

Développement en C++ de logiciels et de bibliothèques modulaires pour le pilotage, l'acquisition, la calibration et le diagnostic de la caméra médicale xemis 2

Olivier Lemaire

lemaire@subatech.in2p3.fr

CNRS / IN2P3 / SUBATECH

Exposé du **Mercredi 14 juin 2023 à 14h15**

Compilé le : 14 juin 2023

1 PRÉSENTATION DE L'EXPÉRIENCE

■ objectif et principe de fonctionnement

- Objectifs
- Principe

■ chaîne d'électronique

- schéma bloc
- schéma visuel de la chaîne
- la réalisation
- l'avancement

2 ENVIRONNEMENT DE DÉVELOPPEMENT

■ logiciels, bibliothèques, langage

■ besoins et contraintes

■ les logiciels développés

- Les projets et leur dépendances

3 STRATÉGIE DE MUTUALISATION

■ création de bibliothèques

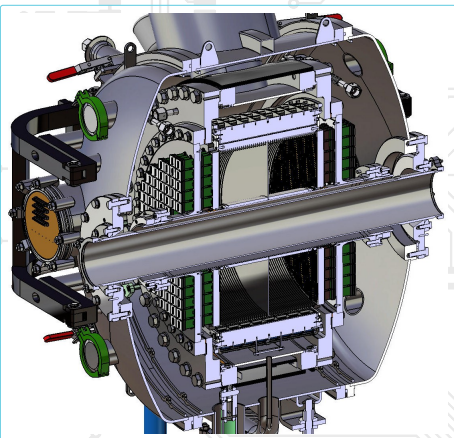
- git
- bibliothèques simples
- bibliothèques hybrides
- bibliothèques externes
- les bibliothèques partagées localement, le cas x2_daq_visu
- la cartographie des relies, le cas tordu ?

4 PRÉSENTATION DE QUELQUES LOGICIELS

■ x2_daq_visu

- spartan
- files
- acquisition
- data display
- plot
- print

5 CONCLUSION



→ Xenon Medical Imaging System

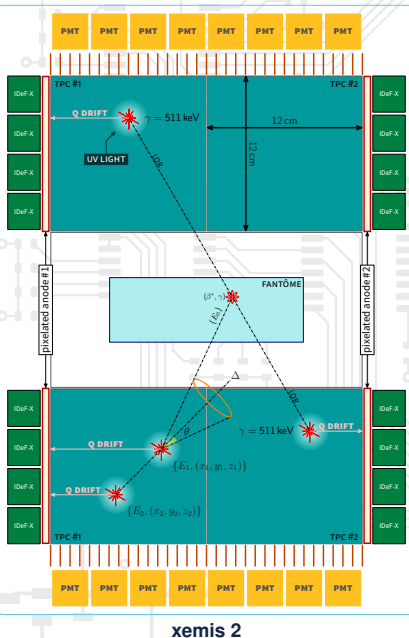
- caméra compton au xénon liquide
- imagerie 3 photons

→ objectif

- produire des images médicales de bonne résolution à partir d'une très faible activité
 - 100 fois moins par rapport au TEP classique

→ technologie

- détecteur chambre à projection temporelle
- milieu de détection xénon liquide
- radio-isotope scandium 44 émetteur 3 photons (β^+ , γ)



→ principe de détection

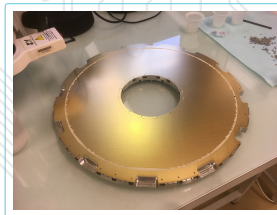
→ 2 photons émis dos à dos

- émission de lumière UV, temps t_0
- dépôts de charge qui dérivent localisation (x, y) et un temps d'arrivée
- $(x, y, \Delta t) \rightarrow (x, y, z)$ car vitesse de dérive connue
- reconstruction d'une ligne de réponse

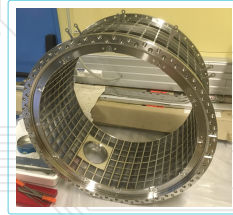
→ un 3^e photon

- même principe, on cherche $\{E_1, (x_1, y_1, z_1)\}$ et $\{E_2, (x_2, y_2, z_2)\}$
- $$\cos(\theta) = 1 - m_e c^2 \frac{E_1}{E_0(E_0 - E_1)}$$

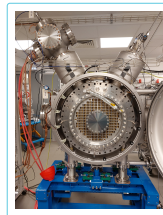
→ en combinant le cone et la ligne de réponse on obtient l'origine de la désintégration



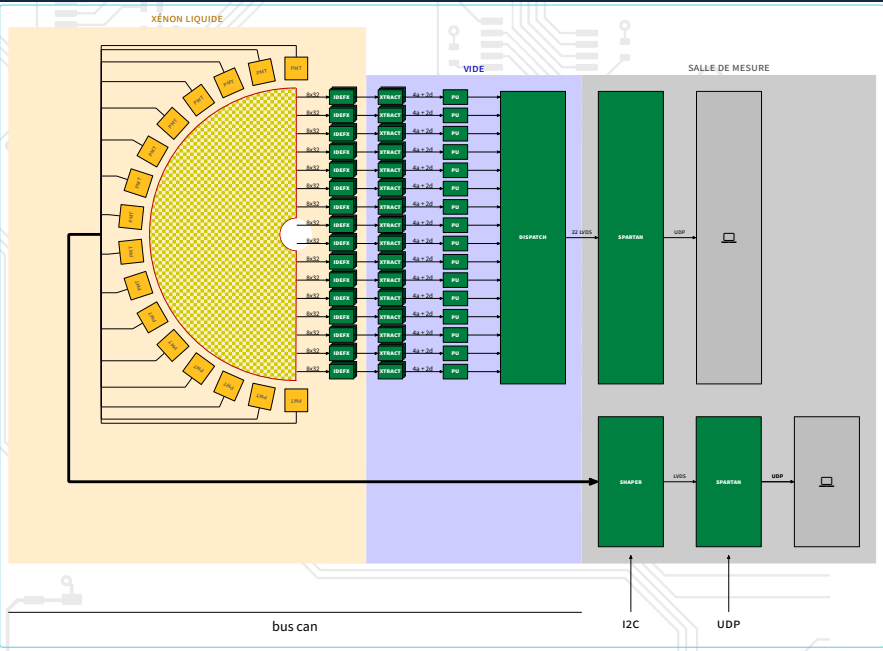
anode

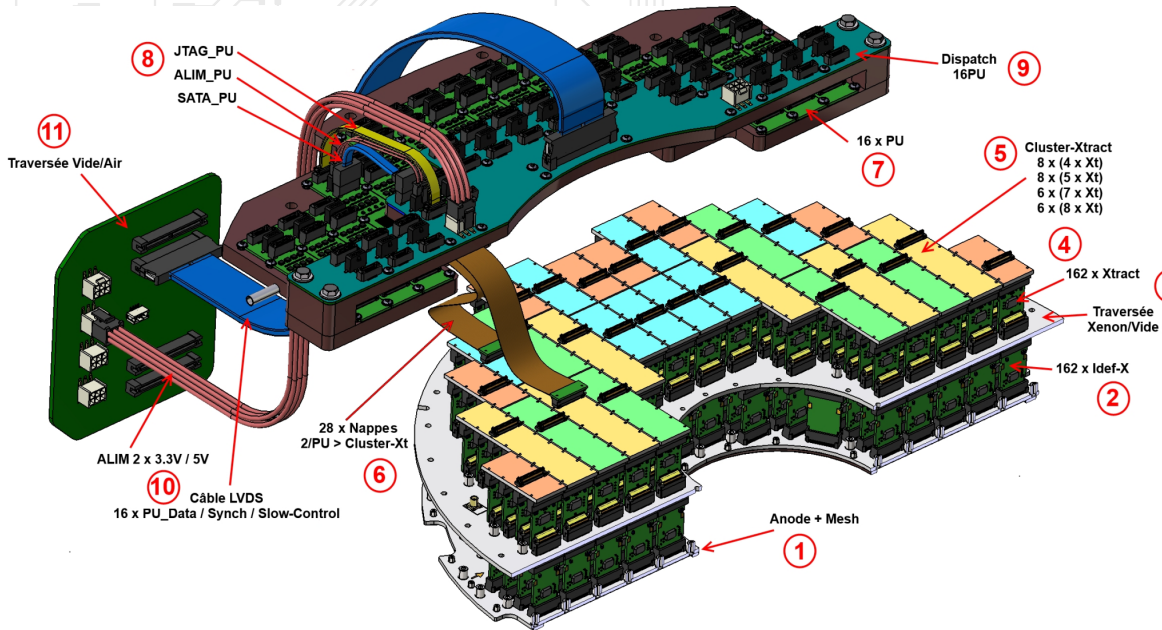


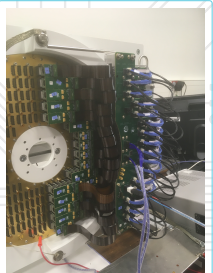
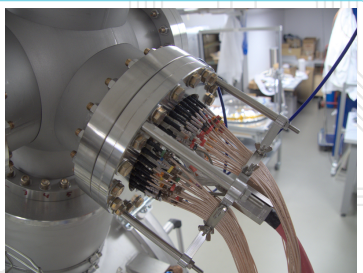
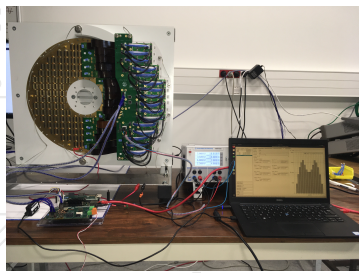
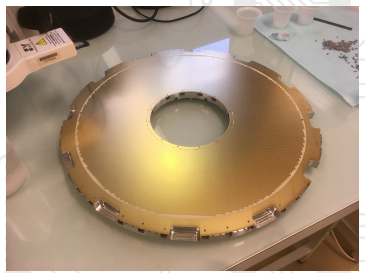
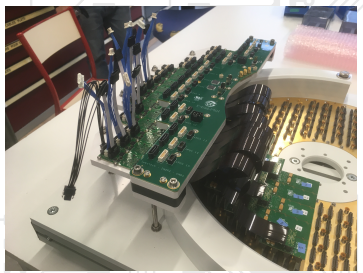
cage pmt



xemis

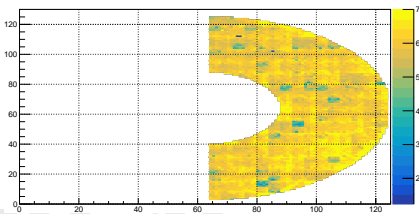








map of pixel noise



- nous avons un emplacement au CHU du Nantes
- Nous en sommes en phase de montage
 - la méca est "terminée"
 - la baie informatique est prête
 - l'électronique PMT est prête
 - l'électronique charge est montée
- pressage et montage d'une anode la semaine dernière

1 PRÉSENTATION DE L'EXPÉRIENCE

■ objectif et principe de fonctionnement

- Objectifs
- Principes

■ chaîne d'électronique

- schéma bloc
- schéma visuel de la chaîne
- la réalisation
- l'avancement

2 ENVIRONNEMENT DE DÉVELOPPEMENT

■ logiciels, bibliothèques, langage

■ besoins et contraintes

■ les logiciels développés

- Les projets et leur dépendances

3 STRATÉGIE DE MUTUALISATION

■ création de bibliothèques

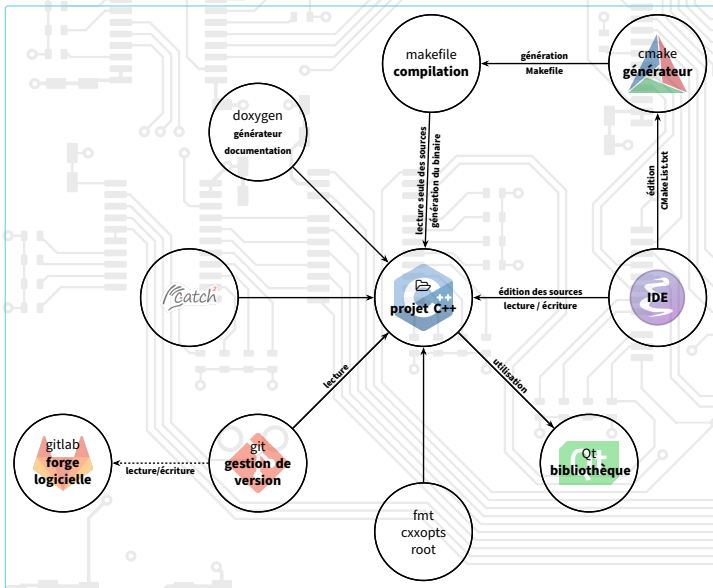
- git
- bibliothèques simples
- bibliothèques hybrides
- bibliothèques externes
- les bibliothèques partagées localement, le cas x2_daq_visu
- la cartographie des relies, le cas tordu ?

4 PRÉSENTATION DE QUELQUES LOGICIELS

■ x2_daq_visu

- spartan
- files
- acquisition
- data display
- plot
- print

5 CONCLUSION



environnement de développement

- langage C++**17**
 - python pour chaîner les procédures
- partie graphique Qt**6**
- construction cmake 3.xx
- calcul root
- fmt, cxxopts
 - essayer de limiter les dépendances
- gestion de version git
 - gitlab in2p3 (hébergement)
 - git (backend)
- test catch2
- documentation doxygen
- OS almaLinux (CERN)

besoins et contraintes::

How Projects Really Work (version 1.0)

Create your own cartoon at www.projectcartoon.com

How the customer explained it



How the project leader understood it



How the analyst designed it



How the programmer wrote it



How the business consultant described it



How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

→ **les besoins**

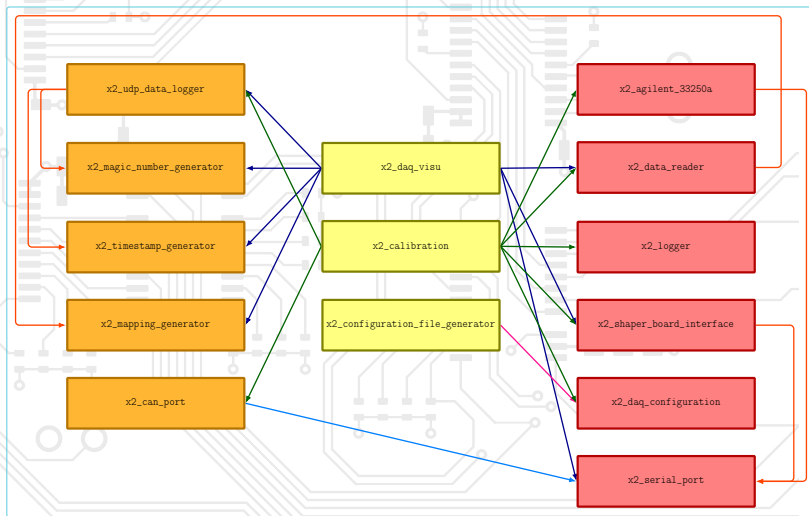
- logiciels de diagnostic
- logiciels de calibration
- logiciels de test
- toujours faire les choses de la même façon

→ **les contraintes**

- développement en parallèle du firmware et du software
 - pas toujours évident de savoir d'où viennent les problèmes
 - beaucoup de choses à développer
 - équipe restreinte
 - des demandes pas toujours faciles

→ **impératif**

- de la reproductibilité
- minimum de maintenance
- transparent pour l'utilisateur (sauf Amaury)



les logiciels développés

- beaucoup de dépendances et de composants mutualisables
- **nécessité de trouver une stratégie...**

1 PRÉSENTATION DE L'EXPÉRIENCE

■ objectif et principe de fonctionnement

- Objectifs
- Principe

■ chaîne d'électronique

- schéma bloc
- schéma visuel globale

- la réalisation
- l'avancement

2 ENVIRONNEMENT DE DÉVELOPPEMENT

■ logiciels, bibliothèques, langage

■ besoins et contraintes

■ les logiciels développés

- Les projets et leur dépendances

3 STRATÉGIE DE MUTUALISATION

■ création de bibliothèques

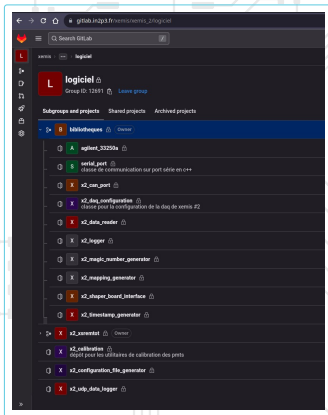
- git
- bibliothèques simples
- bibliothèques hybrides
- bibliothèques externes
- les bibliothèques partagées localement, le cas x2_daq_visu
- la cartographie des relies, le cas tordu ?

4 PRÉSENTATION DE QUELQUES LOGICIELS

■ x2_daq_visu

- spartan
- files
- acquisition
- data display
- plot
- print

5 CONCLUSION



bibliothèques sur le dépôt gitlab



- les logiciels utilisent souvent les mêmes composants
 - nécessité de créer des bibliothèques
 - simplifier la maintenance et assurer de la reproductibilité
- se débarrasser de cpold

```
→ tree --charset unicode --gitignore
|-- foo_dev
|  |-- foo_vi
|  |-- foo_vi.old
|  |-- foo_vi.old.2009_03_29
|  |-- foo_vi.marche_pas
|  |-- foo_vi.todelete
|  |-- foo_vi.OLD.2006_05_12
|  |-- foo_vi.check
foo_save_dev.rar
foo_save20090329_v2_0.tgz
foo_save20080412_v1_2.tgz
foo_save20060509_v1_0.tgz
```

- git / gitlab
 - tout regrouper au même endroit
 - utilisation des groupes et sous-groupes pour hiérarchiser le travail
 - donne la possibilité d'utiliser des sous-modules
 - propagation des modifications contrôlée
 - convention de nommage, exemple x2_sub_project_name

```
tree --charset unicode --gitignore
.
|-- _clang-format
|-- _clang-tidy
|-- cmake
|-- CMakeLists.txt
|-- Doxyfile
|-- hooks
|-- include
|   |-- daq_acquisition.hxx
|   |-- daq_common.hxx
|   |-- daq_data.hxx
|   |-- daq_field.hxx
|   |-- daq_file.hxx
|   |-- daq_reader_charge.hxx
|   |-- daq_reader_light.hxx
|   |-- daq_reader_slice.hxx
|   `-- daq_slice.hxx
|-- lib
|-- readme.md
|-- src
|   |-- daq_common.cxx
|   |-- daq_file.cxx
|   |-- daq_reader_charge.cxx
|   |-- daq_reader_light.cxx
|   |-- daq_reader_slice.cxx
|   `-- lib
|       `-- x2_mapping_generator
-- test
    |-- daq_file_generator.cxx
    |-- daq_file_generator.hxx
    `-- test_x2_data_reader.cxx

9 directories, 22 files
```

→ toutes les bibliothèques sont organisées de la même façon

→ structure du répertoire

- cmake scripts cmake
- doc documentation
- hooks fichier git
- include fichiers header
- src fichiers source
- test fichiers de tests
- lib bibliothèques

→ un fichier CMakeLists.txt primordial dans cette organisation

→ autres fichiers : clang, doxygen, readme ...

→ objectifs :

→ simplifier en ayant une organisation "conventionnelle" et identique dans tous les sous-projets.

→ simplifier l'ajout à un autre projet le plus possible

→ `git submodule add <relative_url> <destination_folder>`

→ `add_subdirectory(/path/to/library)`

→ `target_link_libraries(main_project PRIVATE <target_lib>)`

```

    ### main project
set(X2_FILE_CAN_SENDER_MAIN_PROJECT OFF)
if(CMAKE_CURRENT_SOURCE_DIR STREQUAL CMAKE_SOURCE_DIR)
  set(X2_FILE_CAN_SENDER_MAIN_PROJECT ON)
endif()

# [...]

if(X2_FILE_CAN_SENDER_MAIN_PROJECT)
# executable
add_library(x2_can src/x2_can_port.cxx)
# [récupération automatique du sous-module x2_serial_port]
target_include_directories(x2_can PUBLIC
  "${CMAKE_SOURCE_DIR}/include")
target_link_libraries(x2_can PUBLIC x2_serial_port cxxopts
  fmt::fmt-header-only)
set(CMAKE_INCLUDE_CURRENT_DIR ON)
set(PROJECT_SOURCES main.cxx )
add_executable(x2_file_can_sender ${PROJECT_SOURCES})
# [...]
else()
add_library(x2_can_port src/x2_can_port.cxx )
target_include_directories(x2_can_port PUBLIC
  ${CMAKE_CURRENT_SOURCE_DIR}/include
  ${CMAKE_BINARY_DIR})

# à gérer dans le projet principal
target_link_libraries(x2_can_port PUBLIC
  Threads::Threads
  cxxopts
  fmt::fmt-header-only
  serial_port)
endif()

```

→ projet pouvant être utilisés soit sous en exécutable, soit sous forme de bibliothèques.

→ exemplex2_can_sender

→ bibliothèque CAN

→ exécutable pour envoyer des fichiers sur le bus CAN

→ même structure que le répertoire précédent mais avec un main.cxx en plus à la racine

→ **bibliothèque**

→ `git submodule add ../bibliotheque/x2_can_sender src/lib`

→ `add_subdirectory(src/lib/x2_can_sender)`

→ `target_link_libraries(main_project PRIVATE x2_can_sender)`

→ **projet principal :**

→ `$ git clone --recursive git@gitlab.in2p3.fr:xemis/[...]/x2_can_port.git`

→ `$ cd x2_can_port`

→ `$ mkdir build && cd build`

→ `$ cmake .. && cmake --build .`


```
# -----
# modules
# -----
include(FetchContent)
# -----
# cxxopts
# -----
find_package(cxxopts 3 QUIET)

# téléchargement de la bibliothèque si elle n'est pas disponible
if(NOT cxxopts_FOUND)
  message(STATUS "cxxopts not found; class downloaded from github")
  FetchContent_Declare(
    cxxoptslib
    GIT_REPOSITORY https://github.com/jarro2783/cxxopts.git
    GIT_TAG         c74846a891b3cc3bfa992d588b1295f528d43039 # v3.0.0
    GIT_PROGRESS   ON
  )
  FetchContent_MakeAvailable(cxxoptslib)
  option(CXXOPTS_SYS "Use cxxopts from git" OFF)
else()
  option(CXXOPTS_SYS "Use cxxopts installed on computer" ON)
endif()

configure_file(${CMAKE_SOURCE_DIR}/templates/cxxopts_headers.hxx.in
  cxxopts_headers.hxx)
```

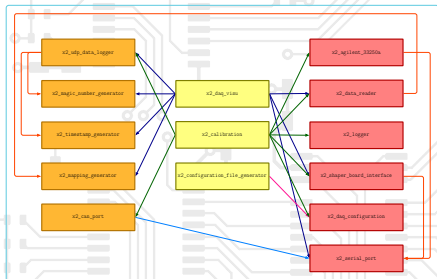
```
#cmakedefine01 CXXOPTS_SYS
// si cxxopts n'est pas installé
#if CXXOPTS_SYS == 0
#include "cxxopts.hpp"
#else
#include <cxxopts.hpp>
#endif
```

```
#define CXXOPTS_SYS 0
// si cxxopts n'est pas installé
#if CXXOPTS_SYS == 0
#include "cxxopts.hpp"
// sinon
#else
#include <cxxopts.hpp>
#endif
```

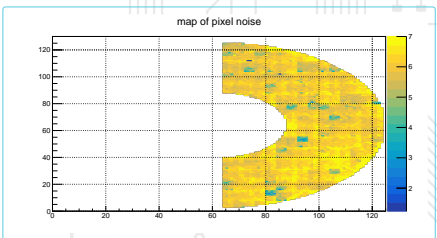
→ récupération de bibliothèques externes

- possibilité de récupération en sous-modules (git)
- téléchargement à l'élaboration (FetchContent)

- 1 vérifier si la paquet existe
- 2 le télécharger dans le cas contraire
- 3 le rendre disponible
- 4 générer un header



les logiciels développés



résultat

- projet regroupant les utilitaires de calibration, test et diagnostic
 - mesure du bruit moyen sur tous les pixels
 - mesure du niveau dc sur tous les pixels
 - calibration du seuil des pmts...
- à chaque procédure correspond un exécutable
- ils partagent tous les même bibliothèques
- certaines bibliothèques sont grosses...
- bibliothèques partagées ou statiques ?
 - statiques → chaque utilitaire embarque sans propre copie
 - partagées → installation ??
- installation d'un rpath sur une bibliothèque partagées
 - le chemin vers les bibliothèques est relatif à l'emplacement de l'utilitaire

création de bibliothèques::les bibliothèques partagées localement, le cas x2_calibration

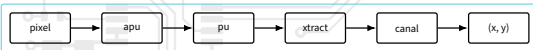
```
# -----
# rpath
# -----
# use, i.e. don't skip the full RPATH for the build tree
set(CMAKE_SKIP_BUILD_RPATH FALSE)
# when building, don't use the install RPATH already (but later on when installing)
set(CMAKE_BUILD_WITH_INSTALL_RPATH FALSE)
set(CMAKE_INSTALL_RPATH "$ORIGIN")
# add the automatically determined parts of the RPATH
set(CMAKE_INSTALL_RPATH_USE_LINK_PATH TRUE)
```

```
build (feature_calibration_zc_th) $$ ldd src/pmt/thresholds/x2_calibration_pmt_thresholds
linux-vdso.so.1 (0x00007ffd5fcd000)
libx2_shaper_board_interface.so => /home/lemaire/code/cxx/xemis/daq/x2_calibration/build/src/lib/x2_shaper_board_interface/libx2_shaper_board_interface.so (0x00007fb03e91000)
libx2_udp_data_logger.so => /home/lemaire/code/cxx/xemis/daq/x2_calibration/build/src/lib/x2_udp_data_logger/libx2_udp_data_logger.so (0x00007fb03e91000)
libx2_data_reader.so => /home/lemaire/code/cxx/xemis/daq/x2_calibration/build/src/lib/x2_data_reader/libx2_data_reader.so (0x00007fb03e1c000)
[...]
```

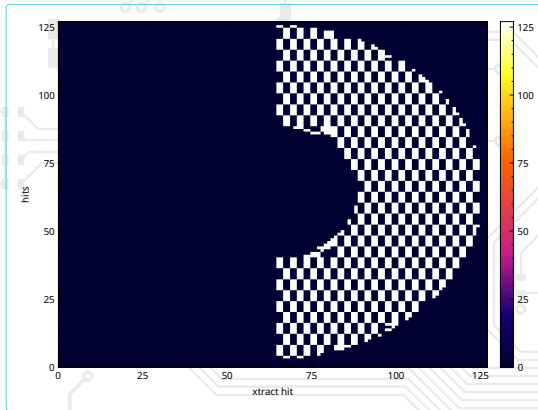
```
cmake --install .
cd ./bin
ldd ./x2_calibration_pmt_thresholds
```

```
total 28M
[...]
-rwxr-xr-x. 1 lemaire lemaire 1.6M May 23 17:44 libx2_data_reader.so
-rwxr-xr-x. 1 lemaire lemaire 246K May 23 17:44 libx2_shaper_board_interface.so
-rwxr-xr-x. 1 lemaire lemaire 2.6M May 24 15:35 libx2_udp_data_logger.so
-rwxr-xr-x. 1 lemaire lemaire 6.0M May 24 15:35 x2_calibration_pmt_thresholds
[...]
```

```
linux-vdso.so.1 (0x00007fff52da0000)
libx2_shaper_board_interface.so => /home/lemaire/code/cxx/xemis/daq/x2_calibration/build/bin/./libx2_shaper_board_interface.so
libx2_udp_data_logger.so => /home/lemaire/code/cxx/xemis/daq/x2_calibration/build/bin/./libx2_udp_data_logger.so (0x00007fbc0692c000)
libx2_data_reader.so => /home/lemaire/code/cxx/xemis/daq/x2_calibration/build/bin/./libx2_data_reader.so (0x00007fbc068b7000)
[...]
```



chaîne d'acquisition simplifiée



exemple de fichier cartographié

→ projet qui génère la cartographie des voies sous forme d'une table de correspondance (lut) dans un fichiers *.hxx

→ à chaque combinaison (d, a, p, x, c) on fait correspondre un pixel (d, x, y)

→ à chaque (d, x, y) → (d, a, p, x, c)

→ petit projet c++ qui fait ce travail

→ conversion de fichiers csv vers mapping.hxx

→ création d'un dépôt, etc.

→ projet simple dans le principe mais l'erreur est humaine...

→ besoin régénérer les tables à chaque modification et de récupérer le nouveau fichier dans les projets qui l'utilisent

→ nécessité de trouver une solution sans pour autant perdre les travail déjà fait...

```

set(X2_MAPPING_EXTRACTOR OFF)
if(CMAKE_CURRENT_SOURCE_DIR STREQUAL CMAKE_SOURCE_DIR)
  set(X2_MAPPING_EXTRACTOR ON)
endif()
[...]
message(STATUS "building project as a library...")

# compilation du générateur
add_custom_target(x2_mapping_generator ALL
  COMMAND g++ "${CMAKE_CURRENT_SOURCE_DIR}/main.cxx" -o generator
  WORKING_DIRECTORY "${CMAKE_BINARY_DIR}"
  COMMENT "building mapping generator"
  VERBATIM)

# execution du générateur et génération du mapping
add_custom_target(x2_mapping_generation ALL
  COMMAND "${CMAKE_BINARY_DIR}/generator"
  DEPENDS x2_mapping_generator
  WORKING_DIRECTORY "${CMAKE_BINARY_DIR}"
  COMMENT "x2 mapping generation"
  VERBATIM
)

# bibliothèque d'interface
add_library(x2_mapping INTERFACE)
target_include_directories(x2_mapping INTERFACE "${CMAKE_BINARY_DIR}")
add_dependencies(x2_mapping x2_mapping_generation)

```

→ bibliothèque hybride

- ce n'est pas vraiment une bibliothèque
- c'est un exécutable qui génère le fichier dont on fera une bibliothèque
- cmake à la rescousse...

- 1 compiler le générateur
- 2 exécuter le générateur pour générer le fichier mapping.hxx
- 3 créer une bibliothèque avec ce nouveau fichier
- 4 créer une chaîne de dépendance

```

inline uint16_t get_index_dapxc_to_dxy(const unsigned int& dispatch, //
  const unsigned int& apus, //
  const unsigned int& pu, //
  const unsigned int& xtract, //
  const unsigned int& channel) //
{
  auto d = dispatch % 2;
  uint16_t index = (d & 0b1) << 13 | // dispatch 1 bit
    ((apus & 0xF) << 9) | // apus -- 4 bits
    ((pu & 0b1) << 8) | // pu -- 1 bit
    ((xtract & 0b111) << 5) | // xtract -- 3 bits
    ((channel & 0x1F)); // canal -- 5 bits

  return index; };

static const std::array<mapping::coordinate_dxy, 16384> lut_dapxc_to_dxy = {
  {0, 77, 7},{0, 78, 7},{0, 76, 7},{0, 79, 7},{0, 77, 6},{0, 78, 6},{0, 76, 6},
  {0, 79, 6}, [...] };

static const std::array<mapping::coordinate_dapxc, 16384> lut_dxy_to_dapxc =
  { [...] };

```

1 PRÉSENTATION DE L'EXPÉRIENCE

■ objectif et principe de fonctionnement

- Objectifs
- Principe

■ chaîne d'électronique

- schéma bloc
- schéma visuel de la chaîne
- la réalisation
- l'avancement

2 ENVIRONNEMENT DE DÉVELOPPEMENT

■ logiciels, bibliothèques, langage

■ besoins et contraintes

■ les logiciels développés

- Les projets et leur dépendances

3 STRATÉGIE DE MUTUALISATION

■ création de bibliothèques

- git
- bibliothèques simples
- bibliothèques hybrides
- bibliothèques externes
- les bibliothèques partagées localement, le cas x2_calibration
- la cartographie des relies, le cas tordu ?

4 PRÉSENTATION DE QUELQUES LOGICIELS

■ x2_daq_visu

- spartan
- files
- acquisition
- data display
- plot
- print

5 CONCLUSION

daq file forensic software -- git commit: 04ab90d34f3c86a158d884eca93b366600549046, git branch: dev

spartan configuration files acquisition raw data display plot textual slices summary pmts

action

configure register(s)
check all registers
uncheck all registers

charge configuration

charge generator configuration

mode: ready

channel: 0

extract: 0

pk: 0

packet: 0

time: 0

value: 0

delay between packets: 0x0

charge input enable

Block 0 -- 0: 0

read: 0x00000000 read: 0x00000000

charge channel enable

Block 0 -- 0: 1

read: 0x00000000 read: 0x00000000

leds

mode: value

polarity: high

color: green

value: 10

timer: 1000

register generic

Block #: 0x0

register #: 0x0

value hex: 0x-80000000

light configuration

light generator configuration

mode: ready

time up: 0

time down: 0

light input enable

Block 7 -- 0: 1

read: 0x00000000 read: 0x00000000

light input inverted

Block 7 -- 0: 2

read: 0x00000000 read: 0x00000000

light channel enable

Block 8 -- 0: 2

read: 0x00000000 read: 0x00000000

acquisition

time high: 10 ms

time low: 80 ms

number of slices: 10

ready to go !!

daq file forensic software -- git commit: 04ab90d34f3c86a158d884eca93b366600549046, git branch: dev

spartan configuration files acquisition raw data display plot textual slices summary pmts

folder /home/lemaire/documents/data/x2_20230414_105031/x2_20230414_105031 open...

add remove file header

Name	Size
.	
..	
x2_20230414_105031_#0.x2daq	428.27 KiB

add to list remove from list clear file list

1 /home/lemaire/documents/data/x2_20230414_105031/x2_20230414_105031/x2_20230414_105031_#0.x2daq

ready to go !!

load files read file header read acquisition header

file Displayed me/lemaire/documents/data/x2_20230414_105031/x2_20230414_105031/x2_20230414_105031_#0.x2daq

file header	acquisition header
file header version (hex) 1	header version 8
machine id (hex) 1ce06475e8c8635d6c33d878e	size 5
magic number (hex) 7359895620264541569	pu in acquisition 0
ms since epoch (hex) 66239a127dd23d81	25 MHz 0
date and time	lx150 1
reset daq time low (dec) 80	pu number 18
reset daq time high (dec) 10	delay between slices 296020
number of slices (dec) 10	firmware version 2
git hash (hex) 7c3bd066e896510f7be5c7346	pm in acquisition
file number (hex) 0	slave 1
	pm number 4
	pu address length fifo dbu 0
	pm address length fifo dbu 0
	udp max length 0
	build date and time XXXXXXXXXXX
	slice number 3456517982
	slice duration 2357605846

daq file forensic software -- git commit: 04ab90d34f3c86a158d884eca93b366600549046, git branch: dev

spartan configuration files acquisition raw data display plot textual slices summary pmts

time base

time high 10

time low 80

number of slices 10

save path /home/lemaire/documents/data/test_banc_lumiere

acquisition mode master

single acquisition

multi acquisition

can configuration

serial port open can port

can baudrate [kbps] close can port

mode get firmware version

number of frame per burst 15 frames get hardware version

delay between frames (ms) 20 ms rescan ports

delay between bursts (ms) 20 ms get serial #

delay end of file (ms) 50 ms send file over can...

restart dingle between bursts don't restart can port closed

firmware version

hardware version

serial number

ready to go !!

folder /home/lemaire/documents/data/x2_20230414_105031/x2_20230414_105031 open...

add to list remove from list clear file list

file name

x2_daq_visu::raw data display

daq file forensic software - git commit: 04ab90d34f3c66115b6884ec93b366600549046, git branch: dev

spartan configuration files acquisition raw data display plot textual slices summary pmts

slice 0 x2_20230414_105031_#0.x2daq

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
2	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
3	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
4	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
5	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
6	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
7	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
8	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
9	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
10	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
11	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
12	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
13	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
14	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
15	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
16	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
17	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
18	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
19	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
20	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
21	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
22	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
23	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
24	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
25	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
26	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
27	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
28	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
29	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
30	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
31	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
32	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
33	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
34	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
35	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
36	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
37	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
38	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820
39	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820	B155C0820

ready to go !!

flags

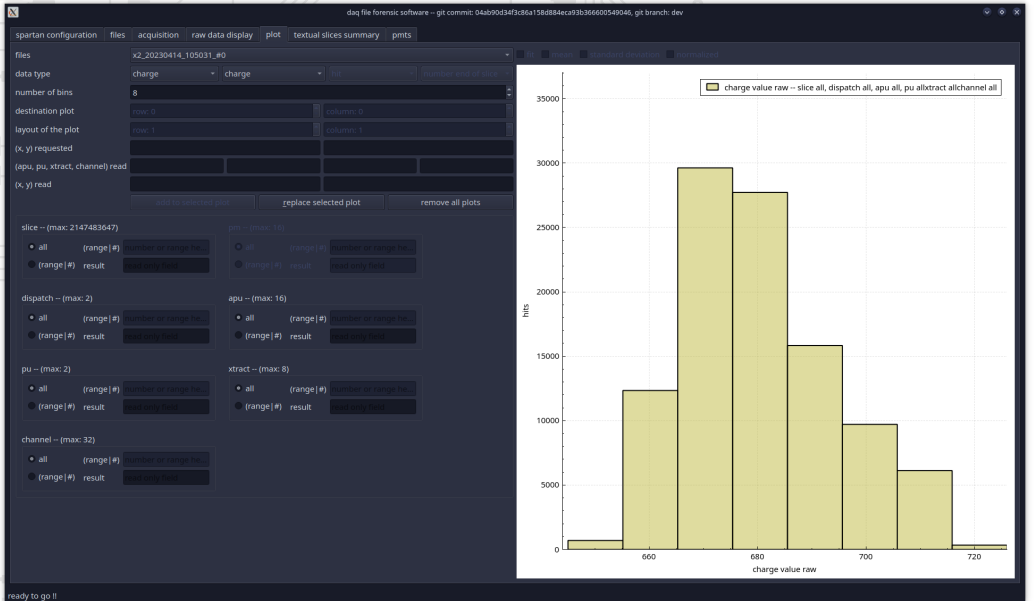
field	header	trailer	
slice number	0d1	0d1	
pu eos	0x0	0x0	
pu mask	0x0	0x0	
pm input inverted	0x0	0x0	
pm mask	0xFFFF	0xFFFF	
pm fifo full	0x0	0x0	
packet full mem d	0b0	0b0	
packet full mem c	0b0	0b0	
fifo full	0b0	0b0	
pu fifo full	0x0	0x0	
complete	0b1	0b1	
size read	0d8	0d8	
0x55C0820	0x00000000	0x00000000	0x0000FFFF
0x55C0820	0x00000000	0x00000000	0x0000FFFF
0x55C0820	0x00000000	0x00000000	0x0000FFFF

slice structure

light header:
light value:
error header:
to end of slice header:
slice ID:
unknown:

value decimal binary hexadecimal

xtract # (3 bits)	2	010	hex2
channel # (5 bits)	6	00110	06
charge (12 bits)	665	0010010011001	299
time (12 bits)	703	001010111111	2bf
word binary	010001100010100110010101011111		



daq file forensic software -- git commit: 04ab90d34f3c86a158d884eca93b366600549046, git branch: dev

spartan configuration files acquisition raw data display plot textual slices summary pmts

pmt bank #12

injection	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	don't inject on any channel
threshold	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	user defined
output	G1	G1	G1	G1	G1	G1	G1	G1	G1	G1	G1	G1	G1	G1	G1	G1	set channels to G1

serial port configuration

serial port:

iss module info:

module id:

firmware version:

iss mode:

serial number:

serial status: serial port closed

ready to go !!

1 PRÉSENTATION DE L'EXPÉRIENCE

■ objectif et principe de fonctionnement

- Objectifs
- Principe

■ chaîne d'électronique

- schéma bloc
- schéma visuel

- la réalisation
- l'avancement

2 ENVIRONNEMENT DE DÉVELOPPEMENT

■ logiciels, bibliothèques, langage

■ besoins et contraintes

■ les logiciels développés

- Les projets et leur dépendances

3 STRATÉGIE DE MUTUALISATION

■ création de bibliothèques


- git
- bibliothèques simples
- bibliothèques hybrides
- bibliothèques externes
- les bibliothèques partagées localement, le cas x2_daq_visu
- la cartographie des relies, le cas tordu ?

4 PRÉSENTATION DE QUELQUES LOGICIELS

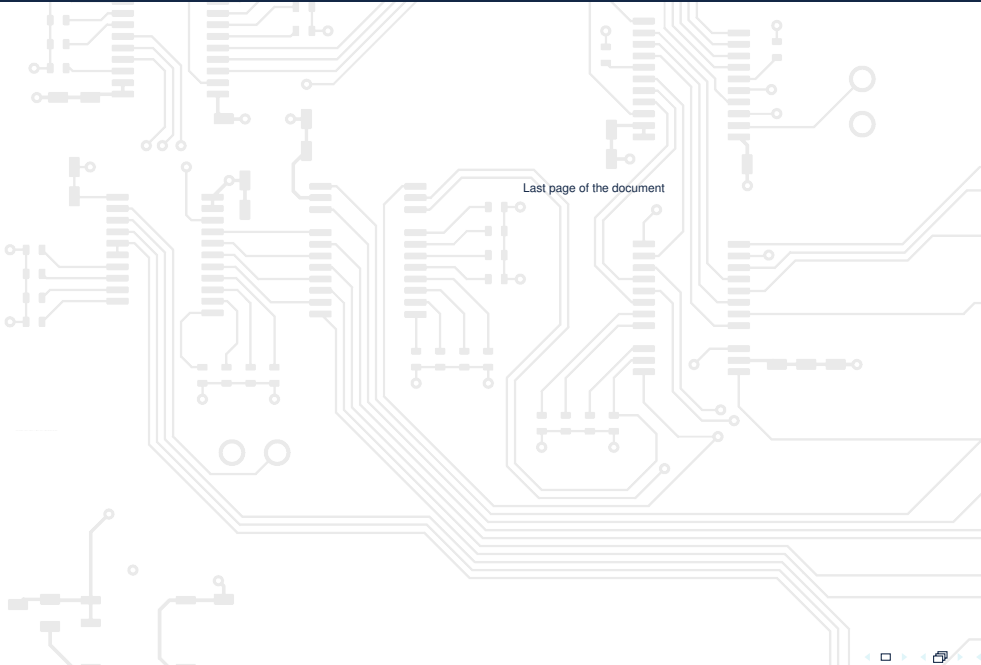
■ x2_daq_visu

- spartan
- files
- acquisition
- data display
- plot
- print

5 CONCLUSION

- 
- **pourquoi je vous parle de tout ça ?**
 - un certain nombre des concepts sont transposables
 - firmware avec hdlmake ?
 - le partage pourrait dépasser le cadre de xemis

That's all folks Question?



Last page of the document