COSYLAB

# User Manual and Test Plan

| | |
|---|---|
| **Revision:** | 1.2 |
| **Status:** | Released |
| **Project:** | EPICS Integration of Teledyne ADQ7 and ADQ14 |
| **Owner:** | Cosylab |
| **Last modification:** | October 21, 2019 |
| **Created:** | July 19, 2018 |

| Prepared by | Reviewed by | Approved by |
|---|---|---|
| Polina Pipp | Ursa Rojec | Ursa Rojec |

# Document History

| Revision | Date | Changed/reviewed | Section(s) | Modification |
|---|---|---|---|---|
| 0.1 | 2018-07-19 | Polina Pipp | All | Document created |
| 0.2 | 2018-07-26 | Polina Pipp | 6.5 | Test results |
| 0.3 | 2018-07-31 | Polina Pipp | 5.1 | Installation |
| 1.0 | 2018-08-02 | Polina Pipp | All | Document release |
| 1.1 | 2019-08-20 | Polina Pipp | 4.1<br>4.2.1, 4.2.2<br>4.2.5, 5.1, 5.2 | Removed duplicated requirement table.<br>Edited: class namings.<br>Added: template description fpr message log; dependencies' versions; CODAC specific installation steps; Sample and Channel decimations descriptions. |
| 1.2 | 2019-10-21 | Polina Pipp | 5.1.2.2, 5.2<br>6 | Updated driver installation in EPICS; updated screenshots<br>Updated test instructions |

# Confidentiality

This document is provided to Teledyne SP Devices, to be distributed along with the EPICS software.

# Scope

This is document that consists of requirements, design description, test plan and user guide document.

# Audience

The audiences of this document are Teledyne SP Devices personnel and their customers using the EPICS device support module.

# 1. Table of Contents

# 2. Abbreviations and Definitions

| Abbreviation | Explanation |
|---|---|
| DAQ | Data acquisition |
| DB | EPICS Database |
| EPICS | Experimental Physics and Industrial Control System |
| GUI | Graphical User Interface |
| IOC | Input Output Controller (EPICS server) |
| NDS | Nominal Device Support v3 |
| PV | Process Variable |
| TSPD | Teledyne Signal Processing Devices |

**Table 1: abbreviations**

# 3. Introduction

EPICS TSPD-ADQ device support module is distributed together with an EPICS test IOC application and a GUI for setting up the data acquisition card and displaying the acquired data.

The software package includes:

- EPICS TSPD-ADQ device support module,
- Engineering Screen (GUI),
- EPICS test IOC application,
- documentation:
  - doxygen describing the API of the device support module;
  - EPICS database files with every record functionality described in DESC field;
  - Technical documentation outlining the architecture, GUI and installation user guide.

# 4. Design description

## 4.1 Functional requirements

| Title | Description |
|---|---|
| Support for ADQ14 and ADQ7 digitizers | The device support module will support ADQ API functions that are valid for ADQ14 and ADQ7 digitizers. |
| Connection to a requested digitizer | The module will be able to establish a communication with any digitizer via its serial number. |
| Data acquisition modes | The following data acquisition modes will be available:<br>• Triggered streaming,<br>• Multi-Record,<br>• Continuous streaming,<br>• Raw streaming. |
| Triggers and clock sources | Data acquisition will be able to run with different trigger modes (Software, External, Level and Internal) and clock settings (reference output and source). |
| Data acquisition parameters | The user will be able to set parameters important to each data acquisition mode. Parameters such as:<br>• Sample skip;<br>• Number of records;<br>• Number of samples per record;<br>• Streaming time. |
| EPICS test application | The test application will allow to test the full functionality of the device support module. |
| GUI capability | The GUI will allow the user to control and monitor the data acquisition parameters. The acquired data will be shown on the screen. The user will be able to control the data acquisition process on the GUI. |
| Technical documentation | The documentation will cover the design of the module, its functional interface (API). The EPICS records will be defined and briefly described in DB files. The GUI guide will describe key elements and areas of the screen. |

**Table 2: requirements**

## 4.2 Architecture

### 4.2.1 Overview

TSPD-ADQ device support module is implemented using the NDS framework.

The NDS device in the module is modelled with ADQDevice, ADQInfo, ADQAIChannelGroup and ADQAIChannel classes. The channel group class handles data acquisition parameters and data acquisition functions. Every physical channel of the device is represented by channel class with a corresponding index.

The NDS device has a tree-like structure. It consists of a root node (created by the ADQDevice class) and child nodes (created by ADQInfo , ADQAIChannel and ADQAIChannelGroup classes).

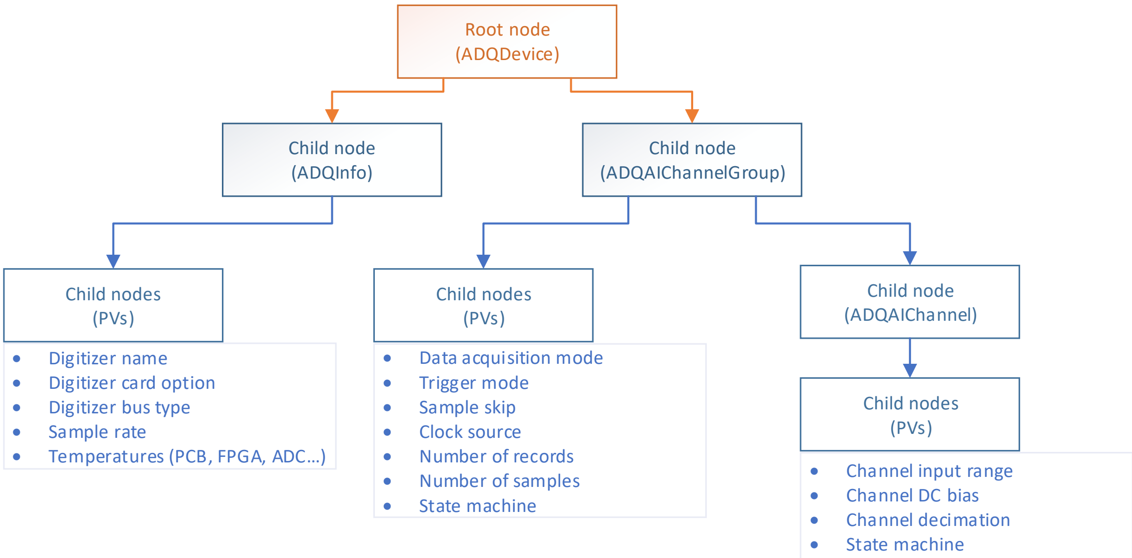


**Figure 1: NDS device structure**

### 4.2.2 Class descriptions

#### 4.2.2.1 ADQDevice

The main functionality of the ADQDevice class is setting up the communication with the requested digitizer.

Before starting the IOC, the user should define the following two parameters (see 4.2.6):

- Serial number of the digitizer he wants to use (defined with macro ADQSN);
- Name of the root node (defined with macro ROOT).

During IOC initialization, the ADQDevice class first creates an NDS device and then searches for the digitizer with the requested serial number. When it is found, the ADQDevice class establishes communication between the NDS device and the digitizer. In the second step of the initialization process, the root node is created, which enables EPICS IOC to communicate with the NDS device. The initialization then proceeds with creation of ADQInfo, ADQAIChannelGroup and ADQAIChannel classes (number of channel class instances depends on the number of channels that the board has).

Each class creates its own child node and adds it to the root node. When the root node and its child nodes are registered, the NDS device is fully set up and an EPICS IOC is created.

#### 4.2.2.2 ADQInfo

This class handles all the information about the digitizer such as its product name, card option, type, product ID, type of connection, sample rate and temperatures (PCB, ADC1, ADC2, FPGA, DCDCA2).

By default, the ADQInfo class creates a child node named “INFO”. PVs of this class are added to this node.

This class is a base class for ADQAIChannelGroup.

#### 4.2.2.3 ADQAIChannelGroup

This class handles the data acquisition parameters such as data acquisition mode, trigger and clock settings, active channels, sample skip, number of samples to acquire, DBS settings. Data acquisition functions (for processing Triggered streaming, Multi-Record, Continuous streaming and Raw streaming) are also handled in this class.

By default, the ADQAIChannelGroup class creates a child node named “COM” to which PVs of this class and each ADQAIChannel class node are added.

This class is a subclass of the ADQInfo class.

#### 4.2.2.4 ADQAIChannel

This class handles channel specific parameters such as input range, DC bias. It also manages the transfer of the acquired data from device support module to the EPICS IOC.

By default, the ADQAIChannel class creates a child node named “CHN” where N is the number of a channel. Each child node is added to the ADQAIChannelGroup class node. This allows EPICS to identify and control parameters and acquired data of every channel.

### 4.2.3 State machine

The state machine handles states and state transitions of a specific class.

Both ADQAIChannelGroup and ADQAIChannel classes have a state machine, the state machine of each channel is dependent on the state machine of the channel group.

When the NDS device is created, both state machines start in OFF state. In this state any change to data acquisition parameters cannot be applied to the digitizer’s configuration.

When the state of the channel group is switched to ON for the first time, the ADQAIChannelGroup class processes every parameter that it handles and sets each channel state to ON. After this, channel specific parameters are also processed

When the state is set to RUNNING, the ADQAIChannelGroup class starts the data acquisition and calls the ADQAIChannel class to push the acquired data to the IOC. Any changes to the parameters are ignored in this state and are only taken into account when the state changes back to ON.

### 4.2.4 Data acquisition

There are four data acquisition modes implemented in the device support module and each mode returns the acquired data differently:

- Multi-Record returns all acquired data at once, where data size is $(Number\ of\ records\ \times Number\ of\ samples)$;
- Continuous streaming returns all acquired data at once, where data size is of predefined buffer size.
- Triggered streaming returns every finished record separately, where data size of a record is $Number\ of\ samples$, until it reaches required $Number\ of\ records$.
- Raw streaming returns all acquired data at once, where data size is of predefined buffer size. This DAQ mode works with one channel per call and doesn't support multichannel acquisition.

When data acquisition is started, the communication with ADQ API is locked. The lock is necessary to make sure that there are no processes that would disrupt the data acquisition and corrupt the data. It is also necessary that all the acquisition related settings, such as number of samples and trigger source do not change while acquisition is in progress.

If a call to the ADQ API is made while the lock is active, the request is queued. For example, it may happen that the ADQInfo class tries to update a temperature value during the data acquisition. In such case the ADQInfo waits until the data acquisition is finished.

### 4.2.5 EPICS databases

During device initialization PVs, that allow one to monitor and control parameters of the connected digitizer, are created. To be able to work with the data (perform calculation, process in any way), the EPICS IOC application uses EPICS record databases. These databases are series of records that are in turn collection of PVs.

PV values are stored in corresponding EPICS records and defined in EPICS DB files. New values can be pushed to PVs through EPICS records as well.

There are four database templates: one for ADQInfo, one for ADQAIChannelGroup, one for ADQAIChannel class and one for message log.

| Record name | Description |
|---|---|
| `ADQInfo.template` | DB for the ADQInfo class. Records store general information about the connected digitizer. |

| | |
|---|---|
| `ADQAIChannelGroup.template` | DB for the ADQAIChannelGroup class. Records manage values of data acquisition parameters and state machine. |
| `ADQAIChannel.template` | DB for the ADQAIChannel class. Records manage the acquired data, values of input range, DC bias, decimation and state machine of each channel. |
| `ADQMessageLog.template` | Records store a limited number of messages sent by the driver to the IOC. |

**Table 3: records in the database**

Each record has a field (DESC) with a brief functional description.

DB files are stored in a Db folder which location is standard for EPICS applications.

### 4.2.6 IOC startup file – st.cmd

EPICS IOC test application relies on records from template files distributed together with the TSPD-ADQ device support module.

EPICS IOC test application startup script can be used as example when a custom st.cmd script needs to be written.

The following macros need to be set in the startup script before the start of the IOC.

| Macro name | Description | Default value |
|---|---|---|
| **ROOT** | Root node name that is passed to the device constructor in the driver. | ADQ |
| **ADQSN** | Serial number of the requested digitizer | 06215 |

**Table 4: st.cmd macros**

# 5. User guide

## 5.1 Installation

### 5.1.1 Dependencies

- EPICS base 7.0.1.1
- CS-Studio BOY 4.5.7
- NDS3 (GitHub version, commit 472049d on Jul 16, 2018)
- NDS3_EPICS (GitHub version, commit 46bf0bb on Jul 7, 2017)
- ADQ SDK (rev.38273)

### 5.1.2 Packages and their installation

#### 5.1.2.1 CODAC Core System

- codac-core-6.0-epics-tspd-adq-*.rpm
  TSPD-ADQ driver packages. Each RPM package contain one of the following: driver support documentation, engineering screens, driver library or DB files.
- adq-test-app.tar.gz
  Test application for CODAC environment.

Installation:

```
$ rpm -U codac-core-6.0-epics-tspd-adq-*.rpm

$ tar -zxvf adq-test-app.tar.gz
$ cd adq-test-app & mvn compile
```

#### 5.1.2.2 EPICS environment

- m-epics-tspd-adq.tar.gz
  Tarball should be installed in a standard EPICS module location (for example, `/opt/epics/modules`).

Installation:

```
$ tar -zxvf m-epics-tspd-adq.tar.gz
$ cd m-epics-tspd-adq
$ vim configure/RELEASE
> # Check the following definitions and update them if needed:
> MODULES = /opt/epics/modules
> ASYN=$(MODULES)/asyn
> NDS3=/usr/local/lib
> NDS3EPICS=$(MODULES)/nds3_epic
> EPICS_BASE = /opt/epics/base
$ make
```

### 5.1.3 TSPD-ADQ module usage

#### 5.1.3.1 CODAC Core System

1. Open the *dbToLoad.cmd* file in *src/main/epics/iocBoot/iocadq-test/* directory. Make sure that the path to the DB file is correct:
   `dbLoadRecords("${EPICS_ROOT}/db/tspd-adq.db","PREFIX=$(ROOT), ADQSN=$(ADQSN)")`
2. Open *st.cmd* file in the same directory and edit the value of macros to needed ones (see 4.2.6).
3. Start the test application:
   `$ mvn run`

#### 5.1.3.2 EPICS environment

4. In your IOC *<top>/configure/RELEASE* file set the path to the TSPD-ADQ module library:
   `ADQ_LIB=${EPICS_MODULES}/m-epics-tspd-adq/lib/linux-x86_64`
5. Configure macros and load the TSPD-ADQ database in your startup script (see 4.2.6). Either copy database file to the db folder or load the database file from the module's folder (`${EPICS_MODULES}/m-epics-tspd-adq/db`):
   `dbLoadRecords("db/tspd-adq.db","PREFIX=${ROOT}, ADQSN=${ADQSN}").`
6. Move to a folder with startup script and run the test application:
   ```
   $ cd iocBoot/iocadq-test
   $ ./st.cmd
   ```

## 5.2 User Interface

### 5.2.1 Opening the GUI

1. Start CS-Studio (CSS).
2. From CSS menu select Window – Open Perspective – Other and select OPI Runtime.
3. Open file `/boy/adqdevice.opi`
4. Before using the GUI, select desired digitizer's serial number from a drop-down menu in the top right corner. If the needed serial number is not in the list, open the panel in "Edit" perspective and add a new serial number to that drop-down menu. Save the panel and open it in "Runtime" perspective.

### 5.2.2 Graphical User Interface

GUI allows a user to monitor and control the data acquisition parameters.

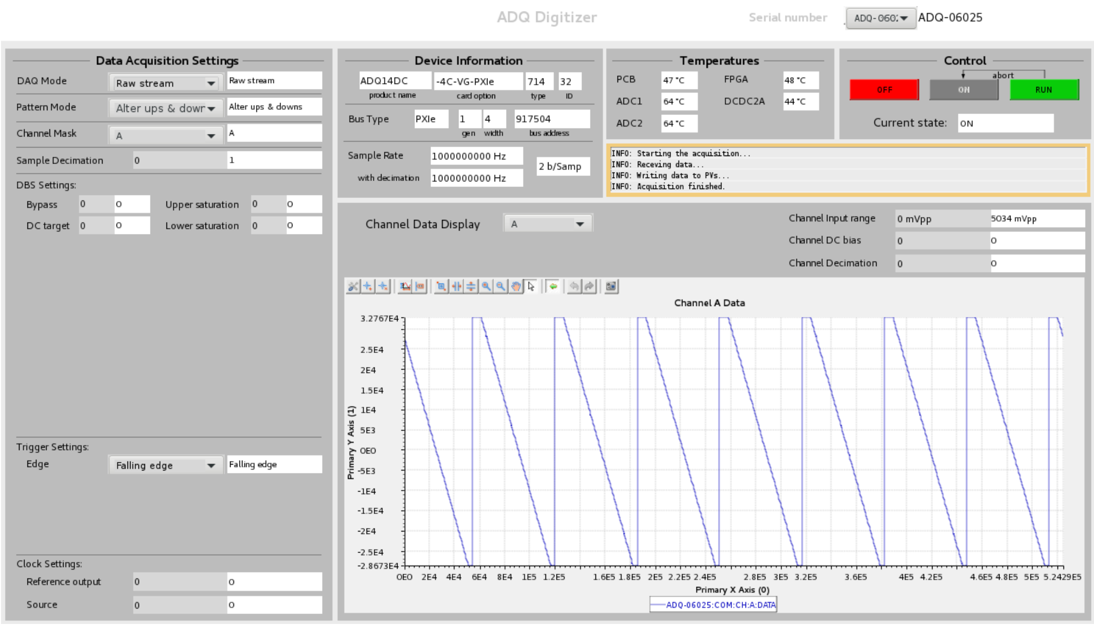


**Figure 2: main screen**

Operator GUI is centralized around the graph displaying acquired data from the digitizer, selected from the drop-down menu in the top right corner of the screen.

The screen is divided into six areas:

- Data Acquisition Settings;
- Device Information;
- Temperatures;
- Control;
- Log messages;
- Channel Data Display.

### 5.2.2.1 Data Acquisition Settings

This area contains all the parameters needed for the correct setup of each data acquisition mode.

There are four supported data acquisition modes:

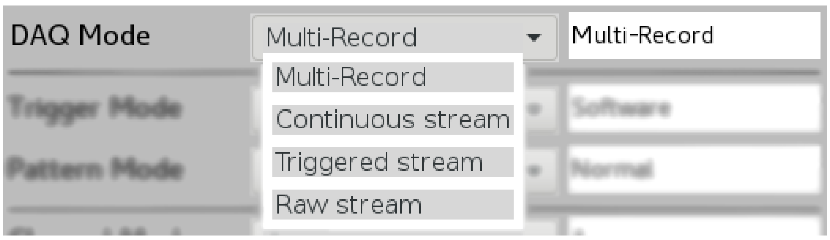


**Figure 3: list of DAQ modes**

Every DAQ mode shares the same a list of parameters:

- Trigger Mode; consists of four options:

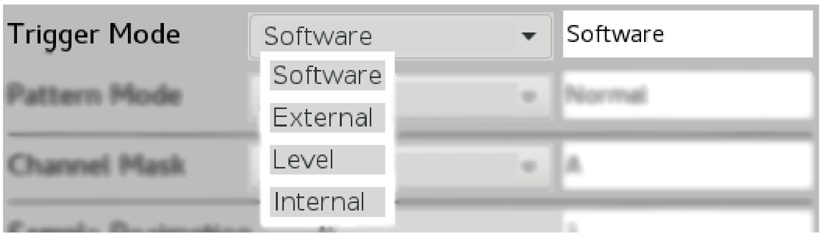


**Figure 4: list of Trigger modes**

Only trigger edge (rising, falling or both edges) of SW trigger can be set.


**Figure 5: Software trigger settings**

External trigger has three parameters: delay, edge and threshold. Delay is set in case there is a delay in the configured logic. Threshold should be set within the high and low levels of the external trigger generator for correct behavior of the data acquisition.

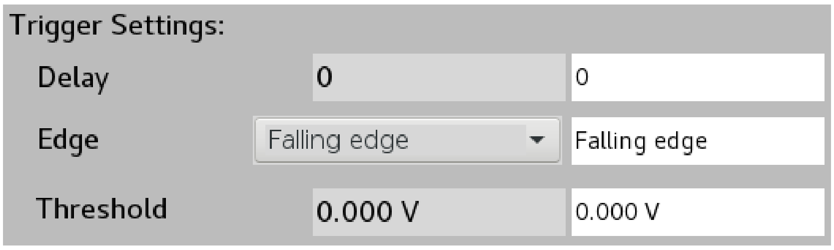


**Figure 6: External trigger settings**

Level trigger has three parameters: level, edge and channel. This trigger is triggered by the input signal's level in codes. Triggering channel can be chosen from the drop-down list.

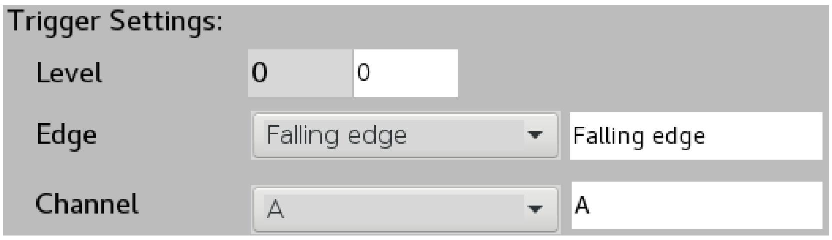


**Figure 7: Level trigger settings**

Internal trigger has four parameters: frequency, edge, lengths of high samples and low samples. Digitizer generates a trigger of a certain frequency and amount of times when the trigger should remain high and low.

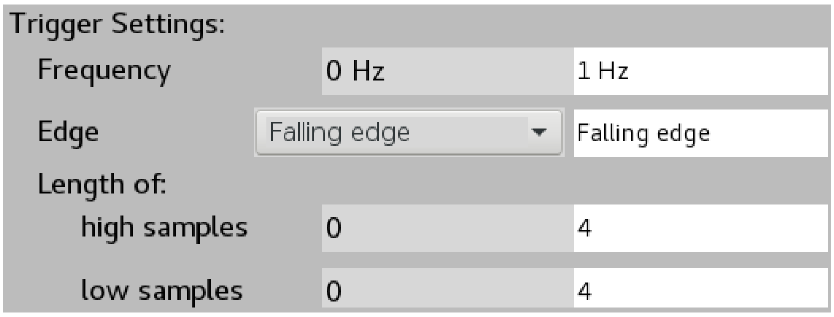


**Figure 8: Internal trigger settings**

- Pattern Mode; default is Normal (direct data):

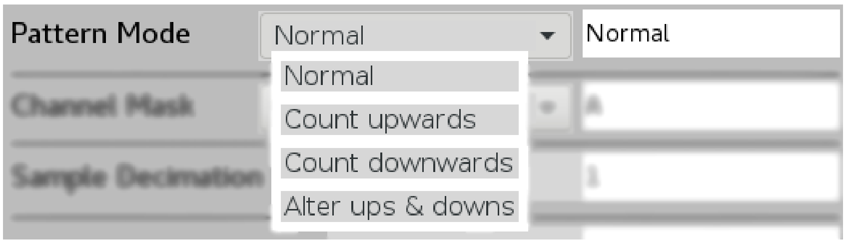


**Figure 9: list of Pattern modes**

- Channel Mask; includes selected channels in data acquisition:


**Figure 10: channel mask parameter**

- Sample Decimation; reduces the sampling rate by $2^N$, where N – is a set value; default is 1;


**Figure 11: sample skip parameter**

- Number of records; a number of waveforms with data samples that will be acquired;
- Number of samples; a number of data samples per record;


**Figure 12: number of records and samples parameters**

- DBS settings – Bypass; enables or disables the DBS instances; default is 0;
- DBS settings – DC target; sets DC target in ADC codes; default is 0;
- DBS settings – Upper saturation; how many codes above baseline can be before the signal ignored in the DC estimation; positive number; default is 0;

- DBS settings – Lower saturation; how many codes below baseline can be before the signal ignored in the DC estimation; negative number; default is 0;

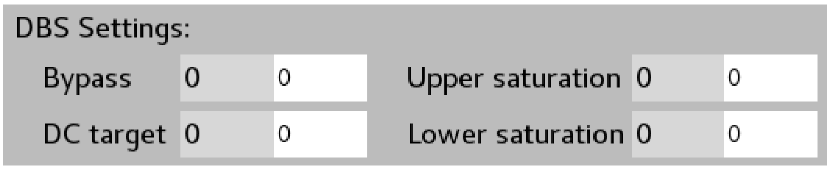


**Figure 13: DBS settings parameters**

- Clock settings – Source; external reference signal should be connected when choosing an external reference; default is 0:

| Clock source | Description |
|---|---|
| **0** | Internal clock source, internal 10 MHz reference |
| **1** | Internal clock source, external 10 MHz reference |
| **2** | External clock source |
| **3** | Internal clock source, external 10 MHz reference from PXIsync |
| **4** | Internal clock source, external TCLKA backplane reference (MTCA units only) |
| **5** | Internal clock source, external TCLKB backplane reference (MTCA units only) |
| **6** | Internal clock source, external 100 MHz reference from PXIe 100 MHz clock |

**Table 5: possible options of Clock source parameter**

- Clock settings – Reference output; enables or disables clock reference output; default is 0.

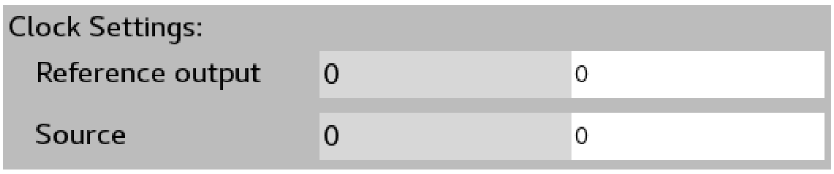


**Figure 14: clock settings parameters**

**Multi-Record** mode has additional following parameters:

- Records to collect; a number of records that will be passed to the system; when set to 0 – none of records will be passed; when set to $(0 < \mathrm{N} \leq \mathrm{Number\ of\ records})$, N records will be passed;
- Max samples; a maximum allowed number of samples per record, depends on the number of records;


**Figure 15: records to collect and maximum number of samples per record, specific to Multi-Record DAQ mode**

- Pre-trigger samples; a number of samples to collect before a trigger arrives; when set, the hold-off samples are reset to 0; when not 0 it slows down the triggering; must be less than the number of samples and it depends on the digitizer's granularity of the pre-trigger buffer; default is 0;
- Hold-off samples; a number of samples to ignore after a trigger arrives; when set, pre-trigger samples are reset to 0; when not 0 it slows down the triggering; must be less than the number of samples and it depends on the digitizer's granularity of the hold-off setting; default is 0.


**Figure 16: pre-trigger and hold-off samples parameters, specific to Multi-Record DAQ mode**

**Continuous streaming** mode:

- Streaming time; time in seconds during which the data acquisition will be taken; if it is set to 0, it automatically updates to 1 second.


**Figure 17: streaming time, specific to Continuous streaming DAQ mode**

**Triggered streaming** mode:

- Pre-trigger samples; same as in Multi-Record;
- Hold-off samples; same as in Multi-Record;
- Flush timeout; time in milliseconds given to prepare the transfer buffers for receiving data; if it is set to 0, it automatically updates to 1000 millisecond.


**Figure 18: flush timeout, specific to Triggered streaming**

**Raw streaming** mode doesn't have any specific parameters. When it is completed, it returns a data of the buffer size which is set in the code.

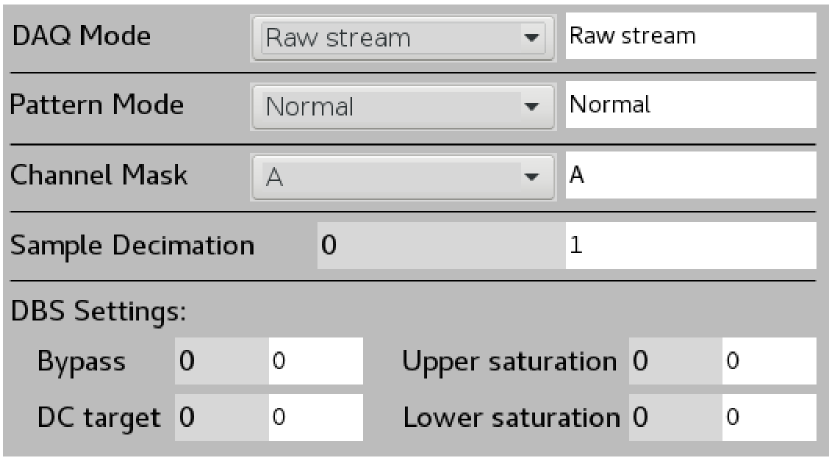


**Figure 19: parameters for Raw streaming DAQ mode**

### 5.2.2.2 Device Information, Temperatures, Log messages

These areas are the informative part of the screen. They display information such as full name of the connected digitizer, its connection type to the system, sample rates and number of bytes per sample.

Temperatures of devices inside the digitizer are updated every second unless a data acquisition is in progress.

Log messages show recent information, warning and error messages.

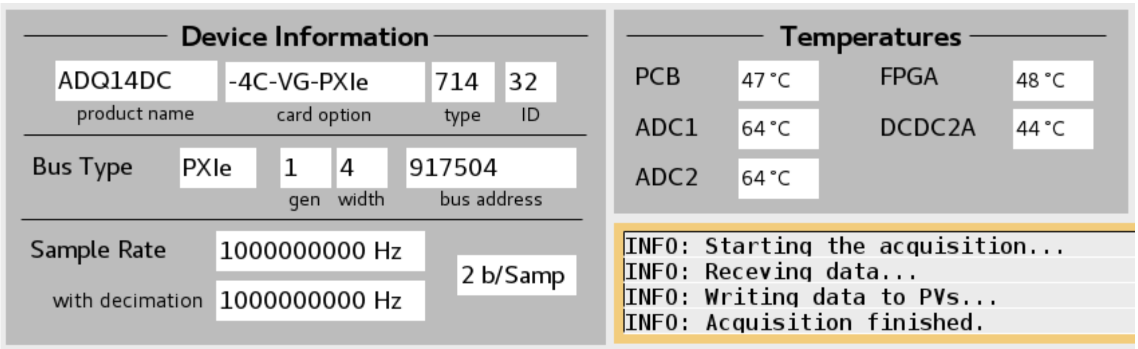


**Figure 20: Device information, Temperatures and Log messages areas of the screen**

### 5.2.2.3 Channel Data Display

This area shows the input range, DC bias and channel decimation of a chosen channel. The channel can be chosen from the drop-down menu.

- Input Range; desired input range in mV (peak-to-peak) for a specific channel;
- DC bias; DC-bias level for a specified input channel;
- Channel Decimation; *ADQ7-FWDSR specific setting*; reduces the sampling rate for a specific channel by $2^N$, where N – is a set value.

Acquired data is displayed on the graph.

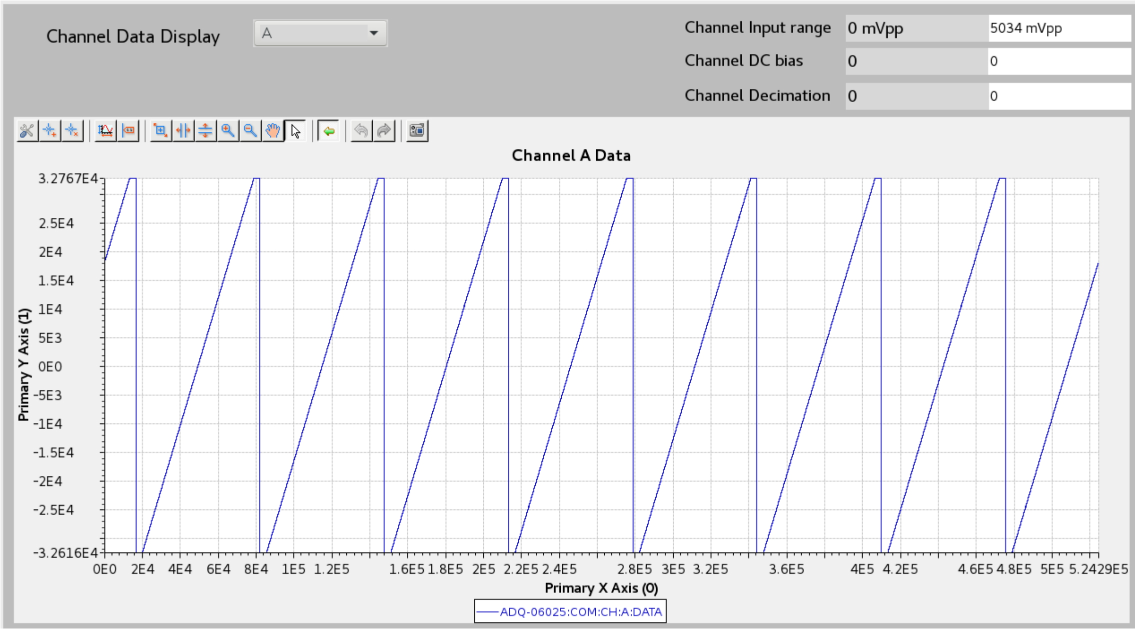


**Figure 21: Channel Data Display area of the screen**

### 5.2.2.4 Control

Control area is used for controlling the device's state machine.

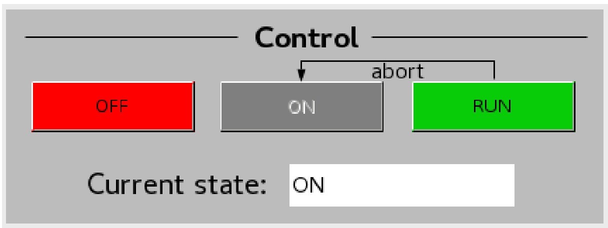


**Figure 22: Control area of the screen**

Button OFF turns the device off. No changes to parameters will be applied.

Button ON sets the device's state to ON. All previously made changes will be processed and set to the digitizer.

Button RUN sets the device to Data Acquisition state. During this state all changes are ignored until the device stops the acquisition. Acquisition either stops automatically due to reached limitations (number of records, streaming time) or it can be stopped/aborted manually by clicking on button ON.

Possible state transitions: OFF ↔ ON ↔ RUN.

Current status is a state readback of the device for this moment.

### 5.2.2.5 Widgets

Widgets (graphical elements on the screen) are used to control and monitor PVs directly. The following widgets were used to create the GUI:

- Text input; for setting values to parameters;
- Text updates; for getting parameters readbacks and log messages:


**Figure 23: Text input and text update widgets**

- Drop-down menus; for setting values to parameters from a limited list:


**Figure 24: Drop-down menu widget**

- Boolean buttons; used for the device state control:


**Figure 25: Boolean button widgets**

- XY Graph; used for plotting the acquired data.

The graph widget has a toolbar for visual data analysis.

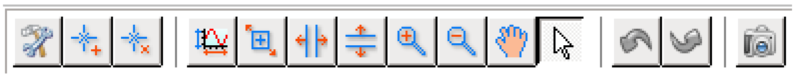

**Figure 26: XY graph toolbar**

The toolbar on the graph can be used for zooming in and out, setting graph properties and capturing a snapshot of the graph. Buttons from left to right are:

- Configure Settings…; Configure the properties of graph, axes or traces.
- Add Annotation…; Add a movable annotation to the graph.
- Remove Annotation…;
- Perform Auto Scale;
- Rubberband Zoom; zoom to selected window.
- Horizontal Zoom; zoom to selected horizontal interval.
- Vertical Zoom; zoom to selected vertical interval.
- Zoom In
- Zoom Out
- Panning; pan on plotting area or axes.
- None
- Undo; undo last operation up to 30 steps.
- Redo; redo last operation.
- Save Snapshot to PNG file; save snapshot to image file.

### 5.2.2.6 Widget states

At the GUI startup widgets may have “Invalid” state and have “Disconnected” state:

- “Invalid” state on a widget means that the PV it is connected to was not yet updated or initialized.

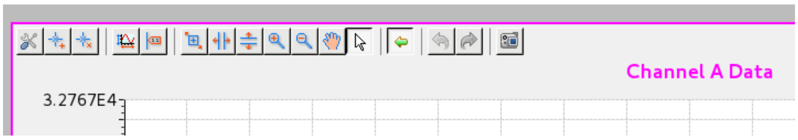


**Figure 27: widget with “Invalid” state**

- “Disconnected” state means that widgets cannot connect to PVs. It can be happening because the IOC application is not running.

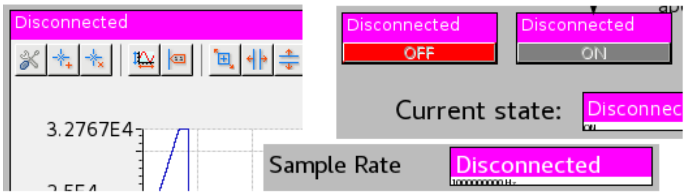


**Figure 28: widgets with “Disconnected” state**

# 6. Test plan

Tests must be done with the provided GUI.

## 6.1 Setting up the hardware

In case the digitizer will be tested with a signal generator, its output power should be within allowed channel input range.

If an external trigger will be used, its output power should be within the allowed values of the digitizer's trigger input range.

## 6.2 Startup

1. Execute the EPICS test IOC application.
2. Open GUI.
3. Some readback fields (text updates) may have INVALID alarm state.
4. Switch the device to ON in the Control area. Readbacks should get a new value (either the same as input or automatically assigned one). The **XY graph widget** may still have the "Invalid" state (for each channel) because the PVs it is connected to were not yet updated.

## 6.3 Data acquisition test with limited data to acquire

All four data acquisition modes (Multi-Record, Continuous streaming, Triggered streaming and Raw streaming) should be tested with each trigger mode (Software, External, Level and Internal), with two different numbers of samples (100 and 100000) and with the highest sampling rate.

A successful test will return a message "Acquisition finished" in the log. No widgets should get an "Invalid" or "Disconnected" state. The graph is updated with the acquired data.

Sometimes it is helpful to use **Perform Auto Scale** and **Horizontal Zoom** tools from the graph toolbar to check the form of the input signal and the total amount of received samples.

To perform the test, follow these steps:

1. Set **DAQ mode**.
2. Choose **Trigger mode** (not available in Raw streaming) and set the trigger **edge**. Set other settings if applicable:
   - External trigger **threshold** to 0.5;
   - Level trigger **channel** to channel(s) that receive data and trigger **level** to 0;
   - Internal trigger **frequency** to 10, **high samples** to 10, **low samples** to 20.
3. Set **Pattern mode** to Normal.
4. Set **Sample skip** to 1.

5. When Multi-Record mode or Triggered streaming mode is chosen:
   - Set **Number of records** to acquire to 50,
   - Set **Number of samples** per record to 100 (the second time to 10000).
6. When Multi-Record mode is chosen, set **Number of records to collect** to 50.
7. When Multi-Record mode or Triggered streaming mode is chosen, set **Pre-trigger** and **Hold-off** samples to 0.
8. When Triggered streaming mode is chosen, set **Flush timeout** to 1000.
9. In **DBS Settings** set **Bypass** to 1.
10. Set **Clock source** and **Clock reference output** to 0.
11. If a signal generator is used, set appropriate **Channel input range** for each active channel. Channel is changed via the drop-down list: CH0, CH1 and so on.
12. Switch the device to **RUN** in the Control area.
13. To stop or abort the data acquisition, switch the device to **ON** state in the Control area.

## 6.4 Data acquisition test with infinite collection

Infinite data acquisition is supported in Triggered streaming mode.

During the infinite collection the log should be constantly updated with "Receiving data…" message. No widgets should get an INVALID state. The graph should be updated with the acquired data of size 10000 after each record collection.

1. Set **DAQ mode** to Triggered streaming.
2. Choose **Trigger mode** and set the trigger **edge**. Set other settings if applicable:
   - External trigger **threshold** to 0.5;
   - Level trigger **channel** to any active channel(s) and trigger **level** to 0;
   - Internal trigger **frequency** to 1000, **high samples** to 10, **low samples** to 20.
3. Set **Pattern mode** to Normal.
4. Set **Sample skip** to 1.
5. When Continuous streaming mode is chosen, set **Streaming time** to 3.
6. Set **Number of records** to -1 (infinite collection).
7. Set **Number of samples** per record to 10000.
8. When Multi-Record mode or Triggered streaming mode is chosen, set **Pre-trigger** and **Hold-off** samples to 0.
9. Set **Flush timeout** to 1000.
10. In **DBS Settings** set **Bypass** to 1.

11. Set **Clock source** and **Clock reference output** to 0.
12. If a signal generator is used, set appropriate **Channel input range** for each active channel. Channel is changed via the drop-down list: CH0, CH1 and so on.
13. Switch the device to **RUN** in the Control area.
14. To stop or abort the data acquisition, switch the device to **ON** state in the Control area.

## 6.5 Test results

Tested with **ADQ14DC-4C-VG-PXIe**:

| DAQ Mode | Trigger / Samples | Software | External | Level | Internal |
|---|---|---|---|---|---|
| Multi-Record | 100 | + | + | + | + |
| | 100000 | + | + | + | + |
| Continuous streaming | | + | + | + | + |
| Triggered streaming | 100 | + | + | + | + |
| | 100000 | + | + | + | + |
| | -1 | N/A | + | + | + |
| Raw streaming | | N/A | N/A | N/A | N/A |

Tested with **ADQ7DC-PXIe**:

| DAQ Mode | Trigger / Samples | Software | External | Level | Internal |
|---|---|---|---|---|---|
| Multi-Record | 100 | + | + | + | + |
| | 100000 | + | + | + | + |
| Continuous streaming | | N/A | + | + | + |
| Triggered streaming | 100 | + | + | + | Overflow (works with 1000) |

| | | | | | |
|---|---|---|---|---|---|
| | 100000 | + | + | + | + |
| | -1 | N/A | + | + | + |
| **Raw streaming** | | N/A | N/A | N/A | N/A |