Remote Measurement Setup with Tektronix DPO4054 and Python

This guide explains how to remotely configure, trigger, and measure signals on a Tektronix MSO4000/DPO4000 oscilloscope using the DPO4054 Python class and SCPI commands.

1. Connecting to the Oscilloscope

The oscilloscope can be controlled via USB, Ethernet, or GPIB. Example for USB connection:

```python
from Instruments import DPO4054
osc = DPO4054('USB0::0x0699::0x0401::C020132::INSTR')
```

- Replace the VISA resource string with your instrument's address.

2. Setting Up the Timebase

Set the horizontal timebase scale (seconds/division):

```python
osc.set_timebase_scale(2/(5e3))  # e.g., for a 5 kHz signal, 2 cycles on screen
```

3. Configuring a Channel

Configure vertical settings for a channel:

```python
osc.setup_channel(
    3,                    # Channel number (e.g., CH3)
    display=True,         # Show the channel trace
    scale=0.1,            # Volts/div (e.g., 0.1 V/div)
    position=0.0,         # Vertical offset (divisions)
    coupling='AC',        # Coupling mode ('AC' or 'DC')
    bandwidth_limit='20MHz'  # Bandwidth limit (optional)
)
```

- This sets up the vertical scale, position, coupling, and bandwidth for CH3.

4. Setting Up the Trigger

Configure an edge trigger:

```python
osc.setup_edge_trigger(
    source_channel=3,   # Trigger on channel 3
    level=50e-3,        # Trigger level (e.g., 50 mV)
    slope='RISE'        # Trigger on rising edge
)
```

5. **Adding a Measurement**

Add a measurement (e.g., frequency on CH3, using slot 1):

```python
osc.add_measurement('FREQ', 3, 1)
```

- `meas_type`: Measurement type (see Table 2-8 in the manual, e.g., 'FREQ', 'MEAN', 'RMS', etc.)
- `channel`: Channel number (1–4)
- `slot`: Measurement slot (1–8)

6. **Reading a Measurement Value**

Wait for the oscilloscope to update, then read the measurement:

```python
import time
time.sleep(2)  # Wait for measurement to settle
print(osc.get_measurement(3, 1))  # Get the measurement value from slot 1, channel 3
```

7. **Clearing Measurements**

To remove all measurements:

```python
osc.clear_all_measurements()
```

This disables and clears all measurement slots (1–8).

8. **Relevant SCPI Commands**

- **Set measurement type:** `MEASUREMENT:MEAS{slot}:TYPE {type}`
- **Set measurement source:** `MEASUREMENT:MEAS{slot}:SOURCE1 CH{channel}`
- **Enable measurement:** `MEASUREMENT:MEAS{slot}:STATE ON`
- **Query measurement value:** `MEASUREMENT:MEAS{slot}:VALUE?`

- **Clear measurement:** `MEASUREMENT:MEAS{slot}:TYPE NONE`

Refer to Table 2-26: Measurement Commands in the Programmer Manual for full details.

---

9. **Example Full Measurement Script**

```python
from Instruments import DPO4054
osc = DPO4054('USB0::0x0699::0x0401::C020132::INSTR')

osc.set_timebase_scale(2/(5e3))
osc.setup_channel(3, display=True, scale=0.1, position=0.0, coupling='AC',
bandwidth_limit='20MHz')
osc.setup_edge_trigger(source_channel=3, level=50e-3, slope='RISE')
osc.add_measurement('FREQ', 3, 1)

import time
time.sleep(2)
print("Frequency on CH3:", osc.get_measurement(3, 1))
```

---

10. **References**

- Tektronix MSO4000/DPO4000 Programmer Manual, Table 2-26
- Tektronix Support