

ARB FUNCTION FOR R&S[®]NGU

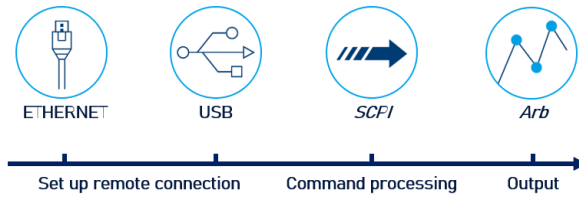
SCPI and python cheat sheet

Arbitrary procedure

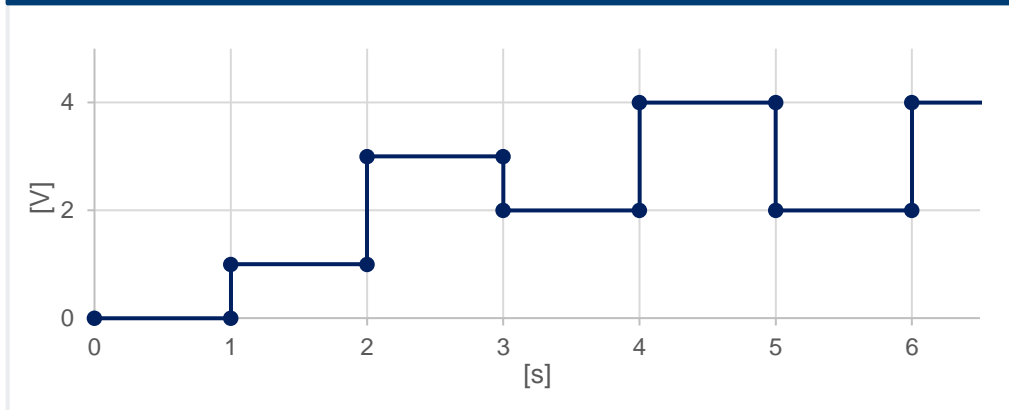
Steps

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

Process



Sample graph of the arbitrary function



Arbitrary SCPI commands (example)

```
>>> ARB:PRI:MODE CPM           #choose current priority mode
>>> ARB:DATA 1,1,1,0,3,1,1,0,2,1,1,0,4,1,1,0,2,1,1,0,4,1,1,0 #v1,c1,t1, int, v2,...
>>> ARB:REP 1                   #repetition of this sequence only once
>>> ARB:BEH:END OFF            #Switch off channel after sequence
>>> ARB:TRAN 1                 #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 RsInstrument library provides a connection for python to 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 the connection to your device:

```
RsInstrument.assert_minimum_version('1.10.0') #set a minimum version
ngu = 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():
    ngu.write('ARbitrary:PRIority:MODE CPM') # Choose Current Priority Mode
    # Define Arb Data: Voltage, Current +, Current -, Time, Interpolation
    ngu.write('ARB:DATA 1,1,1,0,3,1,1,0,2,1,1,0,4,1,1,0,2,1,1,0,4,1,1,0')
    ngu.write('ARB:REP 1') # Arb sequence is repeated once
    ngu.write('ARbitrary:BEH:END OFF') #Switch off channel after sequence
    ngu.write('ARB:TRAN 1') # Transfer Arb sequence into memory
    ngu.query_opc() # Check for command completion
```

Start the arbitrary function:

```
def arb_start():
    ngu.write('ARB ON') # Arb is active now
    ngu.write('OUTP ON') # CH1 on (is still chosen from former sequence)
    ngu.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 = ngu.query_int('OUTPut:STATe?') # Request CH1 state
    ngu.write('OUTPut:GENeral:STATe OFF') # Switch off main output
    ngu.close() # Close the connection finally
```

Save and reload:

```
def save_setup():
    """Save and reload the ARB file on the instrument"""
    ngu.write('ARbitrary:FNAME "ARB01.CSV", INT')
    ngu.write('ARbitrary:SAVE')
    ngu.write('ARbitrary:FNAME "ARB01.CSV", INT')
    ngu.write('ARbitrary:LOAD')
```