XRF Scanner Software Changelog

INFN CHNet - NYUAD

Abstract

This change-log outlines the source code modifications to the INFN CHNet XRF scanner's software made at the New York University Abu Dhabi (NYUAD) research branch. In particular, versions 10.0 and above feature modifications to the software's structure that allow for a multi-detector scanner architecture

Contents

1 XRFMultidet_v10.0

Changelog with respect to versions 9.x

1. Several rearrangements to the GUI for medium resolutions.

   · Port assignment and initialization members for the linear stages are now all arranged under the same tab.

   · All laser-tracking and Arduino port controls are now under the same tab.

   · Several modifications were made to the status labels default values.

   · Support for different motor initialization protocols has been removed — the new mode of operation is to set a default variable in the source code, and to use the initialization parameters for the motor of interest.

2. The *ScreenDetector* external program has been deprecated. The relevant code for screen resolution detection is now embedded in main.cpp

3. Several source modifications to main.cpp

   · Dependencies already included in the project's headers have been commented out.

   · The timeout(int seconds) function has been deprecated as it is not called anywhere in the program.

   · CAEN kernel modules addition has been deprecated as it is improper syntax. The relevant libraries are automatically loaded into the kernel if the computer is physically connected to the DAC and the appropriate drivers are installed.

4. Source code changes to X_init.cpp, Y_init.cpp and Z_init.cpp

   · New local functions to load M404-8PD and M404-2PD parameters have been introduced so as to cut down on redundant code.

   · A new global function to move X, Y, and Z motors to their reference positions has been introduced.

   · Some global and local variables have been refactored to appeal more to English-speaking users.

   · Junk code has been removed, and the three files have been condensed onto one file name motors_init.cpp

5. Several modifications were made throughout the source code to account for the changes specified in the item above.

6. The file move_motors.cpp has been renamed to motors_move.cpp for consistency.

7. Source code changes to mainwindow.cpp

   · X, Y, and Z specific functions for read_answer_XXX() and send_command_XXX() have been replaced by new global functions which admit a port as one of its arguments, i.e. read_answer_XXX() is now read_answer(serialXXX)

   · The VelocityZ() function has been removed from source code as the Z motor velocity is controlled by the feedback loop and the motor initialization routine.

   · StepZ members in the GUI have also been deprecated for the same reasons.

8. The *XRayTable* external program has been implemented, the respective button in the main window should launch a pop-up with a table of the X-ray energies.

2 XRFMultidet_v10.1

Changelog with respecto to version 10.0

1. Source code changes to ADCXRF.c in the Digitizer_USB project

   · A random number generator function *ranqd1()* has been implemented in the ADCXRF_USB data acquisition protocol — now the multidetector software randomly allocates events from larger non-unitary bins into smaller unitary bins, making independent histogram addition possible.

   · A seed function for *ranqd1()* has been implemented, it calls for the system time to generate a seed after every 50E3 randomly generated numbers.

2. Various minor syntax bugs inherited from version 10.0 have been addressed.

3 XRFMultidet_v10.2

Changelog with respect to version 10.1

1. In project *Spectrum.pro*,

   · Modified the value of *ArraySize* variable to 16385, to avoid undefined behavior.

   · Added a shared memory array declaration for *(shared_memory+30000+i) dedicated to the multidetector sum.

2. In *mainwindow_mouse.cpp* sourec file,

   · Added a conditional statement in the pixel selection switch to pass the spectrum to the appropriate shared memory array, depending on the acquisition detector (or sum) chosen in the acquisition menu. E.g., if detector A is chosen, the values found from the rectangle selection in the main window map are passed to the shared memory array for detector A.

   · The previous allows for visualization of summed spectra obtained directly from pixel selection in the main window map.

   · Added the *ranqd1()* protocol in ADCXRF.c to permit the addition of spectra in point acquisition mode.

3. In ADCXRF.c, the required architecture to automatically sum the spectra and pass them to the dedicated shared memory array has been added, i.e., now every acquisition writes the spectra sum automatically to the array in *(shared_memory+30000+i)

4 XRFMultidet_v10.3

Changelog with respect to version 10.2

1. The protocol to save .txt files in mainwindow.cpp has been modified.

2. In project *Spectrum.pro*,

   · The protocol to load new .txr files for both point and scan mode acquisitions has been modified to match the format required for the multidetector sum protocol

   · The mode of display of the XRF spectra has been modified to show a histogram as opposed to connected lines.

4

- The color of the plot display has been changed from light blue against a dark blue background, to black against white with a red marker to facilitate visualization of the data.

- A new *QLineEdit* member has been added to the bottom dock to display the name of the currently loaded file.

- A refresh button has been added to the menu to avoid reloading spectra from the main window if the mode of visualization has changed.

- The *Print* GUI button and its associated *print()* function have been deprecated

- Global boolean variables signals to load and set calibration parameters have been replaced with a more natural signal and slot functions approach.

- The following signal functions have been added as members to the *Mainwindow* class: *toggleLog()*, and *toggleEnergy(bool active)*.

- Calibration parameters are now persistent, i.e. they will be automatically loaded every time the XRF spectra display window is opened.

- Display mode (as in Channels vs. Energy) is automatically toggled and selected if calibration parameters other than '1' for gradient and '0' for offset are found.

3. In *SHM_Creator.cpp* and *Shm.h*, the size of the shared memory segment with *shmid* identifier has been updated from a size of 50 system pages (204800 bytes) to 100 pages (409600 bytes) to avoid overlap between the spectra of the second detector and the detector sum. Overall, the total stack memory usage of this segment is under 0.5 Mbytes.

4. The *(shared_memory+30000+i) array is now *(shared_memory+40000+i) to avoid overlap between spectra of different detectors.

5. In *laser.cpp* source file,

- The *autofocus_timer* is now connected directly to the *readKeyence()* void function through its timeout signal, avoiding the need to call for a proxy function.

- The structure of *readKeyence()* has changed to account for the update rate from the Keyence serial, now no average is taken and if the Keyence serial returns '0'm the function awaits for the next update befor passing onto the tracking function.

· A port flushing protocol has been added for the Arduino port, to guarantee up-to-date distance tracking values.

· The interval for the *autofocus_ timer* has changed to 500 ms.

5   XRFMultidet_v10.4

Changelog with respect to version 10.3

In *mainwindow_ loadSHM.cpp* source file,

· The body of *LoadFileWithCorrection_ SHM()* is now empty as the function cannot be called from anywhere in the source code.

· The *randqd1()* random number generator function has been updated to take type double parameters for its lower and upper limits, some static_cast<type> rules have been added to guarantee proper data type conversion, and a new protocol has been added to round up or down the double type variable returned.

· The calibration gradients and offsets are calculated at the beginning of the *LoadFileWithCorrection_ SHM()* as doubles by type converting the parameters stored in shared memory.

· Lower and upper limits passed as parameters to *ranqd1()* are calculated in *LoadFileWithCorrection_ SHM* from the local type long *dataread* by typecasting it to a double first.

6   XRFMultidet_v10.5

Changelog with respect to version 10.4