Interface Control Document

R&S®SMW-K503/-K504

Pulse descriptor word (PDW) and timed control descriptor word (TCDW)

Products:

► R&S®SMW200A

R&S® Munich | ICD | Version 2.4 | 02.2025

https://www.rohde-schwarz.com/appnote/1GP133

Version	Date	Author	Comments
2.4	01.02.2025	R&S Munich	► Add List Mode Timed Control Descriptor Word ► Update of chapter Network Interface Properties
2.3	15.05.2022	R&S Munich	► Correct error in chapter Pulse Descriptor Word Data Content (Body, Payload)
2.2	23.03.2022	R&S Munich	► Correct error in chapter Pulse Descriptor Word Data Content (Header)
2.1	22.12.2021	R&S Munich	► Update of chapter Network Interface Properties
2.0	02.09.2021	R&S Munich	► New ICD structure ► New descriptor word naming: Control PDW → Timed control descriptor word (TCDW) ► Removed RTDATA flag → RSVD ► New additional PDW format for new features ► Shaped pulses ► Burst PDWs
1.5	17.5.2018	R&S Munich	► Added chapter about the relation between streaming rate of ARB snippets and memory usage ► Changed maximum chirp with for real-time signals (25Bit) ► Updated Appendix2 minimum addressable RAM granularity from 64Samples to 1024 samples ► Added PATH bit inside control PDWs to select affected RF channel
1.4	20.04.17	R&S Munich	► Changed control PDW format. frequency and level changes are now possible simultaneously
1.3	27.02.17	R&S Munich	► Included control PDWs
1.2	18.01.17	R&S Munich	► Changed PDW Format to fixed length
1.1	13.12.16	R&S Munich	► Added appendix for calculation of maximum number of individual waveforms
1.0	09.12.16	R&S Munich	► First draft

Contents

List of Tables

1 Overview

1.1 Document Scope

The present R&S®SMW-K503/-K504 Interface Control Document contains information on

- ► the R&S descriptor word format, including pulse descriptor words and timed control descriptor words in basic and expert mode
- ► Timing requirements and limitations of the real-time control interface
- ► Properties of the ADV DATA/CTRL network interface

It is intended for use by customers using descriptor words to control the R&S®SMW200A in real-time. The interface control document specifies the interface between the customers hardware used for provision of descriptor words and the R&S®SMW200A ADV DATA/CTRL interface. Additional information on descriptor word processing inside the R&S®SMW200A is provided.

1.2 Document Overview

The present document is organized as follows:

- ► Chapter 1 is this introduction which provides the scope of the document, further reference and introduces a list of abbreviations and definitions
- ► Chapter 2 provides the R&S descriptor word structure, including pulse descriptor words and timed control descriptor words in basic and expert mode
- ► Chapter 3 provides timing requirements for PDW and TCDW streaming
- ► Chapter 4 provides information on the network interface

1.3 Further/Reference Documents

[1] "R&S®SMW-K501/-K502/-K503/-K504/-K315 Extended and Real Time Sequencing, Pulse-on-Pulse Simulation User Manual," Rohde & Schwarz.

[2] "R&S®SMW200A Vector Signal Generator User Manual," Rohde & Schwarz.

[3] "Generation of Radar Signals in a Hardware in the Loop (HIL) Environment," Rohde & Schwarz, Application Note.

1.4 Abbreviations and Definitions

The abbreviation “SMW” is used in this document for the Rohde & Schwarz product R&S®SMW200A.

The SMW is a general-purpose vector signal generator with outstanding RF performance. It is capable of generating signals for all main communication, radio and avionic standards and simulating GNSS or radar signals.

Extended Sequencer:

The Extended Sequencer is a firmware application that allows the SMW to generate complex signal sequences in real-time.

Further abbreviations:

PDW	Pulse Descriptor Word
TCDW	Timed Control Descriptor Word
xDW	Descriptor Word

Table 1: Abbreviations

2 R&S Descriptor Word Structure

2.1 General Descriptor Word Format Specification

2.1.1 General Descriptor Word Data Content

The SMW provides a dedicated interface to receive and process R&S descriptor words. R&S Pulse Descriptor Words (PDW) can be used to generate pulsed signals in real-time or replay pre-calculated waveform segments. R&S Timed Control Descriptor Words (TCDW) can be used to change instrument RF frequency and/or level or re-arm the Extended Sequencer.

Descriptor Word Type	Purpose
PDW	Generate pulsed signals in real-time. Replay pre-calculated waveform segments.
TCDW	Control RF parameters. Re-Arm Extended Sequencer.

Table 2: General Descriptor Word Data Content

Descriptor words are transmitted as sequence of bytes. For all descriptor words, their type is determined by flags in the header and flags section. The descriptor word size and content depend on the type.

2.1.1.1 Times

All times are given as number of clock cycles of the internal 2.4 GHz clock signal.

2.1.2 Bit and Byte Ordering Criteria

All data values are encoded using the following bit and byte ordering criteria:

- For numbering, the most significant bit/byte is numbered as bit/byte 0
- For bit/byte ordering, the most significant bit/byte is transmitted first (big-endian)

2.1.3 Reserved and Spare Bits

Reserved and spare bits may be used for evolution and can be defined in future updates of this ICD. In order to assure compatibility with future updates, these bits must be set to 0. The same applies for stuffing bits.

2.2 Pulse Descriptor Word (PDW)

Each pulse descriptor word consists of header, flags, body and payload. Depending on the operating mode (basic/expert) and whether a signal is generated in real-time based on the signal description or a pre-calculated waveform segment, also called ARB segment, is replayed, the content of each part is different. The expert pulse descriptor word format additionally can contain a parameter section or extensions. A mix of formats during a running simulation is not possible. The selection of the desired format can be done locally via the SMW GUI or remotely (SCPI commands).

In this section all PDW components are introduced. The basic and expert pulse descriptor word formats are defined. A comparison of basic and expert pulse descriptor words is provided.

2.2.1 Pulse Descriptor Word Data Content

The header and flags section of each PDW contain information about the timing and the content (structure) of the PDW. Information about the RF characteristics of the desired signal are given in the body section. The payload section either addresses a pre-calculated ARB segment or contains information to generate a pulsed signal in real-time. In expert mode, the params section allows to perform basic pulse shaping on real-time generated signals. Or the extensions section can be used to perform pulse shaping on real-time generated signals and/or generate multiple, identical pulsed signals with a single PDW (bursts).

2.2.1.1 Header

The PDW header section contains the time of arrival (TOA) and flags which define the content of the PDW.

PDW header			**48 Bit**	**56 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
TOA	unsigned int	Timestamp relative to scenario start trigger event TOA = (timestamp in seconds) * 2.4e9	44 Bit	52 Bit
SEG	Boolean	Flag that indicates if the PDW is used to address pre-calculated waveform segments or the PDW is used to generate pulsed signals in real-time without a waveform segment 0 = real-time signal 1 = ARB segment	1 Bit	1 Bit
USE_EXTENSION	Boolean	0 = PDW extension block is not used 1 = PDW extension block is used	-	1 Bit
PARAMS	unsigned int	Additional parameters 0 = no params 1 = use basic edge shaping 2 = RSVD 3 = RSVD	-	2 Bit
RSVD	-	Reserved for future use	3 Bit	-

Table 3: PDW Header Structure

2.2.1.2 Flags

The PDW flags section contains information about the xDW type, phase mode and marker settings.

PDW flags			**8 Bit**	**8 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
CTRL	Boolean	Indicates whether the descriptor word is a PDW or a TCDW 0 = PDW 1 = TCDW	1 Bit	1 Bit
RSVD	-	Reserved for future use	1 Bit	1 Bit
PHASE_MOD	Boolean	Indicates phase mode 0 = value in PHS field inside PDW body is absolute 1 = value in PHS field inside PDW body is relative to the phase value of the last sample of the previous signal	1 Bit	1 Bit
IGNORE_PDW	Boolean	PDW is ignored (no signal output)	1 Bit	1 Bit
M4	Boolean	Reserved for future use	1 Bit	1 Bit
M3	Boolean	Set Marker 3	1 Bit	1 Bit
M2	Boolean	Set Marker 2	1 Bit	1 Bit
M1	Boolean	Set Marker 1	1 Bit	1 Bit

Table 4: PDW Flags Structure

2.2.1.3 Body

The PDW body section contains offset values for frequency, level and phase relative to the instrument RF settings.

PDW Body			**64 Bit**	**64 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
FREQ_OFFSET	int	Frequency offset added to instrument RF frequency. -1 GHz <= frequency_offset <= 1 GHz FREQ_OFFSET= (frequency_offset / 2.4e9) * 2^{32}	32 Bit	32 Bit
LEVEL_OFFSET	unsigned int	Level offset subtracted from instrument RF level. level_offset >= 0 dB LEVEL_OFFSET = $10^{(-level\_offset / 20)} * 2^{15}$	16 Bit	16 Bit
PHASE_OFFSET	unsigned int	Phase offset 0° <= phase_offset < 360° PHASE_OFFSET = phase_offset/360° * 2^{16}	16 Bit	16 Bit

Table 5: PDW Body Structure

2.2.1.4 Params (only in expert mode)

Depending on the PARAMS bits in the PDW header, this 4 Byte block can be used for basic edge shaping. If USE_EXTENSION is set (USE_EXTENSION = 1), this block has to be omitted completely.

PDW params [PARAMS = 0]			**32 Bit**
Parameter	**Data type**	**Description**	**Size**
STUFFING	-	Params block unused. Set to 0.	32 Bit

Table 6: PDW Params Structure for unused Params field

PDW params [PARAMS = 1] (only supported for real-time signals: SEG=0)			**32 Bit**
Parameter	**Data type**	**Description**	**Size**
EDGE_TYPE	unsigned int	Describes rising/falling edge type 0 = Linear 1 = Cosine	3 Bit
MULTIPLIER	Boolean	Multiplier of Rise/Fall time to increase setting range. 0 = x1 1 = x8	1 Bit
RSVD	-	Reserved for future use	6 Bit
RISE_FALL_TIME	unsigned int	Rise/Fall time (first sample to last sample; is added to pulse width) RISE_FALL_TIME = ((Rise/Fall time in seconds) / Multiplier) * 2.4e9	22 Bit

Table 7: PDW Params Structure for basic edge shaping

2.2.1.5 Payload

Depending on the SEG flag settings and the type of real-time signal, different payloads apply.

The payload section either contains the segment index of a pre-calculated waveform or data that describes a real-time signal.

The following types of signals can be generated in real-time and do not require a pre-calculated waveform segment:

- Rectangular pulses with variable pulse width
- Barker codes (R2a, R2b, R3, R4a, R4b, R5, R7, R11, R13)
- Frequency chirps (Up, Down, Triangle)
- Optional edge shaping on real-time pulses

For waveform playback the SEG flag in the PDW HEADER has to be set to ‘1’.

ARB Segment [SEG = 1]			**136 Bit**	**96 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
SEGMENT_IDX	unsigned int	Index of the pre-calculated waveform, which was loaded into the SMW memory in advance	24 Bit	24 Bit
RSVD	-	Reserved for future use	112 Bit	72 Bit

Table 8: PDW Payload Structure for ARB segments

For a real-time signal the SEG flag in the PDW HEADER has to be set to '0'.

Real-Time Signal Unmod [SEG = 0 and MOD=0]			**136 Bit**	**96 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
MOD	unsigned int	Type of modulation 0 = Rectangular pulse	4 Bit	4 Bit
TON	unsigned int	Time on = Pulse width (first sample to last sample) TON = (Time on in seconds) * 2.4e9	44 Bit	44 Bit
RSVD	-	Reserved for future use	88 Bit	48 Bit

Table 9: PDW Payload Structure for unmodulated real-time signals

Real-Time Signal Linear Chirp [SEG = 0 and MOD=1]			**136 Bit**	**96 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
MOD	unsigned int	Type of modulation 1 = Linear Chirp	4 Bit	4 Bit
RSVD	-	Reserved for future use	19 Bit	3 Bit
TON	unsigned int	Time on = Pulse width (first sample to last sample) TON = (Time on in seconds) * 2.4e9	25 Bit	25 Bit
FREQ_INC	int	Frequency step in Hz/Sample freq_step = (bandwidth in Hz)/ (N_samples – 1) When using edge shaping, N_samples includes rising and falling edges. Without edge shaping, N_samples = TON. FREQ_INC = (freq_step/2.4e9) * 2^{64}	64 Bit	64 Bit
RSVD	-	Reserved for future use	24 Bit	-

Table 10: PDW Payload Structure for linear real-time chirps

Real-Time Signal Triangular Chirp [SEG = 0 and MOD=2]			**136 Bit**	**96 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
MOD	unsigned int	Type of modulation 2 = Triangular Chirp	4 Bit	4 Bit
RSVD	-	Reserved for future use	19 Bit	3 Bit
TON	unsigned int	Time on = Pulse width (first sample to last sample) TON = (Time on in seconds) * 2.4e9	25 Bit	25 Bit
FREQ_INC	int	Frequency step in Hz/Sample freq_step = (bandwidth in Hz)/ (N_samples - 1) When using edge shaping, N_samples includes rising and falling edges. Without edge shaping, N_samples = TON. FREQ_INC = (freq_step/2.4e9) * 2^{64}	64 Bit	64 Bit
RSVD	-	Reserved for future use	24 Bit	-

Table 11: PDW Payload Structure for triangular real-time chirps

<table>
<tr><th colspan="3">Real-Time Signal Barker [SEG = 0 and MOD=3]</th><th>136 Bit</th><th>96 Bit</th></tr>
<tr><th>Parameter</th><th>Data type</th><th>Description</th><th>Basic</th><th>Expert</th></tr>
<tr><td>MOD</td><td>unsigned int</td><td>Type of modulation
3 = Barker code</td><td>4 Bit</td><td>4 Bit</td></tr>
<tr><td>CHIP_WIDTH</td><td>unsigned int</td><td>Chip width of one Barker code chip
CHIP_WIDTH = (chip_width in seconds) * 2.4e9
The minimum supported chip width of a barker coded pulse is currently limited to 3.75 ns
(CHIP_WIDTH = 9).
The total pulse width is determined by the Barker code length times the chip width.</td><td>44 Bit</td><td>44 Bit</td></tr>
<tr><td>CODE</td><td>unsigned int</td><td>Code to select the type of Barker code
<table>
<tr><th>CODE</th><th>Length</th><th>Sequence</th></tr>
<tr><td>0</td><td>2</td><td>+-</td></tr>
<tr><td>1</td><td>2</td><td>++</td></tr>
<tr><td>2</td><td>3</td><td>++-</td></tr>
<tr><td>3</td><td>4</td><td>+-++</td></tr>
<tr><td>4</td><td>4</td><td>+---</td></tr>
<tr><td>5</td><td>5</td><td>+++-+</td></tr>
<tr><td>6</td><td>7</td><td>+++--+-</td></tr>
<tr><td>7</td><td>11</td><td>+++---+--+-</td></tr>
<tr><td>8</td><td>13</td><td>+++++--++-+-+</td></tr>
</table></td><td>4 Bit</td><td>4 Bit</td></tr>
<tr><td>RSVD</td><td>-</td><td>Reserved for future use</td><td>4 Bit</td><td>4 Bit</td></tr>
<tr><td>STUFFING</td><td>-</td><td>Stuffing bits</td><td>16 Bit</td><td>16 Bit</td></tr>
<tr><td>RSVD</td><td>-</td><td>Reserved for future use</td><td>64 Bit</td><td>24 Bit</td></tr>
</table>

Table 12: PDW Payload Structure for Barker coded real-time signals

2.2.1.6 Extension (only in Expert mode)

The 20 Byte extension is evaluated if the USE_EXTENSION bit in the header is set to 1. This Block has three extension fields which can be used by setting the corresponding extension flags.

2.2.1.6.1 Extension Flags

The extension flags section allows the user to define the type of extension. Up to three extension fields can be used.

PDW extension flags			**16 Bit**
Parameter	**Data type**	**Description**	**Size**
FIELD_1_TYPE	unsigned int	Describes type of field 1 0 = Unused 1 = Edge field 2 = Burst field 3-7 = RSVD	3 Bit
FIELD_2_TYPE	unsigned int	Describes type of field 2 0 = Unused 1 = Edge field 2 = Burst field 3-7 = RSVD	3 Bit

FIELD_3_TYPE	unsigned int	Describes type of field 3 0 = Unused 1 = Edge field 2 = Burst field 3-7 = RSVD	3 Bit
RSVD	-	Reserved for future use	7 Bit

Table 13: PDW Extension Flags Structure

2.2.1.6.2 Extension Fields

Unused Field [FIELD_x_TYPE = 0]			**48 Bit**
Parameter	**Data type**	**Description**	**Size**
STUFFING	-	Fill with 0	48 Bit

Table 14: PDW Extension Field Structure for Unused Extension Fields

Edge Field [FIELD_x_TYPE = 1] (only supported for real-time signals: SEG=0)			**48 Bit**
Parameter	**Data type**	**Description**	**Size**
EDGE_TYPE	unsigned int	Describes rising/falling edge type 0 = Linear 1 = Cosine	3 Bit
MULTIPLIER	Boolean	Multiplier of rise/fall time to increase setting range. 0 = x1 1 = x8	1 Bit
RISE_TIME	unsigned int	Rise time (first sample to last sample; is added to pulse width) RISE_TIME = ((Rise time in seconds) / Multiplier) *2.4e9	22 Bit
FALL_TIME	unsigned int	Fall time (first sample to last sample; is added to pulse width) FALL_TIME = ((Fall time in seconds) / Multiplier) *2.4e9	22 Bit

Table 15: PDW Extension Field Structure for Edge Shaping

Burst Field [FIELD_x_TYPE = 2]			**48 Bit**
Parameter	**Data type**	**Description**	**Size**
BURST_PRI	unsigned int	Pulse repetition interval (PRI) BURST_PRI = (PRI in seconds) * 2.4e9	32 Bit
BURST_ADD_PULSES	unsigned int	Number of repetitions in addition to the initial signal (real-time signal or ARB segment)	16 Bit

Table 16: PDW Extension Field Structure for Bursts

2.2.2 Pulse Descriptor Word Bits Allocation

The following tables illustrate the PDW bit allocation for basic and expert PDWs without and with extensions.

Header	**Flags**	**Body**	**Payload**	**Total (bits)**
48	8	64	136	256

Table 17: Bits allocation of basic PDW

Header	Flags	Body	Params	Payload	Total (bits)
56	8	64	32	96	256

Table 18: Bits allocation of expert PDW without extensions

Header	Flags	Body	Payload	Extension Flags	Extension 1	Extension 2	Extension 3	Total (bits)
56	8	64	96	16	48	48	48	384

Table 19: Bits allocation of expert PDW with extensions

2.2.3 Comparison of Basic and Expert Pulse Descriptor Word Format

The following table illustrates the differences between basic and expert PDW format. Expert PDW format is recommended.

	Basic mode	**Expert mode**	
		w/o Extensions	With Extensions
Size	32 Byte	32 Byte	48 Byte
Max. Playtime	~ 2 h	> 520 h (~21.5 days)	
Min TOA difference of consecutive PDWs • Real-time signals • ARB segments	 0.5 µs 1.0 µs	 0.5 µs 1.0 µs	 1.0 µs 1.0 µs
(Basic) Edge shaping	✗	(✓)	✓
Burst PDWs	✗	✗	✓

Table 20: Comparison of Basic and Expert Pulse Descriptor Word Format

2.2.4 PDW based ARB segment streaming (rates and memory usage)

To address pre-calculated waveforms with PDWs, ARB segments have to be uploaded to the SMW before starting the simulation. This can be performed via the SMW GUI or remotely with SCPI commands.

A table showing all preloaded segments can be accessed via the SMW GUI.

Segment Index	Filename	Clock Rate	Samples	Length	Path	Info
0	Pulse_10us_1000MHz.wv	1.000 GHz	100000	100.000 µs	/var/user/	Info...
1	Pulse_20us_1000MHz.wv	1.000 GHz	100000	100.000 µs	/var/user/	Info...
2	Pulse_30us_1000MHz.wv	1.000 GHz	100000	100.000 µs	/var/user/	Info...
3	Pulse_30us_500MHz.wv	500.000 MHz	50000	100.000 µs	/var/user/	Info...
4	Pulse_30us_50MHz.wv	50.000 MHz	5000	100.000 µs	/var/user/	Info...

Figure 1: Pre-calculated segment table view in the SMW200A GUI

Each individual waveform is assigned a segment index (first column) which is used inside the PDW (SEGMENT in PDW payload) to address the respective waveform segment.

All waveforms appended to this list are internally resampled to a common clock rate. A container file is automatically created, which is downloaded to the memory of the coder board. After this, the SMW is ready to receive PDWs with a segment index to select and play a waveform.

In order to reach high PDW streaming rates with ARB segments, the SMW firmware up samples the user waveforms before processing them in hardware to minimize the hardware resampling delay. This in turn leads to a higher memory usage.

The desired ARB streaming rate can be selected in the SMW200A GUI:

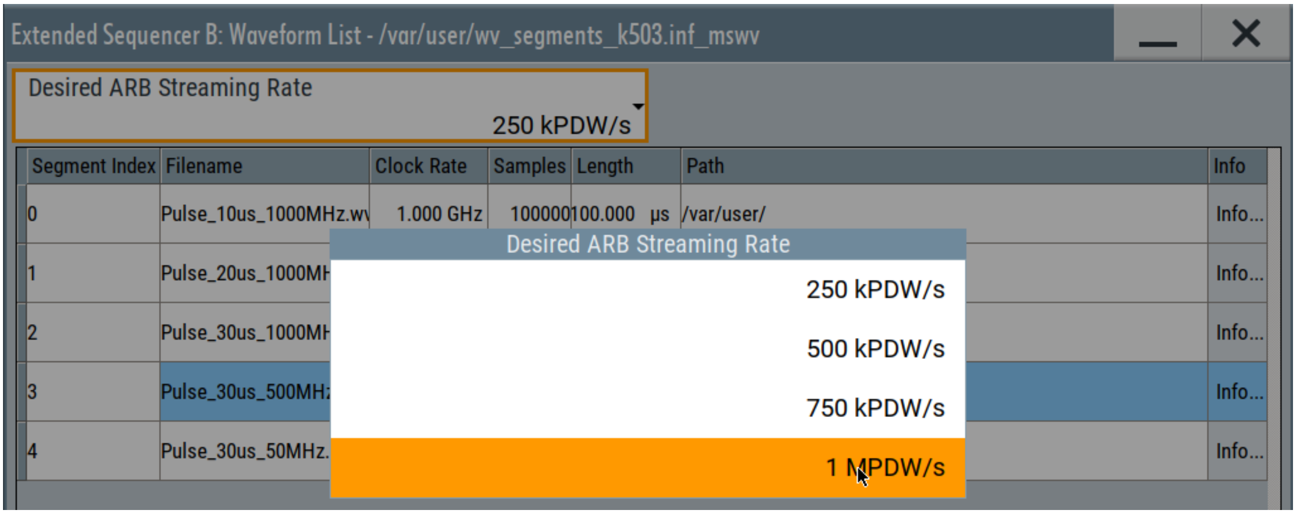


Figure 2: Selection of desired ARB Streaming Rate in SMW200A GUI

The selected streaming rate determines the waveform clock rate after resampling and consequently the memory usage. The following table provides an overview about desired ARB segment streaming rate, waveform clock rate, minimum segment length and as an example minimum memory usage for 1000 segments with a lengths of 100 µs each.

ARB Segment Streaming Rate	Waveform Clock Rate	Minimum segment length[1]	Minimum Memory Usage (1000 segments; 100 µs per segment)[2]
250 kPDW/s	max(37.5 MHz, highest clock rate of loaded segments)	27.3 µs	15 MByte
500 kPDW/s	max(75 MHz, highest clock rate of loaded segments)	13.7 µs	30 MByte
750 kPDW/s	max(300 MHz, highest clock rate of loaded segments)	3.41 µs	120 MByte
1 MPDW/s	2.4 GHz	427 ns	960 Mbyte

Table 21: Overview about relation of max. ARB streaming rate, waveform clock rate, min. segment length and memory usage

The 24 Bit SEGMENT field inside the R&S PDW theoretically allows to address 2^{24} = 16.777.216 individual waveforms. The maximum number of segments with minimum segment size is 2 million.[2]

A mix of real-time PDWs and PDWs which address ARB segments during simulation is supported.

[1] Minimum RAM granularity = 1024 Samples
[2] Max. Memory size = 2 GSamples (requires SMW-K515)

2.3 Timed Control Descriptor Word (TCDW)

By setting the CTRL flag in the xDW flags section, the user can issue commands such as changing the instrument RF frequency and/or level of the signal generator directly from the descriptor word stream or arm the Extended Sequencer, where otherwise a SCPI command would have been necessary.

By embedding the control commands directly in the descriptor word stream, the start of the frequency or level change procedure can be exactly determined by the TOA.

Mechanical step attenuators for level setting are subject to wear and tear when frequently switched.

2.3.1 Timed Control Descriptor Word Data Content

2.3.1.1 Header

The TCDW header section contains the TOA and flags which define the command type.

TCDW header			**48 Bit**	**56 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
TOA	unsigned int	Timestamp relative to scenario start trigger event	44 Bit	52 Bit
PATH	Boolean	Specifies RF path which is affected by TCDW 0 = Path A 1 = Path B	1 Bit	1 Bit
CMD	unsigned int	Specifies command type 0 = Frequency change 1 = Level change 2 = Frequency and level change 3 = Arm Extended Sequencer in Trigger Mode 'Armed Auto' (Stop internal counter and set to zero, xDWs in buffer are still available) 4 = List Mode Frequency Change	3 Bit	3 Bit

Table 22: TCDW Header Structure

2.3.1.2 Flags

The TCDW flags section contains information about the xDW type.

TCDW flags			**16 Bit**	**8 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
CTRL	Boolean	Indicates whether descriptor word is a PDW or a TCDW 0 = PDW 1 = TCDW	1 Bit	1 Bit
RSVD	-	Reserved for future use	15 Bit	7 Bit

Table 23: TCDW Flags Structure

2.3.1.3 Body

The TCDW body section contains values for instrument RF frequency and/or level or list index.

TCDW Body (CMD = 0)			**64 Bit**	**64 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
FVAL	unsigned int	RF frequency setting of signal generator in Hz	40 Bit	40 Bit
STUFFING	-	Stuffing bits	24 Bit	24 Bit

Table 24: TCDW Body Structure for frequency change

<table>
<tr><th colspan="3">TCDW Body (CMD = 1)</th><th>64 Bit</th><th>64 Bit</th></tr>
<tr><th>Parameter</th><th>Data type</th><th>Description</th><th>Basic</th><th>Expert</th></tr>
<tr><td>STUFFING</td><td>-</td><td>Stuffing bits</td><td>40 Bit</td><td>40 Bit</td></tr>
<tr><td>LVAL</td><td>signed fixed point BCD</td><td>RF level setting of signal generator in dBm<table><tr><th>Bit</th><th>0</th><th>1-7</th><th>8-11</th><th>12-15</th><th>16-23</th></tr><tr><td>Value</td><td>Sign
0=pos
1=neg</td><td>Integer part</td><td>Tenth part</td><td>Hundredth part</td><td>Unused (Set to 0)</td></tr></table></td><td>24 Bit</td><td>24 Bit</td></tr>
</table>

Table 25: TCDW Body Structure for level change

<table>
<tr><th colspan="3">TCDW Body (CMD = 2)</th><th>64 Bit</th><th>64 Bit</th></tr>
<tr><th>Parameter</th><th>Data type</th><th>Description</th><th>Basic</th><th>Expert</th></tr>
<tr><td>FVAL</td><td>unsigned int</td><td>RF frequency setting of signal generator in Hz</td><td>40 Bit</td><td>40 Bit</td></tr>
<tr><td>LVAL</td><td>signed fixed point BCD</td><td>RF level setting of signal generator in dBm<table><tr><th>Bit</th><th>0</th><th>1-7</th><th>8-11</th><th>12-15</th><th>16-23</th></tr><tr><td>Value</td><td>Sign
0=pos
1=neg</td><td>Integer part</td><td>Tenth part</td><td>Hundredth part</td><td>Unused (Set to 0)</td></tr></table></td><td>24 Bit</td><td>24 Bit</td></tr>
</table>

Table 26: TCDW Body Structure for frequency and level change

TCDW Body (CMD = 3)			**64 Bit**	**64 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
STUFFING	-	Stuffing bits	40 Bit	40 Bit
STUFFING	-	Stuffing bits	24 Bit	24 Bit

Table 27: TCDW Body Structure for arming

TCDW Body (CMD = 4)			**64 Bit**	**64 Bit**
Parameter	**Data type**	**Description**	**Basic**	**Expert**
FVAL	unsigned int	List index of RF frequency	40 Bit	40 Bit
STUFFING	-	Stuffing bits	24 Bit	24 Bit

Table 28: TCDW Body Structure for list mode frequency change

2.3.2 Timed Control Descriptor Word Bits Allocation

The following tables illustrate the TCDW bit allocation for basic and expert TCDWs.

Header	Flags	Body	Total (bits)
48	16	64	128

Table 29: Bits allocation of basic TCDW

Header	Flags	Body	Total (bits)
56	8	64	128

Table 30: Bits allocation of expert TCDW

2.3.3 Comparison of Basic and Expert Timed Control Descriptor Word Format

The following table illustrates the differences between basic and expert TCDW format. Expert TCDW format is recommended.

	Basic mode	Expert mode
Size	16 Byte	16 Byte
Max. Playtime	~ 2 h	> 520 h (~21.5 days)

Table 31: Comparison of Basic and Expert Timed Control Descriptor Word Format

3 Timing Requirements

3.1 Descriptor Word Processing

A descriptor word arriving at the ADV DATA / CTRL interface is unpacked from the TCP or UDP packet and written to a buffer.

Based on an internal counter that is increased by 1 at a clock rate of 2.4 GHz (resolution of $1/(2.4\cdot10^9)$ s) the first xDW in the buffer is processed. The TOA of a descriptor word is given in clock ticks.

In each clock cycle the current counter state t is compared to the TOA of the first descriptor word in the buffer. The following applies:

- counter state = TOA of the next xDW in buffer: xDW is processed. Example: t =10 and TOA = 10
- counter state < TOA of the next xDW in buffer: do nothing. Example: t =10 and TOA = 140
- counter state > TOA of the next xDW in buffer: xDW is ignored and removed from the buffer. Example: t =10 and TOA = 5.

Info: Current counter state and buffer level (filled/remain) are indicated in the Statistics tab of the Extended Sequencer.

If the next descriptor word in the buffer is a timed control descriptor word the instrument RF settings specified in the descriptor word are applied or the Extended Sequencer is re-armed at the specified TOA.

If a pre-calculated ARB segment is addressed, the respective segment is loaded from the ARB memory (Max. memory size: 2 GSamples (requires SMW-K515)).

In case the descriptor word contains a signal description, the corresponding I/Q signal is generated in real-time.

A multiplexer interleaves pre-calculated ARB segments and real-time signals. Frequency, phase and/or level offsets are applied as specified in the xDW.

If applicable, the pulse edges are shaped according to the edge parameters in the PDW.

*Info: Starting with the SMW FW 5.30.175.14, the xDW buffer can be cleared using the SCPI command SOURce:*BB:ESEQuencer:RTCI:STReam:BUFReset

Please note that after clearing the buffer, up to 16 PDWs (the first ones in the FIFO pipeline) may remain in the pipeline and will be played once the system time reaches their TOAs.

> *The command resets the statistics parameters "Write count", "Read count", "Buffer filled", "Buffer remain".*
>
> *The command does not reset the parameters "System Time", "Executed", "Dropped".*

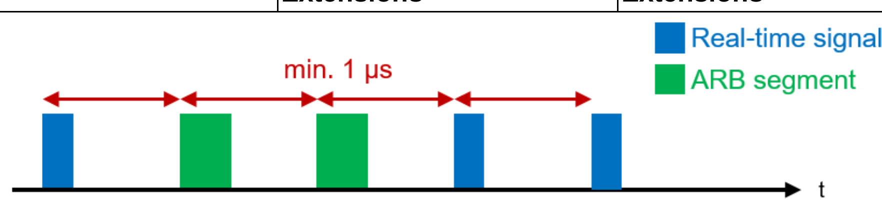
3.2 Minimum TOA difference of consecutive PDWs

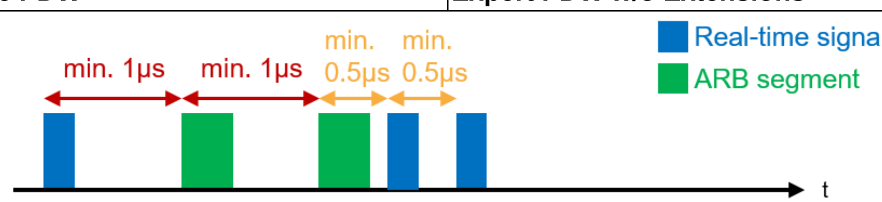
The minimum TOA difference applies to the current PDW in relation to the previous PDW.

Installed Option	Basic PDW	Expert PDW w/o Extensions	Expert PDW with Extensions
SMW-K503	1.0 µs	1.0 µs	1.0 µs
SMW-K504 • Real-time signals • ARB segments	 0.5 µs 1.0 µs	 0.5 µs 1.0 µs	 1.0 µs 1.0 µs

Table 32: Min. TOA difference of consecutive PDWs

The following diagram explains the different restrictions in basic and expert mode for all possible combinations of real-time signals and ARB segments.

<table>
<tr><th>Installed Options</th><th>Basic PDW</th><th>Expert PDW w/o Extensions</th><th>Expert PDW with Extensions</th></tr>
<tr><td>SMW-K503
• Real-time signals
• ARB segments</td><td colspan="3">
</td></tr>
</table>

<table>
<tr><th>Installed Options</th><th>Basic PDW</th><th>Expert PDW w/o Extensions</th></tr>
<tr><td>SMW-K504
• Real-time signals
• ARB segments</td><td colspan="2">
</td></tr>
</table>

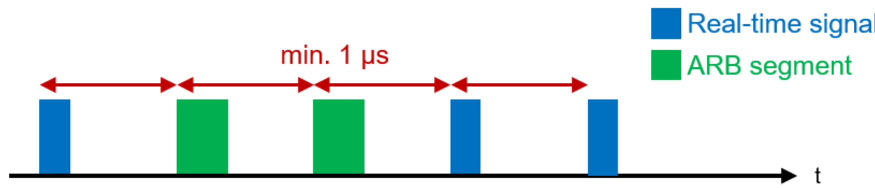
Installed Option	Expert PDW with Extensions
SMW-K504 • Real-time signals • ARB segments	

Table 33: Illustration of min. TOA difference of consecutive PDWs

3.3 Dropping of Descriptor Words

There are three reasons for a descriptor word getting dropped:

1. **Internal counter state > TOA of xDW**

   If a xDW arrives too late at the ADV DATA/CTRL interface, it is dropped.

2. **Two xDWs with identical TOA are received at ADV DATA/CTRL interface**

   In this case, the xDW that is received first (xDW_1) is processed, the second xDW (xDW_2) is dropped

	TOA	Received	Processing
xDW_1	10	first	executed
xDW_2	10	second	dropped

Table 34: Pulse dropping: PDWs with identical TOA

3. **TOA of successive xDW < (TOA of current xDW + (TON + RISE_TIME + FALL_TIME) of current xDW)**

   or

   TOA of successive xDW < (TOA of current xDW + duration of segment addressed by current xDW)

   In this case, playback of the successive xDW_2 starts, before the previous pulse (described in xDW_1) is finished. The signal described in xDW_1, even though not finished, is aborted at the TOA of xDW_2 and xDW_2 is executed

	TOA	TON	Aborted at counter state	Processing
xDW_1	10	10	15	aborted
xDW_2	15	10	-	executed

Table 35: Pulse dropping: TOA of successive xDW < TOA of current xDW + TON of current xDW

This is also valid in case xDW_1 addresses a pre-calculated ARB segment. In this case, TON can be replaced by the segment playtime in the formula above.

Note: xDWs have to be sent in the correct order of their TOA. There is no sorting of xDWs according to TOA in the SMW.

Info: The number of executed and dropped xDWs since the last counter reset are indicated in the Statistics tab of the Extended Sequencer in the SMW GUI.

4 Network Interface Properties

Connector designation	ADV DATA/CTRL
Mechanical connector	RJ45
Supported data rates	10/100/1000 Mbit/s
Supported network protocols	TCP, UDP[3,4]

Table 36: Network interface properties

Ethernet

The Maximum Transmission Unit (MTU) size of the Ethernet controller must be set to 1500 (Bytes). This is the default value for most network interface controllers.

TCP

When using TCP to send xDWs to the SMW, it is highly recommended to switch off Nagle´s algorithm for the sender, since this algorithm interacts badly with TCP delayed acknowledgments. This can be done by enabling the socket option TCP_NODELAY.

UDP

When using UDP to send PDWs to the SMW, it must be considered that the sender application is responsible for fragmenting the PDW data into UDP datagrams of an appropriate size.

To improve real-time capability, the Address Resolution Protocol (ARP) table of the sender should be up-to-date before starting to send UDP datagrams. Otherwise, the first datagram will be possibly sent delayed, since the sender has to send an ARP request and wait for the response to update his ARP table before. This can be done by sending empty UDP datagrams to the SMW ADV DATA/CTRL interface. The procedure has to be repeated depending on the ARP table timeout of the sender.

Note: The ADV DATA/CTRL interface works reliable with TCP- and UDP-packets with a minimum packet (segment) size of 640 Byte. Sending smaller packets can cause packet loss. It is recommended to send packets of twenty PDWs. When sending single PDWs, the packet size has to be artificially increased by sending copies of the PDW and setting the IGNORE_PDW flag in the PDW copies.

Using TCP, packet losses are corrected through re-transmissions. If timing restrictions allow for retransmissions, sending PDW copies is not required.

[3] Can be selected in the SMW200A GUI
[4] Packet overhead: TCP: 54 Byte, UDP: 42 Byte

5 Appendix

A Examples

A.1 Basic PDW

Parameter	Parameter Value	Meaning	Binary Data
PDW Header			
TOA	40000	100 µs	0x0000003a9800
SEG	0	False	
RSVD	0	-	
PDW Flags			
CTRL	0	False	0x01
RSVD	0	-	
PHASE_MOD	0	Absolute	
IGNORE_PDW	0	False	
M4	0	-	
M3	0	Marker 3 off	
M2	0	Marker 2 off	
M1	1	Marker 1 on	
PDW Body			
FRQ	894784854	-500 MHz	0xcaaaaaaa40261555
LEV	6422	6 dB	
PHS	461	30°	
PDW Payload			
MOD	linear FMCW = 1	linear FMCW	0x100000005dc00001234882ef6b75000000
RSVD	0	-	
TON	24000	10 µs	
FREQ_INC	320269318056821	1 GHz / 23999 Sa	
RSVD	0	-	

Table 37: Basic PDW example to generate linear FMCW real-time signal

PDW:

0x0000003a 0x980001ca 0xaaaaaa40 0x26155510 0x0000005d 0xc0000123 0x4882ef6b 0x75000000

A.2 Basic TCDW

Parameter	Parameter Value	Meaning	Bytes
PDW Header			
TOA	40000	100 µs	0x0000003a9802
PATH	0	Path A	
CMD	2	Frequency and level change	
PDW Flags			
CTRL	1	True	0x8000
RSVD	0	-	
PDW Body			
FVAL	10900000000	10.9 GHz	0x0289b0cd008d0000
LVAL	7536640	-13 dBm	

Table 38: Basic TCDW example to switch instrument RF level and frequency

TCDW:

0x0000003a 0x98028000 0x0289b0cd 0x008d0000

A.3 Expert PDW

Parameter	Parameter Value	Meaning	Bytes
PDW Header			
TOA	120000	50 µs	0x000000001d4c04
SEG	0	False	
USE_EXTENSION	1	True	
PARAMS	0	No params	
PDW Flags			
CTRL	0	False	0x01
RSVD	0	-	
PHASE_MOD	0	Absolute	
IGNORE_PDW	0	False	
M4	0	-	
M3	0	Marker 3 off	
M2	0	Marker 2 off	
M1	1	Marker 1 on	
PDW Body			
FRQ	-223696214	-125 MHz	0xe55555555a9d5555
LEV	23197	3 dB	
PHS	21845	120°	
PDW Payload			
MOD	2	triangular chirp	0x2000bb8000003803bb0c6860
RSVD	0	-	
TON	48000	20 µs	
FREQ_INC	1588674209888	500 MHz / 62399 Sa	
PDW Extension Flags			
FIELD_1_TYPE	1	Edge field	0x2800
FIELD_2_TYPE	2	Burst field	
FIELD_3_TYPE	0	Unused	
RSVD	0	-	
PDW Extension Fields			
EDGE_TYPE	0	Linear	0x000708001c20
MULTIPLIER	0	x1	
RISE_TIME	7200	3 µs	
FALL_TIME	7200	3 µs	
BURST_PRI	192000	80 µs	0x0002ee000009
BURST_ADD_PULSES	9	9	
STUFFING	0	stuffing	0x000000000000

Table 39: Expert PDW example to generate multiple triangular FMCW real-time signals with linear edges

PDW:

0x00000000 0x1d4c0441 0xf2aaaaaa 0x5a9d5555 0x2000bb80 0x00003803 0xbb0c6860 0x28000007 0x08001c20 0x0002ee00 0x00090000 0x00000000

A.4 Expert TCDW

Parameter	Parameter Value	Meaning	Bytes
PDW Header			
TOA	240000	100 µs	0x000000003a9802
PATH	0	Path A	
CMD	2	Frequency and level change	
PDW Flags			
CTRL	1	True	0x80
RSVD	0	-	
PDW Body			
FVAL	10900000000	10.9 GHz	0x0289b0cd008d0000
LVAL	-7536640	-13 dBm	

Table 40: Expert TCDW example to switch instrument RF level and frequency

TCDW:

0x00000000 0x3a980280 0x0289b0cd 0x008d0000

Rohde & Schwarz

The Rohde & Schwarz electronics group offers innovative solutions in the following business fields: test and measurement, broadcast and media, secure communications, cybersecurity, monitoring and network testing. Founded more than 80 years ago, the independent company which is headquartered in Munich, Germany, has an extensive sales and service network with locations in more than 70 countries.

www.rohde-schwarz.com

Certified Quality Management
ISO 9001

Rohde & Schwarz training

http://www.training.rohde-schwarz.com/

Rohde & Schwarz customer support

www.rohde-schwarz.com/support

PAD-T-M: 3572.7186.00/01.00/EN

ICD | Version 2.4 | 02.2025
Interface Control Document | R&S®SMW-K503/-K504
Data without tolerance limits is not binding | Subject to change