is that the data is multiplied with the existing envelope from the timing parameters within the rise, on and fall time period. In an ideal case rise and fall time is set to zero and the pulse on time defines the length of the arbitrary pulse shape.

Since arbitrary pulse shape numbers are multiplied it is suggested to use a scale from 0 to 1.0 to obtain correct levels.

The Pulse Sequencer uses linear interpolation between data points to form the resulting pulse envelope. Statistics about the minimum, maximum and number of data points are shown next to the data table.

The **Import** button imports any amount of arbitrary data from an ASCII text file. The import column can be specified in the entry field below the import button.

Clear removes all data from memory and clears the list.



Note about arbitrary envelope data:

Once arbitrary data is imported it becomes a permanent part of the project file. Importing a large number of data points may therefore grow the project file to a very large size. Arbitrary data remains present even if the 'Add Arbitrary Envelope' button is disabled. This allows the user to flexibly switch between both modes without the need to reload any data. If arbitrary envelope data is not required anymore it is suggested to clear it from memory and the project file by using the 'Clear' button. Copying a pulse also copies all arbitrary envelope data.

Level and Other Parameters

Level settings are used to control the RF output power level during all phases of the pulse. The pulse sequencer uses two main settings to do so. One is the attenuation during the on period whereas the other is the attenuation during the off time.

Usually the attenuation during the off-time is much larger than during the on-time which causes a typical RF pulse. If in contrast the attenuation during the on-time is higher than during the off-time the result will be an inverse pulse.

Attenuation numbers must always be positive numbers.

Droop specifies a decay of RF power during the on period of the pulse. A positive number decreases the RF power by this amount between the end of the rising edge and the beginning of the falling edge. The level change is logarithmic is absolute RF power and thus linear in dB versus time.

In theory a 16 bit DAC offers $20 \cdot \log_{10}(65536) = 96$ dB of resolution. Thus, an attenuation level of 100 dB results in the 16 bit output value to be zero. However, the effective number of bits (ENOB) of the vector signal generators ARB reduces the range to maximum numbers in the range of 80 dB. Other effects such as carrier leakage may reduce this number further. Please refer to a more detailed discussion about these effects at the end of this manual.

The **phase** parameter influences the relative phase shift of the resulting RF wave. The permissible number range is -360.0 degrees to +360.0 degrees. The phase setting refers to the starting point of the pulse and modulation may change this phase during the pulse.

Continue phase keeps the phase from the end of the previous pulse and adds it to the actual phase setting. This allows the user to continue a modulation with differential coding from one pulse to the next.

The **frequency offset** shifts the pulse in frequency away from the RF carrier. It is important that large enough ARB clock rates are used if this feature is enabled. As a rule of thumb at least double the ARB clock rate is required. A value of zero disables the frequency offset.

The checkbox **Hide Item In Tree** may be used to hide this pulse entry in the project tree. This might be useful if a large number of pulses exist and the user only needs to generate sequences or multi segment waveforms without seeing the pulse definitions.

The lock just above the project tree is used to toggle the view mode between all entries and the visible ones only. A red lock indicates that hidden items are not shown.

Level Settings

Level Att (On) 0.00 dB

Level Att (Off) 100.00 dB

Droop 0.00 dB

Phase and Frequency

Phase 0.00 degrees

Continue Phase

Frequency Offset 0.000000 MHz

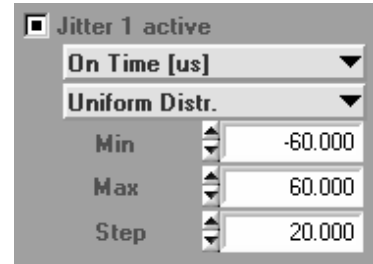
Other Settings

Show

Jitter Settings

Applying jitter is one of R&S Pulse Sequencers most powerful functions. This paragraph discusses jitter settings in detail and highlights possible areas of use.

In general a jitter is understood as the change of a pulse parameter in a random or ordered fashion. This allows simulating real world scenarios, such as imperfections of hardware, real data transmission channels or moving targets for radar pulses. It may also be used to blank parts of a waveform or generate step functions. Generally speaking jitter allows the user to vary pulse parameters within a series of repetitions.



The following parameters can be modified by jitter:

- Delay Time [us]
- Rise Time [us]
- On Time [us]
- Fall Time [us]
- Off Time [us]
- PRF or PRI [us, Hz]
- Level attenuation during on period [dB]
- Level attenuation during off period [dB]
- Frequency offset [MHz]
- Phase shift [degrees]
- Chirp deviation [MHz]

It is possible to assign up to three different jitter types individually and simultaneously. Each jitter type affects one particular pulse setting and can be one of the following distributions:

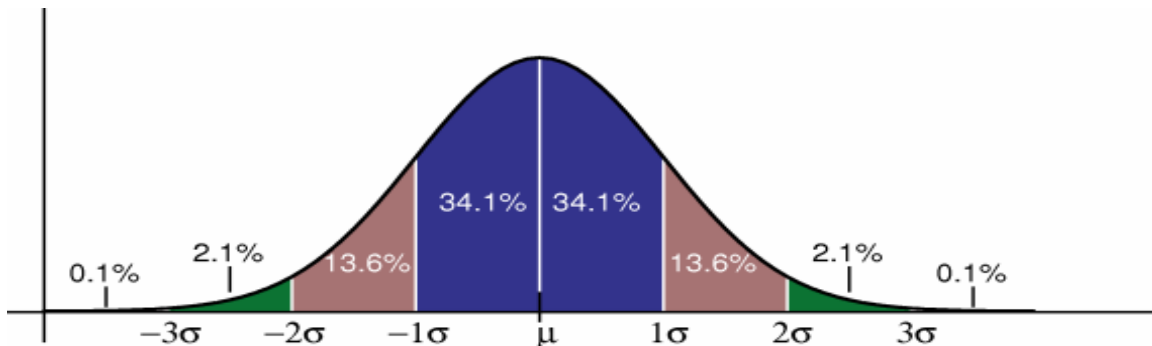
- Uniform distribution
- Gaussian distribution
- Uniform from value list
- Ordered from value list
- Interpolated shape

Uniform Distribution

The uniform distribution is characterized by the values Min, Max and Step Size. Values occur with the same probability in the range $\text{Min} \leq \text{value} \leq \text{Max}$.

Normal Distribution

The gauss or normal distribution is characterized by the parameters location, standard deviation and Min/Max. The following figure illustrates the probability at which values occur related to the standard deviation.



It can be seen that 68.2 % of the resulting values are located in the range Location \pm Variance and 99.6 % are located within \pm three times the variance.

At low probabilities any value may occur. If this is not desired the Pulse Sequencer allows the user to cut off the distribution at both ends by setting Min and Max limits. All values beyond this point are set to the according value. This in fact violates the normal distribution but is required to avoid invalid settings such as negative times or frequencies.

Value List (Uniform)

This setting uses a table of values as source for jitter. Numbers are drawn on a random base and at uniform distribution from the table.

Value List (Ordered)

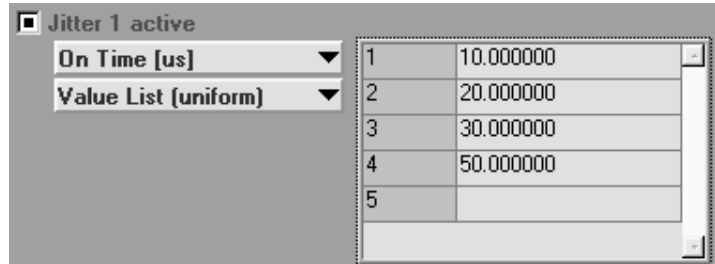
In this case jitter values are taken ordered from a table of possible values. The first pulse uses the very first list entry whereas the second pulse uses the second list entry. This mechanism continues until the end of the list is reached. A further pulse causes a wrap around and taking values starts over at the very beginning of the list.

Shape (Interpolated)

In this case values from the table are interpreted as shape that jitter values follow. Depending on the required number of jitter values linear interpolation may be used. There is no wrap around and thus, table values are always used from beginning to the end.

Using Tables as Source for Jitter Values

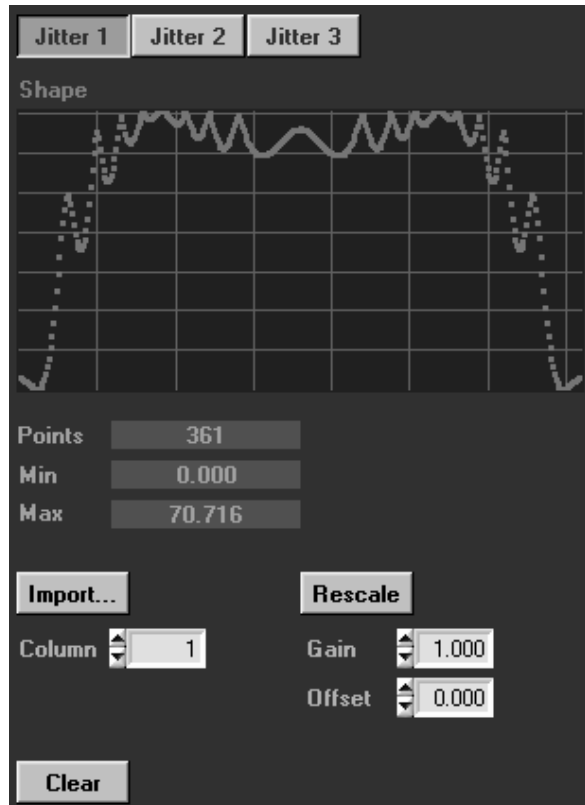
Tables may be used as source for discrete jitter values. Depending on the selected distribution type jitter values are taken randomly or in an ordered manner from the table. The table data is stored with the pulse in the project file. The maximum number of table entries is not limited but for speed and memory reasons extremely large tables should be avoided.



Double clicking into the second column allows the user to edit an existing entry. Blank entries are removed from the list entirely. The very last blank entry may be used to append to the list.

The top row of buttons (**Jitter 1**, **Jitter 2**, **Jitter 3**) is used to select for which jitter profile the panel shall be used. In the example the panel is used with jitter profile one.

The **graphics area** below plots all table values and thus allows a simple visual inspection of the import result. In addition Min and Max value as well as the number of data points are shown.



The **import** filter can process pure ASCII text files as data source. Data needs to be present in columns that are separated by at least one space. The example shows an import of data from the very first column of the file.

Once data is imported it is possible to **rescale** all values to fit the desired jitter range. Rescaling is done by first applying a gain factor and then adding an offset. This step may be executed repeatedly until the desired result is reached.

The **clear** button removes all values from the table and frees all associated memory.

Imported data becomes a permanent part of the pulse and the current project. If a pulse is copied Pulse Sequencer also creates a copy of all jitter values (deep copy). Turning jitter off does not free any associate memory or remove jitter data from the project file. Jitter shape data can only be removed by using the Clear button on the selected profile one, two or three.

Modulation Settings

The Pulse Sequencer allows to apply modulation to the area of rise, on and fall time of a pulse. A selection of most commonly used modulation types are built into the software but custom plug-ins may be used to extend the Pulse Sequencer's capabilities.



The **modulation** drop down box is used to select the pulse modulation. This setting needs to be applied first because it may affect other controls on the modulation panel. If the modulation is set to 'User' the software uses a custom plug-in to compute the intra pulse modulation. All plugging that were discovered during start-up and probed to be a modulation plug-in are listed in the **plug-in** radio box.

Depending on the type of modulation the user may choose a **data source**. In general one can select between various PRBS sequences or fixed data bits. All PRBS sequences are initialized randomly. If a fixed bit sequence is to be used the '**Data bits**' entry box allows the user to enter zeros and ones. Blank characters as well as line feeds are permissible.

The example shows a QPSK modulated pulse that uses 400 bits (200 symbols) of a PRBS 23 data stream.

In addition to data bits some modulation schemes may require additional **configuration parameters**. These parameters are displayed and can be edited on the right side of the modulation panel. All parameters are initialized with default values that are typical for the type of modulation. Once parameters are changed they become part of the pulse definition.

Configuration parameters		Reset
Setting	Value	
Bc [MHz]	0.800	
Bl [MHz]	1000.000	
Fmax [MHz]	10.000	

This note may sound trivial but it becomes very important when external plugging are used with configuration parameters. In this case the plug-in is queried for its default and limit settings. The user may alter these settings for each pulse individually and thus reuse the same plug-in for multiple pulse configurations. Once the actual pulse is compiled into wave data the plug-in is reinitialized with the appropriate user data for the current pulse.

The **Reset** button sets all configuration parameters back to the modulation or plug-in default settings.

Out of range entries are marked **yellow**. The limits for each entry are either determined by the built in modulation or in case of plugging are requested from the plug-in at program start.

Built In Modulation Types

ASK

ASK stands for Amplitude Shift Keying. The amplitude of the RF carrier is attenuated for a bit value of zero and remains at full level for bit values of one. The level of attenuation is specified as depth in percent.

Parameters:

Depth [%]	0...100	Modulation depth
Invert	Yes / no	Invert bits

FSK

FSK stands for Frequency Shift Keying. High bits set the frequency to +Dev/2 whereas low bits set the frequency to -Dev/2.

Parameters:

Deviation [MHz]	1 HZ...100.0 MHz	Modulation depth
-----------------	------------------	------------------

BPSK (Barker)

BPSK stands for Binary Phase Shift Keying. A bit value of one sets the phase to pi whereas zero bits leave the phase at zero. An additional phase offset may be specified to rotate the constellation points.

The following table lists existing Barker Codes:

Length	Code
2	+ - + +
3	+ + -
4	+ - + + + - - -
5	+ + + - +
7	+ + + - - + -
11	+ + + - - - + - - + -
13	+ + + + - - + + - + - +

Barker codes of length 11 and 13 are mainly used in pulse compression radar systems because of their good autocorrelation properties. A logic 1 is used for the + state whereas a logic zero indicates the - state.

Parameters: N/A

CBPSK (Constant envelope BPSK)

This type of modulation is similar to BPSK but transitions from one point to the next happen at constant envelope, thus only phase changes occur. A logic 1 shifts the phase by +pi whereas a logic zero shifts by -pi.

QPSK

QPSK stands for Quadrature Phase Shift Keying.

Two bits are required per sample and mapping is done using the following configuration:

00	+ $\frac{1}{4}$ pi
01	+ $\frac{3}{4}$ pi
10	- $\frac{3}{4}$ pi
11	- $\frac{1}{4}$ pi

Parameters: N/A

CQPSK (Constant envelope QPSK)

This type of modulation is similar to QPSK but transitions from one constellation point to the other happen at constant amplitude, thus, only phase changes occur. The following list shows how data bits are translated into phase changes:

00	- $\frac{1}{2}$ pi
01	- $\frac{1}{4}$ pi
10	+ $\frac{1}{4}$ pi
11	+ $\frac{1}{2}$ pi

8VSB

8VSB stands for Vestigial Side Band. VSB is a special type of phase modulation with 8 constellation points in a straight line. Three bits are required to form one symbol. Data bits are mapped according to the following table:

	<i>Phase</i>	<i>Amplitude</i>
000	+ $\frac{1}{4}$ pi	1
001	+ $\frac{1}{4}$ pi	0.714
010	+ $\frac{1}{4}$ pi	0.429
011	+ $\frac{1}{4}$ pi	0.143
100	- $\frac{3}{4}$ pi	0.143
101	- $\frac{3}{4}$ pi	0.429
110	- $\frac{3}{4}$ pi	0.714
111	- $\frac{3}{4}$ pi	1

FM Chirp

The FM chirp sweeps the RF across a certain frequency range. The deviation and shape may be specified. The total sweep range is $-Dev/2$ to $+Dev/2$.

Parameters:

Deviation [MHz]	1 Hz...100.0 MHz	Total Deviation
Shape	Linear ascending	The frequency is ascending linear from $-Dev/2$ to $+Dev/2$
	Linear descending	The frequency is descending linear from $+Dev/2$ to $-Dev/2$
	Sine	The frequency follows a full sine wave period
	Exp	The frequency is ascending exponentially according to 2.718281828^x From $-Dev/2$ to $Dev/2$
	Exp 10	The frequency is ascending exponentially according to 10.0^x From $-Dev/2$ to $Dev/2$

FM

FM stands for Frequency Modulation with a single tone.

Parameters:

Mod Freq [kHz]	0...25.0 MHz	Modulation frequency
Deviation [kHz]	0...100.0 MHz	Total deviation

AM

AM stands for Amplitude Modulation with a single tone.

Parameters:

AM Type	AM	Regular AM
	AM (LSB)	AM with only lower side band
	AM (USB)	AM with only upper side band
Mod Freq [kHz]	0...100.0 MHz	Modulation frequency
Depth [%]	0...100	Modulation depth

External Plug-in

The modulation may be defined by an external plug-in. This mechanism allows proprietary modulation of envelope types. Examples are bundled with the R&S Pulse Sequencer software.

Marker Settings

Markers 1 through 4 can be freely assigned to any section of a pulse by using the checkbox matrix.

The example shows that Marker 1 is assigned to the rising edge, on period and falling edge of a pulse whereas Marker 2 is setup as a restart flag.

	Marker 1	Marker 2	Marker 3	Marker 4
Delay	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Rise	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
On	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fall	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Off	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Restart	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Restart sets up the Marker to be active for the first 10 % of the entire pulse cycle duration.

Marker information is directly added to the resulting waveform and is therefore tied to the entire pulse sequence.

The Marker flag definitions from this matrix are the underlying marker definitions when pulses are used in sequences. It is possible to further restrict markers in the sequence editor but if a marker is not tied to any pulse section it cannot be used in a sequence. In addition the preferences panel allows to inhibit markers entirely.

If a marker flag is set the according marker becomes active for the entire number of samples used up for this section. Thus, activating a marker for the rising edge generates an output from the very beginning of the ramp until the very end of the ramp.

The same marker can be assigned to multiple sections of a pulse such as rise time, on time and fall time. In this case the marker remains active for all of the selected areas.

Creating Sequences

The sequence is a concatenation of pulse entries together with additional information, such as ARB clock rate, base band filters, jitter as well as marker information. They describe the content of the ARB waveform and thus describe the logical structure of the Pulse Sequencers waveform output.



A new sequence is created by either selecting 'Create -> New Sequence' from the menu bar or clicking the sequence creation button on top of the project tree (second button from left). In either case a new sequence is created and its name set to new<n>.

Clicking on a sequence in the project tree brings up the sequence editor. A prerequisite for the generation of a useful sequence is the existence of at least one pulse that is used with the new sequence.

The sequence editor is a blank table at the very beginning and must be populated with pulse entries. To do so, simply click on the 'Create New Sequence Entry' button just above the table. This button has multiple functions depending on how it is used:

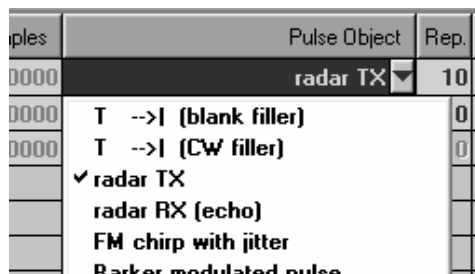
1. If no entries exist in the table a new sequence entry is added.
2. If entries exist but none of them is selected a new sequence entry is appended at the very end of the list. The new entry is a copy of the last one in the list.
3. If entries exist and an entry is selected a copy of the entry is inserted into the list below the marked item.

The Sequence Editor

Each pulse entry is assigned one line in the sequence editor and additional settings such as jitter modes and markers can be applied to each entry individually.

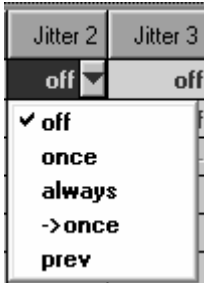
Item	Mode	T start [us]	T stop [us]	Samples	Pulse Object	Rep.	Jitter 1	Jitter 2	Jitter 3	M 1	M 2	M 3	M 4
1	---	0.00	1000.00	50000	radar TX	10	off	off	off	1ST	OFF	OFF	OFF
2	ADD	0.00	1000.00	50000	radar RX (echo)	10	always	always	off	OFF	ALL	OFF	OFF
3	---	1000.00	1200.00	10000	T --> (blank filler)	0	---	---	---	OFF	OFF	OFF	OFF

- Item**
 The very first field indicates the actual entry number. All pulse entries are numbered starting at index one. This column is only for reference and cannot be edited.
- Mode**
 This field can be toggled between the following states: '---', 'ADD', 'MULT' and describes how the actual line item is added to the final sequence. The default is '---' and indicates that the line item is simply appended. The other two modes are overlay modes and either add or multiply the pulse with previous data.
- Tstart[us]**
 Indicates the estimated starting time of this pulse entry in microseconds. This number does not take into account any alterations due to jitter. Thus, final numbers may be different if jitter is applied.
- Tstop [us]**
 Indicates the estimated stop time of this entry. The number is read only for regular pulse entries. In case blank or CW fillers are used this number can be set to the desired value.
- Samples**
 This field shows the required number of samples as estimation for the actual pulse entry. In case timing is altered by jitter the final number maybe different. The number also depends on the selected clock rate.



- Pulse Object**
 This field is the pulse entry selected from the pulse library. Clicking this entry field brings up a drop down box which can be used to select one of the existing pulse entries. Two special pulse entries exist at the top of the drop down box. These entries are not true pulses but act as time fillers. They either add a blank or CW signal up to a certain point in time. This mechanism may be used to create synchronization points. Filler pulses are highlighted in light blue.

- **Rep**
This number controls the count of repetitions that shall apply to this pulse entry. The default value of one adds the pulse entry exactly once to the sequence. Higher numbers repeat the pulse entry 'Rep' times.
- **Vary Jitter 1,2,3**
These settings controls how jitter values are applied in case of multiple repetitions. The selection '**always**' means that new jitter values are calculated and applied to each of the repetitions. Thus, all pulses that are generated from one line item are individual. The setting '**once**' calculates jitter values once for each line item and then reuses the value for all repetitions. The setting '**off**' disables changes caused by jitter entirely. The selection '**->once**' and is similar to '**once**'. The same jitter values are used within one line item and for the set number of repetitions. In contrast to once the next line item continues to use the next jitter value from a list of values specified within a pulse. This setting only makes sense if jitter values are taken from a list and subsequent line items use the same pulse definition. The selection '**prev**' repeats the same jitter values as they were calculated for the previous line item.

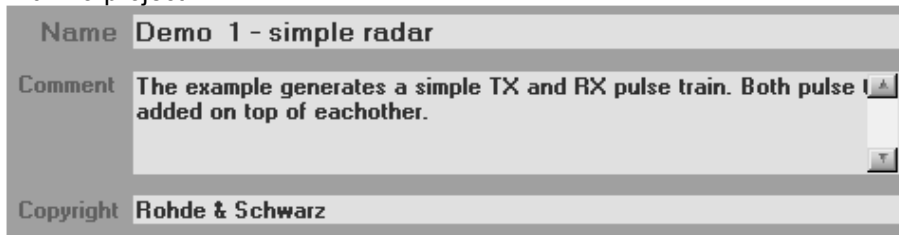


- **Mkr1...Mkr4**
These fields can be cycled through the states 'ALL' → 'OFF' → '1ST' → 'LAST' and are used to mask out marker traces. 'ALL' means that marker information is added for all repetitions as they are defined in the pulse settings. 'OFF' disables markers entirely for this entry. '1ST' allows the marker information only for the very first pulse from the repetitions whereas 'LAST' enables the markers solely for the very last pulse.

Other Sequence Settings

Besides the actual sequence editor table other parameters exist that are tied to a sequence. At the very top of the editor panel general information can be added as well as the name of the sequence specified. Below the sequence editor panel the waveform output name can be specified. The following section explains all controls in more detail:

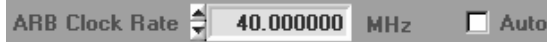
- **Name**
This field sets the name of this sequence as shown in the project tree. Names should be unique within a project.



The screenshot shows a settings panel with three fields:

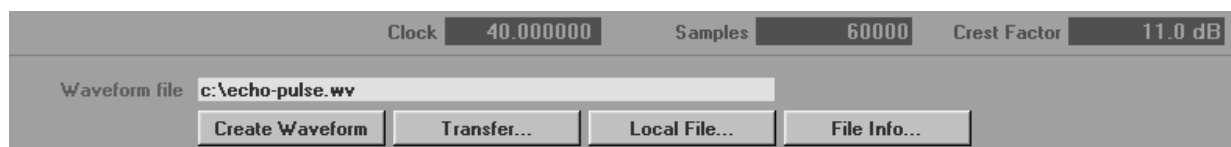
- Name:** Demo 1 - simple radar
- Comment:** The example generates a simple TX and RX pulse train. Both pulse added on top of eachother.
- Copyright:** Rohde & Schwarz

- **Comment**
The comment is optional and added to the final waveform. The entry field can be left empty if no comment is required.
- **Copyright**
The copyright information is optional and added to the final waveform. The entry field can be left blank if no copyright information is required.
- **ARB Clock**
The desired ARB clock rate. See the section clock rate considerations for more details. If the clock rate is changed the sequence need to be created again.



The screenshot shows the ARB Clock Rate control with a value of 40.000000 MHz and an unchecked checkbox labeled 'Auto'.

- **AUTO ARB Clock**
Automatically determines the minimum required ARB clock rate. Use this setting with great care. Particularly for custom modulation the Pulse Sequencer software does not know the over sampling requirements and the proposed clock rate might not work correctly.



The screenshot shows a settings panel with the following fields and buttons:

- Clock:** 40.000000
- Samples:** 60000
- Crest Factor:** 11.0 dB
- Waveform file:** c:\echo-pulse.wv
- Buttons: Create Waveform, Transfer..., Local File..., File Info...

- **Waveform File**
This is the local file name of the output waveform. It must be ensured that the path exists and the file can be written.
- **Create Waveform**
This button starts the waveform creation process. The button becomes active once a file name has been specified.

- **Transfer**
This button takes the user to the transfer window that allows to upload the locally generated wave file to the vector signal generator.
- **Local File...**
This button starts a file browser on the local file system. This browser may be used to locate existing files or set the pathname for the destination file.
- **File Info**
This button evaluates an existing waveform file for information such as duration, sample count, crest factors, etc.
- **Clock Rate, Samples, Crest Factor**
After a waveform is created statistics are shown just below the sequence editor panel. The clock rate is the final clock rate after over sampling. The samples are the true sample count of the waveform.

While a waveform is generated two progress bars and a stop button appear in the bottom part of the sequence panel. One progress bar shows the overall processing state whereas the other indicates the progress of the current task. The 'STOP' buttons interrupts the process and may be useful to stop extremely long processing times.



The Base Band Filter

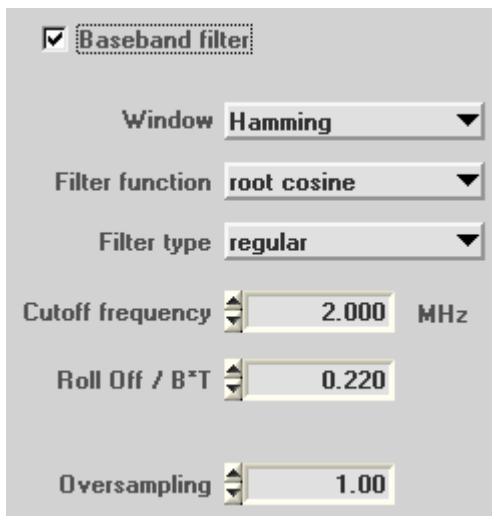
The Pulse Sequencer software can run its waveform output through a baseband filter. Each sequence can have an individual baseband filter configuration but all pulse entries within the sequence are processed using the same filter. Use the baseband filter button from the sequence editor to show the filter configuration dialog. A green LED next to the baseband filter button indicates that the filter is active. The following filter settings exist:



Base Band Filter

The check box enables the base band filter. Once the filter is activated the creation process happens in three steps:

1. compute raw waveform data
2. apply baseband filter to entire output data
3. rescale entire waveform to maximum vector length of 1.0



Window

The window function is multiplied with the filter function and thus influences the resulting output spectrum.

Filter function

The type of filter curve used for the baseband filter.

Filter	Alpha / BT	Impulse response
Rectangular	-	$h(t) = \text{sinc}\left(\frac{t}{T}\right) = \begin{cases} \sin(t/T) & t/T \neq 0 \\ 1 & t/T = 0 \end{cases}$
Root Cosine	Alpha 0.001 – 1.000	$h(t) = \frac{4\alpha}{\pi\sqrt{T}} \frac{\cos\left(\frac{(1+\alpha)\pi t}{T}\right) + \frac{T}{4\alpha t} \sin\left(\frac{(1-\alpha)\pi t}{T}\right)}{1 - \left(\frac{4\alpha t}{T}\right)^2}$
Cosine	Alpha 0.001 – 1.000	$h(t) = \frac{\text{sinc}\left(\frac{t}{T}\right) \cos\left(\frac{\pi\alpha t}{T}\right)}{1 - 4\left(\frac{\alpha t}{T}\right)^2}$
Gaussian	BT 0.001 – 1.000	$h(t) = \frac{\exp\left(\frac{-t^2}{2\delta^2}\right)}{\sqrt{2\pi} \cdot \delta} \quad \delta = \frac{\sqrt{\ln(2)}}{2\pi BT}$

Filter type

The filter type setting is used to modify the filter functionality.

1. regular filter is used as standard FIR filter
2. constant envelope filter creates a signal with almost constant envelope (e.g. GMSK modulation used for GSM)
3. single sideband filter creates a signal with only one sideband (e.g. 16VSB signal)

Cut-off frequency

This setting is the cut-off frequency of the baseband filter. A cut-off frequency of 2 MHz results in a total RF bandwidth of 4 MHz.

Alpha / BT

Some filters such as cosine or root cosine require an additional alpha-factor to determine the excess bandwidth. Gauss filters require the parameter B*T instead of alpha and the cutoff frequency.

B = Filter 3 db bandwidth

T = Symbol period

B*T is related to the ARB clock rate before any oversampling is performed. The Pulse Sequencer software estimates the filter bandwidth by the equation $f = BT * \text{samplerate}$ assuming that one symbol requires a single sample.

Over sampling

Pulse Sequencer allows the user to set any over sampling factor to fit the waveform output a certain number of samples. If fractional over sampling rates are used marker flag changes may not fall precisely on one data point.

Data Logging

In order to keep track of the created waveforms the Pulse Sequencer allows the user to write pulse and sequence parameters to a text file. All data is usually appended in form of columns that are separated by spaces. A headline is added on top of each table to indicate the meaning of each column.

The data log button in the sequence editor is used to open the dialog to configure data logging parameters. Data logging settings are saved with each sequence and can thus be defined differently for each sequence. A green LED next to the data logging button indicates that the log is active.



Data logging may slow down sequence generation since a single line of text is added to the log file for each pulse.

The **generate text report** checkbox enables or disables data logging for the current sequence.

Generate text report

File:

Pulse timing	Pulse settings	Sequence
<input type="checkbox"/> Delay	<input type="checkbox"/> Frequency	<input type="checkbox"/> Sequence entry no.
<input checked="" type="checkbox"/> Rise	<input type="checkbox"/> Phase	<input type="checkbox"/> Fillers
<input checked="" type="checkbox"/> On	<input type="checkbox"/> Level (on)	<input type="checkbox"/> Repetition count
<input checked="" type="checkbox"/> Fall	<input type="checkbox"/> Level (off)	<input checked="" type="checkbox"/> Repetition no.
<input type="checkbox"/> Off	<input type="checkbox"/> Droop	
<input type="checkbox"/> PRF/PRI	<input checked="" type="checkbox"/> FM deviation	
	<input type="checkbox"/> Start time	<input type="checkbox"/> Multi segment no.

The log **file** is written to the file specified under file. New data is always appended to this file. It is possible to use a different file for each sequence.

The checkboxes under the **pulse timing** section enable logging of the according value as one column in the log file. The recorded parameters are always the true values including jitter and possible errors due to the ARB clock rate setting. The values do not contain possible errors caused by baseband filter up sampling.

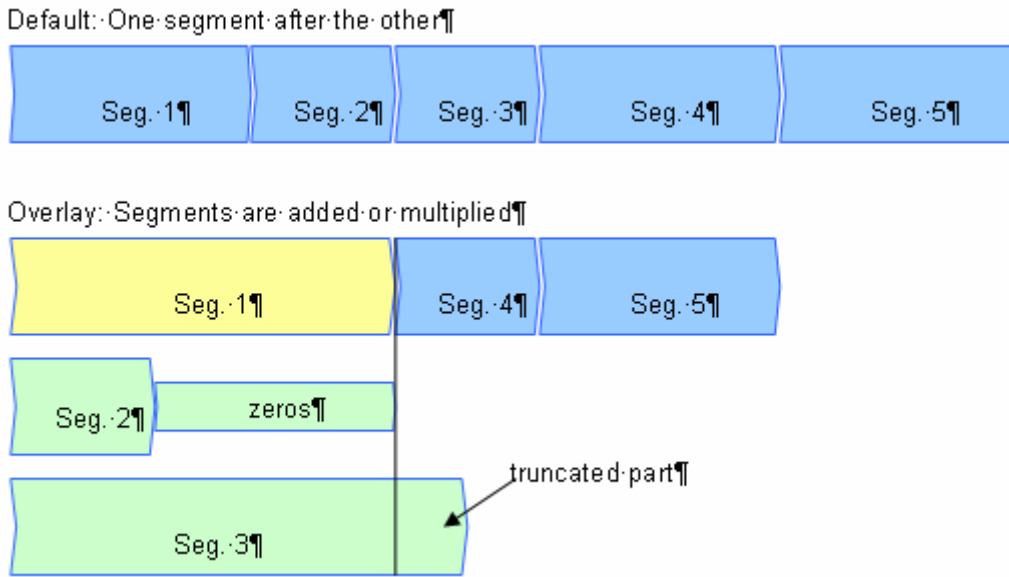
The other settings record parameters such as level, frequency offset or phase. A specific entry is the **start time**. This figure is calculated from the absolute start point of a pulse entry within a waveform plus a pre delay time if enabled. It is therefore useful to identify the absolute point in time of the rising edge of a pulse.

All entries in the sequence section are related to sequences and multi segment waveforms. **Sequence entry number** reports the line item within a sequence from one to the number of lines present within a sequence. **Fillers** reports the amount of time that was added by either a CW or blank filler entry. **Repetition count** is the number of repetitions that are specified for a line item in the sequence editor. The **repetition number** is used to identify each repetition individually. The number counts from one to the total number of repetitions set for a line item. **Multi segment number** records the line item in a multi segment waveform definition and can thus be used to identify one segment within a MSW.

Overlaying Multiple Entries

By default all entries from the sequence editor are added sequentially to the final waveform but under certain conditions it may be required to add multiple waveforms to each other. The process of adding waveforms is referred to as overlay mode by the Pulse Sequencer software.

Each line item within a sequence can be interpreted as sections on a timeline. In overlay mode, however the beginning of a section remains constant and new data is added to existing data. Multiple entries of a sequence can be defined to form one overlay group.



The picture shows the layout of an overlay group. The first segment (yellow) defines the length in time of the entire overlay period. Subsequent segments are added to the existing data starting at the beginning. If the new segment is shorter than the length of the overlay group zeros are appended. Longer segments are truncated. The Pulse Sequencer software also allows the usage of blank or CW fillers as first segments of an overlay group. This allows to set a defined length or end point for the entire group. Besides adding waveforms to each other it is also possible to multiply waveforms. This mechanism might be useful for blanking signals.

The picture below shows an example of an overlay of two waveforms.

Item	Mode	Tstart [us]	Tstop [us]	Samples	Pulse Object	Rep.	Jitter 1	Jitter 2	Jitter 3	M 1	M 2	M 3	M 4
1	---	0.00	1000.00	50000	radar TX	10	off	off	off	1ST	OFF	OFF	OFF
2	ADD	0.00	1000.00	50000	radar RX [echo]	10	always	always	off	OFF	ALL	OFF	OFF
3	---	1000.00	1200.00	9999	T --> (blank filler)	0	---	---	---	OFF	OFF	OFF	OFF

Item 1 starts the overlay by adding 10 pulses within a time period of 1 ms. The second item creates a waveform of another 10 pulses which are added to the first waveform starting at t=0. Both waveforms occupy the same amount of time and thus the second item in neither truncated nor zeros are appended. Item 3 starts adding blank filler starting at the end of the overlay period. The example shows how two pulse trains can be added.

Overlay Application Examples

Radar Antenna TX, RX Simulation

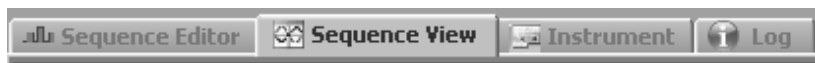
For radar receiver testing it might be desirable to generate signals that contain the transmit pulse as well as multiple receive pulses. In this case one sequence entry could be used to generate multiple repetitions of the TX pulse as they would be transmitted during one turn of a radar antenna. The following line items are set to overlay and add the receive pulse with a time delay, frequency offset or phase shift caused by jitter profiles. This allows to generate complex pulse return patterns as they might be caused by multiple reflections or antenna side lobes.

Blanking of Radar Receive Signal

Some radar systems blank their signals during fractions of one antenna turn. In order to simulate blanking one line item of the sequence could be set up to generate multiple repetitions of a pulse as it would result from one antenna turn without blanking. The next line item is then set to overlay and multiplies a single on / off pulse to generate the blanking.

Viewing the Sequence

Once an ARB waveform is successfully generated the result can be viewed using the sequence view panel. This panel displays the true I/Q values as they are used in the output waveform.



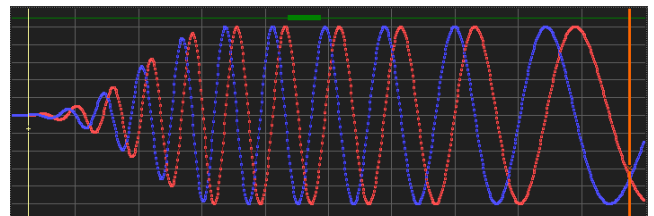
The sequence view tab is only visible if valid wave data exists and disappears again when changing back to pulse editor mode or selecting a new sequence. Interpreting the I/Q data correctly requires some basic knowledge about the principles of vector modulation. The most important pieces of information are provided in the sections below.

I/Q versus Time

The top window displays the I and Q traces versus time and is referred to a view port. The way data is presented changes depending on the amount of data looked at. If more data points than screen points exist the view mode shows straight lines between the minimum and maximum value. This ensures that no peak envelope data is missed. If less data points than screen points are looked at (zoomed in) the Pulse Sequencer software shows individual sample points and connecting lines. The two pictures below demonstrate the difference when looking at a chirped signal.



Zoomed out



Zoomed in

Both, I and Q signals are used in the modulator stage of the vector signal generator to modulate the RF carrier signal. Thus, it should be mentioned here that both, the I and Q signals shall not be misinterpreted as the generator's RF output signal.

At the top of the I and Q trace window navigation buttons are located to move back and forth or zoom in and out.

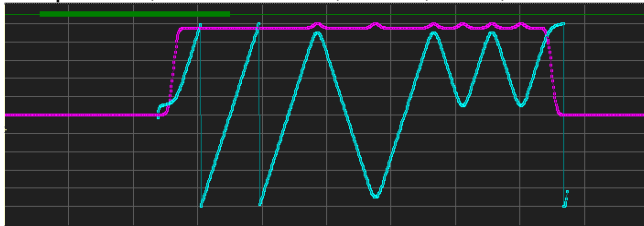
←	This button moves to the very beginning of the waveform.
←	Move left by half the display length
→	Move right by half the display length
Slider	Reposition within the trace
Lens -	Zoom out by a factor of two
Lens +	Zoom in by a factor of two
Lens	Zoom to the boundaries set by the two cursors

The buttons on the right side just above the I and Q trace window set the desired view mode. Each of the view modes has advantages and disadvantages and the use depends on the actual waveform and the zoom level.

- **Polar View**

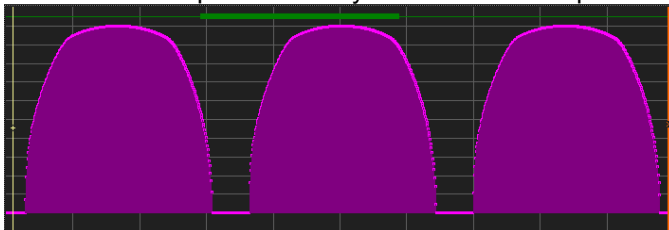
Switches the view port to polar view. In polar view data is not displayed as I and Q versus time but as magnitude and angle versus time. The angle is expressed as phase (green curve) in the range from $-\pi$ to $+\pi$. A change in phase is equivalent to rotating left or right at constant radius in the constellation diagram and thus changing the RF frequency. Once minus or plus π is reached the curve wraps around. The magenta trace indicates the magnitude which is equivalent to the RF envelope of the pulse.

This view mode may be helpful to inspect the pulse envelope or pulses with phase modulation components, such as BPSK, QPSK, QAM, etc.



- **Log Magnitude**

Data is displayed as envelope in dB ($20 \log[\sqrt{I^2+Q^2}]$). The scale ranges from 0 dB down to -100 dB and allows to display the full 16 bit dynamic range of the AD converters. This view mode is helpful because it allows to inspect even very low signal levels that would not be visible in the I/Q versus time mode. In addition the graphical representation of the envelope in dB would match a spectrum analyzer trace in zero span mode (logarithmic scale).



Cursor Functions

The I/Q versus time view port provides two vertical cursor lines that may be used for measurements or to zoom into the signal. Each cursor (yellow and orange) has two text boxes associated with that display the actual position in time and samples. The two fields in the line below display the difference between the two cursor positions in time and as frequency.

Time	353.297 us	704.959 us
Sample	17664	35247
dT / f	351.662 us	2843.6 Hz

I/Q Plane

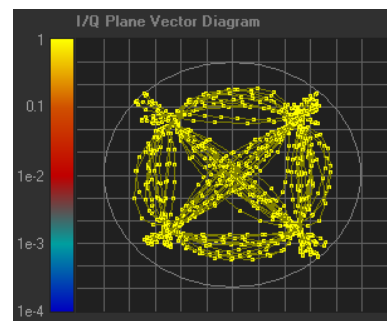
The Pulse Sequencer software always shows the view port content (I/Q versus time window) as vector or density diagram in the I/Q plane. The I/Q plane uses the I signal on the x-axis and the Q signal on the y-axis. The total range of the coordinate system is $-1.2/-1.2$ to $+1.2/+1.2$. Depending on the number of points that are displayed in the view port the display mode changes between two states:

- **I/Q vector diagram**

This mode shows I/Q data points with connecting lines in between. The displayed content is limited to the part of signal displayed in the I/Q versus time view port.

The example shows the vector diagram of a QPSK modulated and baseband filtered signal.

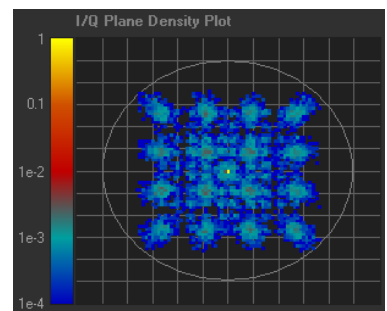
The grey circle marks the envelope level of 1.0. Clipping occurs if this limit is exceeded and thus the circle indicates the safe signal range.



- **I/Q density plot**

If a larger number of samples is looked at the Pulse Sequencer automatically switches to a density view. The density plot shows the probability at which I/Q points occur in the waveform. The display is relative, meaning that the point with the maximum probability is set to 1. In many cases of pulsed signals with long idle times this will be the origin. The color scale is logarithmic and ranges from 1 (bright yellow) down to a probability of 1/10000 (dark blue). The scale on the left side can be used to interpret the colors.

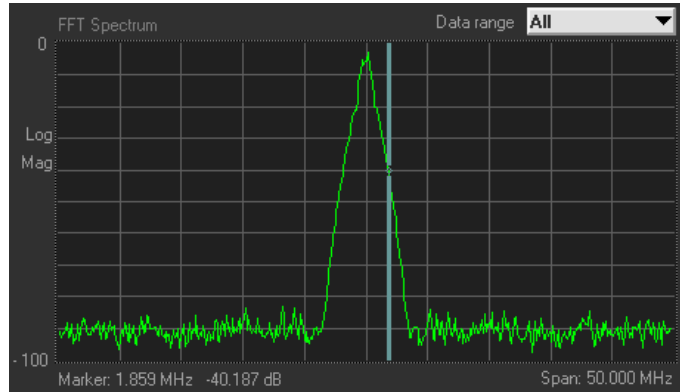
The example shows a larger piece of the same signal shown in the vector diagram above.



FFT

The FFT view allows the user to determine if the ARB clock rate is sufficient for the required signal bandwidth. The scale of the FFT is logarithmic from 0 down to -100 dB. The frequency span covers the range from minus sample rate / 2 to plus sample rate / 2.

Two view modes exist and specify the amount of data that is to be evaluated for the FFT calculation. By default the FFT is calculated across all samples in the output waveform. However, if the number of samples is too large the view mode can be set to view port only which limits the FFT view to only the content that is shown in the I/Q versus time window.



Multi Segment Waveforms

The Pulse Sequencer allows the user to create multi segment waveforms. These waveforms can be understood as a concatenation of multiple sequences with jump marks in between. The benefit is to allow fast arbitrary jumps between any of the sequences.

Assembling multi segment waveforms manually would require the following steps:

- Create one segment of the MSW locally (e.g. one sequence)
- Transfer file to instrument and append to the MSW
- Delete local wave file
- Repeat the above steps for all segments
- Compute multi segment waveform on the instrument
- Set playback parameters (trigger, etc)

All the above steps are automated by the Pulse Sequencer software. The following paragraph briefly outlines the settings on this panel.

A new MSW is created by calling *Create -> New Multi Segment Waveform* from the menu bar. This adds a new MSW to the project tree and calls the MSW editor.

Name specifies the name of the multi segment waveform. This name is used for reference only (project tree) and does not affect the waveform itself. Use unique names to identify your MSWs in the project tree.

The **table** contains all sequences in the order which they occur in the MSW. The icons on the left side above the table can be used to add new line items or remove a selected one.

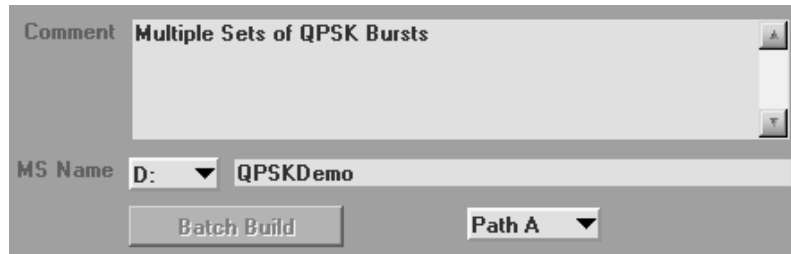
The columns **samples** and **CRF** (crest factor) are populated once a segment was calculated.

The column **sequence** is used to select entries from the sequence library.

The **up** and **down** buttons above the table may be used to move selected line items up or down in the list.

Item	Samples	CRF	Sequence
1	0	0.00	Radar Type 5 - 2, Path A
2	0	0.00	Radar Type 5 - 1, Path A
3	0	0.00	Radar Type 5 - 3, Path A
4	0	0.00	Radar Type 5 - 1, Path A
5	0	0.00	Radar Type 5 - 1, Path A
6	0	0.00	Radar Type 5 - 3, Path A
7	0	0.00	Radar Type 5 - 2, Path A
8	0	0.00	Radar Type 5 - 3, Path A
9	0	0.00	Radar Type 5 - 2, Path A
10	0	0.00	Radar Type 5 - 3, Path A
11	0	0.00	Radar Type 5 - 1, Path A
12	0	0.00	Radar Type 5 - 2, Path A

Multi Segment Waveforms are assembled on the instrument and thus additional information about the instrument file names need to be provided. Pulse Sequencer automates most of the steps of generating the MSW.



The **Comment** is optional and allows the user to add explaining text to the MSW definition. The text is not added to the MSW.

The **MS Name** sets up the drive and name of all multi segment related files, such as the configuration file (.inf_mswv), the file names of all segments and the resulting multi segment waveform file name (.wv).

The **Path** drop down box sets the destination path on the instrument for using the MSW with. This option is only supported on two path instrument configurations.

The button '**Batch Build**' starts the creation and assembling process. The process may take some time depending on the segment lengths and the number of segments. However, it fully automates the creation of all segments, transfer and assembly. The button is disabled if one of the following conditions are met:

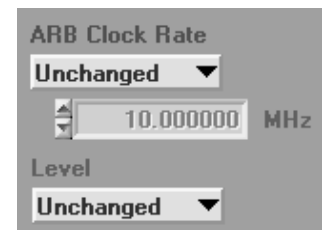
1. No instrument connection is established
2. No name is specified
3. The Instrument is an R&S AFQ100A.

Note:

The firmware of the AFQ100A does not support assembling multi segment waveforms. This task is usually done using WiniQSIM2 when operating this instrument.

In addition to the above file name related settings a range of parameters exist that influence the process of MSW creation on the instrument.

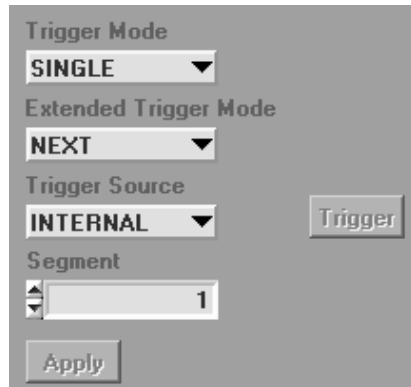
The **ARB Clock Rate** mode specifies how the segments are concatenated regarding the resulting ARB clock rate. If 'Unchanged' is selected the instrument leaves all clock rates as they are. Thus, switching clock rates may occur when jumping from one segment to the next. If 'Highest' is selected the instrument resamples all segments to the highest of all clock rates. The 'User' setting allows the user to set the target clock rate to a fixed value. Compiling a MSW takes more time when using User or Highest since the instrument needs to resample all waveform data.



The **Level** may also be altered when assembling a MSW. 'Equal RMS' forces the instrument to rescale all segments to the same RMS level whereas 'Unchanged' leaves all segments as they are.

Operating Multi Segment Waveforms

The Pulse Sequencer can remote control some functions related to multi segment waveforms. This may be useful for manual tests of to verify waveform segments.



All controls on the MSW panel are related to instrument functions that are described in more detail in the operating manual. The following section briefly explains the functionality. The controls become active only if an instrument connection is established.

Trigger Mode

This drop down box selects the trigger mode for the MSW. Valid choices are AUTO and SINGLE.

Extended Trigger Mode

This drop down box selects the extended trigger mode for the MSW. Valid choices are SAME, NEXT and SEAMLESS.

Trigger Source

This drop down box selects the trigger source for the MSW. Valid choices are INTERNAL, EXT1, EXT2 and PATH2.

Segment

This number selects the next segment that is to be selected when the Apply button is pressed.

Apply Button

This button writes all of the above settings to the instrument. Please refer to the operating manual of the instrument for more details about the options.

Trigger

This button becomes active if an instrument connection is set up and the trigger source is set to INTERNAL. Together with the trigger modes SINGLE + NEXT this may be used to manually switch the list to play back the next entry once.

RF Lists

RF Lists directly affect the up conversion of the baseband signal and thus may be used with any type of signal. An RF List typically contains pairs of level and frequency settings that are used for the up conversion. The list may be used for fast switching across wide level or frequency ranges.

Typical switching times are in the range of 400 us. Please see the instrument manual for further information about operating RF lists.

The Pulse Sequencer software can simplify and automate the creation of RF lists. New RF Lists are created by calling Create -> New RF List from the menu bar. This adds a new RF list entry to the project tree and calls the RF list editor.

Entry	Frequency [GHz]	Level [dBm]
1	1.400000	-30.00
2	1.600000	-30.00
3	1.500000	-30.00
4	2.000000	-30.00
5	1.300000	-30.00
6	1.800000	-30.00
7	1.200000	-30.00
8	1.700000	-30.00
9	1.100000	-30.00
10	1.900000	-30.00
11	0.000000	-100.00
12	0.000000	-100.00
13	0.000000	-100.00
14	0.000000	-100.00
15	0.000000	-100.00
16	0.000000	-100.00
17	0.000000	-100.00

The **Name** field is used to setup the name of the RF list. It is suggested to use unique names throughout projects since the name is used in the project tree to identify the RF list.

By default the list length is set to zero and no lines can be edited. The first thing is therefore to setup the list length using the **Set Length** button. The number left to the button defines the length between 1 and 10000 entries.

The **dwell time** sets the amount of time that each item is active once the RF list is activated on the instrument.

Frequency and **Level** values may be edited at any time within the enabled table rows. Using the **up** and **down** buttons line items may be moved up or down in the list.

It is possible to automatically populate the list using all the same or random numbers. The **value** drop down box defines whether frequency or level values are to be populated. The **Data Mode** drop down box selects how list items are populated. Valid choices are 'All Same', 'Uniform' or 'Unique Rnd'. The first option fills the list with the same values whereas Uniform uses a uniform distribution to populate the list. 'Unique Rnd' uses a uniform distribution but ensures that no item is used twice.

The Log Panel



Most of the Pulse Sequencers activities such as waveform creation or GPIB traffic are logged to a log panel. The log panel is always available from the tab controls on the right side of the main program window. The log window is read only and data can be marked and copied to the clipboard using Ctrl-C. The log panel output may be useful to determine the cause for an error or unexpected program behaviour. It also displays all GPIB communication between the Pulse Sequencer software and the instrument.

The following section shows a typical example of a waveform creation process. For each line item of a sequence information is written to the log file. The following entry provides information about the peak vector length of the output signal. If the number is larger than 1.0 Pulse Sequencer automatically rescales the entire waveform to safe values in order to avoid clipping. The following block provides statistics about the generated waveform file, such as crest factors and clock rate.

```
09:49:58  ===== Wave File Generation Start =====
09:49:58      REP      START  PULSE
09:49:58      10      0.000  radar TX
09:49:59      REP      START  PULSE
09:49:59      10      0.000  radar RX (echo)
09:49:59      BLANK 500.000 us
09:49:59  Waveform peak vector is  1.000.
09:50:00  Creating wave file c:\echo-pulse.wv...
09:50:00  -----
09:50:00  Clock rate   :40000000.000 Hz
09:50:00  Signal RMS   : 0.28 / RMSOfs:10.95 dBfs
09:50:00  Signal PEAK  : 1.00 / PKOfs : 0.00 dBfs
09:50:00  Signal CRF   :10.95 dB
09:50:00  -----
09:50:00  Creating marker trace from temporary marker file ...
09:50:01  Waveform file generated.
```

A typical section for a waveform upload is shown below. All GPIB traffic from Pulse Sequencer to the instrument is indicated by the prefix 'GPIB >>' whereas incoming information is marked with 'GPIB <<'. The log information provides the full GPIB traffic with the exception of binary transfers that are truncated to only a status line.

```

09:50:20 -----
09:50:20      File Upload
09:50:20 -----
09:50:20 Source       : 'c:\echo-pulse.wv'
09:50:20 Destination  : 'd:\test.wv'
09:50:20 GPIB >> 'SOUR:FREQ:MODE CW; *OPC?'
09:50:20 GPIB << '1'
09:50:20 GPIB >> 'SOUR:FREQ 1.000000 GHz; :POW -30.000000 dBm; *OPC?'
09:50:20 GPIB << '1'
09:50:20 GPIB >> 'OUTP:STAT ON; *OPC?'
09:50:20 GPIB << '1'
09:50:20 -----
09:50:20 Transferring wave file c:\echo-pulse.wv.
09:50:20 GPIB >> ':MMEM:MSIS 'd:('; *OPC?'
09:50:20 GPIB << '1'
09:50:20 GPIB >> ':MMEM:CDIR '\'; *OPC?'
09:50:20 GPIB << '1'
09:50:20 GPIB >> ':MMEM:CAT?'
09:50:20 GPIB << '2044225779,10520203264,".restore.lnk,...'
09:50:20 Destination file already exists on instrument.
09:50:21 GPIB >> ':MMEM:DEL 'd:\test.wv'; *OPC?'
09:50:21 GPIB << '1'
09:50:21 Opening input file c:\echo-pulse.wv.
09:50:21 Starting transfer: :MMEM:DATA 'd:\test.wv',#6240548 ...
09:50:21 GPIB >> 240548 bytes of binary data.
09:50:21 Data transfer complete.
09:50:21 GPIB >> 'SOUR1:BB:ARB:PRES'
09:50:21 GPIB >> 'SOUR1:BB:ARB:WAV:SEL 'd:\test.wv'; *OPC?'
09:50:21 GPIB << '1'
09:50:21 GPIB >> 'SOUR1:BB:ARB:STAT ON'
09:50:24 Transfer complete.
09:50:24 GPIB >> 'STAT:OPER:COND?'
09:50:24 GPIB << '0'
09:50:24 GPIB >> 'CAL:IQM:LOC?'
09:50:24 GPIB << '0'

```

Use this information in case an error is shown in the status line as it might unveil the root cause of the unexpected program behaviour.

Plug-ins

Plug-ins can be used to extend the Pulse Sequencers built in modulation capabilities. Some plug-ins are shipped with the software as binary and source code and may serve as a starting point for own applications. The following sections discuss the plug-in mechanism in more detail and provide information on the programming interface. In principle it is possible to use any programming language of choice but this documentation focuses on ANSI C.

The Plug-in Mechanism

Plug-ins are Microsoft Windows DLLs and need to be located in the sub directory \Plug-ins under the installation directory of the Pulse Sequencer software. This sub directory is searched during program start and useful plug-ins are loaded into memory for later use.

Every plug-in needs to provide a certain range of functions to identify itself and perform the calculations required for the intra pulse modulation. These functions are described further in the programming interface section of this manual. In addition, plug-ins may register a set of configuration parameters with the Pulse Sequencer software. These parameters become part of a pulse definition and may be used as parameters for equations used inside the plug-in. This allows to reuse plug-ins with different configurations.

The Programming API

The following paragraph lists all functions that need to be provided by the plug-in. It explains the interface as well as the functionality that needs to be provided by each function.

Requesting the Plug-in Type

```
void __declspec(dllexport) __cdecl mod_type (char szModType[1024]);
```

This function is mandatory. It provides a string that is used to determine the purpose of the plug-in. Currently plugins can only be used for intra pulse modulation but further extensions may serve a different purpose.

Parameters:

szModType	out	“modulation” = the plug-in is used for intra pulse modulation.
-----------	-----	--

Query Plug-in Version

```
void __declspec(dllexport) __cdecl mod_ver (char szModVer[1024]);
```

This function is mandatory. It shall return the version string of the plug-in.

Parameters:

szModVer	out	Format: X.Y.Z where X,Y and Z are numeric values, e.g. 2.34.01
----------	-----	--

Define Plug-in Name

```
void __declspec(dllexport) __cdecl mod_name (char szModName[1024]);
```

This function is mandatory. It provides the Pulse Sequencer software with the name of the plug-in. The name serves multiple purposes. It is used to reference the plug-in from a pulse descriptions and it is used in the project tree to identify the plug-in. Particularly, the first statement requires that plug-in names are unique and do not change at a later time. If the plug-in name changes the pulse definition becomes invalid and the pulse cannot be calculated anymore.

Parameters:

<code>szModName</code>	<code>out</code>	Plug-in Name. Must not be an empty string. Name must be unique.
------------------------	------------------	---

Query Plug-in Comment / Explanation

```
void __declspec(dllexport) __cdecl mod_comment (char szModComment[4096]);
```

This function is mandatory. It is used to return explaining text regarding the plug-in functionality.

Parameters:

<code>szModComment</code>	<code>out</code>	String with explaining text. Multiple lines are possible. May be an empty string.
---------------------------	------------------	---

Query Plug-in Author

```
void __declspec(dllexport) __cdecl mod_author (char szModAuthor[1024]);
```

This function is mandatory. It is used to return information about the author of a plug-in.

Parameters:

<code>szModAuthor</code>	<code>out</code>	String with author information. May be an empty string.
--------------------------	------------------	---

Query Plug-in Error

```
void __declspec(dllexport) __cdecl mod_error (char szModError[1024]);
```

This function is mandatory. It is called from Pulse Sequencer whenever another function returns FALSE and may return additional error information.

Parameter:

<code>szModError</code>	<code>out</code>	Explaining error text. It is suggested to clear any internal error text after is it queried through this function.
-------------------------	------------------	--

Plug-in Modulation Engine

```
int __declspec(dllexport) __cdecl mod_engine ( double *dAM,
                                             double *dPhase,
                                             int iActSample);
```

This function is mandatory. It is the core function of the plug-in and transforms samples into I/Q data.

Parameters:

dAM	out	Amplitude, range 0 ... 1.0
dPhase	out	Phase, range $-\pi$... $+\pi$
iActSample	in	Sample number. The number always starts at zero with the very first sample of the rising edge.

Return:

TRUE	Calculation was successful
FALSE	Error during calculation. Pulse Sequencer subsequently calls the error string function and terminates any further calculation.

It is required that this function returns useful numbers for amplitude and phase.

Query Plug-in Configuration Parameters

```
int __declspec(dllexport) __cdecl mod_getconf (int iIndex,
                                              char szType[256],
                                              char szName[256],
                                              void *pDefaultVal,
                                              void *pMin,
                                              void *pMax);
```

This function is optional. It may be used to register configuration parameters with the Pulse Sequencer software.

Parameter:

	iIndex	in	Index number of parameter, starting at zero
	szType	out	parameter data type identifier
	szName	out	name string associated with parameter
	pDefaultVal	out	pointer to default value
	pMin	out	pointer to minimum value
	pMax	out	pointer to maximum value

For the default, minimum and maximum value settings the Pulse Sequencer software provides the function with a pointer that could hold up to 1024 bytes. The functions needs to type cast this pointer to the required data type.

Available Data Types:

<u>szType</u>	<u>Data Type</u>	<u>Bytes</u>	<u>Precision</u>	<u>Purpose</u>
DBL	double	8	3	double precision value
DBL6	double	8	6	double precision value
INT	integer	4	0	integer value
BOOL	integer	4	0	boolean value (yes, no)
STR	string	255	0	zero terminated string

Setting Plug-in Configuration Parameters

```
int __declspec(dllexport) __cdecl mod_setconf (    int iIndex,
                                                void *pDat);
```

The function is optional but needs to exist if plug-in parameters were registered with the Pulse Sequencer software. It is used to set configuration parameters before a pulse calculation is started. All configuration values are referenced to by the index that was used when requesting parameters from the plug-in.

Parameters:

Parameter	Direction	Description
iIndex	in	Index of configuration parameter, starting at zero
pDat	in	Data from Pulse Sequencer. The pointer needs to type casted into the correct data type.

Return:

TRUE	Parameter was set successfully.
FALSE	An error occurred setting the parameter. Pulse Sequencer does subsequently query the plug-in error string and stop all further processing.

Clock Rate Considerations

This paragraph discusses issues that may arise from false clock rate settings. It gives advice for correct settings and points out limitations for pulse timing and other parameters.

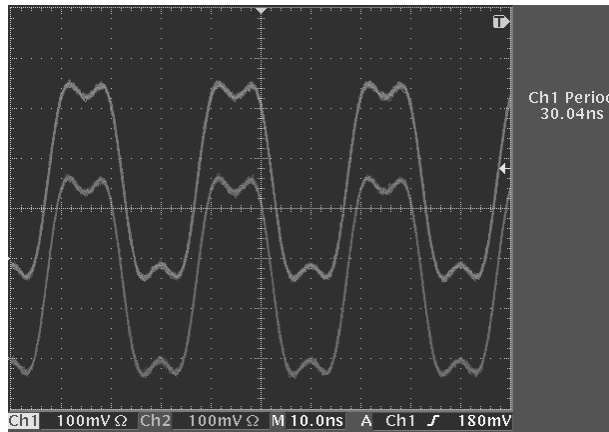
The R&S Pulse Sequencer is not directly dependant on the actual instrument when rendering waveform data. It therefore allows to use parameters within wide ranges event if the actual instrument is not capable of playing back the waveform file correctly. This behaviour is implemented intentionally to allow room for future hardware and provide means for experimentation with settings. However, special care must be taken and basic understanding of the ARB operation is required to determine optimum settings.

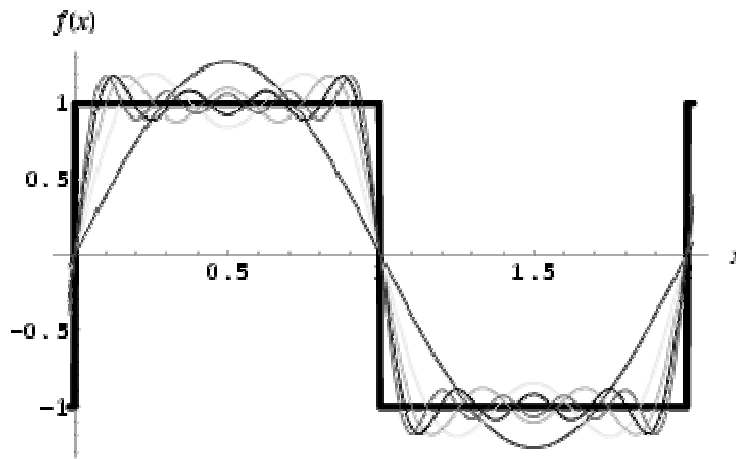
Minimum Pulse Width

The minimum pulse width is determined by the bandwidth of the instrument I/Q modulator. An AFQ 100A for example is rated at 100 MHz of maximum I/Q bandwidth. This bandwidth translates into $1 / 100 \text{ MHz} = 10 \text{ ns}$ period time. This number must be regarded as the shortest possible time at which a waveform of alternating zeros and ones playing back at 200 MHz generates a perfect sine wave.

In most cases pulses require defined shapes for their rising and falling edge, e.g. trapezoid or cosine. In this case a series of harmonics are required to achieve the desired shape. The quality of the shape increases with the number of harmonics that are available and thus the useful bandwidth decreases by the same amount.

The example below shows the I and Q output (into 50 Ohm load) of an AFQ100A playing back a waveform at 200 MHz ARB clock rate. The waveform consists of three alternating ones and zeros which generates a period time of 30 ns. Since the maximum bandwidth of the instrument is 100 MHz (10 ns) we can make use of frequencies up to the third harmonic.





The figure shows a Fourier series of a square wave and can be described by the following equation:

$$f(x) = \frac{4}{\pi} \sum_{n=1,3,5,\dots}^{\infty} \frac{1}{n} \sin\left(\frac{\pi nx}{L}\right)$$

It can be seen that even providing the 7th harmonic does not generate a very good square wave. This basic math should be taken into consideration when designing very short pulses or considering fast rise or fall times.

Timing Error

Timing is a discrete number when dealing with ARBs where the clock rate defines the granularity on the time axis. An AFQ 100A for example is specified at a maximum ARB clock rate of 300 MHz. This maximum clock rate results in a timing granularity of $1 / 300 \text{ MHz} = 3.333 \text{ ns}$. R&S Pulse Sequencer computes the number of samples from the timing figures as well as the clock rate setting.

Example:

The rising time is set to 25 ns and the ARB clock rate is set to 300 MHz. The number of samples is calculated as $25 \text{ ns} / 3.333 \text{ ns} = 7.50075$. The R&S Pulse Sequencer uses 7 samples leaving an error of $25 \text{ ns} - 7 * 3.333 \text{ ns} = 1.669 \text{ ns}$. If a clock rate of 200 MHz was used the granularity would be 5 ns and the timing error therefore zero.

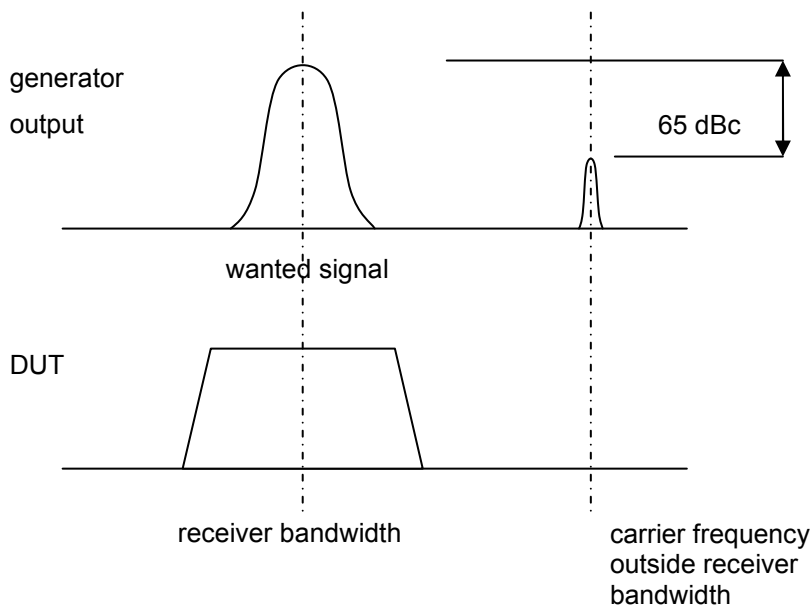
Dynamic Range

The full dynamic range of the R&S vector signal generators ARBs provide a total of 16 bits for both, the I and the Q signal. However, the effective number of bits is less due to multiple reasons. The following example explains the effect of the carrier leakage through the I/Q modulator and points out possible solutions on how to achieve higher dynamic ranges.

The instrument specification of the R&S SMU200A lists a typical carrier leakage value of -65 dBc for the I/Q modulator. This means that even if there is no ARB signal applied to the I/Q modulator and the output level is set to 0 dBm we still see a carrier at -65 dBm at the generator output. In some applications higher dynamic ranges may be required and additional effort is required to achieve this dynamic range.

Adding a frequency offset:

If the receiver bandwidth is narrow it is possible to add a frequency offset to the pulse definitions used in the R&S Pulse Sequencer software. The device under test would then see a carrier leakage of typical -65 dBc outside of its receiver bandwidth.



Using pulse modulation in parallel:

The R&S Pulse Sequencer software allows the flexible generation of marker signals. It is possible to tie a marker signal to the active part of the pulse and route this signal to the pulse modulator input of the vector signal generator. The pulse modulator in the R&S SMU200A offers an on/off ratio of greater than 70 dBc.

Index

8

8VSB..... 22

A

Alpha..... 31

AM 23

API..... 44

Arbitrary Pulse Envelope..... 15

ASK..... 21

B

Barker 21

Base Band Filter..... 30

Batch Build..... 39

BPSK..... 21

C

Chirp 23

Clock Rate Considerations 49

Cursor Functions..... 36

Cut-off frequency..... 31

D

Data Logging..... 32

Data source..... 20

Delay Time..... 14

Density plot 36

Droop..... 16

E

Extended Trigger Mode..... 40

F

Features..... 6

FFT 37

Files

 temporary 8

Filter function..... 31

Filter type..... 31

FM 23

Frequency..... 41

Frequency offset..... 16

FSK 21

H

Hardware Requirements 7

I

I/Q Plane..... 36

Installation 7

Instrument..... 11

Introduction..... 6

J

Jitter..... 17

L

Level..... 16,41

Library

 pulse 9

 sequence 9

Log panel..... 42

M

Magnitude View 35

Marker 24

Minimum Pulse Width 49

Mode 26

Modulation Settings 20

Multi Segment Waveforms..... 38

N

Normal Distribution 18

O

Off Time..... 14

On Time..... 14

Overlaying 33

P

Phase..... 16

Plug-in Author 45

Plug-in Comment..... 45

Plug-in Configuration Parameters..... 47

Plug-in Error..... 45

Plug-in Initialization 46

Plug-in Modulation Engine..... 47

Plug-in Name 45

Plug-in Parameters..... 46

Plug-in Type..... 44

Plug-in Version..... 44

Plug-ins..... 44

Polar View..... 35

Preferences..... 8

PRI / PRF..... 15

Project Tree..... 9

Pulse library 9

Pulse Object..... 26

Q

QPSK..... 22

R

RF Lists..... 41

S

Segment..... 40

Sequence Editor..... 26

Sequence library 9

Sequences..... 25

Shape..... 18

Software Requirements..... 7

T

Temporary files..... 8

Text report..... 32

Timing Error 50

Timing Parameters..... 14

Trigger..... 40

Trigger Mode..... 40

Trigger Source 40

Tstart.....26

Tstop.....26

U

Uniform Distribution 17

V

Value List..... 18

Vector diagram 36

Viewing the Sequence 34

VISA resource..... 11

W

Window.....30

Z

Zoom in.....35

Zoom out35