Report: Continuous Acquisition and Triggered Measurement Loop on Tektronix DPO4054

Introduction

This report discusses a Python-based approach to configure the Tektronix DPO4054 oscilloscope for continuous acquisition with trigger-based measurement. The code demonstrates:

- Channel configuration
- Trigger setup for an analog input signal
- Adding a frequency measurement in a measurement slot
- Running the oscilloscope in sequence acquisition mode
- Polling for trigger completion and retrieving measurement results
- Returning the oscilloscope to continuous acquisition mode

Code Overview

```
# Configure channel 3 for measurement
osc.setup_channel(
    channel=3,
    label='clk',
    display=True,
    scale=100e-3,      # 100 mV/div vertical scale
    position=0.0,
    offset=0.0,
    coupling='AC',     # AC coupling for analog signal
    bandwidth='FULL',
    invert=False
)

# Setup trigger on channel 3
osc.trigger_setup(
    channel=3,
    level=80e-3,       # Trigger level at 80 mV (approx 80% of 100 mV scale)
    slope='RISE',
    mode='EDGE'
)

# Add frequency measurement on channel 3, assigned to slot 1
osc.add_measurement(channel=3, meas_type='FREQ', slot=1, source=1)

# Set oscilloscope to sequence acquisition mode (triggered acquisitions stored in
memory segments)
osc.set_acquire_sequence()

volt = 0
while volt < 3.3:  # Simulate voltage ramp from 0 to 3.3 V
```

```
    if osc.get_operation_status():  # Poll for trigger completion
        sleep(0.5)  # Wait for acquisition to complete
        freq = osc.get_measurement(channel=3, slot=1, source=1)  # Retrieve
frequency measurement
        print(f'FREQ immediate : {freq}')
        break
    volt += 0.1  # Increment simulated voltage

# Return oscilloscope to continuous acquisition mode
osc.set_acquire_continuous()
```

Explanation of Key Components

1. Channel Setup

- Channel 3 is configured with a label `"clk"` and enabled for display.
- The vertical scale is set to 100 mV/div, suitable for small analog signals.
- AC coupling is selected to block DC offset and focus on AC signal components.
- Full bandwidth is enabled to avoid bandwidth limiting.

2. Trigger Configuration

- Trigger source is channel 3.
- Trigger level is set to 80 mV, approximately 80% of the vertical scale, appropriate for detecting signal edges.
- Rising edge trigger mode is selected to capture signal transitions from low to high.

3. Measurement Setup

- A frequency measurement (`'FREQ'`) is assigned to measurement slot 1 on channel 3.
- This slot-based measurement allows efficient repeated queries of frequency after each trigger event.

4. Acquisition Mode

- The oscilloscope is set to **sequence acquisition mode** via `set_acquire_sequence()`.
- Sequence mode allows multiple triggered acquisitions to be stored sequentially in segmented memory.

5. Trigger Loop and Measurement Retrieval

- A simulated voltage variable `volt` increments from 0 to 3.3 V to mimic an external condition or stimulus.
- In each loop iteration, the code polls the oscilloscope with `get_operation_status()` to check if the trigger acquisition is complete.
- Upon trigger completion, the code waits 0.5 seconds to ensure data is ready.
- The frequency measurement is retrieved from slot 1 and printed.
- The loop breaks after the first successful trigger acquisition.

6. Return to Continuous Mode

- After measurement, the oscilloscope is set back to continuous acquisition mode with `set_acquire_continuous()`.
- This allows the scope to run freely and continuously acquire data.

Operational Flow Summary

Step	Description
Channel setup	Configure channel 3 for AC coupled analog input
Trigger setup	Set trigger on channel 3, rising edge, 80 mV
Measurement assignment	Add frequency measurement on channel 3, slot 1
Acquisition mode	Switch to sequence mode for triggered captures
Trigger loop	Poll for trigger completion, retrieve frequency
Exit loop	Break after first trigger acquisition
Reset acquisition mode	Return to continuous acquisition mode

Code Snippets for Key Operations

Channel Setup

```
osc.setup_channel(channel=3, label='clk', display=True, scale=100e-3,
position=0.0,
                  offset=0.0, coupling='AC', bandwidth='FULL', invert=False)
```

Trigger Setup

```
osc.trigger_setup(channel=3, level=80e-3, slope='RISE', mode='EDGE')
```

Add Measurement

```
osc.add_measurement(channel=3, meas_type='FREQ', slot=1, source=1)
```

Set Sequence Acquisition Mode

```
osc.set_acquire_sequence()
```

Trigger Polling Loop

```
volt = 0
while volt < 3.3:
    if osc.get_operation_status():
        sleep(0.5)
        print(f'FREQ immediate : {osc.get_measurement(channel=3, slot=1,
source=1)}')
        break
    volt += 0.1
```

Return to Continuous Mode

```
osc.set_acquire_continuous()
```

Conclusion

This example demonstrates a practical approach to:

- Configure an oscilloscope channel and trigger for analog signal measurement.
- Use sequence acquisition mode to capture triggered events.
- Poll the instrument for trigger completion in a loop.
- Retrieve frequency measurements assigned to a measurement slot.
- Return the instrument to continuous acquisition after measurement.

This approach is suitable for automated testing scenarios where triggered measurements are synchronized with external events or stimuli.