Test automation utility for event timing modules

Di WANG

LINAC Control Group

KEK

20250807

Background: The Timing System Upgrade

- EVG & EVR
  - Event Generator (EVG) sends event codes to Event Receiver (EVR)
  - EVR generates trigger pulse to accelerator devices (BPM, magnet, etc.)

- Current Landscape:
  - KEK LINAC relies on a large, distributed timing system with over 100 Event Receivers (EVRs).

- The VME platform is becoming obsolete; key components like the MVME5500/MVME6100 CPU are being discontinued.

- We are migrating our timing system from VME to the modern MicroTCA (MTCA) standard. This necessitates rigorous validation of all new hardware.
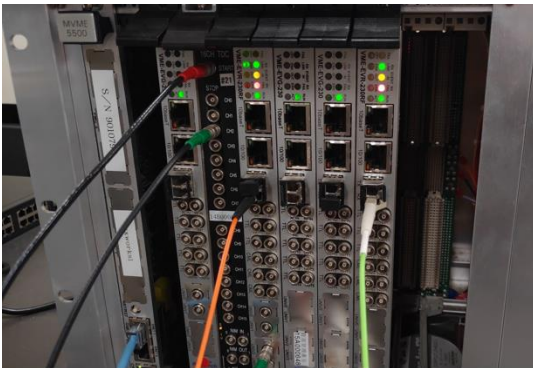
Overview of timing stations at LINAC
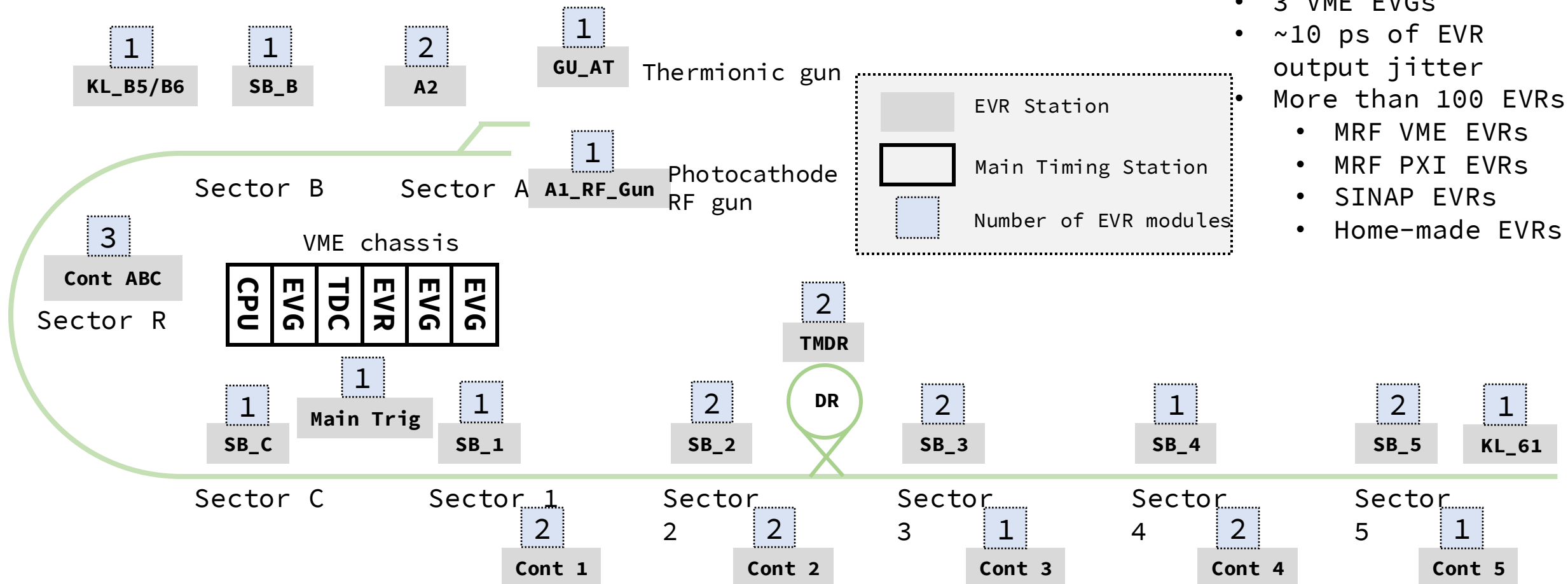
- 21 timing stations
- 3 VME EVGs
- ~10 ps of EVR output jitter
- More than 100 EVRs
  - MRF VME EVRs
  - MRF PXI EVRs
  - SINAP EVRs
  - Home-made EVRs

1 KL_B5/B6

1 SB_B

2 A2

1 GU_AT — Thermionic gun

1 A1_RF_Gun — Photocathode RF gun

3 Cont ABC

Sector B   Sector A

Sector R

VME chassis

| CPU | EVG | TDC | EVR | EVG | EVG |

2 TMDR

DR

1 Main Trig

1 SB_C

1 SB_1

2 SB_2

2 SB_3

1 SB_4

2 SB_5

1 KL_61

Sector C   Sector 1   Sector 2   Sector 3   Sector 4   Sector 5

2 Cont 1

2 Cont 2

1 Cont 3

2 Cont 4

1 Cont 5

Legend:
- EVR Station
- Main Timing Station
- Number of EVR modules

# Event Timing Modules: Evolving System

- **Hardware Diversity:**
  - MRF provides various of products across several HW form factors, e.g., VME, CPCI/PXI, PCIe, PMC, MTCA

- **Constant Evolution: The timing modules are a moving target.**
  - New Features: Newer module series (e.g., the 300-series) add advanced capabilities like delay compensation, segmented data buffer and flip-flop output.
  - Frequent Updates: Continuous firmware and EPICS driver releases introduce improvements and bug fixes that require re-validation.

From	To	Note
VME-EVG-230	VME-EVR-230	In Ope
VME-EVG-230	VME-EVR-230RF	In Ope
VME-EVG-230	PXI-EVR-230	In Ope
VME-EVG-230	Home-made EVR	In Ope
VME-EVG-230	MTCA-EVR-300RF	??
VME-EVG-230	MTCA-EVR-300U	??
VME-EVG-230	VME-EVR-300	??
MTCA-EVM-300	MTCA-EVR-300U	○
MTCA-EVM-300	MTCA-EVR-300RF	○
MTCA-EVM-300	VME-EVR-300	○
MTCA-EVM-300	VME-EVR-230RF	??
MTCA-EVM-300	VME-EVR-230	??
MTCA-EVM-300	PXI-EVR-230	??
MTCA-EVM-300	Home-made EVR	??

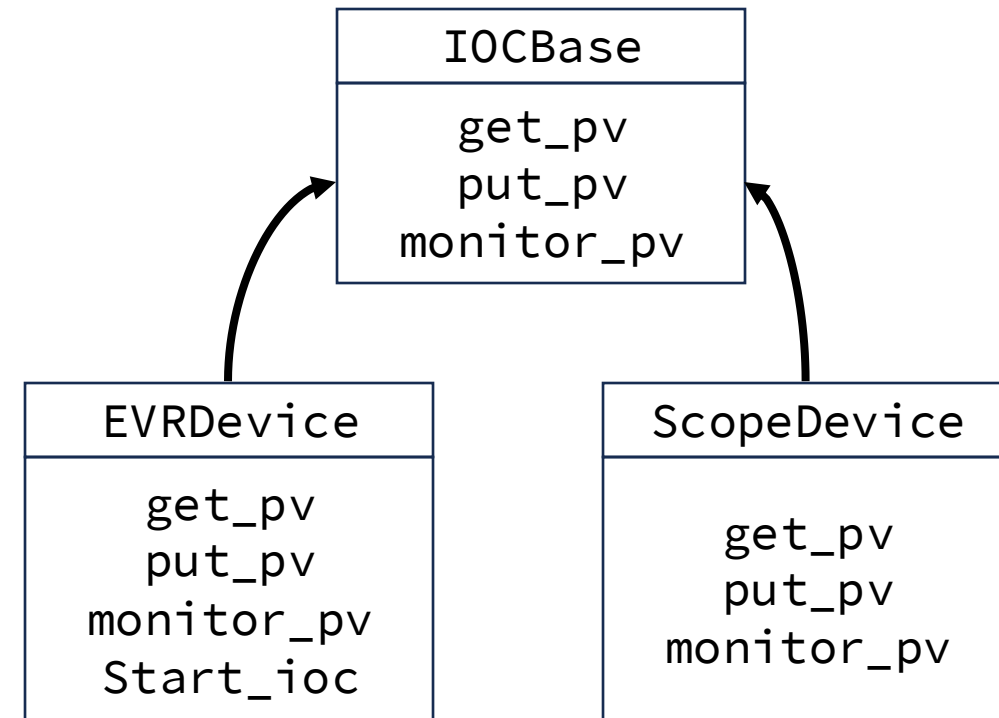| 140207 | 11.5.2022 | - Added pulse generator masking capability for HW set and reset | mTCA-EVR-300DC |
150207	15.6.2022	- Added VME-UTB-64x support for VME-EVR-300	VME-EVR-300
160207	6.10.2022	- Changed pulse with and delay to 32 bits on all pulse generators.	mTCA-EVR-300DC and VME-EVR-300
170207	6.10.2022	- Use LOS from SFP transceiver to reset receiver.	mTCA-EVR-300DC
180207	17.01.2023	- Added support for new mTCA-EVR-300DC hardware with improved backplane triggers	mTCA-EVR-300DC

Challenge

- ## The Upgrade Challenge:
  - We are in a long-term transition, phasing out obsolete VME hardware.
  - This results in a complex, mixed-generation environment where legacy and new modules must operate together.

- ## The Bottleneck:
  - Manually testing hundreds of timing modules in this dynamic environment is time-consuming, prone to human error, and lacks repeatability.

- ## An automated solution is beneficial:
  - Develop a robust, automated, and reliable Site Acceptance Test (SAT) suite.
  - Increase test coverage and consistency.
  - Reduce manual intervention and testing time.
  - Generate automated test reports for validation and records.

Test Automation Utility Architecture
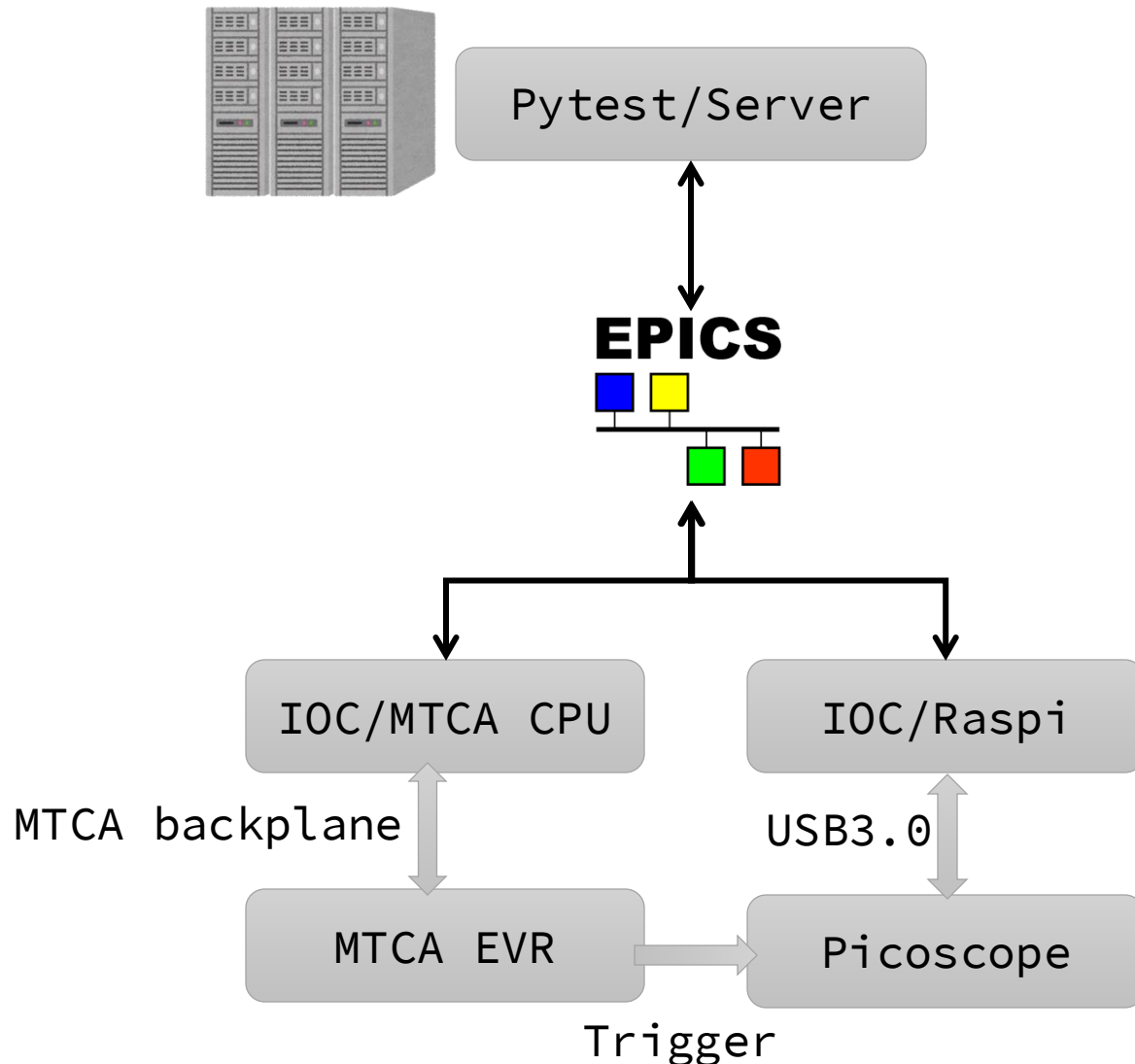
- Hardware:
  - MicroTCA
    - VadaTech Chassis: VT814
    - VadaTech MCH: UTC004
    - VadaTech CPU: AMC725
    - MRF MTCA-EVR-300RF
  - Raspberry Pi
  - Picoscope
- Software:
  - uv for python management
  - Pytest
- Workflow: pytest script sends commands via EPICS Channel Access (CA) to both the EVR and Picoscope to execute and verify tests.

- Why pytest? Fixtures for resource management, powerful assert statements, extensive plugins (e.g., pytest-html for reports), and simple syntax

Framework design

- Framework Logic: Leveraging pytest fixtures to manage connections and hardware states.
  - IOCBase: A base class for inheritance
  - EVRDevice: Establishes connection to the EVR IOC, ensures it's in a known state.
  - ScopeDevice: Connects to the Picoscope IOC, configures capture settings (trigger, voltage range, timebase).

- Test Organization:
  - test_basic.py: For basic functions, EVR status checks, event code reception etc.
  - test_fpout.py: For validation of front-panel trigger output generation.
  - test_fpinp.py: For validation of interrupt from front-panel input signal.

IOCBase

get_pv
put_pv
monitor_pv

EVRDevice

get_pv
put_pv
monitor_pv
Start_ioc

ScopeDevice

get_pv
put_pv
monitor_pv

Configurable Tests

- Tests are described in a Human-readable configuration YAML file

- Config items
  - PV prefix for EVR IOC and Picoscope IOC
  - Whether start a new IOC using python subprocess or using existing one?
  - If start IOC using python, where is binary program for IOC, start command, startup path?
  - Which EVR output port connect to which Picoscope channel? E.g., front panel universal output 1 connect to Picoscope channel A
  - Using PV Access or Channel Access to connect to IOC (python module confliction, still working on)

```yaml
mtca-evr-300rf-example:
  prefix: "TT:MTCA-EVR-300:"
  ioc_start: "YES"
  ioc_exe: "/path/to/epics/modules/mrfioc2/bin/linux-x86_64/mrf"
  ioc_cmd: "evr300.cmd"
  cmd_path: "/path/to/epics/ioc/mrf300-test-ioc/mtca"
  hardware: "MTCA-EVR-300RF"
  use_scope: "YES"
  scope_prefix: "TT:PS3405D:"
  scope_port: "FPUV1"
  scope_channel: "A"
  evr_name: "EVR"
  use_pva: "NO"

vme-evr-230-default:
  prefix: "TT:VME-EVR-230:"
  ioc_start: "NO"
  hardware: "VME-EVR-230"
  use_scope: "YES"
  scope_prefix: "TT:PS3405D:"
  scope_port: "FP1"
  scope_channel: "B"
  evr_name: "EVR230"
  use_pva: "NO"
```

Basic EVR function tests

- Verify basic functions of the EVR.

- Use @pytest.mark.parametrize to define multiple sets of arguments and fixtures at the test function.

- Test item examples:
  - Event Code Receiving: Configure an event code counter to a specific event and verify event rate.
  - Heartbeat & Status: Poll status PVs to ensure the module is alive and reporting no errors.

```python
@pytest.mark.parametrize(
    "suffix, value",
    [
        ("DCEna-Sel", "Enable"),
        ("DCLck-Sts", "Locked"),
        ("DCLckTrk-Sts", "Lock"),
        ("DCLckDat-Sts", "Fine"),
    ],
)
def test_dc(evr_device, suffix, value):
    """
    Check Delay Compensation function
    """
    if "230" in evr_device.hardware:
        pytest.skip("This test only applies to EVR 300 series")
    ret = execute_and_assert(lambda: evr_device.get_pv(suffix, as_string=True))
    assert ret == value
```

- Confirm that receiving a specific event code produces the correct electrical pulse at the physical output ports.
- This requires synchronizing an action (sending an event) with a measurement (capturing a waveform).
- Method:
  - Configure: Set up the EVR to output a pulse (with different pulse width) on a specific channel for a given event code.
  - Arm: Arm the Picoscope to trigger on the expected pulse.
  - Execute: Send the event command to the event receiver via EPICS.
  - Verify: The Picoscope triggers, captures the waveform, and the test script fetches and analyzes the data.
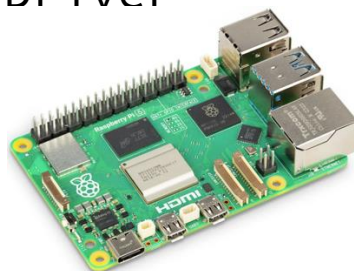
EPICS driver for Picoscope

- Raspberry Pi 5
  - Although AMC725 has 2 mini-USB, A portable Raspi is preferred
- Picoscope 3405D, 4-ch, 100 MHz, 8-bit at 1 GS/s
  - Only require USB & Linux Support
- Software
  - Picoscope Linux drivers and API for 32-bit Raspberry Pi OS
  - Asynps3000a epics module (C++) based on asynPortDriver


```
#!../../bin/linux-arm/testasynps3000a
< envPaths
epicsEnvSet("IOC","ioctestasynps3000a")
epicsEnvSet("TOP","/home/pi/epics/R7.0.9/modules/asynps3000a")
epicsEnvSet("ASYN","/home/pi/epics/R7.0.9/modules/asyn-4-45")
epicsEnvSet("EPICS_BASE","/home/pi/epics/R7.0.9/base")
cd "/home/pi/epics/R7.0.9/modules/asynps3000a"
## Register all support components
dbLoadDatabase "dbd/testasynps3000a.dbd"
testasynps3000a_registerRecordDeviceDriver pdbbase
epicsEnvSet("PP",                        "TT")
epicsEnvSet("DEVICE_NAME",               "PS3405D")
epicsEnvSet("EPICS_CA_MAX_ARRAY_BYTES",  "10000000")
epicsEnvSet("PREFIX",                    "TT:PS3405D:")
epicsEnvSet("PORT",                      "PICO")
epicsEnvSet("MAX_SAMPLES",               "10000")
PS3000AConfig("PICO", "10000")
[INF] PS3000A:PS3000A:229 PicoScope 3000A Series Driver
[INF] PS3000A:getInfo:322 device information:
Driver Version: PS3000A Linux Driver, 2.2.153.7541
USB Version: 3.0
Hardware Version: 1
Variant Info: 3405DMSO
Serial: KP361/0179
Cal Date: 08Feb24
Kernel: 0.0
Digital H/W: 1
Analogue H/W: 1
Firmware 1: 1.7.5.0
Firmware 2: 1.0.67.0
dbLoadRecords("db/ps3000a.template", "P=TT:PS3405D:,R=,PORT=PICO,ADDR=0,TIMEOUT=1,MAX_SAMPLES=10000")
dbLoadRecords("db/ps3000aN.template","P=TT:PS3405D:,R=A-,PORT=PICO,ADDR=0,TIMEOUT=1,NAME=Channel A,MAX_SAMPLES=10000")
dbLoadRecords("db/ps3000aN.template","P=TT:PS3405D:,R=B-,PORT=PICO,ADDR=1,TIMEOUT=1,NAME=Channel B,MAX_SAMPLES=10000")
dbLoadRecords("db/ps3000aN.template","P=TT:PS3405D:,R=C-,PORT=PICO,ADDR=2,TIMEOUT=1,NAME=Channel C,MAX_SAMPLES=10000")
dbLoadRecords("db/ps3000aN.template","P=TT:PS3405D:,R=D-,PORT=PICO,ADDR=3,TIMEOUT=1,NAME=Channel D,MAX_SAMPLES=10000")
cd "/home/pi/epics/R7.0.9/modules/asynps3000a/iocBoot/ioctestasynps3000a"
iocInit
Starting iocInit
#############################################################################
## EPICS R7.0.9
## Rev. 2025-06-04T18:00+0900
## Rev. Date build date/time:
#############################################################################
iocRun: All initialization complete
dbpf "TT:PS3405D:A-Enabled" "1"
```


IOC for Picoscope

OPI for Picoscope IOC

Info

EPICS name	TT:PS3405D:
Model	3405DMSO
EPICS Driver Ver	0.1.0
Pico Driver Ver	PS3000A Linux Driver, 2.2.153.7541
Pico USB Ver	3.0
Pico HW Ver	1
Pico Serial	KP361/0179
Pico Cal Date	08Feb24
Pico Kernel	0.0
Pico Firmware 1	1.7.5.0
Pico Firmware 2	1.0.67.0
Connection	Connect Disconnect
	Connected
Debugging	Open asynRecord

Acq Control

Time tick		1000.000 ns
Time base	127	127
Nr. pre trig samples	500	500
Nr. post trig samples	4500	4500
Total nr. samples		5000
Max. nr. samples		10000
Trigger repeat	No	No
Acquire	Start Stop	
Status	Idle	
Maximum segments		500000
Maximum value		32512
Minimum value		-32512
Trigger Enabled		Yes
PWQ Enabled		No

Channels

Trigger mode   Simple   Simple

| Ch A | Ch B | Ch C | Ch D |
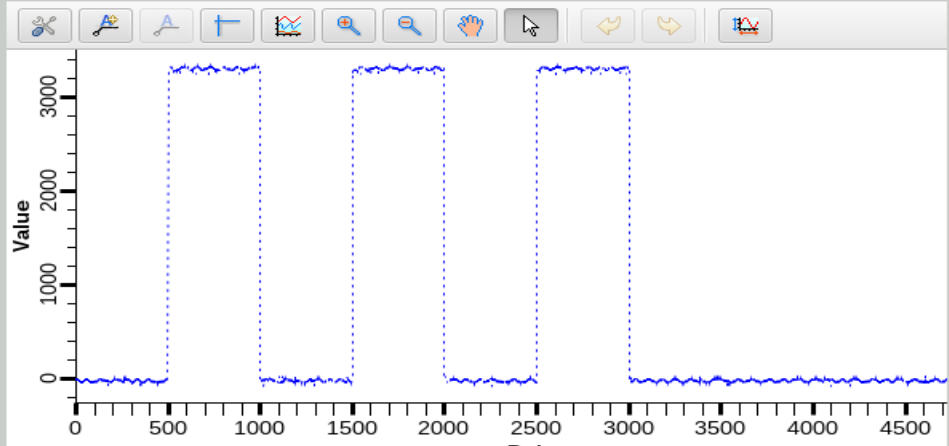
Channel A

Enabled	Yes	Yes
Use trigger	Yes	Yes
Coupling	DC	DC
Range	5 V	5 V
Convert	mV	mV

Simple trigger

Threshold (ADC)	200	200
Threshold (mV)	3000	3000
Direction	Above	Above
Delay	0	0
Timeout	0	0

Advanced trigger

Condition	Don't Care	Don't Care
Direction	Above	Above
Thresh. Upper	200	200
Thresh. Upper Hyst.	180	180
Thresh. Lower	180	180
Thresh. Lower Hyst.	200	200
Thresh. Mode	Level	Level
Delay	0	0

Pulse Width Qualifier

Condition	Don't Care	Don't Care
Direction	Above	Above
Type	None	None
Limit Upper	50	50
Limit Lower	10	10

TT:PS3405D:A-Waveform_RBV

Test Examples

GitLab Automation

1. Hardware installation and cable connection

2. Add new tests/new config

3. Push commits to GitLab

4. Specify IOC name using GitLab CI/CD variables

5. GitLab Runner picks up the task
   1. Prepare environment
   2. Login to MTCA server
   3. (Optional) start IOC
   4. Run tests
   5. Collect and upload artifacts to GitLab

6. Examine test report

Jobs / #1374 / **Artifacts**

⚠ failed (allowed to fail) Job **#1374** in pipeline **#877** for bbe89318 from main by Di Wang 1 hour ago

Artifacts / reports   ⬇ Download artifacts archive

Name	Size
📁 ..	
📄 report.html	99.5 KiB

```
377     def test_dc(evr_device, suffix, value):
378         """
379         Check Delay Compensation function
380         """
381         if "230" in evr_device.hardware:
382             pytest.skip("This test only applies to EVR 300 series")
383         ret = execute_and_assert(lambda: evr_device.get_pv(suffix, as_string=True))
384 >       assert ret == value
385 E       AssertionError: assert 'Medium' == 'Fine'
386 E
387 E         - Fine
388 E         + Medium
389 tests/test_basic.py:139: AssertionError
390 - Generated html report: file:///nfs/linacfs-users/sdcswd/tmp/mrfioc2-tests/reports/report.html -
391 =========================== short test summary info ===========================
392 FAILED tests/test_basic.py::test_stable_errors[CntLinkTimo-I] - assert 71 == 64
393 FAILED tests/test_basic.py::test_event_interval - assert 1.3353432038700624e-...
394 FAILED tests/test_basic.py::test_dc[DCLck-Sts-Locked] - AssertionError: asser...
395 FAILED tests/test_basic.py::test_dc[DCLckDat-Sts-Fine] - AssertionError: asse...
396 ==================== 4 failed, 102 passed in 296.49s (0:04:56) ====================
397 Running after_script
398 Running after script...
399 $ which ssh-agent || ( apt-get update -y && apt-get install openssh-client -y )
400 /usr/bin/ssh-agent
401 $ eval $(ssh-agent -s)
402 Agent pid 15
403 $ echo "$SSH_PRIVATE_KEY" | tr -d '\r' | ssh-add - > /dev/null
404 Identity added: (stdin) (gitlab-kekb_runner)
405 $ mkdir -p ~/.ssh
406 $ chmod 700 ~/.ssh
407 $ scp -r -o StrictHostKeyChecking=no $SSH_USER@$SSH_HOST:~/tmp/$CI_PROJECT_NAME/reports $CI_PROJECT_DIR
408 $ ssh -o StrictHostKeyChecking=no $SSH_USER@$SSH_HOST "rm -rf ~/tmp/$CI_PROJECT_NAME"
409 $ rm -rf ~/.ssh
410 Uploading artifacts for failed job
411 Uploading artifacts...
412 reports/: found 2 matching artifact files and directories
413 Uploading artifacts as "archive" to coordinator... 201 Created  id=1374 responseStatus=201 Created token=glcbt-64
414 Cleaning up project directory and file based variables
415 ERROR: Job failed: exit code 1
```

```
285 tests/test_fpout.py::test_pulser_consecutive_trigger[6] PASSED          [ 83%]
286 tests/test_fpout.py::test_pulser_consecutive_trigger[7] PASSED          [ 84%]
287 tests/test_fpout.py::test_pulser_consecutive_trigger[8] PASSED          [ 85%]
288 tests/test_fpout.py::test_pulser_consecutive_trigger[9] PASSED          [ 86%]
289 tests/test_fpout.py::test_pulser_consecutive_trigger[10] PASSED         [ 87%]
290 tests/test_fpout.py::test_pulser_consecutive_trigger[11] PASSED         [ 88%]
291 tests/test_fpout.py::test_pulser_consecutive_trigger[12] PASSED         [ 89%]
292 tests/test_fpout.py::test_pulser_consecutive_trigger[13] PASSED         [ 90%]
293 tests/test_fpout.py::test_pulser_consecutive_trigger[14] PASSED         [ 91%]
294 tests/test_fpout.py::test_pulser_consecutive_trigger[15] PASSED         [ 92%]
295 tests/test_fpout.py::test_pulser_ff[0] PASSED                           [ 93%]
296 tests/test_fpout.py::test_pulser_ff[1] PASSED                           [ 94%]
297 tests/test_fpout.py::test_pulser_ff[2] PASSED                           [ 95%]
298 tests/test_fpout.py::test_pulser_ff[3] PASSED                           [ 96%]
299 tests/test_fpout.py::test_pulser_ff[4] PASSED                           [ 97%]
300 tests/test_fpout.py::test_pulser_ff[5] PASSED                           [ 98%]
301 tests/test_fpout.py::test_pulser_ff[6] PASSED                           [ 99%]
302 tests/test_fpout.py::test_pulser_ff[7] PASSED                           [100%]
303 - Generated html report: file:///nfs/linacfs-users/sdcswd/tmp/mrfioc2-tests/reports/report.html -
304 ==================== 106 passed in 296.67s (0:04:56) ====================
305 Running after_script
306 Running after script...
307 $ which ssh-agent || ( apt-get update -y && apt-get install openssh-client -y )
308 /usr/bin/ssh-agent
309 $ eval $(ssh-agent -s)
310 Agent pid 15
311 $ echo "$SSH_PRIVATE_KEY" | tr -d '\r' | ssh-add - > /dev/null
312 Identity added: (stdin) (gitlab-kekb_runner)
313 $ mkdir -p ~/.ssh
314 $ chmod 700 ~/.ssh
315 $ scp -r -o StrictHostKeyChecking=no $SSH_USER@$SSH_HOST:~/tmp/$CI_PROJECT_NAME/reports $CI_PROJECT_DIR
316 $ ssh -o StrictHostKeyChecking=no $SSH_USER@$SSH_HOST "rm -rf ~/tmp/$CI_PROJECT_NAME"
317 $ rm -rf ~/.ssh
318 Uploading artifacts for successful job
319 Uploading artifacts...
320 reports/: found 2 matching artifact files and directories
321 Uploading artifacts as "archive" to coordinator... 201 Created  id=1375 responseStatus=201 Created token=glcbt-64
322 Cleaning up project directory and file based variables
323 Job succeeded
```

GitLab Automation

```
                                     '4.1.1', 'metadata': '3.1.1'}}
185  rootdir: /nfs/linacfs-users/sdcswd/tmp/mrfioc2-tests
186  configfile: pyproject.toml
187  plugins: html-4.1.1, metadata-3.1.1
188  collecting ... collected 106 items
189  tests/test_basic.py::test_ioc_up
190  -------------------------------- live log call --------------------------------
191  2025-07-29T15:19:36+0900, INFO, root: [test_basic.py:test_ioc_up: line 28]
192  Sw Version: 2.7.2
193  Hw Version: mTCA-EVR-300RF
194  Fw Version: 207.20
195  Position: 9:0.0 slot=2
196  Topology ID: 3
197  Commit Hash: NotConfigured
198  PASSED                                                            [  0%]
199  tests/test_basic.py::test_basic[Ena-Sel-Enabled] PASSED           [  1%]
200  tests/test_basic.py::test_basic[Link-Sts-OK] PASSED               [  2%]
201  tests/test_basic.py::test_basic[Pll-Sts-OK] PASSED                [  3%]
202  tests/test_basic.py::test_basic[TimeValid-Sts-Valid] PASSED       [  4%]
203  tests/test_basic.py::test_sfp_pwr PASSED                          [  5%]
204  tests/test_basic.py::test_stable_errors[CntLinkTimo-I] FAILED     [  6%]
205  tests/test_basic.py::test_stable_errors[CntHwOflw-I] PASSED       [  7%]
206  tests/test_basic.py::test_stable_errors[CntSwOflw-I] PASSED       [  8%]
207  tests/test_basic.py::test_stable_errors[CntRxErr-I] PASSED        [  9%]
208  tests/test_basic.py::test_event_receive PASSED                    [ 10%]
209  tests/test_basic.py::test_event_interval FAILED                   [ 11%]
210  tests/test_basic.py::test_soft_event PASSED                       [ 12%]
211  tests/test_basic.py::test_dc[DCEna-Sel-Enable] PASSED             [ 13%]
212  tests/test_basic.py::test_dc[DCLck-Sts-Locked] FAILED             [ 14%]
213  tests/test_basic.py::test_dc[DCLckTrk-Sts-Lock] PASSED            [ 15%]
214  tests/test_basic.py::test_dc[DCLckDat-Sts-Fine] FAILED            [ 16%]
215  tests/test_fpinp.py::test_fp_in[0] PASSED                         [ 16%]
216  tests/test_fpout.py::test_pulser_polarity[0] PASSED               [ 17%]
217  tests/test_fpout.py::test_pulser_polarity[1] PASSED               [ 18%]
218  tests/test_fpout.py::test_pulser_polarity[2] PASSED               [ 19%]
219  tests/test_fpout.py::test_pulser_polarity[3] PASSED               [ 20%]
```

Search visible log output

Duration: 5 minutes 17 seconds
Finished: 8 minutes ago
Queued: 1 second
Timeout: 1h (from project)
Runner: #32 (q-WjM2tz) runs on linac server (lcsar93)
Source: Push
Tags: docker-runner linac

Job artifacts
These artifacts are the latest. They will not be deleted (even if expired) until newer artifacts are available.

Keep   Download   Browse

Commit bbe89318
add more delay

Pipeline #877 ⚠ Warning for main

test

Related jobs
→ ⚠ testing

Example Test Report

Environment

Python	3.13.4
Platform	Linux-5.14.0-503.26.1.el9_5.x86_64-x86_64-with-glibc2.34
Packages	• pytest: 8.4.0 • pluggy: 1.6.0
Plugins	• html: 4.1.1 • metadata: 3.1.1

Summary

17 tests took 00:00:51.

(Un)check the boxes to filter the results.

☑ 4 Failed, ☑ 13 Passed, ☐ 0 Skipped, ☐ 0 Expected failures, ☐ 0 Unexpec

Result ▲	Test	
Failed	tests/test_basic.py::test_stable_errors[CntLinkTimo-I]	00:00:10

```
evr_device = <core.evr_device.EVRDevice object at 0x7fea9fc46cf0>, suffix = 'CntLinkTimo-I'

    @pytest.mark.parametrize(
        "suffix", [("CntLinkTimo-I"), ("CntHwOflw-I"), ("CntSwOflw-I"), ("CntRxErr-I")]
    )
    def test_stable_errors(evr_device, suffix):
        """
        Check if the error counters are stable (not increasing)
        """
        before = execute_and_assert(lambda: evr_device.get_pv(suffix))
        sleep(10)
        after = execute_and_assert(lambda: evr_device.get_pv(suffix))
>       assert after == before
E       assert 1471814 == 1471808

tests/test_basic.py:65: AssertionError
```
                                                                            expand [+]

Failed	tests/test_basic.py::test_event_interval	00:00:06

```
evr_device = <core.evr_device.EVRDevice object at 0x7fea9fc46cf0>

    def test_event_interval(evr_device):
        # monitor event 1 (50 Hz)
        execute_and_assert(lambda: evr_device.put_pv("EvtA-SP", "1"))
        execute_and_assert(lambda: evr_device.put_pv("EvtA-SP.OUT", f"@OBJ={evr_device.evr_name},Code=1"))
        sleep(1)

        avgts = avgTimestamp()
        evr_device.monitor_pv("EvtACnt-I", avgts.readValue)
        sleep(5)
        evr_device.clear_monitor("EvtACnt-I")
        #TODO: check if this limit is good enough
>       assert abs(avgts.getAvg()-0.02) < 1e-05
E       assert 1.7312712458722457e-05 < 1e-05
E        +  where 1.7312712458722457e-05 = abs((0.020017312712458723 - 0.02))
E        +    where 0.020017312712458723 = getAvg()
```
                                                                            expand [+]

Failed	tests/test_basic.py::test_dc[DCLck-Sts-Locked]	28 ms

```
tests/test_basic.py::test_ioc_up
---------------------------------------------------------------------- live log call ----------------------------------------------------------------------
2025-07-28T16:40:43+0900, INFO, root: [test_basic.py:test_ioc_up: line 28]

Sw Version: 2.7.2
Hw Version: mTCA-EVR-300RF
Fw Version: 207.20
Position: 9:0.0 slot=2
Topology ID: 3
Commit Hash: NotConfigured

PASSED                                                                                   [   5%]
tests/test_basic.py::test_basic[Ena-Sel-Enabled] PASSED                                  [  11%]
tests/test_basic.py::test_basic[Link-Sts-OK] PASSED                                      [  17%]
tests/test_basic.py::test_basic[Pll-Sts-OK] PASSED                                       [  23%]
tests/test_basic.py::test_basic[TimeValid-Sts-Valid] PASSED                              [  29%]
tests/test_basic.py::test_sfp_pwr PASSED                                                 [  35%]
tests/test_basic.py::test_stable_errors[CntLinkTimo-I] FAILED                            [  41%]
tests/test_basic.py::test_stable_errors[CntHwOflw-I] PASSED                              [  47%]
tests/test_basic.py::test_stable_errors[CntSwOflw-I] PASSED                              [  52%]
tests/test_basic.py::test_stable_errors[CntRxErr-I] PASSED                               [  58%]
tests/test_basic.py::test_event_receive PASSED                                           [  64%]
tests/test_basic.py::test_event_interval FAILED                                          [  70%]
tests/test_basic.py::test_soft_event PASSED                                              [  76%]
tests/test_basic.py::test_dc[DCEna-Sel-Enable] PASSED                                    [  82%]
tests/test_basic.py::test_dc[DCLck-Sts-Locked] FAILED                                    [  88%]
tests/test_basic.py::test_dc[DCLckTrk-Sts-Lock] PASSED                                   [  94%]
tests/test_basic.py::test_dc[DCLckDat-Sts-Fine] FAILED                                   [ 100%]
```

Example failed test command line output

Example failed test HTML report

Example successful test HTML report

report.html

Report generated on 29-Jul-2025 at 18:10:13 by pytest-html v4.1.1

Environment

Python	3.13.4
Platform	Linux-5.14.0-503.26.1.el9_5.x86_64-x86_64-with-glibc2.34
Packages	• pytest: 8.4.0 • pluggy: 1.6.0
Plugins	• html: 4.1.1 • metadata: 3.1.1

Summary

106 tests took 00:04:56.

(Un)check the boxes to filter the results.

☑ 0 Failed, ☑ 106 Passed, ☑ 0 Skipped, ☑ 0 Expected failures, ☑ 0 Unexpected passes, ☑ 0 Errors, ☑ 0 Reruns   Show all details / Hide all details

Result ▲	Test	Duration	Links
Passed	tests/test_basic.py::test_ioc_up	228 ms	
Passed	tests/test_basic.py::test_basic[Ena-Sel-Enabled]	25 ms	
Passed	tests/test_basic.py::test_basic[Link-Sts-OK]	26 ms	
Passed	tests/test_basic.py::test_basic[Pll-Sts-OK]	26 ms	
Passed	tests/test_basic.py::test_basic[TimeValid-Sts-Valid]	26 ms	
Passed	tests/test_basic.py::test_sfp_pwr	26 ms	
Passed	tests/test_basic.py::test_stable_errors[CntLinkTimo-I]	00:00:10	
Passed	tests/test_basic.py::test_stable_errors[CntHwOflw-I]	00:00:10	
Passed	tests/test_basic.py::test_stable_errors[CntSwOflw-I]	00:00:10	
Passed	tests/test_basic.py::test_stable_errors[CntRxErr-I]	00:00:10	

Achievements & Future Work

- ## Achievements
  - Efficiency: Reduced SAT execution time from several hours manually to several minutes automatically.
  - Reliability: Eliminated human error, ensuring every new module is tested against the exact same criteria.
  - Traceability: Automatically generated HTML reports provide a permanent, time-stamped record of test results for every device.
  - Scalability: The framework is easily extendable to timing modules other than MTCA form factor.

- ## Future Enhancements
  - Developing a GUI wrapper (e.g., using Python) for easier usage by technicians.
  - Expanding the test library to cover long-term stability and stress tests.