Acquisition software

for digital measurement of Doppler broadening

8.8. 2012, version 1.0

Jakub Čížek, Marián Vlček

Faculty of Mathematics and Physics
Charles University in Prague
Czech Republic

1. Introduction

Acquisition software described here was developed for digital measurement of Doppler broadening using the Acqiris (Aglient Technologies) family of high speed digitizers [1]. The acquisition software enables configuration of digitizers and automated acquisition of waveforms.
Two programs for acquisition are available:

- **Dacqn** – program for acquisition in a mode with a single position of trigger (internal or external)
- **cDacqn** – program for acquisition in a mode when two trigger levels are specified and the acquisition program switch between them during measurement, for details see Ref. [2]. Usually one trigger level is for the single mode (i.e. a gamma in one detector starts the acquisition), while the second trigger level is for the coincidence mode (i.e. two gamma rays detected simultaneously in both detectors are required to start the acquisition. The acquisition is performed in sessions consisting always of (i) measurement in the single mode (it enables to create independent energy calibration for each session) and followed by (ii) measurement in the coincidence mode.

The users are however kindly requested to cite the use of our codes in their works using the reference J. Čížek. M. Vlček, I. Procházka, Nuclear Instruments and Methods in Physics Research Section A, 623, 982-994 (2010).

2. Source code and building of executables

The programs Dacqn and cDacqn are written in C++. The source code can be downloaded from the web page http://physics.mff.cuni.cz/kfnt/us/groups/pas/software.html.
For successful compilation of the source code Acqiris libraries `AqDrv4.lib` and `AqDrv4_x64.lib` and Acqiris header files must be included in the project. Moreover to run the programs dynamic library `AqDrv4.dll` must be installed on the computer in a directory included in the environment variable PATH. These libraries are provided by Aglient Technologies on a supporting software CD supplied with the digitizers. The Acqiris header files are located in the sub-folder:
`user_directory\Aglient\Acqiris\include\`,
the Acqiris libraries can be found in the sub-folder:
`user_directory\Aglient\Acqiris\lib\`
and the dynamic library is located in the sub-folder:
`user_directory\Aglient\Acqiris\bin\`,
where `user_directory` is the directory where the Aglient Technologies software was installed.
It is recommended to firstly install the software form the supporting CD and then build the acquisition programs Dacqn or cDacqn.

3. Function of the programs

When the acquisition program Dacqn or cDacqn is started it firstly looks for the input file `Dacqn.ini` or `cDacqn.ini`, for Dacqn and cDacqn, respectively. These files should be located in the directory where the program is started. Subsequently the program writes on the

screen information about digitizer found to be connected to the system. The digitizer is then configured using the settings specified in the input file. After configuration of digitizer the program, reads the settings from the digitizer and writes them on the screen which enables to check if configuration of the digitizer was successful. Finally the acquisition is started.
A point sampled by digitizer is called a *sample*. Digitized pulse consisting of certain number of samples (specified in the input file) is called *waveform*. Acquisition is performed in parts called *sessions*. This enables to correct for possible drift or instabilities of the system which may occur during the measurement. Each session is written to a separate file. The number of waveforms accumulated in each session is specified in the input file. Session number and the percentage of session completed so far are written on the screen. When a session is completed a message is written to the protocol file `name.ptc`, where `name` is specified in the `Target path` entry in the input file.

4. Input files

The input file `Dacqn.ini` must be located in the directory where the program Dacqn is started. Similarly the initial files or `cDacqn.ini`,must be located in the directory where the program cDacqn is started.
The input files are ASCII files containing lines with structure `keyword=parameter`, where `parameter` is a number, numbers separated by coma or a string. Each line of the initial file must start with a keyword (spaces are ignored). Every text after `parameter` up to the end of line (`'\n'`) is considered as a comment and is ignored. The easiest way how to prepare the input file is to edit the templates supplied with the source codes. The acquisition program during reading of the input file writes the `parameters` on the screen which enables user to check if the input file was read correctly. In the following text the meaning of all entries in the input file is explained. It is recommended to consult also the Acqiris Programmer's Reference Manual [3] and the Acqiris User manual [4].

`Dacqn.ini`
The input file for the program Dacqn

`Segments=1`
Acquisition can be performed in so called sequence mode [3], where instead of transfer waveform-by-waveform the sampled waveforms are accumulated in a buffer in the internal memory of digitizer and the buffer is then transferred at once. The sequence mode is faster when a lot of short waveforms have to be acquired. The parameter `Segments` specify how many waveforms is accumulated in the buffer. When `Segments=1` is selected the sequence mode is switched off.

`Waveforms=1000`
Number of waveforms per session

`Samples=1000`
Number of samples in waveform

`Sessions=100`
Total number of session to be measured

`Sampling interval(ns)=15`
Sampling interval (i.e. distance between samples) given in ns

`Delay time(ns)=0`
Time (given in ns) after the occurrence of the trigger signal when the sampling should start, negative values mean pre-triggering, while positive values are used for post-triggering

`Coupling=3,3`
Trigger coupling in channel 1 and 2:
0 – DC coupling
1 – AC coupling
3 – DC coupling 50Ω (external trigger)
4 – AC coupling 50Ω (external trigger)

`Bandwidth=0,0`
Bandwidth limit:
0 – no bandwidth limit
1 – bandwidth limit at 25 MHz
2 – bandwidth limit at 700 MHz
3 – bandwidth limit at 200 MHz
4 – bandwidth limit at 20 MHz
5 – bandwidth limit at 35 MHz

`Full scale(mV)=250,250`
Full scale in the channel 1 and 2 given in mV

`Offset(mV)=0,0`
Offset in the channel 1 and 2 given in mV

`Trigger type=-1`
Type of the trigger used in acquisition:
1 – internal trigger in channel 1
2 – internal trigger in channel 2
-1 – external trigger in channel 1
-2 – external trigger in channel 2

`Trigger coupling=3,3`
Coupling used for trigger:
1 – DC coupling 1 MΩ
2 – AC coupling 1 MΩ
3 – DC coupling 50 Ω
4 – DC coupling 50 Ω

`Trigger slope=0`
Slope of the trigger:
0 – positive
1 – negative

`Trigger level(mV)=100`
Voltage level to fire the trigger given in mV

`TimeOut=1000`
Time interval in ms for which the digitizer waits for trigger signal. If no trigger signal comes during this time interval a message about time out is written on the screen, acquisition loop is resets and waits for next signal.

`Suffix=cdb`
Suffix of files with acquired data

`Write mode=0`
0 – data are written to binary files
1 – data are written to ASCII text files

`Target path=C:/data/name`
Path where the acquired data will be written.

Data files created by the program are named `name-`*`n`*`.Suffix`, where *n* is the current number of session and `name` and `Suffix` are strings specified in the `target path` and `Suffix` section of the input file.
It `Write mode = 1` the data are written to formatted text file containing two columns. The first column contains samples from the channel 1, while the second column contains samples from the channel 2. Waveforms are separated by a blank line. Formatted writing to text file is substantially slower than writing to a binary file and size of text files is larger as well. For these reasons `Write mode = 1` is intended for the testing purposes only, when immediate inspection of sampled waveform is required.
If `Write mode = 0` data are written to binary files. In normal operation binary files should be used since they have smaller size and writing data to binary file is significantly faster.
The Cmfit and DCDB codes for analysis of waveforms expect that sampled data are stored in binary files. Dacqn code creates binary files with the following structure:
`Samples, Segments, Waveforms, w1_ch1[1], w1_ch2[1], ... , w2_ch1[1], w2_ch[1], ......,` i.e. each file begins with three integers (32-bits) `Samples, Segments, Waveforms` representing the number of samples per waveform, the number waveforms per segment and the number of waveforms per session; a list of short integers (16-bits) `w1_ch1[1], w1_ch2[1], ...` representing the first sample of the first waveform in the channel 1 (`w1_ch1[1]`), the first sample of the first waveform in the channel 2 (`w1_ch2[1]`), the second sample of the first waveform in the channel 1 (`w1_ch1[2]`), the second sample of the first waveform in the channel 2 (`w1_ch1[2]`), etc `...` the first waveform is followed by the second waveform `w2_ch1[1], w2_ch2[1], ...` and next waveforms in the same manner.

`Simulation=0`
0 – acquisition is performed using a real digitizer connected to the computer
1 – simulation mode for testing of the program without any real digitizer, waveforms are simulated.

cDacqn.ini
The input file for the program cDacqn

`Segments=1`
Acquisition can be performed in so called sequence mode [3], where instead of transfer waveform-by-waveform the sampled waveforms are accumulated in a buffer in the internal memory of digitizer and the buffer is then transferred at once. The sequence mode is faster when a lot of short waveforms have to be acquired. The parameter `Segments` specify how many waveforms is accumulated in the buffer. When `Segments=1` is selected the sequence mode is switched off.

`Waveforms=1000`
Number of waveforms per session

`Samples=1000`
Number of samples in waveform

`Sessions=100`
Total number of session to be measured

`Sampling interval(ns)=15`
Sampling interval (i.e. distance between samples) given in ns

`Delay time(ns)=0`
Time (given in ns) after the occurrence of the trigger signal when the sampling should start, negative values mean pre-triggering, while positive values are used for post-triggering

`Coupling=3,3`
Trigger coupling in channel 1 and 2:
0 – DC coupling
1 – AC coupling
3 – DC coupling 50Ω (external trigger)
4 – AC coupling 50Ω (external trigger)

`Bandwidth=0,0`
Bandwidth limit:
0 – no bandwidth limit
1 – bandwidth limit at 25 MHz
2 – bandwidth limit at 700 MHz
3 – bandwidth limit at 200 MHz
4 – bandwidth limit at 20 MHz
5 – bandwidth limit at 35 MHz

`Full scale(mV)=250,250`
Full scale in the channel 1 and 2 given in mV

`Offset(mV)=0,0`
Offset in the channel 1 and 2 given in mV

`Trigger type=-1`
Type of the trigger used in acquisition in the coincidence mode:
1 – internal trigger in channel 1
2 – internal trigger in channel 2
-1 – external trigger in channel 1
-2 – external trigger in channel 2

`sTrigger type=-1`
Type of the trigger used in acquisition in the single mode:
1 – internal trigger in channel 1
2 – internal trigger in channel 2
-1 – external trigger in channel 1
-2 – external trigger in channel 2

`Trigger coupling=3,3`
Coupling used for trigger in the coincidence mode:
1 – DC coupling 1 MΩ
2 – AC coupling 1 MΩ
3 – DC coupling 50 Ω
4 – DC coupling 50 Ω

`sTrigger coupling=3,3`
Coupling used for trigger in the single mode:
1 – DC coupling 1 MΩ
2 – AC coupling 1 MΩ
3 – DC coupling 50 Ω
4 – DC coupling 50 Ω

`Trigger slope=0`
Slope of the trigger in the coincidence mode:
0 – positive
1 – negative

`sTrigger slope=0`
Slope of the trigger in the single mode:
0 – positive
1 – negative

`Trigger level(mV)=100`
Voltage level (given in mV) to fire the trigger in the coincidence mode

`sTrigger level(mV)=100`
Voltage level (given in mV) to fire the trigger in the single mode

`TimeOut=1000`
Time interval in ms for which the digitizer waits for trigger signal. If no trigger signal comes during this time interval a message about time out is written on the screen, acquisition loop is resets and waits for next signal.

`Write mode=0`

0 – data are written to binary files
1 – data are written to ASCII text files

`Target path=C:/data/name`
Path where the acquired data will be written.

Data files created by the program in the single and the coincidence mode are named `name-n.sgl` and `name-n.cdb`, respectively. The symbol *n* is the current number of session and `name` is the string specified in the `target path` entry of the input file.
It `Write mode = 1` the data are written to formatted text file containing two columns. The first column contains samples from the channel 1, while the second column contains samples from the channel 2. Waveforms are separated by a blank line. Formatted writing to text file is substantially slower than writing to a binary file and size of text files is larger as well. For these reasons `Write mode = 1` is intended for the testing purposes only, when immediate inspection of sampled waveform is required.
If `Write mode = 0` data are written to binary files. In normal operation binary files should be used since they have smaller size and writing data to binary file is significantly faster.
The Cmfit and DCDB codes for analysis of waveforms expect that sampled data are stored in binary files. Dacqn code creates binary files with the following structure:
`Samples, Segments, Waveforms, w1_ch1[1], w1_ch2[1], ... , w2_ch1[1], w2_ch[1], ......,` i.e. each file begins with three integers (32-bits) `Samples, Segments, Waveforms` representing the number of samples per waveform, the number waveforms per segment and the number of waveforms per session; a list of short integers (16-bits) `w1_ch1[1], w1_ch2[1], ...` representing the first sample of the first waveform in the channel 1 (`w1_ch1[1]`), the first sample of the first waveform in the channel 2 (`w1_ch2[1]`), the second sample of the first waveform in the channel 1 (`w1_ch1[2]`), the second sample of the first waveform in the channel 2 (`w1_ch1[2]`), etc `...` the first waveform is followed by the second waveform `w2_ch1[1], w2_ch2[1], ...` and next waveforms in the same manner.

References

[1] http://www.home.agilent.com
[2] J. Čížek. M. Vlček, I. Procházka, *Nuclear Instruments and Methods in Physics Research Section A* **623**, 982-994 (2010).
[3] http://cp.literature.agilent.com/litweb/pdf/U1092-90002.pdf
[4] http://cp.literature.agilent.com/litweb/pdf/U1092-90004.pdf