ROHDE&SCHWARZ
Make ideas real

ARB FUNCTION FOR R&S®NGP800

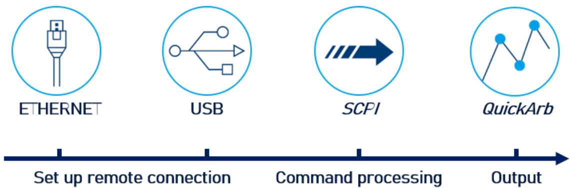
SCPI and python cheat sheet

Arbitrary procedure

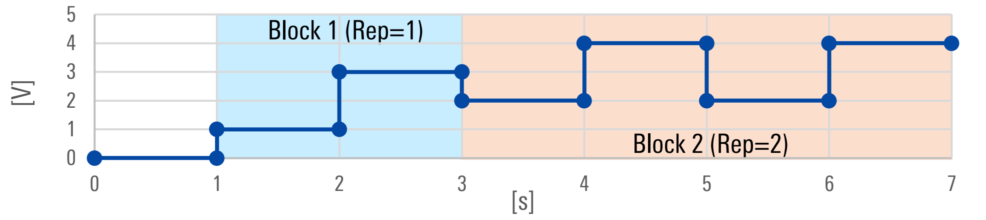
Steps

1. Set up a remote connection via **LAN, GPIB** or **USB**
2. Send SCPI commands to set and enable the arbitrary function
3. Connect your DUT

Process



Graph example for Arbitrary function



Arbitrary SCPI commands for the example

>>> INST 1	#select the output of your device
>>> ARB:BLOC 1	#select the first of eight possible blocks
>>> ARB:BLOC:DATA 1,1,0.1,0,3,3,0.1,0	#voltage1, current1, time1, interpolation, voltage2,...
>>> ARB:BLOC:REP 1	#repetition of this block only once
>>> ARB:BLOC 2	#select the second of eight possible blocks
>>> ARB:BLOC:DATA 2,2,0.1,0,4,4,0.1,0	#voltage1, current1, time1, interpolation, voltage2,...
>>> ARB:BLOC:REP 2	#repetition twice
>>> ARB:SEQ:REP 1	#repeat the sequence of the two blocks once
>>> ARB:SEQ:BEH:END HOLD	#sets end behavior for the voltage of the last block
>>> ARB:SEQ:TRAN	#transfers the arbitrary points to the channel
>>> ARB ON	#enables the arbitrary sequence
>>> OUTP ON	#turns on the output and starts the arb- sequence

Library for connection to the power supply

The library RsInstrument provides the connection between python and the power supply.

Steps	Command
Use the following pip convention to install this package:	pip install RsInstrument
After installing the package, use the following import convention:	from RsInstrument import* from time import sleep

Set up connection to your device:

```
RsInstrument.assert_minimum_version('1.10.0') #set a minimum version
ngp = RsInstrument('TCPIP::xxx.xxx.xxx.xxx::INSTR', True, True,
           "SelectVisa= 'rs', ")  #Standard LAN connection/ Control the device via RsVisa
```

Set up the arbitrary file:

```
def arb_setup():
    ngp.write('INST OUT1')  # Choose CH1
    ngp.write('ARB:BLOC 1')  # Select first block
    ngp.write('ARB:BLOC:DATA 1,1,1,0,3,3,1,0')  # Define Arb Block 1
    ngp.write('ARB:BLOC:REP 1')  # Block is repeated 1 time in sequence
    ngp.write('ARB:BLOC 2')  # Select second block
    ngp.write('ARB:BLOC:DATA 2,2,1,0,4,4,1,0')  # Define Arb Block 1
    ngp.write('ARB:BLOC:REP 2')  # Block is repeated twice
    ngp.write('ARBitrary:SEQ:REP 1')  # Sequence will be repeated once
    ngp.write('ARBitrary:SEQ:BEH:END HOLD')  #End behavior
    ngp.write('ARB:SEQ:TRAN')  # Transfer Arb sequence into memory
    ngp.query_opc()  # Check for command completion using *OPC?
```

Start the arbitrary function:

```
def arb_start():
    ngp.write('ARB ON')  # Arb is active now
    ngp.write('OUT ON')  # CH1 on (is still chosen from former sequence)
    ngp.query_opc()  # Check for command completion
```

Stop the arbitrary function:

```
def off():
    state = 1
    while state == 1: # wait until CH1 changes to OFF state, then switch off main output
        sleep(0.4)
        state = ngp.query_int('OUTPut:GEN?')  # Request CH1 state
    ngp.write('OUTPut:GENeral:STATe OFF')  # Switch off Main Output
    ngp.close()  # Close the connection finally
```

Call functions:

```
if __name__ == "__main__":
    arbsetup()
    arbstart()
    off()
```

Rohde & Schwarz GmbH & Co. KG (www.rohde-schwarz.com)
Rohde & Schwarz customer support (www.rohde-schwarz.com/support) **Rohde & Schwarz training** (www.training.rohde-schwarz.com)
R&S® is a registered trademark of Rohde & Schwarz GmbH & Co. KG | PD 3672.9415.32 | Version 01.00 | December 2023 (sa)
Trade names are trademarks of the owners | ARB function - SCPI and python cheat sheet for R&S®NGP 800 power supplies | Data without tolerance limits is not binding

Firmware Version: -