TELEDYNE SP DEVICES
Everywhereyoulook™

ADQ8

Daisy-Chain Board Synchronization

Author(s): Teledyne SP Devices
Document ID: 19-2246
Classification: Open
Revision: A
Print date: 2019-02-26

Contents

Document History

Revision	Date	Section	Description	Author
A	2019-01-08	-	Initial revision	TSPD
PB1	2019-02-26	-	Extracted from project	TSPD

1 Introduction

This document is a high level description of the ADQ8 Daisy-Chain synchronization system feature.

1.1 Definitions and Abbreviations

Table 1 lists the definitions and abbreviations used in this document.

Table 1: Definitions and abbreviations used in this document.

Item	Description
API	Application programming interface

2 Daisy Chain Triggering

One problem facing a digitizing system consisting of more than one digitizer, where synchronicity is desired, is how to distribute the trigger signal. Regardless of method, the end goal is the same: to trigger all the channels at the same time. One naive solution many be to split the trigger signal and pass that to each digitizer through cables of equal length. However, this solution does not scale with the number of digitizers in the system—eventually the trigger signal is attenuated beyond the sensitivity of the analog input. Moreover, by attenuating the signal, the trigger edge is distorted and precision is lost.

To address these issues this system features a *daisy chain* trigger mechanism. The idea is to pass the trigger signal from one device to the next using the SYNC IN/OUT connectors. However, in order to keep track of the trigger point across all devices, a common time frame is needed. A system consisting of one single chassis may use its built-in 10 MHz clock reference while a system consisting of two of more chassis *must* use an external source provided to each chassis through cables of equal length. Failure to set up the system in this way will result in a skew in the timing grid between chassis.

Fig. 1 presents a block diagram of a system consisting of two chassis in a daisy chain configuration. The first device in the daisy chain is designated as the *master device* and the next is the *first slave device*, the one after that the *second slave device* and so on. Each time the trigger signal is passed through a slave device it is resynchronized to the 10 MHz grid, i.e. each slave delays the trigger signal in a well-defined manner of one 10 MHz period (100 ns). This implies that the relative distance in time between the true trigger point and the trigger point detected by a slave device increases with its distance to the master device in the daisy chain. In turn, this means that each slave device will have to record data *before* its perceived trigger point in order to capture data around the true trigger point. Fig. 2 presents a timing diagram of the start of the trigger process.

When the data is retrieved, following a successful acquisition, the trigger information from the slave devices is adjusted with the trigger information from the master device.

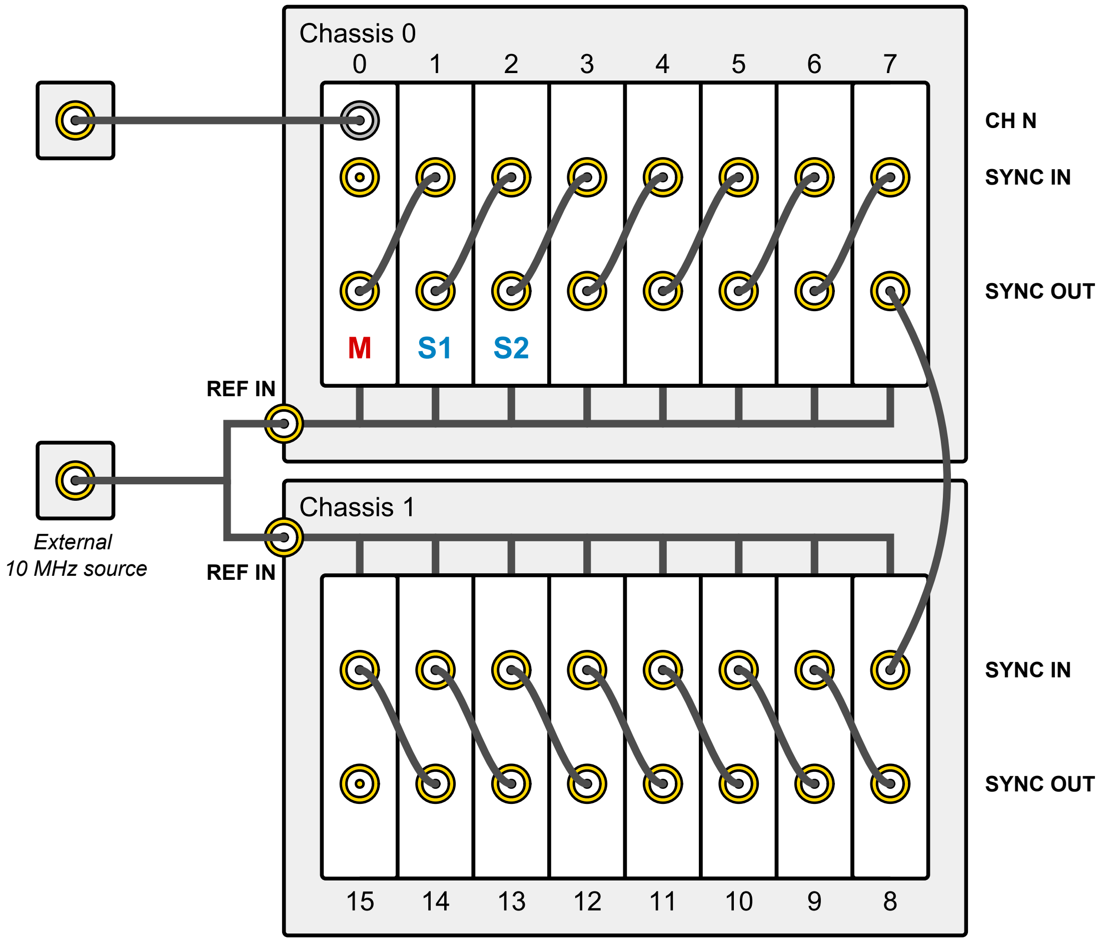

Figure 1: Block diagram of the daisy chain trigger mechanism for a system consisting of 16 devices across two chassis. Each device is annotated with its position in the chain. A common 10 MHz clock reference is distributed to the reference clock input on both chassis. The trigger signal is sampled by one of the data channels (channel N) on the master device.

2.1 Rearm Time

The rearm time is the time from when a record has ended to when the digitizer is able to accept new triggers, i.e. the minimum time between records. For ADQ8, this is given by

$$4000\ \text{ns} + N \cdot 1\ \text{ns} \tag{1}$$

where N is the number of pretrigger samples. When using the daisy chain trigger the number of pretrigger samples depends on the device position in the chain. The number of pretrigger samples for a slave device at position P is given by

$$152 + 100 \cdot P \tag{2}$$

Combining (1) and (2) results in the equation for the rearm time

$$4152\ \text{ns} + P \cdot 100\ \text{ns} \tag{3}$$

where P is the daisy chain position.

The device will ignore triggers occurring during the rearm period. Since the number of pretrigger samples differs between the units in the daisy chain a trigger during the rearm period for one device many cause an earlier device to trigger. Therefore, it is important that no triggers occur during the rearming window.

Note

No triggers may occur during the rearm time when using the daisy chain trigger. The worst-case rearm time is given by (3).

3 Maintenance

This section describes how to perform basic maintenance of the system.

3.1 Replacing a Device

The following should be done after receiving a replacement device:

- Verify that the firmware revision matches the other devices.
- Calibrate the clock reference delay for the new device, see Section 3.2.

3.2 Calibration

The clock reference delay must be calibrated for new digitizers. The calibration is required for the daisy chain trigger timing to be within specification. The calibration is done for a single device in a specific slot. The device should not be moved after the calibration is completed.

The calibration will match the delay of the new digitizer to a reference digitizer. The reference digitizer can be any digitizer located in the same chassis as the new device.

3.2.1 Calibration File

Every chassis has a calibration file at

```
%LOCALAPPDATA%\SP Devices\calibration\delays_chassis_tspd1_tspd2_tspd3.csv
```

This file contains the calibration data for all chassis. Each row in the file contains the calibration value for one digitizer in a specific chassis slot. The calibration file is required when calibrating a new digitizer. After the calibration is complete, this file must be updated manually with the new values.

3.2.2 Requirements

- Square wave generator capable of generating a 1 MHz signal.
- 250 MHz low pass filter
- 50 Ohm signal splitter, at least 1:2.

- One previously calibrated ADQ8 digitizer to be used as a reference. This reference digitizer must be in the same chassis as the uncalibrated device.
- Python 3
- Python calibration scripts
- `delays_chassis_tspd1_tspd2_tspd3.csv` file with calibration values for the reference digitizer.

3.2.3 Calibration Script

The calibration script requires Python 3.6 or later and the Python packages matplotlib and numpy. The script collects 1000 records on each digitizer and measure the relative delay for each record. The mean value of the delay will be used as the calibration value. The results are saved to

```
data/calibrate_delay_single_adq8_<hostname>/
```

where `<hostname>` is the host name of the chassis. The script is split into two files:

`calibrate_delay_single_adq8.py`
: Delay calibration script. Calls functions from `delay_calibration.py`.

`delay_calibration.py`
: Contains functions for interfacing with the digitizer, saving and reading data. Calling this script directly will have no effect.

3.2.4 Calibration Procedure

The calibration procedure is described below. The digitizer should be at its working temperature when the calibration is performed. To reach the working temperature after a cold boot the digitizer must powered on for one hour after initialization. The unit is initialized the first time the software connects to the digitizer.

> **Note**
> The digitizers must have been initialized and running for one hour before an accurate calibration can be performed.

1. If the chassis has been turned off, initialize the digitizers by collecting data once. After initialization, let the digitizers heat up for at least one hour.
2. Shut down any software communicating with the digitizer.
3. Connect the 1 MHz signal to channel 1 on the new and the reference device. The signal should be connected through the low-pass filter and the splitter.
   - The signal must be connected to channel 1.
   - The cables from the splitter must match in length.
   - The amplitude should be around 0.8 Vpp with zero offset. The script will fail if the amplitude is too low or too high.

4. Disconnect the daisy chain cables connecting the chassis to the other chassis. The daisy chain cables connecting the digitizers within the chassis must remain connected for all devices.

5. Update the following values in the calibration script:

   - `new_device_sn`: The serial number of the new digitizer.
   - `reference_device_sn`: The serial number of the reference digitizer.
   - `delayfile_path`: The path to the `delays_chassis_tspd1_tspd2_tspd3.csv` file with the calibration value for the reference device.

6. Run calibration by issuing:

```
> python calibrate_delay_single_adq8.py
```

   from a Python terminal. Example output:

```
Found reference delay 100.0 for device SPD-06512 in delay file
Setting "number_of_records" to 1000
Setting "data_path" to data\calibrate_delay_single_adq8_tspd5
Enumerating devices..
Found device: SPD-06512
Found device: SPD-06784
[...]
Measuring delays
Saving delay metrics to data\calibrate_delay_single_adq8_tspd5\delay_metrics.txt
Max delay: ['176.85', '0.00']
Min delay: ['118.95', '0.00']
Abs diff : ['57.89', '0.00']
Std. dev.: ['11.94', '0.00']
Mean: ['142.82', '0.00']
Delays: ['242.82', '100.00']
Plot delays..
Saving delay figure to data\calibrate_delay_single_adq8_tspd5\delays.png
Saving delays to: data\calibrate_delay_single_adq8_tspd5\
delays_chassis_tspd5_dev_SPD-06784.csv
```

7. Verify the delays in the plot. The delays for the reference device should be zero. The delays for the new device should be within $\pm$ 250 ps.

8. Copy the row with the new device calibration value from the generated `.csv` file to the `delays_chassis_tspd1_tspd2_tspd3.csv` file.

9. Update the calibration file on all chassis.

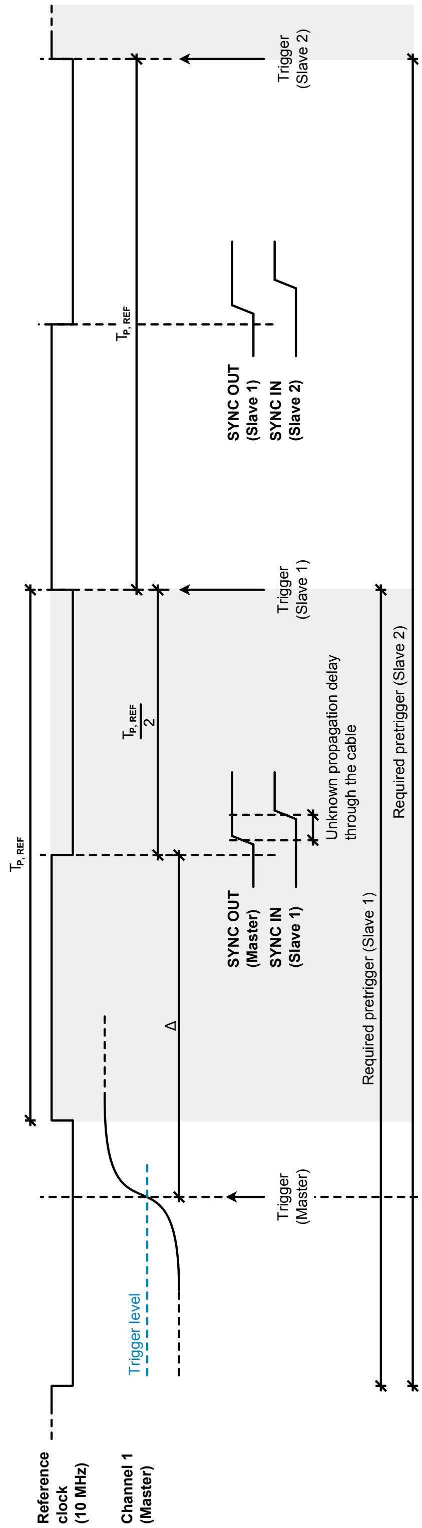

Figure 2: Timing diagram of the daisy chain trigger mechanism for the first three devices in a chain: the master device and two slave devices.

TELEDYNE SP DEVICES
Everywhereyoulook™

Worldwide Sales and Technical Support
www.teledyne-spdevices.com

Teledyne SP Devices Corporate Headquarters
Teknikringen 6
SE-583 30 Linköping
Sweden
Phone: +46 (0)13 645 0600
Fax: +46 (0)13 991 3044
Email: spd_info@teledyne.com