

RAMP FUNCTION FOR R&S[®]NGL AND R&S[®]NGM

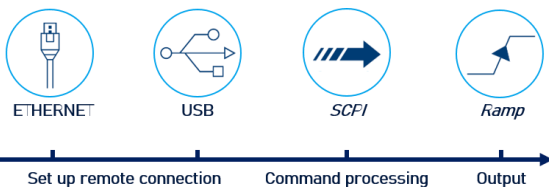
SCPI and python cheat sheet

Ramp procedure

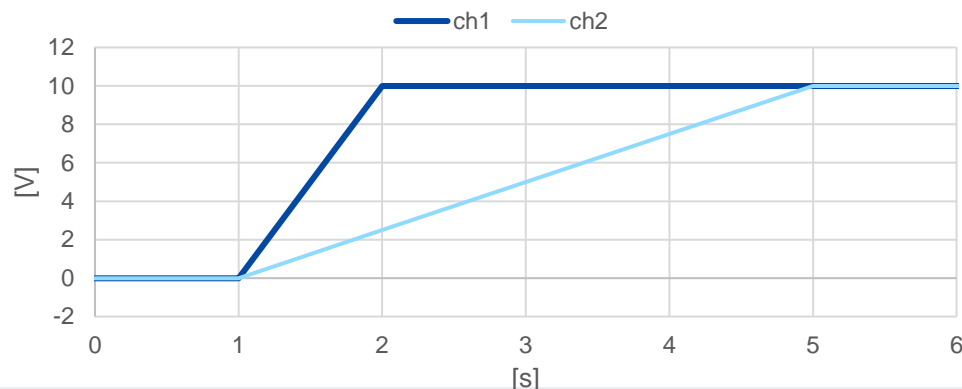
Steps

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

Process



Sample graph for ramp function with the R&S[®]NGM202



Sample ramp SCPI commands

>>> *RST	#sets the instrument to a defined default status
>>> INST OUTX	#select the output X of your device
>>> APPL 10.0, 1.0	#voltage 10V, current 1A
>>> VOLT:RAMP:DUR Y	#ramp duration Y
>>> VOLT:RAMP ON	#enables the ramp function
>>> OUTP:SEL ON	#enables channel X
>>> 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
ngm = 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, channel):
    ngm.write_str(f'INST {channel}') #choose channel
    ngm.write_str(f'APPL {data}') #set voltage and current
    ngm.write_str(f'VOLT:RAMP:DUR {duration}') #set the duration of the ramp
    ngm.write_str("VOLT:RAMP ON") #activate ramp function
    ngm.write_str("OUTP:SEL ON") #activate selected channel
```

Start ramp function:

```
def ramp_start():
    ngm.write_str("OUTP:GEN ON") #switch general output on
    ngm.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 = ngm.query_int('OUTPut:STATe?') # Request CH1 state
    ngm.write('OUTPut:GENeral:STATe OFF') # Switch off Main Output
    ngm.close() # Close the connection finally
```

Call functions:

```
if __name__ == "__main__":
    durations = [1.0, 4.0] #list of durations for each channel
    data = '10.0,1.0' #voltage and current value
    channel = [1, 2] #list of channels, if NGM201 or NGL201 only one channel available
    for i in range(0, len(channel)):
        ramp_setup(data, durations[i], channel[i]) #call ramp_setup for each list item
    ramp_start() #finally start the output
    off()
```