

# R&S®SMM-K507

## ARB Ethernet Upload

### User Manual



1179571802  
Version 03

**ROHDE & SCHWARZ**  
Make ideas real



This document describes the following software options:

- R&S®SMM-K507 ARB Ethernet Upload (1441.0934.xx)

© 2023 Rohde & Schwarz GmbH & Co. KG  
Muehldorfstr. 15, 81671 Muenchen, Germany  
Phone: +49 89 41 29 - 0

Email: [info@rohde-schwarz.com](mailto:info@rohde-schwarz.com)

Internet: [www.rohde-schwarz.com](http://www.rohde-schwarz.com)

Subject to change – data without tolerance limits is not binding.

R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG.

All other trademarks are the properties of their respective owners.

1179.5718.02 | Version 03 | R&S®SMM-K507

The following abbreviations are used throughout this manual: R&S®SMM100A is abbreviated as R&S SMM.

# Contents

<b>1</b>	<b>Welcome to the ARB ethernet upload option.....</b>	<b>5</b>
1.1	Key features.....	5
1.2	Accessing the ARB ethernet upload dialog.....	6
1.3	What's New.....	6
1.4	Documentation overview.....	6
1.4.1	Getting started manual.....	6
1.4.2	User manuals and help.....	7
1.4.3	Service manual.....	7
1.4.4	Instrument security procedures.....	7
1.4.5	Printed safety instructions.....	7
1.4.6	Data sheets and brochures.....	7
1.4.7	Release notes and open source acknowledgment (OSA).....	8
1.4.8	Application notes, application cards, white papers, etc.....	8
1.4.9	Videos.....	8
1.5	Scope.....	9
1.6	Notes on screenshots.....	9
<b>2</b>	<b>About the ARB ethernet upload option.....</b>	<b>10</b>
2.1	Required options.....	10
2.2	About ARB ethernet upload.....	10
<b>3</b>	<b>Uploading waveform files.....</b>	<b>13</b>
3.1	Connecting the R&S SMM100A.....	13
3.2	Configuring the host PC.....	14
3.3	Configuring the R&S SMM100A.....	16
3.4	Activating and monitoring waveform upload.....	16
<b>4</b>	<b>ARB ethernet upload configuration and settings.....</b>	<b>19</b>
4.1	General settings.....	19
4.2	Statistics.....	21
4.3	Local QSFP network settings.....	23
<b>5</b>	<b>Signal generation control.....</b>	<b>26</b>
<b>6</b>	<b>Remote control commands.....</b>	<b>27</b>

6.1	General commands.....	28
6.2	Statistics commands.....	31
6.3	QSFP+ network commands.....	35
6.4	Trigger, marker and clock commands.....	38
	<b>Annex.....</b>	<b>39</b>
<b>A</b>	<b>I/Q over Ethernet (QSFP+) protocol.....</b>	<b>39</b>
<b>A.1</b>	<b>UDP/IP protocol stack.....</b>	<b>39</b>
A.1.1	Main specifications.....	39
A.1.2	OSI model.....	40
A.1.3	I/Q frame and UDP subheader.....	40
A.1.4	ACK frame.....	41
A.1.5	Control and data frame types.....	41
<b>A.2</b>	<b>Waveform file transfer.....</b>	<b>43</b>
<b>A.3</b>	<b>ARB upload flow diagram.....</b>	<b>44</b>
<b>A.4</b>	<b>Waveform upload command sequences.....</b>	<b>45</b>
<b>A.5</b>	<b>Using a python script for waveform data transfer.....</b>	<b>46</b>
	<b>Glossary: Abbreviations.....</b>	<b>51</b>
	<b>List of commands.....</b>	<b>52</b>
	<b>Index.....</b>	<b>53</b>

# 1 Welcome to the ARB ethernet upload option

The R&S SMM-K507 option ARB ethernet upload is a firmware application that allows high-speed streaming of waveform files via the QSFP+ interface of the R&S SMM100A.

This user manual contains a description of the functionality that the application provides, including remote control operation.

All functions not discussed in this manual are the same as in the base unit and are described in the R&S SMM100A user manual. The latest version is available at:

[www.rohde-schwarz.com/manual/SMM100A](http://www.rohde-schwarz.com/manual/SMM100A)

## Installation

You can find detailed installation instructions in the delivery of the option or in the R&S SMM100A service manual.

## 1.1 Key features

- Ultra-fast waveform I/Q data transfer from a host PC to the R&S SMM100A:
  - 10 Gbit Ethernet: Up to 100 times faster data transfer at up to 9 Gbit/s transfer rate
  - 40 Gbit Ethernet: Up to 400 times faster data transfer at up to 36 Gbit/s transfer rate
- Automatic upload and play of waveform files (\*.wv)
- Low latency:
  - No intermediate storage on the HDD required. The IQ samples are directly loaded into the ARB memory.
  - No ARB reconfiguration required when the meta data is already transmitted and not changed.
  - Further time saving when working with smaller waveforms.
- Easy integration:
  - Standard data transfer via UDP/IP
  - Flexible connection between host PC and R&S SMM100A:  
10G electrical connection (RJ45), 10G optical connection (SFP+) or 40G optical connection (QSFP+)
- Monitoring upload characteristics of the upload status and waveform count
- Monitoring upload statistics including receive (Rx) and transmit (Tx) parameters and error statistics:
  - Rx: Upload segments, frames and data
  - Tx: Reply frames
  - Occurred errors during upload

## 1.2 Accessing the ARB ethernet upload dialog

### To open the dialog with ARB ethernet upload settings

1. In the block diagram of the R&S SMM100A, select "Baseband > ARB".
2. Select "Mode > Ethernet Upload".

The dialog provides general settings for ARB Ethernet upload.

The signal generation is not started immediately. To start signal generation with the default settings, select "State > On".

## 1.3 What's New

This manual describes firmware version FW 5.20.043.xx and later of the R&S®SMM100A.

Compared to the previous version, it provides the new features listed below:

- ARB ethernet upload works, if the waveform data is equal or lower than the ARB memory, see ["To prepare waveform data"](#) on page 15.
- Waveform status replaces the former ARB ethernet status ("Status"), see ["Waveform Status"](#) on page 20.
- Python script update for waveform data transfer, see [Example "To transfer waveform data"](#) on page 47.
- Editorial changes.

## 1.4 Documentation overview

This section provides an overview of the R&S SMM100A user documentation. Unless specified otherwise, you find the documents at:

[www.rohde-schwarz.com/manual/smm100a](http://www.rohde-schwarz.com/manual/smm100a)

### 1.4.1 Getting started manual

Introduces the R&S SMM100A and describes how to set up and start working with the product. Includes basic operations, typical measurement examples, and general information, e.g. safety instructions, etc. A printed version is delivered with the instrument.

## 1.4.2 User manuals and help

Separate manuals for the base unit and the software options are provided for download:

- **Base unit manual**  
Contains the description of all instrument modes and functions. It also provides an introduction to remote control, a complete description of the remote control commands with programming examples, and information on maintenance, instrument interfaces and error messages. Includes the contents of the getting started manual.
- **Software option manual**  
Contains the description of the specific functions of an option. Basic information on operating the R&S SMM100A is not included.

The contents of the user manuals are available as help in the R&S SMM100A. The help offers quick, context-sensitive access to the complete information for the base unit and the software options.

All user manuals are also available for download or for immediate display on the Internet.

## 1.4.3 Service manual

Describes the performance test for checking compliance with rated specifications, firmware update, troubleshooting, adjustments, installing options and maintenance.

The service manual is available for registered users on the global Rohde & Schwarz information system (GLORIS):

<https://gloris.rohde-schwarz.com>

## 1.4.4 Instrument security procedures

Deals with security issues when working with the R&S SMM100A in secure areas. It is available for download on the internet.

## 1.4.5 Printed safety instructions

Provides safety information in many languages. The printed document is delivered with the product.

## 1.4.6 Data sheets and brochures

The data sheet contains the technical specifications of the R&S SMM100A. It also lists the options and their order numbers and optional accessories.

The brochure provides an overview of the instrument and deals with the specific characteristics.

See [www.rohde-schwarz.com/brochure-datasheet/smm100a](http://www.rohde-schwarz.com/brochure-datasheet/smm100a)

### 1.4.7 Release notes and open source acknowledgment (OSA)

The release notes list new features, improvements and known issues of the current firmware version, and describe the firmware installation.

The software makes use of several valuable open source software packages. An open-source acknowledgment document provides verbatim license texts of the used open source software.

See [www.rohde-schwarz.com/firmware/smm100a](http://www.rohde-schwarz.com/firmware/smm100a)

### 1.4.8 Application notes, application cards, white papers, etc.

These documents deal with special applications or background information on particular topics.

See [www.rohde-schwarz.com/application/smm100a](http://www.rohde-schwarz.com/application/smm100a)

### 1.4.9 Videos

Find various videos on Rohde & Schwarz products and test and measurement topics on YouTube: <https://www.youtube.com/@RohdeundSchwarz>



On the menu bar, search for your product to find related videos.



*Figure 1-1: Product search on YouTube*



## 1.5 Scope



Tasks (in manual or remote operation) that are also performed in the base unit in the same way are not described here.

In particular, it includes:

- Managing settings and data lists, like saving and loading settings, creating and accessing data lists, or accessing files in a particular directory.
- Information on regular trigger, marker and clock signals and filter settings, if appropriate.
- General instrument configuration, such as checking the system configuration, configuring networks and remote operation
- Using the common status registers

For a description of such tasks, see the R&S SMM100A user manual.

## 1.6 Notes on screenshots

When describing the functions of the product, we use sample screenshots. These screenshots are meant to illustrate as many as possible of the provided functions and possible interdependencies between parameters. The shown values may not represent realistic usage scenarios.

The screenshots usually show a fully equipped product, that is: with all options installed. Thus, some functions shown in the screenshots may not be available in your particular product configuration.

## 2 About the ARB ethernet upload option

### 2.1 Required options

The basic equipment layout for ARB ethernet upload includes the options:

- Base unit
- Wideband baseband generator (R&S SMM-B9)
- ARB ethernet upload (R&S SMM-K507)
- Optional, ARB memory extension to 512 MSample (R&S SMM-K511)
- Optional, ARB memory extension to 1 GSample (R&S SMM-K512)  
Requires R&S SMM-K511.
- Optional, ARB memory extension to 2 GSample (R&S SMM-K513)  
Requires R&S SMM-K511 and R&S SMM-K512.

For more information, see data sheet.

### 2.2 About ARB ethernet upload

Equipped with option R&S SMM-K507 ARB ethernet upload, the R&S SMM100A can upload waveforms significantly faster to its ARB memory. Data rates of 10 Gbit/s and 40 Gbit/s at the QSFP+ connectors "HS Dig I/Q" allow much shorter waveform loading times. These loading times can be up to 100 times shorter compared to waveform upload via the USB or the other I/Q connectors "I/Q" and "Dig I/Q".

#### Test setup

Figure 2-1 shows a typical test setup including a personal computer (PC) as waveform source and the R&S SMM100A that is processing the waveform I/Q data.

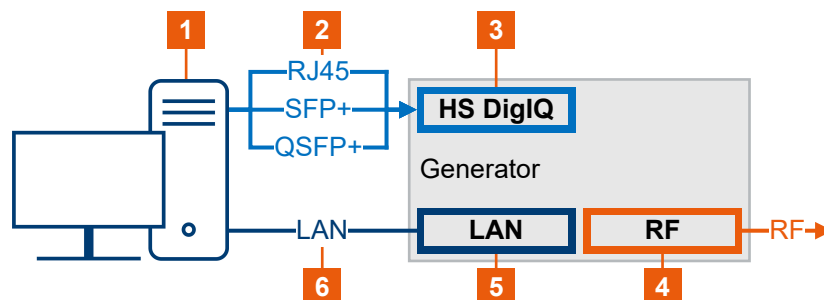


Figure 2-1: ARB Ethernet upload test setup

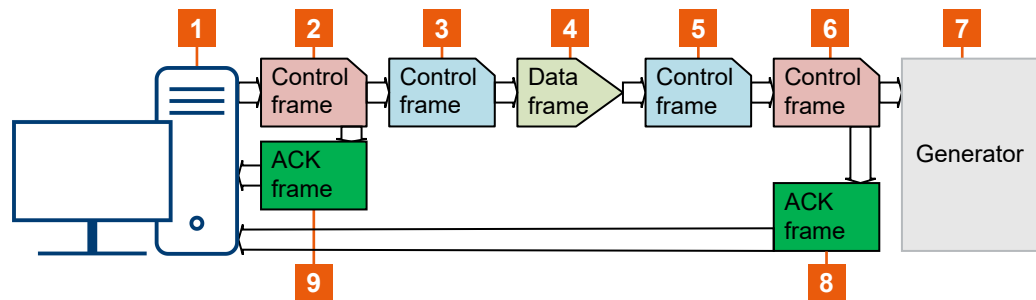
- 1 = PC as waveform data source with one or more network interface cards  
 2 = Ethernet connection using RJ45, SFP+ or QSFP+ cables  
 3 = R&S SMM100A waveform data input: "HS Dig I/Q"

- 4 = R&S SMM100A output: "RF"
- 5 = R&S SMM100A control: "LAN"
- 6 = Optional LAN connection for remote control operation

### Waveform transfer and waveform upload

For waveform data transfer, the PC is equipped with an application that supports and executes waveform data transmission. This application can be a driver or a client software that supports the User Datagram Protocol (UDP IPv4, see [Chapter A, "I/Q over Ethernet \(QSFP+\) protocol"](#), on page 39).

[Figure 2-2](#) shows the different frames that are exchanged between R&S SMM100A and the PC.



**Figure 2-2: Waveform data transfer from PC to R&S SMM100A**

- 1 = PC as waveform data source
- 2 = Control frame: APPL command: restart ARB
- 3 = Control frame: End of waveform data transfer
- 4 = Data frames: Waveform data as UDP datagrams/frames
- 5 = Control frame: start of waveform data transfer
- 6 = Control frame: APPL commands: stop ARB, set waveform parameters (clock, length, etc.) in waveform tags/header
- 7 = R&S SMM100A as waveform data sink
- 8, 9 = Acknowledgement frames

[Table 2-1](#) describes the corresponding sequence of frames sent between the PC to the R&S SMM100A. The application control frames for sequence event 1 and sequence event 6 are mandatory.

**Table 2-1: Waveform data transfer: Frame sequences**

Sequence	Frame type	Description	Label 2-2
1	Control frame	PC: Sends application control frames to stop ARB, set waveform parameters (clock, length, etc.) as waveform tags or waveform header data. R&S SMM100A: Restarts the ARB application.	6
2	ACK frame	R&S SMM100A reply frame.	8
3	Control frame	PC: Defines the start of waveform data transfer.	5
4	Data frame	PC: Transfers waveform data as UDP datagrams/frames. R&S SMM100A: Receives waveform data as UDP datagrams/frames.	4

Sequence	Frame type	Description	Label 2-2
5	Control frame	PC: Defines the end of waveform data transfer.	3
6	Control frame	PC: Sends application control frames to restart ARB. R&S SMM100A: Restarts the ARB application, depending on ARB trigger.	2
7	ACK frame	R&S SMM100A reply frame.	9

## 3 Uploading waveform files

This chapter provides an overview on how to prepare, configure and operate the R&S SMM100A for waveform upload via the QSFP+ interface.

- [Connecting the R&S SMM100A](#)..... 13
- [Configuring the host PC](#)..... 14
- [Configuring the R&S SMM100A](#)..... 16
- [Activating and monitoring waveform upload](#)..... 16

### 3.1 Connecting the R&S SMM100A

This chapter provides step-by-step descriptions for connecting the R&S SMM100A with the host PC. The step-by-step descriptions cover the following topics:

- QSFP+ interface: "[To connect to HS Dig I/Q](#)" on page 13
- LAN interface: "[To connect to LAN](#)" on page 14

#### To connect to HS Dig I/Q

At the host PC, you can connect up to three interface types of the Ethernet card to the QSFP+ interface (HS Dig I/Q) of the R&S SMM100A.

1. For connection, use high-quality cables and adapters depending on the interface that you are using. The table [Table 3-1](#) lists cables and adapters required for each interface type.

**Table 3-1: Cables and adapters for the QSFP+ interface**

Interface type	Host PC interface	Cable	Adapter
Electrical (10G)	10 Gbit Ethernet card	RJ45 CAT7	RJ45 to SFP+ and SFP+ to QSFP+
Optical (10G)	10 Gbit Ethernet card	SFP+ cable	SFP+ to QSFP+
Optical (40G)	40 Gbit Ethernet card	QSFP+ cable (e.g. R&S DIGIQ-HS)	-

2. Hold the QSFP+ plug of the cable or adapter by its panes.
3. Turn the QSFP+ cable or adapter, so that the release tab shows upwards.
4. Insert and push the QSFP+ plug into the QSFP+ cage.

For a detailed description, see chapter "Connecting to HS Dig I/Q" in the R&S SMM100A user manual.

### To connect to LAN

Optionally, connect the host PC and R&S SMM100A to LAN for remote control operation via the SCPI interface. Use this connection to configure RF signal settings of the R&S SMM100A like frequency and power level.

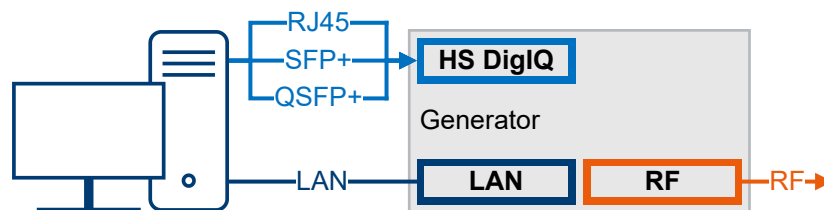
- ▶ Connect the LAN socket via an RJ-45 cable to the LAN.

For a detailed description, see chapter "Connecting to LAN" in the R&S SMM100A user manual.

Table 3-2 and Figure 3-1 provide an overview on connections and the test setup.

**Table 3-2: Host PC and R&S SMM100A connections**

Signal	Host PC	R&S SMM100A
Data	Electrical (10G): RJ45 Optical (10G): SFP+ Optical (40G): QSFP+	HS Dig I/Q
Control (SCPI)	LAN	LAN



**Figure 3-1: Test setup with host PC and R&S SMM100A**

For a detailed description of the test setup, see "Test setup" on page 10.

## 3.2 Configuring the host PC

This chapter provides step-by-step instructions to configure the host PC for waveform transfer. Before configuration, connect the host PC to the R&S SMM100A as described in Chapter 3.1, "Connecting the R&S SMM100A", on page 13. The host PC operates with Windows operating system. The step-by-step instructions cover the following topics:

- "To configure network settings" on page 14
- "To prepare waveform data" on page 15
- "To transfer waveform data" on page 15

### To configure network settings

For correct waveform data transfer from the host PC to the R&S SMM100A, make sure to configure the host PC as follows:

1. Install the latest Windows driver version of the controller of the Ethernet card.

2. Configure the Ethernet network card, e.g., set the correct mode of operation for data transfer.  
See the documentation of the manufacturer of the network card.
3. Configure the Windows network settings of the Ethernet controller: IP address settings of the PC and IP version IPv4
4. Optionally, adjust Windows Firewall settings to allow connection to the R&S SMM100A.

#### To prepare waveform data

Before transferring waveform data, you need to prepare this data first. Also, transfer requires unencrypted waveforms.

1. Add tags to specify characteristics of the waveform.  
The R&S SMM100A uses a simple tag-oriented format, see chapter "Tags for waveforms, data and control lists" in the R&S SMM100A user manual.
2. Check that the size of your waveform data is equal or lower than the ARB memory of the R&S SMM100A.  
Playing files that are larger than the ARB memory is not possible. If necessary, extend the ARB memory. See [Chapter 2.1, "Required options"](#), on page 10.



Uploading waveform data from hard disk drives (HDD) reduces the transfer speed.

#### To transfer waveform data

1. At the host PC, install a suitable application that supports and executes waveform data transmission. This application can be a driver or a client software.  
Make sure that this application supports User Datagram Protocol (UDP IPv4) and protocol stack as specified in [Chapter A, "I/Q over Ethernet \(QSFP+\) protocol"](#), on page 39.
2. At the R&S SMM100A, enable ARB Ethernet upload.  
See ["To activate waveform upload"](#) on page 17.
3. Check that the R&S SMM100A connects to the QSFP+ network.  
See ["To connect to the QSFP+ network"](#) on page 16.
4. At the host PC, execute the application.  
For example, you can run a python script for small waveform files.
  - a) Install the latest version of python available at:  
<https://www.python.org/downloads/>
  - b) Execute the python script example.  
See [Example "To transfer waveform data"](#) on page 47.

### 3.3 Configuring the R&S SMM100A

This chapter provides step-by-step instructions to configure ARB Ethernet upload at the R&S SMM100A. These instructions cover the following topics:

- ["To enable ARB Ethernet upload"](#) on page 16
- ["To connect to the QSFP+ network"](#) on page 16

#### To enable ARB Ethernet upload

1. Select "Baseband > ARB".
2. Select "General > Mode > Ethernet Upload".
3. Select the "Ethernet Mode" compatible with the network card of the host PC:
  - For 10 Gbit (optical or electrical), select "Ethernet Mode" > "10 GbE".
  - For 40 Gbit (optical), select "Ethernet Mode" > "40 GbE".

#### To connect to the QSFP+ network

The R&S SMM100A is configured for ARB Ethernet upload as described in ["To enable ARB Ethernet upload"](#) on page 16. The ARB state is deactivated ("State" > "Off").

1. Select "Local QSFP Network Settings".
2. Configure QSFP network parameters as needed. In particular, specify "IP Address", "Subnet Mask", "IP Address" and "UDP Port".

The figure below shows a valid configuration expecting a 40 Gbit Ethernet connection between host PC and an R&S SMW.

Configuration of the QSFP+ network settings of the R&S SMM100A is analogous.

The screenshot shows the "Local QSFP Network" configuration window. It is divided into two sections: QSFP 1 and QSFP 2.

Section	Parameter	Value
QSFP 1	Network Status	Connected
	Running Application	ARB Upload (40GbE)
QSFP 2	Board Name	SMW200A-999001-HS-DIG-IQ1
	Board Address	
	Address Mode	Static
	IP Address	192.168.179.14
	Subnet Mask	255.255.255.0
	MAC Address	00 90 B8 21 DD 82
	UDP Port	49 152

### 3.4 Activating and monitoring waveform upload

This chapter provides step-by-step instructions for waveform upload at the R&S SMM100A. These instructions cover the following topics:

- ["To activate waveform upload"](#) on page 17



- ["To monitor waveform upload characteristics"](#) on page 17

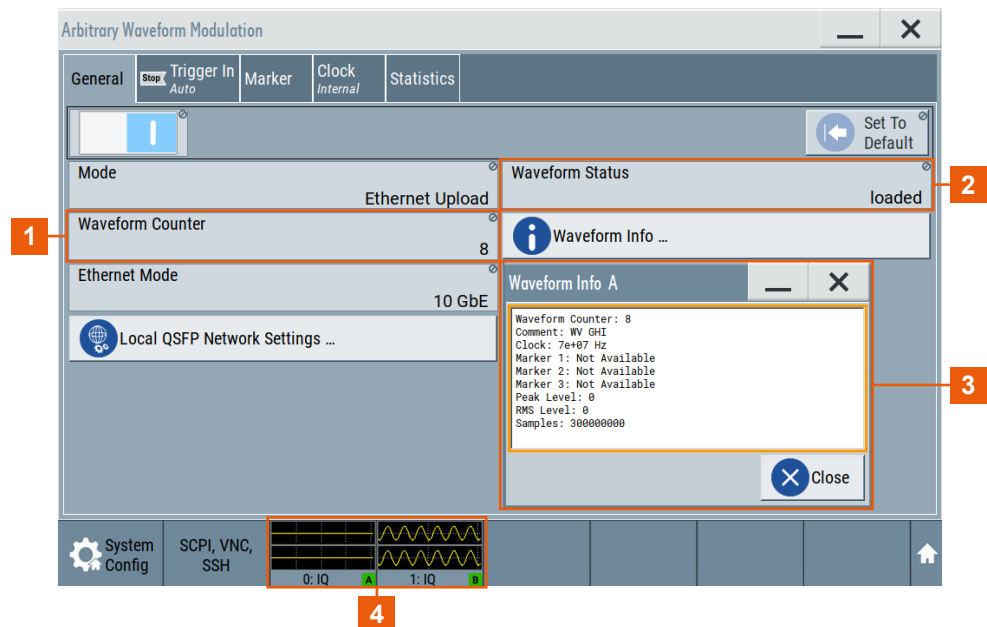
### To activate waveform upload

1. Enable ARB Ethernet upload.  
See ["To enable ARB Ethernet upload"](#) on page 16.
2. Optionally, configure QSFP+ network settings.  
See ["To connect to the QSFP+ network"](#) on page 16.
3. Select ARB "State" > "On".

### To monitor waveform upload characteristics

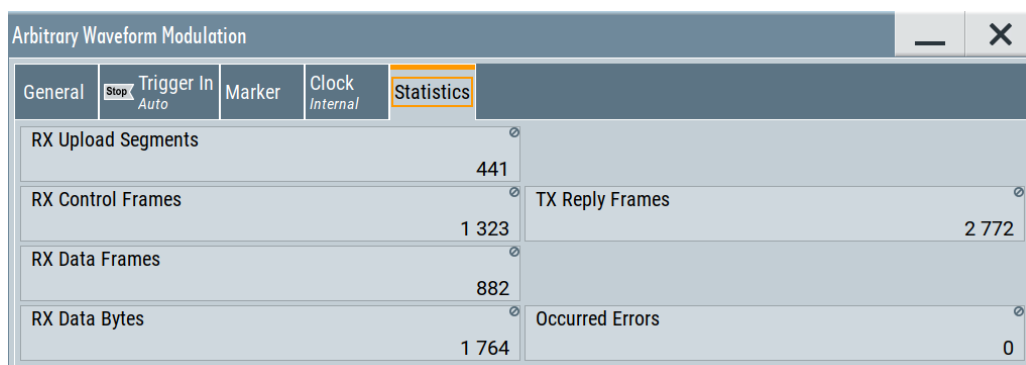
The R&S SMM100A runs with activated waveform upload as described in ["To activate waveform upload"](#) on page 17.

1. In the "General" tab, monitor the status, waveform count and additional information on the currently played waveform.



- 1 = Number of uploaded and played waveforms
- 2 = ARB upload status: ARB is running
- 3 = Waveform info including waveform tags and waveform header data
- 4 = Baseband signal graphical display: First path, no baseband signal. Second path, present baseband signal from waveform I/Q data.

2. In the "Statistics" tab, monitor the waveform upload data and frame characteristics.



Arbitrary Waveform Modulation				
General	Trigger In <i>Auto</i>	Marker	Clock <i>Internal</i>	Statistics
RX Upload Segments	441			
RX Control Frames	1 323			TX Reply Frames 2 772
RX Data Frames	882			
RX Data Bytes	1 764			Occurred Errors 0

## 4 ARB ethernet upload configuration and settings

Access:

1. Select "Baseband" > "ARB".
2. Select "General" > "Mode" > "Ethernet Upload".

The dialog provides settings to configure settings and monitor characteristics of the ARB Ethernet upload.

The remote commands required to define these settings are described in [Chapter 6, "Remote control commands"](#), on page 27.

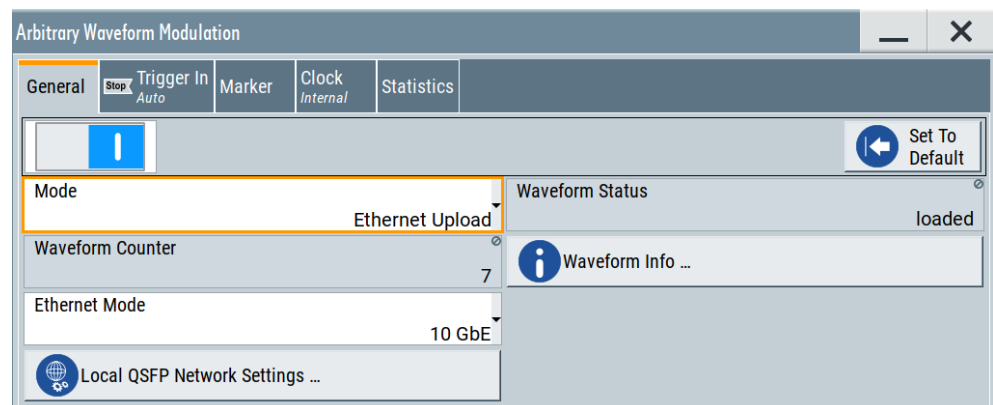
**Settings:**

- [General settings](#)..... 19
- [Statistics](#)..... 21
- [Local QSFP network settings](#)..... 23

### 4.1 General settings

Access:

1. Select "Baseband" > "ARB".
2. Select "General" > "Mode" > "Ethernet Upload".



This tab provides general ARB Ethernet upload settings including setting the application to its default configuration.

**Settings:**

State.....	20
Set To Default.....	20
Mode.....	20
Waveform Status.....	20
Waveform Counter.....	20
Ethernet Mode.....	21
Waveform Info.....	21
Local QSFP Network Settings.....	21

**State**

Activates ARB modulation. Switching on the ARB generator turns off all the other digital standards and digital modulation types in the same signal path.

For more information, see chapter "ARB general settings" in the R&S SMM100A user manual.

**Set To Default**

Calls the default settings.

For more information, see chapter "ARB general settings" in the R&S SMM100A user manual.

**Mode**

Sets the mode of the ARB application.

"Standard"      Standard ARB mode for playing waveform files using the internal baseband source.

"Ethernet Upload"  
                     Requires R&S SMM-K507.  
                     ARB mode for uploading waveform files using the external QSFP+ interface.

Remote command:

`[ :SOURce<hw> ] :BB:ARbitrary:MODE` on page 29

**Waveform Status**

Displays the upload status of the waveform data.

"not loaded"      No waveform data available as source.

"loading"              Loads waveform data into the ARB memory of the R&S SMM100A. The R&S SMM100A receives the data from the external device connected to the QSFP+ network.

"loaded"              Waveform data is loaded into the ARB memory and ready for playback.

Remote command:

`[ :SOURce<hw> ] :BB:ARbitrary:ETHernet [ :WAVEform ] :STATus?` on page 29

**Waveform Counter**

Displays the number of waveforms that are uploaded to and already played from the ARB memory.

Remote command:

`[ :SOURce<hw> ] :BB:ARbitrary:ETHernet:WAVEform:COUNter?` on page 30

### Ethernet Mode

Sets the mode for the waveform data upload via the Ethernet connection.

"10GbE"            10 Gbit Ethernet

"40GbE"           40 Gbit Ethernet

Remote command:

`[ :SOURce<hw> ] :BB:ARbitrary:ETHernet:MODE` on page 29

### Waveform Info

Accesses a dialog that displays further information on uploaded waveforms.



For information on waveform tags, see chapter "How to manually create a waveform using tag file format" in the R&S SMM100A user manual.

Remote command:

`[ :SOURce<hw> ] :BB:ARbitrary:ETHernet:WAVEform:INFO?` on page 30

`[ :SOURce<hw> ] :BB:ARbitrary:ETHernet:WAVEform:TAG?` on page 31

### Local QSFP Network Settings

Accesses a dialog to configure the local QSFP+ network, see [Chapter 4.3, "Local QSFP network settings"](#), on page 23.

## 4.2 Statistics

Access:

1. Select "Baseband > ARB".

## 2. Select "Statistics".

Arbitrary Waveform Modulation	
RX Upload Segments	441
RX Control Frames	1 323
RX Data Frames	882
RX Data Bytes	1 764
TX Reply Frames	2 772
Occurred Errors	0

This tab displays statistics of ARB Ethernet upload characteristics including receive (Rx) statistics, transmit (Tx) statistics and error statistics.

**Settings:**

<a href="#">RX Upload Segments</a> .....	22
<a href="#">RX Control Frames</a> .....	22
<a href="#">RX Data Frames</a> .....	22
<a href="#">RX Data Bytes</a> .....	22
<a href="#">TX Reply Frames</a> .....	23
<a href="#">Occurred Errors</a> .....	23

**RX Upload Segments**

Displays the number of received upload segments.

Remote command:

`[ :SOURCE<hw> ] :BB:ARbitrary:ETHernet:STATistics:RXUsegments?`  
on page 34

**RX Control Frames**

Displays the number of received control frames.

Remote command:

`[ :SOURCE<hw> ] :BB:ARbitrary:ETHernet:STATistics:RXCFrames?`  
on page 33

**RX Data Frames**

Displays the number of received data frames.

Remote command:

`[ :SOURCE<hw> ] :BB:ARbitrary:ETHernet:STATistics:RXDFrames?`  
on page 34

**RX Data Bytes**

Displays the total received data payload in bytes.

Remote command:

`[ :SOURCE<hw> ] :BB:ARbitrary:ETHernet:STATistics:RXDBytes?`  
on page 33

**TX Reply Frames**

Displays the number of transmitted reply frames.

Remote command:

`[ :SOURCE<hw> ] :BB:ARbitrary:ETHernet:STATistics:TXRFrames?`

on page 34

**Occurred Errors**

Displays the number of errors that occurred during ARB Ethernet upload.

These errors include, for example, loss of packets, packet flow errors, ARB configuration errors and protocol errors. For error history details, open the "Info / Notifications" dialog: Select "System Config" > "Info / Notifications" > "History".

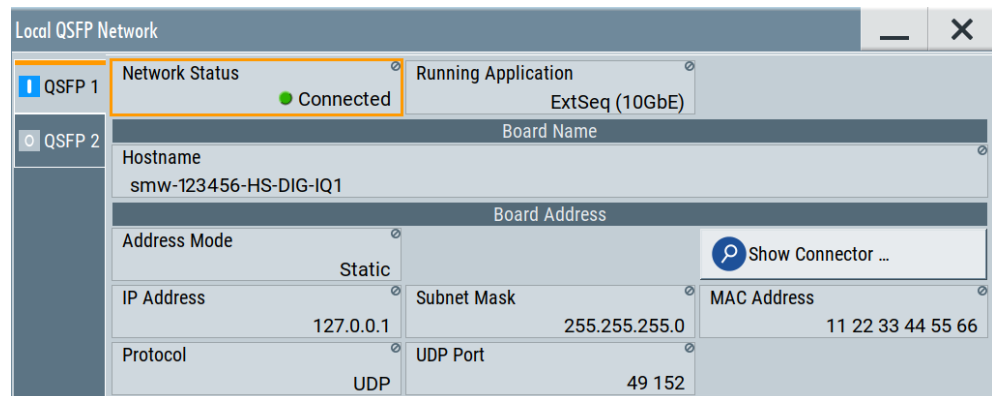
Remote command:

`[ :SOURCE<hw> ] :BB:ARbitrary:ETHernet:STATistics:ERRors?` on page 33

## 4.3 Local QSFP network settings

Access:

1. Select "Baseband" > "ARB".
2. Select "General" > "Mode" > "Ethernet Upload".
3. Select "Local QSFP Network Settings".



The dialog provides settings to configure the QSFP+ network environment and specific identification parameters of the instrument in the network.

The remote commands required to configure these settings are described in [Chapter 6.3, "QSFP+ network commands"](#), on page 35.

**Settings:**

<a href="#">Network Status</a> .....	24
<a href="#">Running Application</a> .....	24
<a href="#">Hostname</a> .....	24

Address Mode.....	24
IP Address.....	24
Subnet Mask.....	25
MAC Address.....	25
Protocol.....	25
UDP Port.....	25

### Network Status

Indicates that the instrument is connected to the network.

Remote command:

`:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:STATus?` on page 35

### Running Application

Indicates the application that currently uses the QSFP+ network configuration.

"ARB Upload (10GbE)/(40GbE)"

ARB ethernet upload using 10 Gbit or 40 Gbit ethernet connection.

Remote command:

`:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:APPLication?` on page 36

### Hostname

Displays the hostname. The hostname consists of the R&S SMM100A short name <instrument>, the serial number <serial number> and the connector of the QSFP+ interface <HS-DIG-IQx>.

Each instrument is delivered with an assigned hostname, a logical name which can be used instead of the IP address. With the default network settings, the IP address is allocated by the DHCP server. This address can change each time the instrument is reconnected. Unlike the IP address, the hostname name does not change.

#### Note:

This function is password-protected. Unlock the protection level 1 to access it.

We recommend that you do not change the default network settings or the hostname to avoid problems with the network connection.

If you change the hostname, be sure to use a unique name.

Remote command:

`:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork[:COMMON]:HOSTname?`  
on page 36

### Address Mode

Indicates that IP addresses are assigned in static mode, i.e. the IP address is assigned manually.

Remote command:

`:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:IPAddress:MODE` on page 36

### IP Address

Assign the IP address manually.

Remote command:

`:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:IPAddress` on page 37



**Subnet Mask**

Assign the subnet mask manually.

Remote command:

```
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork[:IPAddress]:SUBNet:MASK
```

on page 37

**MAC Address**

Indicates the MAC (Media Access Control) address, a unique identifier of the network adapter in the R&S SMM100A.

Remote command:

```
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:MACaddress? on page 37
```

**Protocol**

Displays the communication protocol for the network traffic.

UDP is a connectionless and unreliable transport protocol which is used when a reliable delivery is not required. In general, UDP is faster than TCP in terms of data send over the IP.

Remote command:

```
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:PROTocol on page 37
```

**UDP Port**

Define the port number used for network traffic.

Remote command:

```
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:PORT on page 38
```

## 5 Signal generation control

In this chapter, the signal generation control functionality for ARB Ethernet upload mode is described. It includes trigger and marker settings. In particular, the difference of these settings compared to the standard ARB functionality is described.

### Trigger settings

For trigger mode "Single", signal duration is fixed to one sequence length:

"Signal Duration" = "1 SL", "Signal Duration Unit" = "Sequence Length (SL)"

### Marker settings

Marker modes "Unchanged" and "Restart" are supported for each marker signal.

For a comprehensive description, see chapter "Trigger, marker and clock settings" in the R&S SMM100A user manual.

## 6 Remote control commands

The following commands are required to generate signals with the ARB ethernet upload option in a remote environment. We assume that the R&S SMM100A has already been set up for remote operation in a network as described in the R&S SMM100A documentation. A knowledge about the remote control operation and the SCPI command syntax is assumed.



### Conventions used in SCPI command descriptions

For a description of the conventions used in the remote command descriptions, see section "Remote-Control Commands" in the R&S SMM100A user manual.

### Common suffixes

The following common suffixes are used in the remote commands:

Suffix	Value range	Description
ENTity<ch>	1 to 4	Entity in a multiple entity configuration with separate baseband sources ENTity3 4 require option R&S SMM-K76
SOURce<hw>	1	Available baseband signals Only SOURce1 possible, if the keyword ENTity is used

### Programming examples

This description provides simple programming examples. The purpose of the examples is to present **all** commands for a given task. In real applications, one would rather reduce the examples to an appropriate subset of commands.

The programming examples have been tested with a software tool which provides an environment for the development and execution of remote tests. To keep the example as simple as possible, only the "clean" SCPI syntax elements are reported. Non-executable command lines (e.g. comments) start with two // characters.

At the beginning of the most remote control program, an instrument preset/reset is recommended to set the instrument to a definite state. The commands \*RST and SYSTem:PRESet are equivalent for this purpose. \*CLS also resets the status registers and clears the output buffer.

The following chapters describe the commands specific to the ARB ethernet upload option.

- [General commands](#).....28
- [Statistics commands](#).....31
- [QSFP+ network commands](#).....35
- [Trigger, marker and clock commands](#).....38

## 6.1 General commands

### Example: Activating ARB Ethernet upload

This example requires that the R&S SMM100A is connected to the QSFP+ network, see [Example"Configuring the QSFP+ network"](#) on page 35.

```
// *****
// Enable ARB Ethernet upload mode.
// *****
SOURCEl:BB:ARB:MODE?
// Response: "STAN"
// Standard ARB mode is enabled.
SOURCEl:BB:ARB:MODE EUPL
// Switches to ARB Ethernet upload mode.

// Configure the Ethernet mode.
SOURCEl:BB:ARB:ETHernet:MODE?
// Response: "M10G"
// Sets for 10 Gbit Ethernet mode.
// If your PC also supports 40 Gbit Ethernet mode, switch to this mode:
SOURCEl:BB:ARB:ETHernet:MODE M40G

// Query the status of the waveform upload.
SOURCEl:BB:ARB:ETHernet:WAVEform:STATUs?
// Response: "loaded"
// The waveform data is ready for playback.
// Activate ARB Ethernet upload.
SOURCEl:BB:ARB:STATe 1

// *****
// Optionally, configure trigger settings to specify the start of waveform upload.
// *****
SOURCEl:BB:ARB:TRIGger:SEQuence?
// Response: "SING"
// Single trigger event to start signal generation.
SOURCEl:BB:ARB:TRIGger:SLENgth?
// Response: "1"
// Trigger signal length is one sequence.

// *****
// Optionally, monitor or reconfigure the settings for the first marker signal.
// *****
SOURCEl:BB:ARB:TRIGger:OUTPut1:MODE?
// Response: "UNCH"
// Marker mode unchanged is set. Switch to marker mode restart.
SOURCEl:BB:ARB:TRIGger:OUTPut1:MODE REStart
```

If activated, you can monitor upload characteristics, see [Example"Monitoring ARB Ethernet upload characteristics"](#) on page 31.

**Commands:**

<a href="#">[:SOURce&lt;hw&gt;]:BB:ARbitrary:MODE</a> .....	29
<a href="#">[:SOURce&lt;hw&gt;]:BB:ARbitrary:ETHernet:MODE</a> .....	29
<a href="#">[:SOURce&lt;hw&gt;]:BB:ARbitrary:ETHernet[:WAVEform]:STATus?</a> .....	29
<a href="#">[:SOURce&lt;hw&gt;]:BB:ARbitrary:ETHernet:WAVEform:COUNter?</a> .....	30
<a href="#">[:SOURce&lt;hw&gt;]:BB:ARbitrary:ETHernet:WAVEform:INFO?</a> .....	30
<a href="#">[:SOURce&lt;hw&gt;]:BB:ARbitrary:ETHernet:WAVEform:TAG?</a> .....	31

**[[:SOURce<hw>]:BB:ARbitrary:MODE <ArbMode>**

Defines the mode of the ARB application.

**Parameters:**

<ArbMode>                   STANdard | EUPLoad

**STANdard**

ARB standard mode using a waveform file as internal baseband source.

**EUPLoad**

ARB Ethernet upload mode for uploading waveform files via the external QSFP+ interface.

\*RST:           STANdard

**Example:**                   See [Example"Activating ARB Ethernet upload"](#) on page 28.

**Options:**                   EUPLoad requires R&S SMM-K507.

**Manual operation:**   See ["Mode"](#) on page 20

**[[:SOURce<hw>]:BB:ARbitrary:ETHernet:MODE <Mode>**

Sets the Ethernet mode for the waveform data upload via the Ethernet connection.

**Parameters:**

<Mode>                       M10G | M40G

**M10G**

10 Gbit Ethernet mode

**M40G**

40 Gbit Ethernet mode

\*RST:           M10G

**Example:**                   See [Example"Activating ARB Ethernet upload"](#) on page 28.

**Manual operation:**   See ["Ethernet Mode"](#) on page 21

**[[:SOURce<hw>]:BB:ARbitrary:ETHernet[:WAVEform]:STATus?**

Queries the status of the ARB Ethernet upload application.

**Return values:**

<Status>                   string

**"not loaded"**

No waveform data available as source.

**"loading"**

Loads waveform data into the ARB memory of the R&S SMM100A. The R&S SMM100A receives the data from the external device connected to the QSFP+ network.

**"loaded"**

Waveform data is loaded into the ARB memory and ready for playback.

**Example:** See [Example"Activating ARB Ethernet upload"](#) on page 28.

**Usage:** Query only

**Manual operation:** See ["Waveform Status"](#) on page 20

**[:SOURce<hw>]:BB:ARbitrary:ETHernet:WAVeform:COUNTER?**

Queries the number of waveforms that are uploaded to and already played from the ARB memory.

**Return values:**

<WaveformCounter> integer

**Example:** See [Example"Activating ARB Ethernet upload"](#) on page 28.

**Usage:** Query only

**Manual operation:** See ["Waveform Counter"](#) on page 20

**[:SOURce<hw>]:BB:ARbitrary:ETHernet:WAVeform:INFO?**

Queries information about the currently played waveform in a comma-separated list.

The table below list all parameters that are queried.

Parameter	Description
Waveform counter	Integer number of played waveforms
Comment	Waveform name or comment tag
Clock	Clock rate in Hz
Marker x	Marker mode for marker x. x can be 1 to 3.
Peak level	RF signal peak level
RMS level	RF signal RMS level
Samples	Number of played waveform samples

**Example:** See [Example"Activating ARB Ethernet upload"](#) on page 28.

**Usage:** Query only

**Manual operation:** See ["Waveform Info"](#) on page 21

---

**[:SOURce<hw>]:BB:ARbitrary:ETHernet:WAVEform:TAG? <TagName>**

Queries waveform tags defined in the waveform file with extension \*.wv.

Query results are returned in a comma-separated list. Listed are pairs "<TagName>" "<TagValue>" for each tag.

**Query parameters:**

<TagName>                    string  
                                   Name of the waveform tag.

**Return values:**

<TagValue>                   string  
                                   Value of the waveform tag.

**Example:**                    See [Example "Activating ARB Ethernet upload"](#) on page 28.

**Usage:**                        Query only

**Manual operation:**    See ["Waveform Info"](#) on page 21

## 6.2 Statistics commands

The commands of the BB:ARB:ETHernet:STATistics provide statistical results for waveform upload. Results are available for receive (Rx) direction (PC ⇒ R&S SMM100A) and transmit (Tx) direction (R&S SMM100A ⇒ PC).

**Example: Monitoring ARB Ethernet upload characteristics**

```
// *****
// Query ARB Ethernet upload characteristics in Rx direction.
// *****
SOURcel:BB:ARbitrary:ETHernet:STATistics:RXUsegments?
// Response: "15790"
// The R&S SMM100A receives 15790 Rx upload segments.
SOURcel:BB:ARbitrary:ETHernet:STATistics:RXCFrames?
// Response: "47470"
// The R&S SMM100A receives 47470 Rx control frames.
SOURcel:BB:ARbitrary:ETHernet:STATistics:RXDFrames?
// Response: "31610"
// The R&S SMM100A receives 31610 Rx data frames.
SOURcel:BB:ARbitrary:ETHernet:STATistics:RXDBytes?
// Response: "63220"
// The R&S SMM100A receives 63220 Rx data bytes.

// *****
// Query ARB Ethernet upload characteristics in Tx direction.
// *****
SOURcel:BB:ARbitrary:ETHernet:STATistics:TXCFrames?
// Response: "89120"
// The R&S SMM100A sends 89120 Tx reply/acknowledgment frames.
```

```
// *****
// Query occurred errors.
// *****
SOURCE:BB:ARbitrary:Ethernet:STATistics:ERRORs?
// Response: "0"
// No errors occurred during the ARB Ethernet upload session.

// *****
// Alternatively, query all results.
// *****
SOURCE:BB:ARbitrary:Ethernet:STATistics:ALL?
// Response: "15790","47470","31610","63220","89120","0"
// Returns a comma-separated list of all result parameters:
// "<RxUploadSegments>,<RxControlFrames>,<RxDataFrames>,<RxDataBytes>,<TxReplyFrames>,<OccurredErrors>"
```

**Commands:**

<a href="#">[:SOURCE&lt;hw&gt;]:BB:ARbitrary:Ethernet:STATistics:ALL?</a> .....	32
<a href="#">[:SOURCE&lt;hw&gt;]:BB:ARbitrary:Ethernet:STATistics:ERRORs?</a> .....	33
<a href="#">[:SOURCE&lt;hw&gt;]:BB:ARbitrary:Ethernet:STATistics:RXCFrames?</a> .....	33
<a href="#">[:SOURCE&lt;hw&gt;]:BB:ARbitrary:Ethernet:STATistics:RXDBytes?</a> .....	33
<a href="#">[:SOURCE&lt;hw&gt;]:BB:ARbitrary:Ethernet:STATistics:RXDFrames?</a> .....	34
<a href="#">[:SOURCE&lt;hw&gt;]:BB:ARbitrary:Ethernet:STATistics:RXUSegments?</a> .....	34
<a href="#">[:SOURCE&lt;hw&gt;]:BB:ARbitrary:Ethernet:STATistics:TXRFrames?</a> .....	34

**[\[:SOURCE<hw>\]:BB:ARbitrary:Ethernet:STATistics:ALL?](#)**

Queries all statistic results in a comma-separated list that contains the following parameters:

```
<RxUploadSegments>,<RxControlFrames>,<RxDataFrames>,<RxDataBytes>,<TxReplyFrames>,<OccurredErrors>
```

**Return values:**

<RxUploadSegment> integer

Number of Rx upload segments, see [\[:SOURCE<hw>\]:BB:ARbitrary:Ethernet:STATistics:RXUSegments?](#) on page 34.

<RxControlFrames> integer

Number of Rx control frames, see [\[:SOURCE<hw>\]:BB:ARbitrary:Ethernet:STATistics:RXCFrames?](#) on page 33.

<RxDataFrames> integer

Number of Rx data frames, see [\[:SOURCE<hw>\]:BB:ARbitrary:Ethernet:STATistics:RXCFrames?](#) on page 33.



<RxDataBytes>	integer Number of Rx data bytes, see <a href="#">[:SOURCE&lt;hw&gt;]:BB:ARbitrary:Ethernet:Statistics:RXDBytes?</a> on page 33.
<TxReplyFrames>	integer Number of Tx reply frames, see <a href="#">[:SOURCE&lt;hw&gt;]:BB:ARbitrary:Ethernet:Statistics:TXRFrames?</a> on page 34.
<b>Example:</b>	See <a href="#">Example"Monitoring ARB Ethernet upload characteristics"</a> on page 31.
<b>Usage:</b>	Query only

**[[:SOURCE<hw>]:BB:ARbitrary:Ethernet:Statistics:ERRORS?**

Queries the number of errors that occurred during ARB Ethernet upload.

**Return values:**

<OccurredErrors>	integer
*RST:	0

**Example:** See [Example"Monitoring ARB Ethernet upload characteristics"](#) on page 31.

**Usage:** Query only

**Manual operation:** See ["Occurred Errors"](#) on page 23

**[[:SOURCE<hw>]:BB:ARbitrary:Ethernet:Statistics:RXCFrames?**

Queries the number of Rx control frames, that the R&S SMM100A receives from the PC.

**Return values:**

<ControlFrames>	integer
*RST:	0

**Example:** See [Example"Monitoring ARB Ethernet upload characteristics"](#) on page 31.

**Usage:** Query only

**Manual operation:** See ["RX Control Frames"](#) on page 22

**[[:SOURCE<hw>]:BB:ARbitrary:Ethernet:Statistics:RXDBytes?**

Queries the number of Rx data bytes, that the R&S SMM100A receives from the PC.

**Return values:**

<DataBytes>	integer
*RST:	0

**Example:** See [Example"Monitoring ARB Ethernet upload characteristics"](#) on page 31.

**Usage:** Query only

**Manual operation:** See ["RX Data Bytes"](#) on page 22

#### **[:SOURce<hw>]:BB:ARBitrary:ETHernet:STATistics:RXDFrames?**

Queries the number of data frames, that the R&S SMM100A receives from the PC.

**Return values:**

<DataFrames> integer  
\*RST: 0

**Example:** See [Example"Monitoring ARB Ethernet upload characteristics"](#) on page 31.

**Usage:** Query only

**Manual operation:** See ["RX Data Frames"](#) on page 22

#### **[:SOURce<hw>]:BB:ARBitrary:ETHernet:STATistics:RXUSegments?**

Queries the number of Rx upload segments, that the R&S SMM100A receives from the PC.

**Return values:**

<UploadSegments> integer  
\*RST: 0

**Example:** See [Example"Monitoring ARB Ethernet upload characteristics"](#) on page 31.

**Usage:** Query only

**Manual operation:** See ["RX Upload Segments"](#) on page 22

#### **[:SOURce<hw>]:BB:ARBitrary:ETHernet:STATistics:TXRFrames?**

Queries the number of Tx reply frames sent from R&S SMM100A to the PC.

**Return values:**

<ReplyBytes> integer  
\*RST: 0

**Example:** See [Example"Monitoring ARB Ethernet upload characteristics"](#) on page 31.

**Usage:** Query only

**Manual operation:** See ["TX Reply Frames"](#) on page 23

## 6.3 QSFP+ network commands

### Example: Configuring the QSFP+ network

```
// *****
// Configuring QSFP+ 1 network parameters at the connector HS Dig I/Q1.
// *****
// Set the IP address.
SYSem:COMMunicate:BB1:QSFP:NETWork:IPAdDress "127.0.0.1"
// Set the subnet mask.
SYSem:COMMunicate:BB1:QSFP:NETWork:IPAdDress:SUBNet:MASK 255.255.255.0
// Set the UDP Port.
SYSem:COMMunicate:BB1:QSFP:NETWork:PORT 49152

// Query the QSFP+ network settings.
SYSem:COMMunicate:BB1:QSFP:NETWork:STATus?
// Response: "1"
// The R&S SMM100A is connected.
SYSem:COMMunicate:BB1:QSFP:NETWork:APPLication?
// Response: "ExtSeq (10GbE)" for extended sequencer, 10 GbE Ethernet mode.
// Response: "ARB Upload (10GbE)" for ARB Ethernet upload, 10 GbE Ethernet mode.
SYSem:COMMunicate:BB1:QSFP:NETWork:HOSTname?
// Returns the hostname consisting of the R&S SMM100A short name, the
// serial number and the connector HS-DIG-IQx.
SYSem:COMMunicate:BB1:QSFP:NETWork:IPAdDress:MODE?
// Response: "STATic"
SYSem:COMMunicate:BB1:QSFP:NETWork:MACAdDress?
SYSem:COMMunicate:BB1:QSFP:NETWork:PROTOcol?
// Response: "UDP"
```

### Commands:

:SYSem:COMMunicate:BB<hw>:QSFP:NETWork:STATus?	35
:SYSem:COMMunicate:BB<hw>:QSFP:NETWork:APPLication?	36
:SYSem:COMMunicate:BB<hw>:QSFP:NETWork[:COMMON]:HOSTname?	36
:SYSem:COMMunicate:BB<hw>:QSFP:NETWork:IPAdDress:MODE	36
:SYSem:COMMunicate:BB<hw>:QSFP:NETWork:IPAdDress	37
:SYSem:COMMunicate:BB<hw>:QSFP:NETWork[:IPAdDress]:SUBNet:MASK	37
:SYSem:COMMunicate:BB<hw>:QSFP:NETWork:MACAdDress?	37
:SYSem:COMMunicate:BB<hw>:QSFP:NETWork:PROTOcol	37
:SYSem:COMMunicate:BB<hw>:QSFP:NETWork:PORT	38

---

### :SYSem:COMMunicate:BB<hw>:QSFP:NETWork:STATus?

Queries the network configuration state.

#### Return values:

<NetworkStatus> 1 | ON | 0 | OFF

\*RST: n.a. (no preset. default: 0)

- Example:** See [Example"Configuring the QSFP+ network"](#) on page 35.
- Usage:** Query only
- Manual operation:** See ["Network Status"](#) on page 24

**:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:APPLication?**

Queries a running application.

**Return values:**

<RunningApp> string  
Returns the name of the running application.

**ARB Upload (10/40GbE)**

ARB Ethernet upload 10/40 Gbit Ethernet mode

- Example:** See [Example"Configuring the QSFP+ network"](#) on page 35.

**Usage:** Query only

**Options:** ARB Upload (10GbE)|ARB Upload (40GbE) require R&S SMM-K507.  
ExtSeq (10GbE) requires R&S SMM-K506.

- Manual operation:** See ["Running Application"](#) on page 24

**:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork[:COMMON]:HOSTname?**

Queries an individual hostname for the vector signal generator.

**Note:**We recommend that you do not change the hostname to avoid problems with the network connection. If you change the hostname, be sure to use a unique name.

This is a password-protected function. Unlock the protection level 1 to access it, see `:SYSTem:PROTECT<ch>[:STATE]`.

**Return values:**

<Hostname> string

- Example:** See [Example"Configuring the QSFP+ network"](#) on page 35.

**Usage:** Query only

- Manual operation:** See ["Hostname"](#) on page 24

**:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:IPADdress:MODE**  
<IpAddressMode>

Displays the mode of the IP address.

**Parameters:**

<IpAddressMode> STATic  
\*RST: n.a. (factory preset: STATic)

- Example:** See [Example"Configuring the QSFP+ network"](#) on page 35.

**Manual operation:** See ["Address Mode"](#) on page 24

---

**:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:IPADdress <IpAddress>**

Sets the IP address.

**Parameters:**

<IpAddress> String  
Range: 0.0.0.0 to ff.ff.ff.ff

**Example:** See [Example"Configuring the QSFP+ network"](#) on page 35.

**Manual operation:** See ["IP Address"](#) on page 24

---

**:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork[:IPADdress]:SUBNet:MASK  
<SubnetMask>**

Sets the subnet mask.

**Parameters:**

<SubnetMask> String

**Example:** See [Example"Configuring the QSFP+ network"](#) on page 35.

**Manual operation:** See ["Subnet Mask"](#) on page 25

---

**:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:MACAddress?**

Queries the MAC address of the network adapter.

This is a password-protected function. Unlock the protection level 1 to access it, see `:SYSTem:PROTect<ch>[:STATe]` in the R&S SMM100A user manual.

**Return values:**

<MacAddress> string

**Example:** See [Example"Configuring the QSFP+ network"](#) on page 35.

**Usage:** Query only

**Manual operation:** See ["MAC Address"](#) on page 25

---

**:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:PROTocol <Protocol>**

Displays the communication protocol for the network traffic.

**Parameters:**

<Protocol> UDP  
\*RST: n.a. (factory preset: UDP)

**Example:** See [Example"Configuring the QSFP+ network"](#) on page 35.

**Manual operation:** See ["Protocol"](#) on page 25

---

**:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:PORT <PortNumber>**

Sets the port address used for network traffic.

**Parameters:**

<PortNumber>	integer
Range:	0 to 65536
*RST:	n.a. (factory preset: 49152)

**Example:** See [Example "Configuring the QSFP+ network"](#) on page 35.

**Manual operation:** See ["UDP Port"](#) on page 25

## 6.4 Trigger, marker and clock commands

For trigger, marker and clock commands, see the corresponding subchapters of chapter "SOURCE:BB:ARbitrary subsystem" in the R&S SMM100A user manual.

# Annex

## A I/Q over Ethernet (QSFP+) protocol

- [UDP/IP protocol stack](#).....39
- [Waveform file transfer](#).....43
- [ARB upload flow diagram](#).....44
- [Waveform upload command sequences](#).....45
- [Using a python script for waveform data transfer](#).....46

### A.1 UDP/IP protocol stack

This chapter provides background information on the protocol stack to transfer I/Q waveform data over Ethernet.

- [Main specifications](#).....39
- [OSI model](#).....40
- [I/Q frame and UDP subheader](#).....40
- [ACK frame](#).....41
- [Control and data frame types](#).....41

#### A.1.1 Main specifications

##### Protocol characteristics

The UDP/IP protocol stack offers the following main characteristics:

- Protocol: User Datagram Protocol (UDP) with IP version 4 (IPv4)
- IP addresses: Static
- Primary device (client): Host PC with 10G/40G Ethernet network card
- Secondary device: R&S SMM100A
- Endianness: LittleEndian for UDP packet header and payload

##### Protocol stack specification

The UDP/IP protocol stack is specified as follows:

- At the beginning of the UDP payload, the proprietary I/Q frame header is inserted. That means that every datagram is one I/Q frame.
- Main frame types: Control frame and data frame
- The client always starts a session with an initial control frame using the command C\_START\_SESSION.
- Control frames also initiate or mark the end of every data transfer session.
- Data transfer sessions contain one or more data frames.

- The header of the I/Q classifies the frame type used in the I/Q frame.
- A waveform consists of one or more waveform segments.
- The segment length is a multiple of 512 bytes. If a segment is shorter, missing bytes are added to the payload (padding).

### A.1.2 OSI model

OSI layer		Classification	Description
5	Application (level 1)	"ARB Upload" protocol	String based, e.g. for transmission of waveform parameter.
4	Application (level 0)	"SigGen - I/Q over Ethernet" protocol	Proprietary Rohde & Schwarz protocol used for upload
3	Network / transport	UDP/IP	-
2	Data link	Ethernet, IEEE802.3	-
1	Physical	QSFP+ (10GbE, 40GbE)	Optical or copper

### A.1.3 I/Q frame and UDP subheader

The I/Q frame carries 64-bit information divided into six frame fields. [Figure A-1](#) displays the frame composition and [Table A-1](#) describes frame field bit information.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
<b>Bit [0:31]</b>	<b>Flow Control Counter (0..65535)</b> (can be restarted with zero every time)																<b>Coder instance</b>				<b>Payload Definition Code</b> Cmd Code/Data Format				<b>Frame Type</b>							
<b>Bit [32:63]</b>	Payload Size (byte, except sub-header)																Protocol Version / reserved															

*Figure A-1: I/Q frame or UDP subheader*

*Table A-1: Bit number and I/Q frame or UDP subheader fields*

Bit number	I/Q frame or UDP subheader fields
[0:15]	<b>Flow control counter</b> (0 to 65535) Restart with zero every time.
[16:23]	<b>Coder instance</b>
[24:29]	<b>Payload definition code</b> Cmd code/data format
[30:31]	<b>Frame type</b> 0: Ctrl frame   1: Data frame   2: Not defined   3: Idle frame
[32:47]	<b>Payload size</b> (byte) Except for subheader.
[48:63]	<b>Protocol version / reserved</b>



### A.1.4 ACK frame

The acknowledgement (ACK) frame carries 64-bit information divided into three frame fields. [Figure A-2](#) displays the frame composition and [Table A-2](#) describes frame field bit information.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Bit [0:31]	Reserved (Default value: 0x0200)																Error code (OK = 0x0000)															
Bit [32:63]	additional State/Info field																															

*Figure A-2: ACK frame and frame fields*

*Table A-2: Bit number and ACK frame field*

Bit number	ACK frame fields
[0:15]	Reserved (Default value: 0x0200)
[16:31]	Error code (OK = 0x0000)
[32:63]	Additional state/info field

The UDP payload is 18 bytes in total. This implementation ignores the last 10 bytes.

### A.1.5 Control and data frame types

This chapter provides information on control frames and data frames. The information includes the frame types and names, payload, frame field information and requirements.

#### A.1.5.1 Control frame

There are five types of control frames, see [Table A-3](#).

Table A-3: Control frame types

Frame type/command	Cmd/ payload code (6 Bits)	Payload (Bytes)	Reply	Description and requirements
C_START_SESSION	0	8	ACK	The host PC sends this frame type at least once when initiating the upload session.  For example, if you change the ARB mode to "Standard", you need to send C_START_SESSION again.  For Payload-Byte[0], set 0x00.
C_START_WV_TRANSFER	1	16	-	Initializes receiving of waveform data and sends the command to the application layer (APPL). Command payload: <ul style="list-style-type: none"> <li>• 4 Bytes: Segment ID</li> <li>• 4 Bytes: Memory offset = n x 512 Bytes One segment offset is a multiple of 512 bytes (maximum number of bytes = 2 TB – 512 bytes)</li> <li>• 8 Bytes: number of 32-bit samples.</li> </ul>
C_TRANSFER_FINISHED	2	0	-	Indicates the end of data transfer.  Use this frame type, for example, to indicate the end of waveform data transfer.
C_APPL_DATA	3	8 to 4096	ACK	Use this frame type for data exchange in the application layer. Sends the command to the application layer (APPL).  Command payload is a zero-terminated string with maximum string size = 4096 bytes – 1 * payload size. The payload is a multiple of 8 bytes. If it does not fit, add zeros.  Example for APPL commands from host PC to SMW (BBW):  STOP_ARB_AND_SET_ARB_PARAMS: {TYPE:SMU-WV}{SAMPLES:10000000}{CLOCK: 1.1e+08}{LEVEL_OFFS:0,0}  CHECK_STATE_AND_RESTART_ARB  CHECK_STATE_AFTER_UPLOAD
C_GET_STATE	5	8	ACK (IF-State)	Transmission-status-check to check the IF-state. Checks also errors in layer 3 (network/transport) and layer 4 (application level 0).  Returns "0", if no errors occurred during last check.  ACK frame field: Additional info = number of last received 32-bit samples.

### A.1.5.2 Data frame

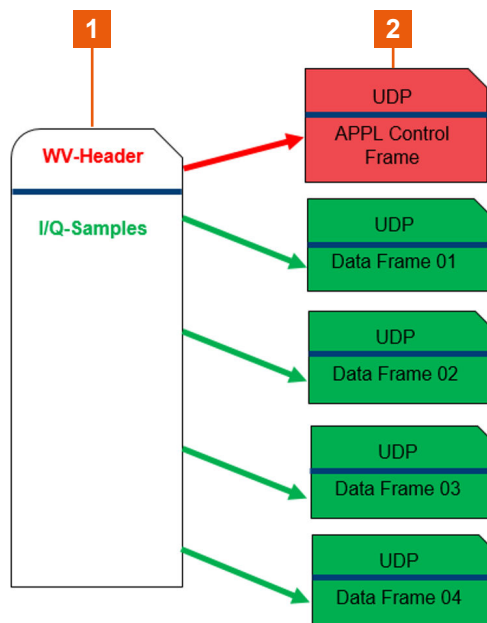
There is one data frame type, see [Table A-4](#).

Table A-4: Data frame types

Frame type/command	Payload code (6 Bits)	Payload
D_WV_SAMPLES	1	n x 32 bits waveform I/Q samples (16-bit I value, 16-bit Q value) (n = payload size / 4 = Samples per frame)

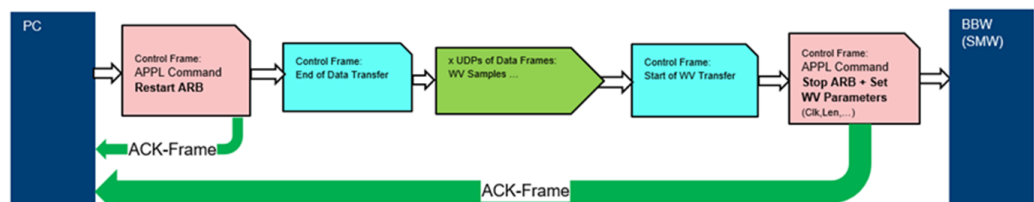
## A.2 Waveform file transfer

### Waveform file and UDP frames

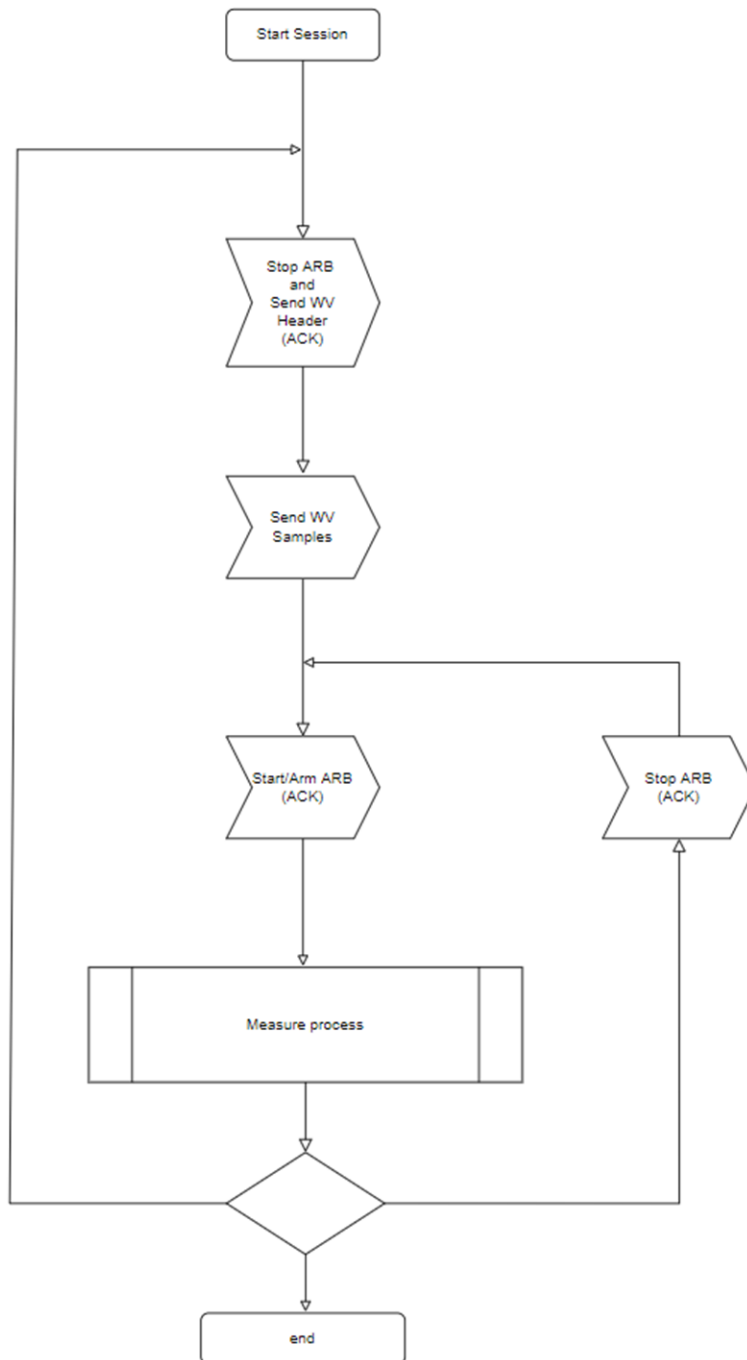


1 = Waveform file and file content  
2 = UDP frames

### Waveform transfer (PC) and waveform download (SMW)



### A.3 ARB upload flow diagram



## A.4 Waveform upload command sequences

### APPL command

Use the application layer (APPL) command `APPL-Command (CmdCode= ,3')` to send text-based command sequences to the SMW ARB application. For example, use the APPL command `„STOP_ARB_AND_SET_ARB_PARAMS:“`.

After this string, insert the waveform header of a waveform file. For example, insert the following:

```
"STOP_ARB_AND_SET_ARB_PARAMS:{TYPE:SMU-WV}{COMMENT:
"5G_signal_01.wv"}{SAMPLES:300000000}{CLOCK:7.0e+07}{LEVEL OFFS:
0,0}"
```

You can set the level offset RMS value and peak value `LEVEL OFFS: [RMS, PEAK]`. For more information, see chapter "Tags for waveforms, data and control lists" in the R&S SMM100A user manual.

The following sections provide examples for waveform upload command sequences in tables:

- ["WV transfer with meta data and upload check and ARB restart"](#) on page 45
- ["WV transfer with meta data and upload check and no ARB restart"](#) on page 45
- ["WV play stop and start via protocol"](#) on page 46

### WV transfer with meta data and upload check and ARB restart

The table shows a command sequence for transferring waveforms with meta data, checking the waveform upload check and restarting the ARB application.

Seq. No.	Command / frame	Description
(1,1)	APPL command (CmdCode= ,3'): "STOP_ARB_AND_SET_ARB_PARAMS:"	Stops the ARB generator, configures waveform parameter. Replies with ACK/NAK. If errors occur, repeat this command.
(1,2)	Ctrl frame	Starts waveform transfer: "START_WV_TRANSFER"
(1,3)	Data frames	Transfer waveform data.
(1,4)	Ctrl frame	Terminates waveform transfer: "TRANSFER_FINISHED"
(1,5)	APPL-Command (CmdCode= ,3'): "CHECK_STATE_AND_RESTART_ARB"	If no errors occurred during transfer, the ARB application restarts and ACK is sent. If errors occur, the ARB application does not restart and NAK is sent. Repeat the transfer with sequence number (1,2) and following.

### WV transfer with meta data and upload check and no ARB restart

The table shows a command sequence for transferring waveforms with meta data, checking the waveform upload check and no restart of the ARB application.

## Using a python script for waveform data transfer

Seq. No.	Command / frame	Description
(2,1)	APPL command (CmdCode= ,3'): "STOP_ARB_AND_SET_ARB_PARAMS:"	Stops the ARB generator, configures waveform parameter. Replies with ACK/NAK. If errors occur, repeat this command.
(2,2)	Ctrl frame	Starts waveform transfer: "START_WV_TRANSFER"
(2,3)	Data frames	Transfer waveform data.
(2,4)	Ctrl frame	Terminates waveform transfer: "TRANSFER_FINISHED"
(2,5)	APPL-Command (CmdCode= ,3'): "CHECK_STATE_AFTER_UPLOAD"	If no errors occurred during transfer, ACK is sent with automatic restart trigger. If errors occur, NAK is sent. Repeat the transfer with sequence number (2,1) and following.

**WV play stop and start via protocol**

The table shows a command sequence for arbitrary playing waveforms via the protocol without reconfiguring or transferring waveform data again.

Seq. No.	Command / frame	Description
(3,1)	APPL command (CmdCode= ,3'): "STOP_ARB"	Stops the ARB generator.
(3,2)	APPL-Command (CmdCode= ,3'): "CHECK_STATE_AND_RESTART_ARB"	Waveform play restarts. If a trigger is active, you can start palying/replaying the waveform with an external/internal trigger signal.

**A.5 Using a python script for waveform data transfer**

The following example describes how to transfer waveform data from the host PC to the R&S SMM100A via a python script. You can run the script to transfer small waveform files of several hundred Msample.

The script uses waveform upload command sequence as in the table [WV transfer with meta data and upload check and ARB restart](#).

**Example: To transfer waveform data**

```

# script version: 1.01
# this program sends waveform samples (2x16byte) to the Rohde & Schwarz signal
# generator via ARB-ETH-UPLOAD (UDP/QSFP)
# file format example:
{TYPE:SMU-WV}{COMMENT:"wv-info or filename"}{SAMPLES:400000}{CLOCK:200000000}
{LEVEL OFFS:0,0}{WAVEFORM-1600001:#<--binary samples-->}

# General, network and port specifications
import socket
import ctypes, sys
from ctypes import *

# host IP address
UDP_IP_CLIENT = "192.168.179.9"
UDP_PORT_CLIENT = 32551

# Siggen QSFP IP address
UDP_IP_ADDRESS = "192.168.179.14"
UDP_PORT_NO = 49152

# UDP payload max
frm_size = 63624

# application commands
strCmdWvConfig      = "STOP_ARB_AND_SET_ARB_PARAMS:"
strCmdWvRestart     = "CHECK_STATE_AND_RESTART_ARB"
strCmdWvChkStateOnly = "CHECK_STATE_AFTER_UPLOAD" # if using of external
                                                    # start trigger for trigger
                                                    # mode 'single' or 'armed'

strCmdWvConfig      = "STOP_ARB" # stops ARB, restart with external
                                # start trigger or
                                # "CHECK_STATE_AND_RESTART_ARB"
                                # for trigger mode 'single' or 'armed'

# RX Ack frame
class rxAck_t(Structure): # size = 20
    _fields_ = [("frm_cnt", c_uint16), # 0x0200
                ("reserve", c_uint8), # 0
                ("frm_err", c_uint8), # 0: no errors, other: error bit mask
                ("info", c_uint32)] # number of received samples

# TX data frame
class txFrm_t(Structure):
    _fields_ = [("frm_cnt", c_uint16), # flow cntnl counter
                ("reserved", c_uint8),
                ("frm_type", c_uint8), # frame type (command/data)
                ("pld_sz", c_uint16), # frame payload without header in bytes
                ("prt_ver", c_uint16), # protocol version
                ("payload", c_uint8 * frm_size)]

```

## Using a python script for waveform data transfer

```

# TX transmission start frame
class startWvTransmFrm_t(Structure):
    _fields_ = [("frm_cnt", c_uint16), # flow cntrl counter
               ("reserve", c_uint8),
               ("frm_type", c_uint8), # frame type (command/data)
               ("pld_sz", c_uint16), # frame payload without header in bytes
               ("prt_ver", c_uint16), # protocol version
               ("segm_id", c_uint32), # 0
               ("segm_addr", c_uint32), # 0
               ("samples", c_uint64)] # number of 32bit (2x16bit, I/Q) samples
                                     # number is a multiple of 128 samples

# reads command confirmation from Siggen
def readAck(clientSock):
    clientSock.settimeout(3.0)
    resp = clientSock.recvfrom(18) # read 18 bytes
    clientSock.settimeout(None)
    return rxAck_t.from_buffer(bytearray(resp[0]))

# sends any application command and waits for the answer
def wrtApplCmdAndReadAck(clientSock, frm_cnt, applCmdStr):
    toSend = len(applCmdStr)+1 # max string length is 4095
    addStaffing = toSend % 8
    if addStaffing>0:
        toSend += 8 - addStaffing
    applCmd = txFrm_t(frm_cnt,0,0x03,toSend,0x0100)
    applCmd.payload[:len(applCmdStr)] = bytearray(applCmdStr,"ascii")
    [:len(applCmdStr)]
    applCmd.payload[len(applCmdStr)] = 0
    clientSock.sendto(bytearray(applCmd)[:toSend+8], (UDP_IP_ADDRESS,
    UDP_PORT_NO))
    return readAck(clientSock)

# read complete file into a buffer
file = open(sys.argv[1], 'rb') # *.wv
wv = bytearray(file.read())
file.close()

# extract the waveform header
sHdr = str()
wvPos = wvPosSmpl = wv.find(b'{WAVEFORM-}')
if wvPos != -1:
    empPos = wv.find(b'{EMPTYTAG-}')
    if empPos != -1:
        wvPos = empPos
    wvPos = wv.rfind(b'}', 0, wvPos)
if wvPos != -1:
    sHdr = str(wv[wvPos + 1], 'UTF-8')

```



## Using a python script for waveform data transfer

```

print('HDR: ' + sHdr)

# read value of sample number
samples = 0
wvPos = sHdr.find('{SAMPLES:}')
if wvPos != -1:
    samples = int(sHdr[wvPos+9:sHdr.find('}',wvPos)])
print(f'number of samples: {samples}')

currSmplPos = wv.find(b'##',wvPosSmpl) + 1

# always send complete 128 samples blocks
samples_128 = samples
smplMod = samples % 128
if smplMod>0:
    smplMod = 128 - smplMod
    samples_128 += smplMod
    wv += bytearray(smplMod*4)

clientSock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
clientSock.bind((UDP_IP_CLIENT, UDP_PORT_CLIENT));

# send Open-Session-Cmd:
ctrlFrm = txFrm_t(0,0,0x00,8,0x0100) # open session
ctrlFrm.payload[:8] = bytearray([0, 0, 0, 0, 0, 0, 0, 0]);
clientSock.sendto(bytearray(ctrlFrm)[:8+8], (UDP_IP_ADDRESS, UDP_PORT_NO))
ackFrm = readAck(clientSock)
print('Session-Open-Cmd-result: ' , ackFrm.frm_err)

# send Send-WV-Config/Header-Cmd:
ackFrm = wrtApplCmdAndReadAck(clientSock, 0, strCmdWvConfig + sHdr)
print('Send_WV-Header-Ack-result: ' , ackFrm.frm_err)

# send Start-WV-Transmission-Cmd:
frm_cnt = 1
# start wv transmission
startWvTransmFrm = startWvTransmFrm_t(frm_cnt, 0, 0x01,16,0x0100, 0, 0,
samples_128)
clientSock.sendto(bytearray(startWvTransmFrm)[:16+8], (UDP_IP_ADDRESS,
UDP_PORT_NO))

# send I/Q samples:
leftSmpl = samples_128
smplPerFrmMax = frm_size>>2
while leftSmpl>0:
    toSend = smplPerFrmMax if leftSmpl>=smplPerFrmMax else leftSmpl
    leftSmpl -= toSend
    frm_cnt += 1
    dataFrm = txFrm_t(frm_cnt,0,0x80,toSend*4,0x0100)
    dataFrm.payload[:(toSend*4)] = wv[currSmplPos:currSmplPos+(toSend*4)]

```

## Using a python script for waveform data transfer

```
currSmplPos += toSend*4
clientSock.sendto(bytearray(dataFrm)[:8+(toSend*4)], (UDP_IP_ADDRESS,
UDP_PORT_NO))

# send End-of-WV-Transmission-Cmd:
frm_cnt += 1
ctrlFrm = txFrm_t(frm_cnt,0,0x02,0,0x0100) # open session
clientSock.sendto(bytearray(ctrlFrm)[:8], (UDP_IP_ADDRESS, UDP_PORT_NO))

# send Check-Transm-and-Start-ARB-Cmd:
frm_cnt += 1
ackFrm = wrtApplCmdAndReadAck(clientSock, frm_cnt, strCmdWvRestart)
print('Check-Transm-and-Start-ARB-result: ', ackFrm.frm_err, 'sample recv:',
ackFrm.info) # check: ackFrm.info == samples_128?

clientSock.close()
```

## Glossary: Abbreviations

### A

**ARB:** Arbitrary waveform generator

### Q

**QSFP:** Quad Small Form-factor Pluggable

### S

**SFP:** Small Form-factor Pluggable

## List of commands

:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:APPLication?	36
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:IPADdress	37
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:IPADdress:MODE	36
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:MACAddress?	37
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:PORT	38
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:PROTocol	37
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork:STATus?	35
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork[:COMMon]:HOSTname?	36
:SYSTem:COMMunicate:BB<hw>:QSFP:NETWork[:IPADdress]:SUBNet:MASK	37
[:SOURce<hw>]:BB:ARBitrary:ETHernet:MODE	29
[:SOURce<hw>]:BB:ARBitrary:ETHernet:STATistics:ALL?	32
[:SOURce<hw>]:BB:ARBitrary:ETHernet:STATistics:ERRors?	33
[:SOURce<hw>]:BB:ARBitrary:ETHernet:STATistics:RXCFrames?	33
[:SOURce<hw>]:BB:ARBitrary:ETHernet:STATistics:RXDBytes?	33
[:SOURce<hw>]:BB:ARBitrary:ETHernet:STATistics:RXDFrames?	34
[:SOURce<hw>]:BB:ARBitrary:ETHernet:STATistics:RXUSegments?	34
[:SOURce<hw>]:BB:ARBitrary:ETHernet:STATistics:TXRFrames?	34
[:SOURce<hw>]:BB:ARBitrary:ETHernet:WAVeform:COUNter?	30
[:SOURce<hw>]:BB:ARBitrary:ETHernet:WAVeform:INFO?	30
[:SOURce<hw>]:BB:ARBitrary:ETHernet:WAVeform:TAG?	31
[:SOURce<hw>]:BB:ARBitrary:ETHernet[:WAVeform]:STATus?	29
[:SOURce<hw>]:BB:ARBitrary:MODE	29

# Index

## A

About ARB ethernet upload .....	10
Application cards .....	8
Application notes .....	8
ARB Ethernet statistics commands	
Remote-control commands .....	31
ARB general Ethernet commands	
Remote-control commands .....	28

## B

Brochures .....	7
-----------------	---

## C

Conventions	
SCPI commands .....	27

## D

Data sheets .....	7
Default settings	
ARB .....	20
Documentation overview .....	6

## G

General	
Ethernet mode .....	21
Mode .....	20
Status .....	20
Waveform counter .....	20, 21
Getting started .....	6

## H

Help .....	7
------------	---

## I

I/Q over Ethernet (QSFP+) protocol	
ARB flow diagram .....	44
UDP protocol stack .....	39
Waveform file transfer .....	43
Waveform upload command sequences .....	45
Installation .....	5
Instrument help .....	7
Instrument security procedures .....	7
IP address	
Dynamic .....	24
Zeroconf .....	24

## O

Open source acknowledgment (OSA) .....	8
--	---

## Q

QSFP+ network commands	
Remote-control commands .....	35
QSFP+ network settings	
Hostname .....	24
IP address mode .....	24
Network IP address .....	24

Network Mac address .....	25
Network status .....	24
Protocol .....	25
Running Application .....	24
Subnet mask .....	25
UDP port .....	25

## R

Release notes .....	8
Remote control	
Programming examples .....	27
Remote-control commands	
ARB Ethernet general commands .....	28
ARB Ethernet statistics commands .....	31
QSFP+ network commands .....	35

## S

Safety instructions .....	7
Security procedures .....	7
Service manual .....	7
Set to default	
ARB .....	20
State	
ARB .....	20
Statistics .....	21
Occurred errors .....	23
Rx control frames .....	22
Rx data bytes .....	22
Rx data frames .....	22
Rx upload segments .....	22
Tx reply frames .....	23

## U

UDP protocol stack .....	39
ACK frame .....	41
Control frame .....	41
Data frame .....	42
Frame types .....	41
I/Q frame .....	40
Main specifications .....	39
OSI model .....	40
UDP subheader .....	40
User manual .....	7

## V

Videos .....	8
--------------	---

## W

White papers .....	8
--------------------	---