

## Application Note AN6003

# APVSG Memory Segmentation

### Purpose

This Application Note gives an overview of the definitions for IQ samples, segments, and sequences and then describes the different options for storing and replaying multiple sets of IQ modulation data from the memory.

### Table of Contents

Introduction.....	2
Nomenclature.....	3
IQ Sample .....	3
Segment .....	4
Sequence.....	4
Store Segments .....	6
Upload files with the APVSG GUI .....	6
Load Block Data with SCPI Commands .....	6
Choose and Replay Segments .....	7
Pick a Segment in the GUI .....	7
Select Segments with SCPI Commands.....	8
Replaying Segments in Sequences .....	9
Sequencing with SCPI commands .....	9
[Option] Sequencing with the Fast Control Port .....	11
Further Documentation .....	12

## Introduction

The memory segmentation in APVSG devices allows the user to freely switch between stored waveforms. To that end, multiple sets of IQ modulation data are stored separately in the memory of the device. Each set of IQ data can then be replayed and employed to modulate the carrier frequency to result in the desired RF output. The user is presented with several playback options to select which set of stored IQ modulation data is replayed.

This Application Note first gives a brief introduction into the nomenclature of samples, segments and sequences and then describes the different options for storing and replaying the IQ data in the memory. As for storing options, the user can choose between

- The APVSG GUI or
- SCPI commands.

Selecting the replayed IQ data can be done with the same means as storing. Additionally, if the device supports the fast control port (FCP) interface option, the FCP can be configured to control the selected data for real-time switching of the replayed segments which can be important for time critical applications.

The most important SCPI commands for memory segments are:

**BB:ARbitrary:CLOCK <frequency>**

Set the playback rate for all replayed samples.

**BB:ARbitrary:WSEgment <segment ID>**

Select an IQ waveform segment with the specific segment ID in the memory for replay.

Further related commands can be found at the end of this Application Note. Further associated documentation is:

- Programmer's Manual [1]  
with a description of all SCPI commands
- APVSG Datasheet [2]  
with specifications on the APVSG devices.
- AN6001 [3]  
Application Note on the AnaPico IQ file format.
- AN6002 [4]  
Application Note on the Fast Control Port (FCP) interface

## Nomenclature

### IQ Sample

With AnaPico APVSG devices, an IQ sample is defined to consist of three parts. These are:

- 16 bits I (in-phase) data
- 16 bits Q (quadrature) data
- Optionally 8 marker bits (0 or 8 bits, depending on the configuration)

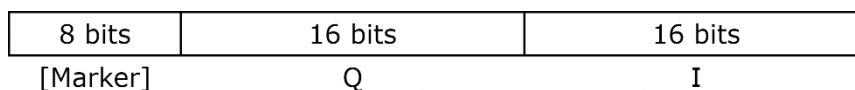


Figure 1: IQ Sample format

The optional marker bits are optional and can be utilized to mark specific samples of IQ samples within a set of samples. The different markers can each be set individually for specific samples in one waveform to then be used to wait for trigger events or to generate trigger signals. More information on the IQ sample format, marker bits and how to store samples in a file can be found in AN6001 [3].

### Marker Example

Imagine for instance a set of consecutive IQ samples that represent a period of a sine wave. One could then configure the IQ samples to contain 8 marker bits each, out of which one marks the start of the period and one to denote the sample with the maximum amplitude of Q. The former marker bit will then be 0 in all samples except for the one that starts the period. Similarly, the latter will only be 1 for the samples with maximal amplitude. In Figure 2 this example is illustrated with M0 and M1 representing the markers corresponding to the start and the maximum amplitude of Q respectively.

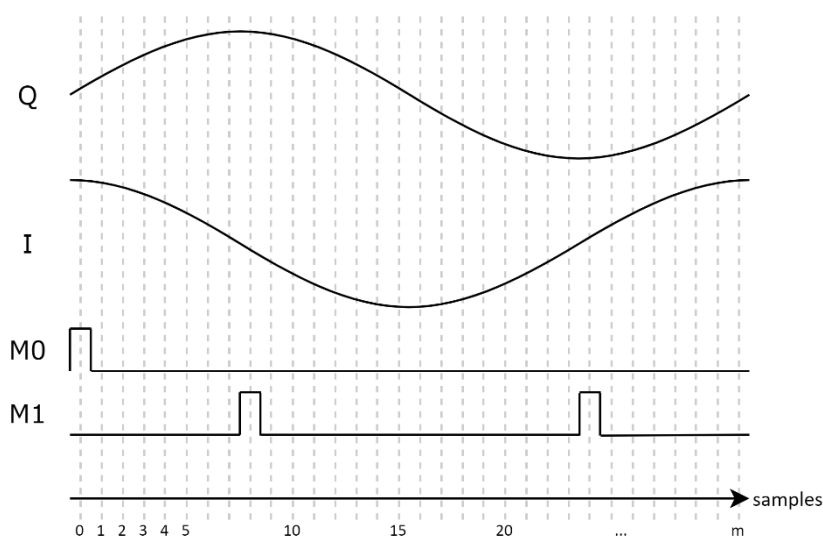


Figure 2: Example of an IQ sample that represents a period of a sin wave with marker bits. Only two out of eight marker bits are displayed, the others are omitted.

## Segment

With AnaPico devices and applications, a segment is defined as a set of multiple IQ samples. Typically, one segment represents one period of a waveform for IQ modulation of a carrier frequency. Each segment has the following properties:

- Variable size of  $m$  samples (see datasheet [2] for specifics on min/max samples per segment)
- Assigned segment ID

The segments are stored in the memory of the APVSG. Figure 3 visualizes how several segments are stored in the memory, with each segment containing multiple IQ samples.

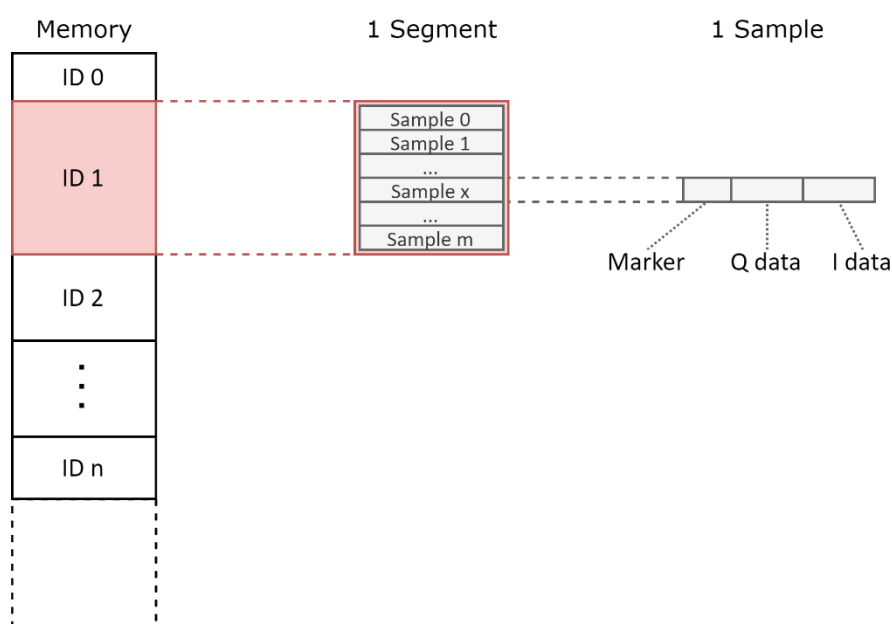


Figure 3: Illustration of a memory containing segments that each consist of multiple IQ samples.

## Sequence

For replaying different segments sequentially, they can be combined into a sequence. A sequence determines which segments are to be replayed how often and in which order. Hence, segments are repeated a specified number of times before starting with the following segment.

Additionally, sequences can be nested and combined with other sequences to be replayed consecutively in any desired order. Thereby, complex arrangements of successively replayed segment can be created. The following two examples give ideas of a possible scenario for each a simple sequencing and a sequencing with nested definitions. For more information on replaying sequences, have a look at the section *Replaying Segments in Sequences*.

**Example 1:**

Two segments are stored in the memory with segment IDs 3 and 5 respectively. A sequence then defines in what order these two segments are replayed. Figure 4 illustrates an example of a sequence that can be defined for the two stored segments. This particular sequence will have segment 3 repeated 5 times, then replay segment 5 a total of 2500 times before finally switching back to segment 3 for 40 repetitions.

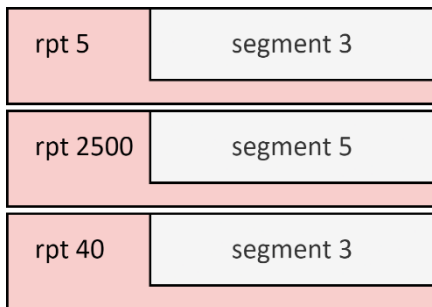


Figure 4: Example 1 - one sequence

**Example 2:**

The segments with IDs 3, 5 and 10 are stored in memory. As depicted in Figure 5 **Fehler! Verweisquelle konnte nicht gefunden werden.**, the nested sequence repeats segment 10 twice before three times replaying the whole sequence of example 1. This whole process is then repeated a total of 100 times.

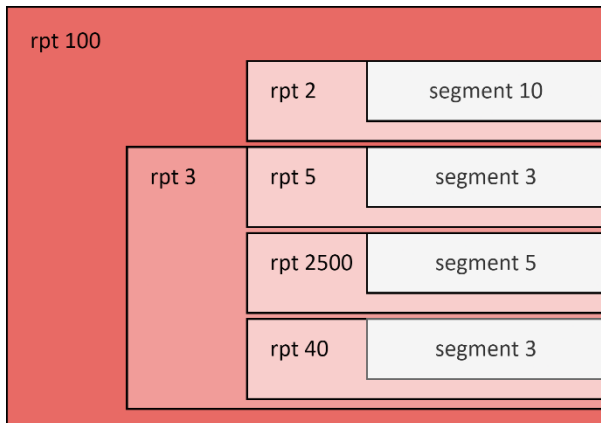


Figure 5: Example 2 - nested sequences

## Store Segments

There are two possible ways to write segments to the memory of the signal generator. Either the segments are uploaded using the APVSG graphical user interface (GUI) or the segment data is sent to the device with SCPI commands.

### Requirements and limitations:

- Each segment that is stored on the device requires a unique segment ID. It is the responsibility of the user to keep track of the IDs and assure they are existing and unique for each segment.
- Segments have a minimal number of samples required for them to be successfully stored in the memory. In case the number of samples in a segment go below this limit however, the segment is simply repeated periodically until the requirement is met and then written to the memory. Users should be aware of this limitation, especially if the concerning segment is to be replayed in a sequence.

For information on the data format used for IQ sample files, please refer to AN6001 [3].

### Upload files with the APVSG GUI

Segments can conveniently be loaded to the memory of the signal generator by loading a file with the APVSG GUI.

Note that the QI files need to be loaded in the segment tab in the RF modulation section to be assigned a unique segment ID. The AnaPico APVSG GUI supports the following features related to segment loading:

- Upload QI files (see AN6001 [3] for details on the file format) to the memory.
- Upload all QI files of a folder to a segment each.
- Progress bar with full and empty space of the memory.
- ID of each segment displayed.

### Load Block Data with SCPI Commands

The Application Note AN6001 on QI file formats [3] contains both explanations and an example on how to load data to the device's memory with SCPI commands. The most important SCPI commands for this endeavour are listed below for the reader's convenience.

#### **BB:ARbitrary:WAVEform:MARKer:STATE ON|OFF|1|0**

Enable/disable the usage of optional marker bits. It is very important to set the state of the marker bits before any data is uploaded or the samples stored in memory will not be interpreted correctly.

### **BB:ARbitrary:WAVeform:DATA [<segment ID>] <data block>**

Transfer the IQ modulation samples of the data block to the device and store them with the segment ID. If the optional segment ID is not set, it is set to zero per default. Note that segments are appended in the memory, so it is important to keep track of the remaining empty space. Please consult AN6001 [3] for a detailed explanation of the data block format, which includes a digit and byte count.

### **BB:ARbitrary:WAVeform:DATA:FREE?**

Query that returns the free space in bytes that may be written to the device.

### **BB:ARbitrary:WAVeform:DATA:DELeTe ALL**

Clears the entire waveforms memory.

## **Choose and Replay Segments**

The segments stored in the memory can be replayed to modulate the carrier frequency with the IQ data. The modulated signal can then for example be monitored at the RF output of the signal generator. Please note that the IQ modulation must be enabled for the selected segment to modulate the carrier frequency. If marker mode is enabled, two user-defined markers can be connected to the multi-function output.

Exactly one segment at a time is selected for replaying the IQ data. The user is responsible to make sure only segments existing in the memory are replayed. Selecting segments with IDs that have not been previously loaded to the memory may result in unexpected memory behaviour and unpredictable modulation of the carrier frequency. When a new segment is selected, the transition to the new segment is seamless. This means that the former segment is fully replayed up to the last sample before switching to the first sample of the consecutive segment. Switching segments involves an undefined latency during which the state of the memory is undefined.

There are three different ways to select a new segment for replay:

- in the APVSG GUI
- SCPI commands
- Directly through the fast control port (FCP)

### **Pick a Segment in the GUI**

Before selecting a segment in the APVSG GUI, make sure all the desired segments are uploaded. The successfully loaded segments are shown in the APVSG GUI and can be selected for replay. To modulate the output frequency with the selected segment, the modulation must be turned on.

## Select Segments with SCPI Commands

For selecting a segment with SCPI commands, use the following SCPI commands:

### **BB:ARbitrary:CLOCK <frequency>**

Set the playback rate to <frequency> for all replayed samples. Should be configured before selecting the memory segment.

### **BB:ARbitrary:WSEgment:SOURce INTERNAL|FCPort**

Set the segment selection source to either of the following:

Internal: Segments are selected individually in the APVSG GUI or with SCPI commands.

FCP: Segments can either be selected like in the internal mode and through segments IDs transferred through the FCP interface. New selections automatically overwrite any prior existing ones.

### **BB:ARbitrary:WSEgment <segment ID>**

Select a segment in the memory with the specific segment ID for replay.

### **BB:ARbitrary:WSEgment?**

Query that returns the currently selected segment, or zero if no segment is selected.

### **BB:ARbitrary:WSEgment:COUNt?**

Query that returns the number of currently stored segments in the memory

### Example

1. Load a segment to the memory with your favourite method. Let us assume the ID of the uploaded segment is 3.
2. SCPI command sequence for selecting a segment and modulating the carrier frequency:

<b>SOUR:SEL 1</b>	Sets the source to channel 1
<b>FREQ 50e6</b>	Sets initial RF output frequency to 50 MHz
<b>BB:ARB:WSEG:SOUR INT</b>	Set the segment selection to internal
<b>BB:ARB:CLOC 500e6</b>	Set the sample clock to 500 MHz
<b>BB:ARB:WSEG 3</b>	Select segment with ID 3
<b>BB:ARB:WSEG?</b>	Expected answer: 3 (ID of selected segment)
<b>BB:ARB:WSEG:COUN?</b>	Expected answer: 1 (stored segments)
<b>BB:ARB:WAV:STAT ON</b>	Enable the IQ modulation with memory data
<b>OUTP:STAT ON</b>	Enables the RF output

For more information on the individual SCPI commands, consult the Programmer's Manual [1].



## Replaying Segments in Sequences

### Sequencing with SCPI commands

The following SCPI commands are needed for controlling sequences with SCPI commands. Data sent or received has IEEE488.2 definite block data format:

#### **BB:ARbitrary:WSEquence:LOAD #<num\_digits><byte\_count><data\_byte>**

Send a script file that describes the sequences to the device as block data.

See the sequence script part of this chapter for details on these scripts.

<num\_digits> specifies how many digits are contained in <byte\_count>.

<byte\_count> specifies how many data bytes follow in <data\_bytes>.

Have a look at the AN6001 [3] for a detailed explanation of the data block format.

#### **BB:ARbitrary:WSEquence:LOAD:ERROR?**

Return a descriptive string of the parser error for the last sequence LOAD command. The command returns an empty string if there is no error.

#### **BB:ARbitrary:WSEquence:RUN 0 | 1**

Stop (0) or start (1) the sequencer with the loaded script.

### Sequence Script Format (.qis files)

A sequence script is used to send a description of a user defined sequence to the device. These scripts must be saved in a file ending with *.qis*. The content of the script file is typically sent to the device in block data with the *LOAD* SCPI command. As described above, the block data includes the number of digits, byte count and the script file data.

Given below are the concept and rules for sequence scripts that must be strictly adhered to.

#### Example Script

```
SEQUENCE version=0.1
# Simple test sequence with a nested sequence

Loop #repeat endlessly
  Loop repeat=2 #repeat the inner part twice
    Segment ID=2 repeat=1
    Segment ID=1 repeat=1
  End
  Segment ID=0 repeat=4
End
```

Table 1: Allowed keywords for sequence scripts

Keyword	Valid Parameters	Description
Sequence	version: script format version [date]: creation date, YYYY-MM-DD	meta information about the file
Loop / End	[repeat]: number of repeats	
Segment	id: segment id [repeat=1]: repeats the segment	plays the segment <id>

The following rules apply to all sequence scripts:

- Ignore case  
i.e. Loop = loop = LOOP = LoOp
- Ignore indentation  
Can be used for readability but has no effect on the result.
- # mark comments  
All text subsequent to a # on the same line is ignored.
- 0 or 1 commands per line  
Empty line or line with just comment is valid.
- Parameters are added in a named fashion.  
i.e. id=3
- Parameters are separated with a space from each other and from the keyword.
- Loop and End behave like brackets.  
There needs to be the same amount of End and Loop.
- Segment id=<id>  
The id parameter is mandatory.
- Script versions supported thus far:  
0.1

### Example of Segment Sequencing

1. Load the three segments with IDs 0, 1 and 2 to the memory with your favourite method.  
See chapter *Store Segments* on how to load segments.
2. Prepare a script like the one depicted in Figure 6.
3. SCPI command example for sequencing the loaded segments and modulating the carrier frequency:

<b>SOUR:SEL 1</b>	Set the source to channel 1
<b>BB:ARB:WSEG:COUN?</b>	Expected answer: 3 (stored segments, at least)
<b>BB:ARB:WSEG:LOAD &lt;data&gt;</b>	Send the script as block data
<b>FREQ 50e6</b>	Set initial RF output frequency to 50 MHz
<b>BB:ARB:WSEG:SOUR SEQ</b>	Set the segment selection to sequence
<b>BB:ARB:CLOC 500e6</b>	Set the sample clock to 500 MHz
<b>BB:ARB:WAV:STAT ON</b>	Enable the IQ modulation with memory data

**BB:ARB:WSEQ:RUN 1**  
**OUTP:STAT ON**

Start the sequencer with the loaded script  
 Enable the RF output

```
SEQUENCE version=0.1
# Simple test sequence with a nested sequence

Loop #repeat endlessly
  Loop repeat=2 #repeat the inner part twice
    Segment ID=2 repeat=1
    Segment ID=1 repeat=1
  End
  Segment ID=0 repeat=4
End
```

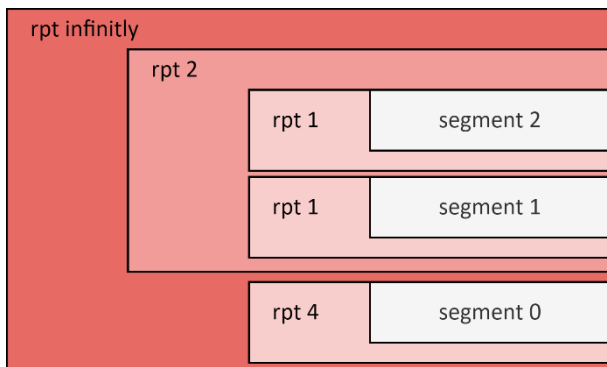


Figure 6: Script and visualization of an example sequence

### [Option] Sequencing with the Fast Control Port

One way of controlling the sequences<sup>1</sup> of segments is by deploying the fast control port (FCP) interface. FCP is an optional feature that is not included in the basic APVSG devices, so please check first if your signal generator supports it. The FCP can be used to select segments in conjunction with selecting the segments internally. The most recent selection always determines the selected segment. Note that the source of the segment selection must be set correctly, or the selection will not be acknowledged as a valid command.

Sending segment IDs to the APVSG through the FCP interface allows the user to replay a desired sequence of segments.

For details on the transmission over FCP, see the Application Note AN6002 for FCP [4] and refer to the APVSG datasheet [2] for specifications.

For a successful transmission of segment IDs over FCP, consider:

1. Load at least one segment to the memory (see section *Store Segments*) and check that the segment IDs are set as intended.
2. Connect the FCP interface to an external source that sends Segment IDs.
3. Use the APVSG GUI or SCPI commands to enable the FCP segment mode as well as IQ modulation.

SCPI commands sequence example:

**FREQ 1e9**

Set initial RF output frequency to 1 GHz

**BB:ARB:WSEG:SOUR FCP**

Set the segment selection source to FCP

**FCP:STR:SEG ON**

Enable segment ID streaming through FCP

**BB:ARB:CLOC 500e6**

Set the sample clock to 500 MHz

**BB:ARB:WAV:STAT ON**

Enable the IQ modulation with memory data

**OUTP:STAT ON**

Enable the RF output

4. Stream an arbitrary sequence of segment IDs to the FCP interface in order to switch the selected segment in the memory remotely.

<sup>1</sup> Refer to section *Sequence* at the beginning of this Application Note for more information about sequences.

## Further Documentation

- [1] AnaPico Programmer's Manual for Signal Generators  
<https://www.anapico.com/downloads/manuals/>  
*Includes description of all SCPI commands.*
- [2] APVSG Datasheet  
<https://www.anapico.com/downloads/brochures-and-data-sheets/>  
*Specifications of APVSG devices*
- [3] AN6001 APVSG - AnaPico IQ File Format  
<https://www.anapico.com/downloads/application-notes-and-videos/>  
*Includes information on the IQ data file formats and their applications.*
- [4] AN6002 APVSG – Fast Control Port Interface  
<https://www.anapico.com/downloads/application-notes-and-videos/>  
*Includes Segment Transmission over the Fast Control Port Interface*