

OUTPUT DELAY FUNCTION FOR R&S®NGM202

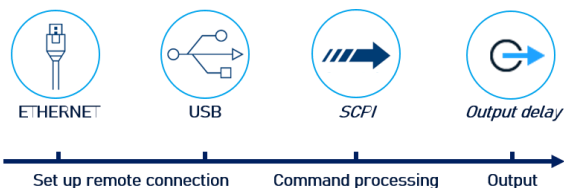
SCPI and python cheat sheet

Ramp procedure with the R&S®NGM202

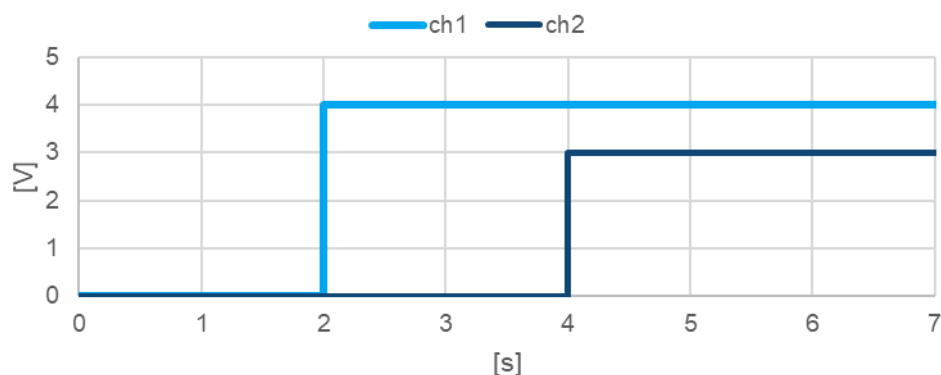
Steps

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

Process



Sample graph of output delay function



Output delay SCPI commands (example)

>>> *RST	#sets the instrument to a defined default status
>>> INST X	#select the output X for your device
>>> APPL A, B	#voltage A Volts, current B Amps
>>> OUTP:DEL:DUR Y	#delay Y seconds
>>> OUTP:DEL ON	#enables delay 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 the 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 output delay function:

```
def delay_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'OUTP:DEL:DUR {duration}') #set the duration of the delay
    ngm.write_str("OUTP:DEL ON") #activate delay function
    ngm.write_str("OUTP:SEL ON") #activate selected channel
```

Start delay function:

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

Stop delay function:

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

Call functions:

```
if __name__ == "__main__":
    duration = [2.0, 4.0] #list of durations for each channel
    data = ['4.0,1.0', '3.0,1.0'] #voltage/current values
    for i in range(0,len(data)):
        delay_setup(data[i], duration[i],i+1) #call delay_setup for each list item
    delay_start() #finally start the output
    off()
```