Developer Guide v0.1

Viljar Eikeland

September 2019

This document contains technical information regarding the test procedure.

1 Requirements

Access to wp3, data-format-sw & alpide-sw on https://git.app.uib.no/pct

Requires: g++(*includes gcc*), boost, cmake, python3(*pip3 and numpy*), doxygen, Graphviz(*optional*) and root.

For ubuntu the libraries can be installed as followed:

g++:

- sudo apt-get install g++

Boost:

- sudo apt-get install libboost-all-dev

Cmake:

- sudo apt-get install cmake

Python3:

- sudo apt-get install python3-dev
- sudo apt install python3-pip
- pip3 install numpy

Doxygen:

- sudo apt-get install doxygen

Graphviz(*optional*):

- sudo apt-get install graphviz

Root:

- https://root.cern/building-root
- install dependencies: sudo apt-get install dpkg-dev binutils libx11-dev libxpm-dev libxft-dev libxext-dev libpng-dev libjpeg-dev
- move to folder where you want to install root.
- sudo git clone http://github.com/root-project/root.git
- move to build folder: cd root/build/
- cmake ..
- make -jn (n is the nuber of CPU cores to use for the building)
- source bin/thisroot.sh
- If cmake or compiler can not find root or thisroot:
- add ". path/to/root/build/bin/thisroot.sh" to your .bashrc file

2 Cloning a git-app repository

Move to directory for pCT repositories

- Find the desired repository on git-app
- Find the desired branch and copy the ssh url
- use: git clone *git.app.ssh.url*
- Check current branch: git branch –list
- switch branch: git checkout *branch name*

3 Building data-format-sw

Move to data-format-sw directory

- mkdir bin/output
- cd build
- cmake ..
- make EventDict
- sudo make lib
- (*Optional*) make doc

make EventDict will create a file named libEventDict_rdict.pcm. This file must be placed in alpide-sw/build/bin in order to generate the root file when a test is performed.

4 Building alpide-sw

Move to alpide-sw directory

- cd build
- cmake ..
- make name_of_test.cpp
- cd bin
- ./name_of_test

All tests are named main_name_of_test.cpp
If a script has the prefix main_ ,the cmake command will automatically generate the makefile needed to make the test executable.

5 Available tests

- FIFO scan
  Checks the FIFO memories of the Alpide
- Analog Scan
  Checks the analog part of the Alpide, looks for noisy or dead pixels.
- Digital Scan
  Checks the digital part of the Alpide, looks for stuck, unmaskable and unpulsable pixels.
- DAC scan
  Checks the electrical components of the Alpide.
- Root file
  Generates a root file when provided a .txt documents containing pRU data
- Threshold scan
  Checks the threshold for individual pixels on the Alpide
- Single Event
  Collects hit data from the Alpide