ROHDE&SCHWARZ
Make ideas real

OUTPUT DELAY FUNCTION FOR THE R&S®NGU201

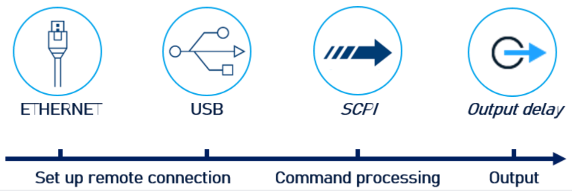
SCPI and python cheat sheet

Ramp procedure with the R&S®NGU201

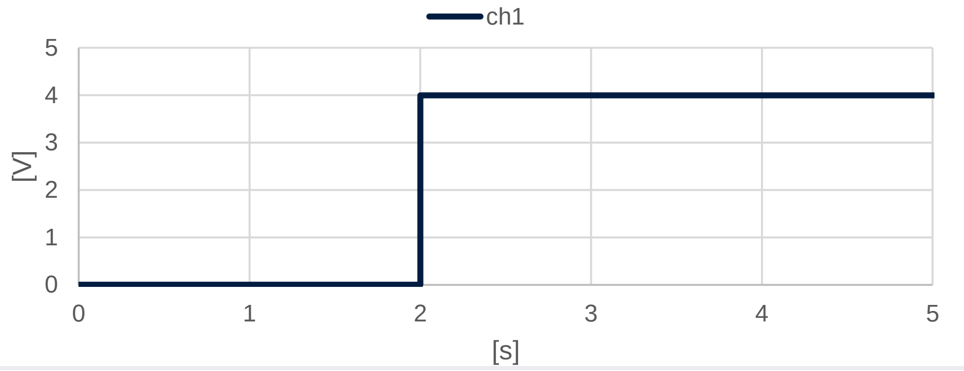
Steps

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

Process



Sample graph for output delay function



Output delay SCPI commands (example)

>>> *RST	#sets the instrument to a defined default status
>>> INST 1	#select the output 1 of your device
>>> APPL 4.0, 1.0	#voltage 4 Volts, current 1 Amps
>>> OUTP:DEL:DUR 2.0	#delay of 2.0 seconds
>>> OUTP:DEL ON	#enables the delay function
>>> OUTP:SEL ON	#enables channel 1
>>> OUTP:GEN ON	#enables output for selected channel

Library for connection to the power supply

The RsInstrument library provides a 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 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 output delay function:

```
def delay_setup(data, duration):
    ngu.write_str(f'INST 1') #choose channel
    ngu.write_str(f'APPL {data}') #set voltage and current
    ngu.write_str(f'OUTP:DEL:DUR {duration}') #set the duration of the delay
    ngu.write_str("OUTP:DEL ON") #activate delay function
    ngu.write_str("OUTP:SEL ON") #activate selected channel
```

Start the delay function:

```
def delay_start():
    ngu.write_str("OUTP:GEN ON") #switch general output on
    ngu.query_opc()  # Check for command completion
```

Stop the delay function:

```
def off(duration):
    state = 1
    sleep(duration+0.4)
    while state == 1: # wait until CH1 changes to OFF state, then switch off main output
      state = ngu.query_int('OUTP?')  # Request CH1 state
    ngu.write('OUT OFF')  # Switch off Main Output
    ngu.close()  # Close the connection finally
```

Call functions:

```
if __name__ == "__main__":
    duration = 2.0 #list of durations for each channel
    data = '4.0,1.0' #voltage values
    delay_setup(data, duration) #call delay_setup for each list item
    delay_start() #finally start the output
    off(duration)
```

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 3672.9450.32 | Version 01.00 | December 2023 (sa)
Trade names are trademarks of the owners | Output delay function - SCPI and python cheat sheet for R&S®NGU | Data without tolerance limits is not binding

Firmware Version: 03.081