

ARB FUNCTION FOR R&S®NGM

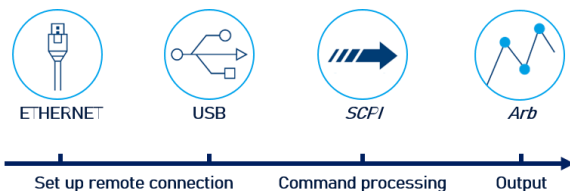
SCPI and python cheat sheet

Arbitrary procedure

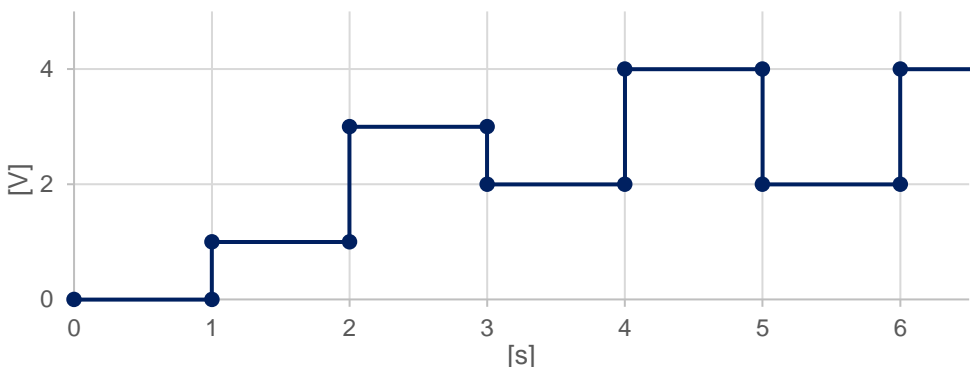
Steps

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

Process



Graph of arbitrary function



Arbitrary SCPI commands for example

```
>>> INST 1 #select the output of your device
>>> ARB:DATA 1,1,1,0,3,3,1,0,2,2,1,0,4,4,1,0,2,2,1,0,4,4,1,0 #v1, c1, t1, interpolation, v2,...
>>> ARB:REP 1 #repetition of this block only once
>>> ARB:BEH:END HOLD #sets end behavior for the voltage of the last block
>>> ARB:TRAN #transfers 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 the package:	<code>pip install RsInstrument</code>
After installing the package, use the following import convention:	<code>from RsInstrument import*</code> <code>from time import sleep</code>

Setup connection to your device:

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

Setup arbitrary file:

```
def arb_setup():
    ngm.write('INST OUT1') # Choose CH1
    ngm.write('ARB:DATA 1,1,1,0,3,3,1,0,2,2,1,0,4,4,1,0,2,2,1,0,4,4,1,0') # Define Arb Data
    ngm.write('ARB:REP 1') # Arb sequence is repeated once
    ngm.write('ARbitrary:BEH:END HOLD') #End behavior
    ngm.write('ARB:TRAN 1') # Transfer Arb sequence into memory
    ngm.query_opc() # Check for command completion using *OPC?
```

Start arbitrary function:

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

Stop 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 = ngm.query_int('OUTPut:STATe?') # Request CH1 state
    ngm.write('OUTPut:GENeral:STATe OFF') # Switch off main output
    ngm.close() # Close the connection finally
```

Save and reload:

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