ROHDE&SCHWARZ
Make ideas real

# OUTPUT DELAY FUNCTION FOR R&S®NGP804

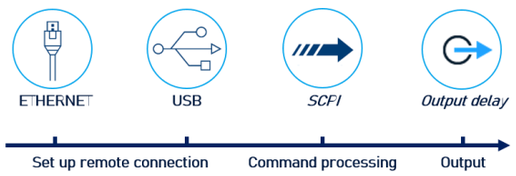
## SCPI and python cheat sheet

### Ramp procedure with the R&S®NGP804

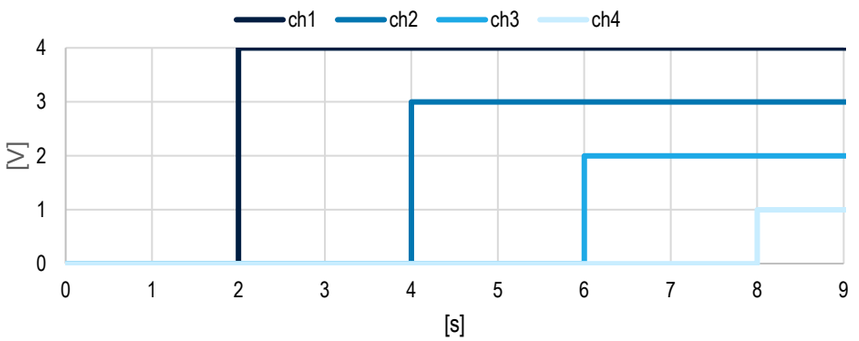
**Steps**

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

**Process**




### Graph example for output delay function




### Output delay SCPI commands (example)

| | |
|---|---|
| >>> *RST | #sets instrument to a defined default status |
| >>> INST X | #select output X of your device |
| >>> APPL A, B | #voltage A Volts, current B Amps |
| >>> OUTP:DEL:DUR Y | #delay of 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 connects 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*<br>from time import sleep |

### Set up connection to your device:

```
RsInstrument.assert_minimum_version('1.10.0') #set a minimum version
ngp = 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(voltage, current, duration, channel):
    ngp.write_str(f'INST {channel}') #choose channel
    ngp.write_str(f'APPL {voltage},{current}') #set voltage and current
    ngp.write_str(f'OUTP:DEL:DUR {duration}') #set the duration of the delay
    ngp.write_str("OUTP:DEL ON") #activate delay function
    ngp.write_str("OUTP:SEL ON") #activate selected channel
```

### Start delay function:

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

### Stop delay function:

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

### Call functions:

```
if __name__ == "__main__":
    durations = [1.0, 2.0, 4.0, 8.0] #list of durations for each channel
    data = [4,3,2,1] #voltage values
    channel = [1, 2, 3, 4] #list of channels
    for i in range(0, len(channel)):
        delay_setup(data[i], data[i], durations[i], channel[i]) #call ramp_setup for each list item
    delay_start() #finally start the output
    off()
```


**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 3672.9444.32 | Version 01.00 | December 2023 (sa)



Firmware Version: 02.015