

RAMP FUNCTION FOR R&S®NGU

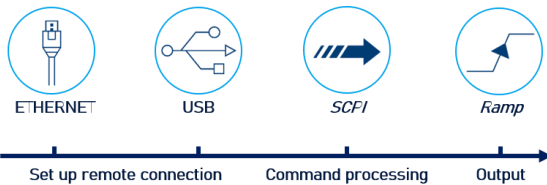
SCPI and python cheat sheet

Ramp procedure

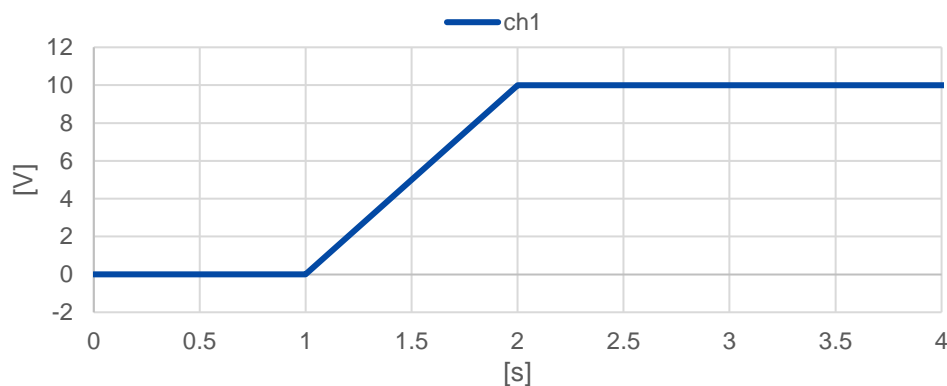
Steps

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

Process



Graph example for Ramp function



SCPI commands for the example

>>> *RST	#sets the instrument to a defined default status
>>> INST OUT1	#select the output 1 of your device
>>> APPL 10.0, 1.0	#voltage 10V, current 1A
>>> VOLT:RAMP:DUR 1.0	#ramp duration of 1 second
>>> VOLT:RAMP ON	#enables the ramp 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 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 ramp function:

```
def ramp_setup(data, duration):
    ngu.write_str(f'INST1') #choose channel
    ngu.write_str(f'APPL {data}') #set voltage and current
    ngu.write_str(f'VOLT:RAMP:DUR {duration}') #set the duration of the ramp
    ngu.write_str("VOLT:RAMP ON") #activate ramp function
    ngu.write_str("OUTP:SEL ON") #activate selected channel
```

Start ramp function:

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

Stop ramp 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('OUTP OFF') # Switch off Main Output
    ngu.close() # Close the connection finally
```

Call functions:

```
if __name__ == "__main__":
    duration = 1.0
    data = '10.0,1.0' #voltage and current value
    channel = 1 #with NGU only one channel available
    ramp_setup(data, duration) #call ramp_setup
    ramp_start() #finally start the output
    off()
```