MARVELL

SPICA5 FW and API interwork

Spica5 SDK team

13 May 2022

MARVELL

Spica5 FW & API interwork

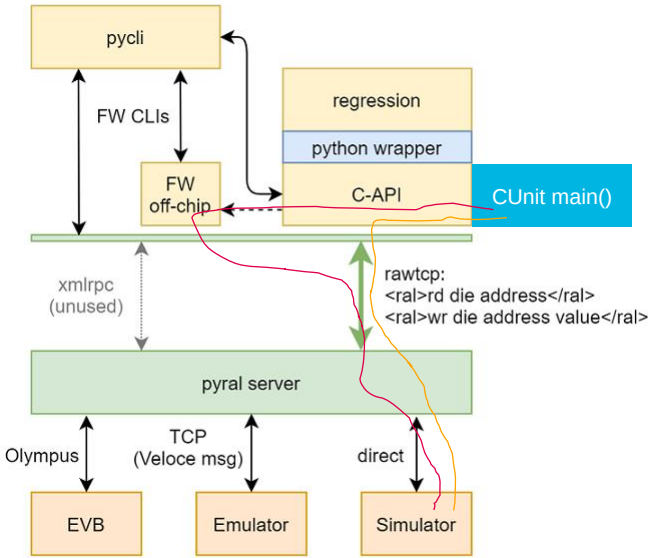
Spica5 SDK team

04 May 2022

Spica5 Overview

API Unit test:
check unittest c code – api – fw – simulator debug
(in dc5 chamber: https://dc5lp-vinetx000.marvell.com/)

CUnit Test Models



- Start pyral along with simulator:
  - [venv] [dle@dc5lp-vinetx076 ~/python3_venv]$ python ${HSC_DIR}/spica5/tools/pyral/scripts/ral_server --chip spica5 --port 50000 simulator
- Start off-chip FW:
  - [venv] [dle@dc5lp-vinetx076 ~/python3_venv]$ ${HSC_DIR}/spica5/firmware/application/.x86/spica5_fw_app --ip localhost --port 50000 --cli-port 52000
- Either start unit test:
  - ${HSC_DIR}/spica5/api/capi/build-output/tests/tests --rawtcp --ip 127.0.0.1 --port 50000 --product spica5
- Or start gdb to run the unit test

Ref: overview.md

Setup ral_server, simulator and x86 FW

- Start the ral_server along with the simulator in python3 environment:
  - [venv] [dle@dc5lp-vinetx076 ~/python3_venv]$ python ${HSC_DIR}/spica5/tools/pyral/scripts/ral_server --chip spica5 --port 50000 simulator
- Compile the FW in x86:

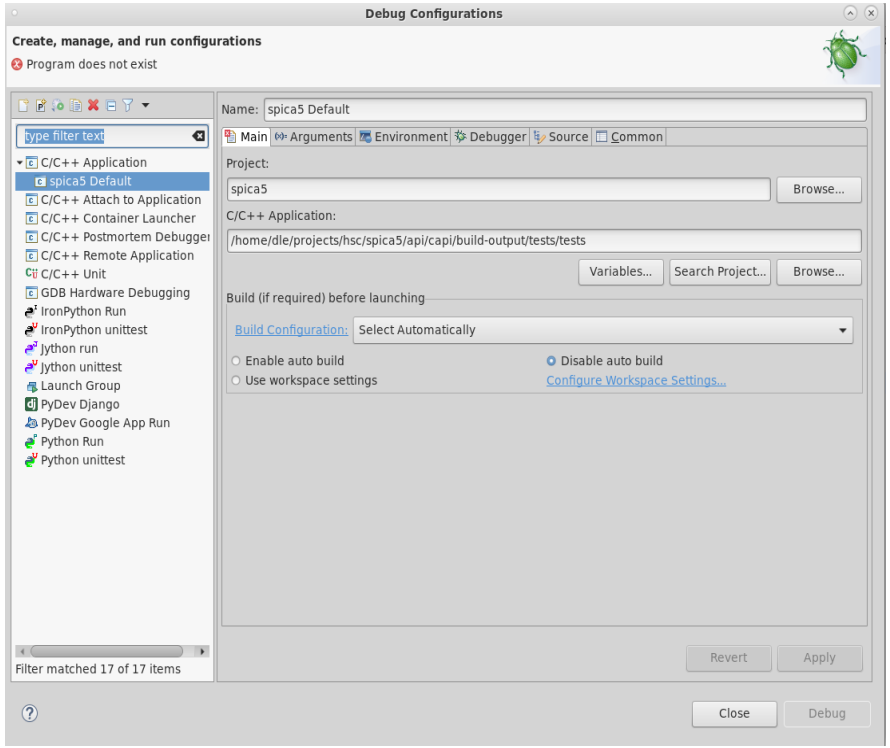
```
cd ${HSC_DIR}/spica5/firmware/application

# Debug mode:
# We invoke Meson with the setup command, giving it the location of the build directory. Meson uses out of source builds.
# Meson setup just need to be done once
meson setup .x86 --buildtype=plain
meson setup .x86 -Dxmlrpc=false –reconfigure

# To start the build, simply type the following command.
# Meson will use the Ninja backend to build project
meson compile -C .x86 -j 3

# If you want to clean, following command can be used.
ninja -C .x86 clean
```

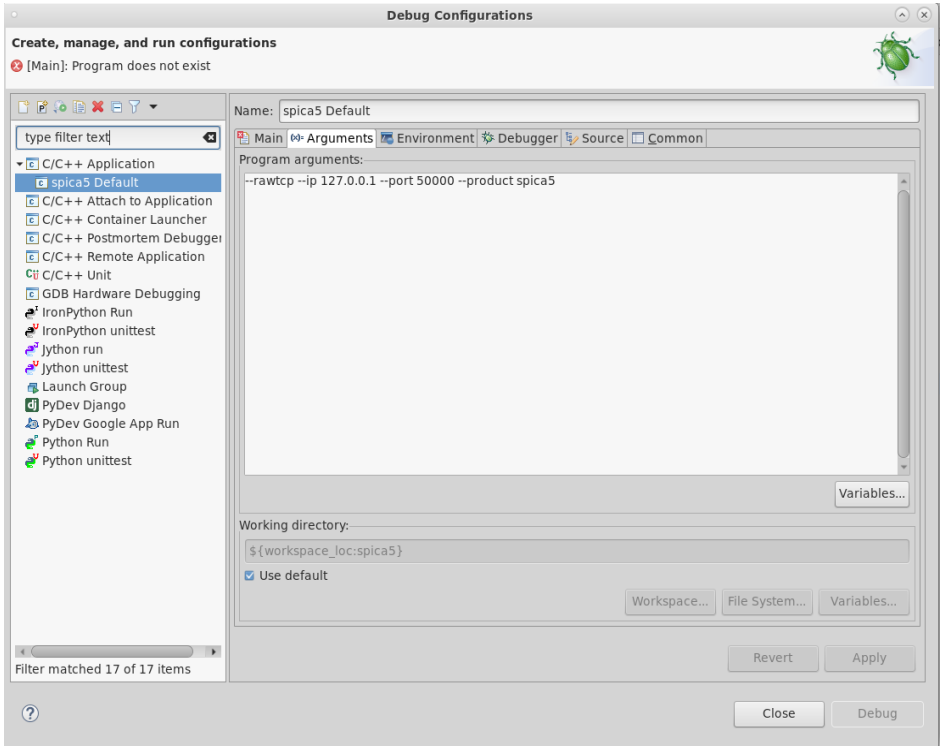
- Start the FW in x86:
  - ${HSC_DIR}/spica5/firmware/application/.x86/spica5_fw_app --ip localhost --port 50000 --cli-port 52000

Compile the API and CUNIT test with debug flag

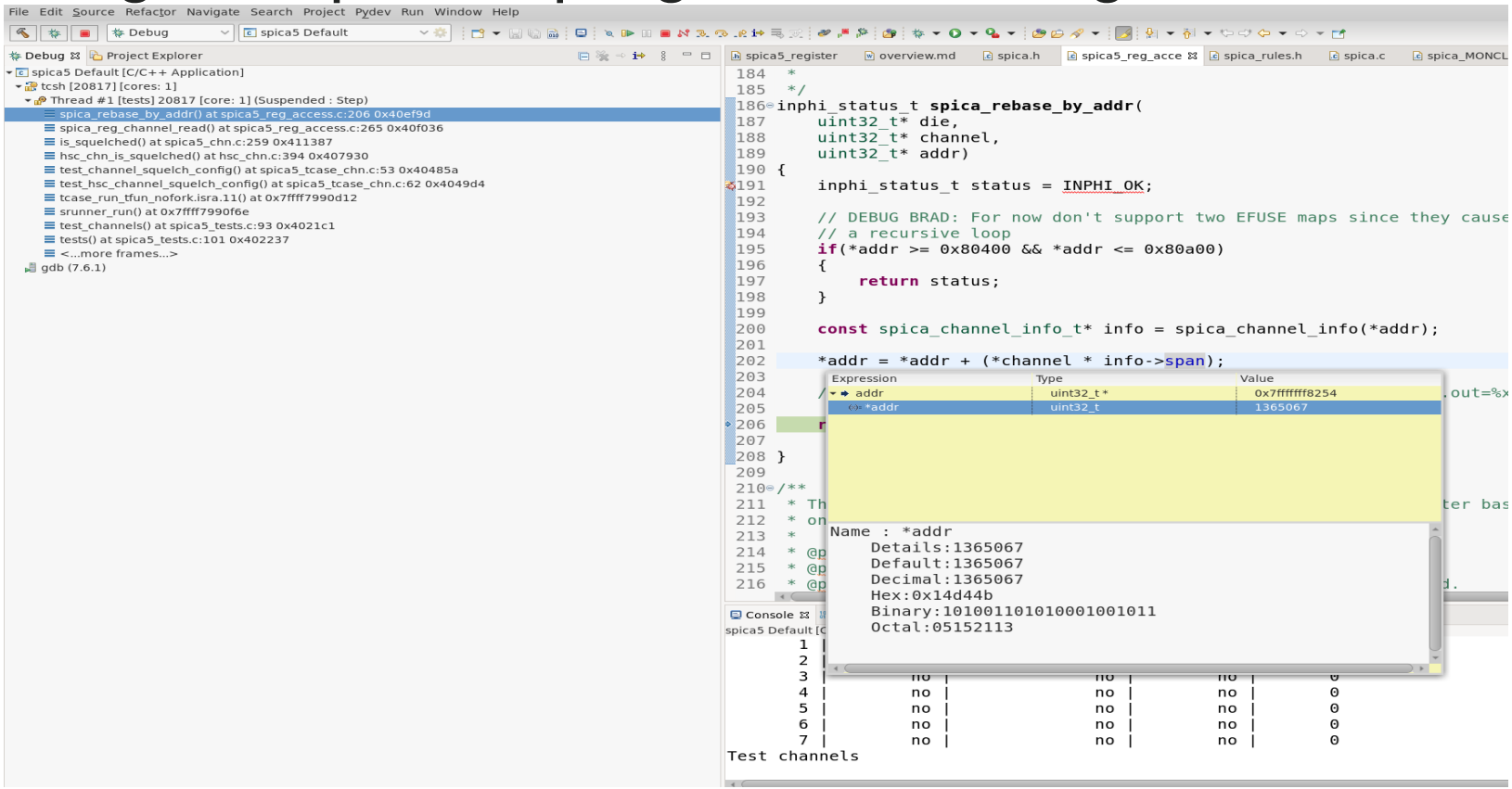
- Clean CAPI and test:
  - [dle@dc5lp-vinetx076 capi]$ make -C ${HSC_DIR}/spica5/api/capi clean
  - [dle@dc5lp-vinetx076 capi]$ make -C ${HSC_DIR}/spica5/api/capi/tests clean
- Make universal api clib:
  - [dle@dc5lp-vinetx076 capi]$ make -C ${HSC_DIR}/spica5/api/capi PRODUCTS=spica5 PROJECT_PREFIX=spica5 CC_OPTIONS="-g3 -O0" legacy -j5
- Make legacy api clib:
  - [dle@dc5lp-vinetx076 capi]$ make -C ${HSC_DIR}/spica5/api/capi PRODUCTS=spica5_spicap PROJECT_PREFIX=spica5 CC_OPTIONS="-g3 -O0" legacy -j5
- Make the unit test:
  - [dle@dc5lp-vinetx076 capi]$ make -C ${HSC_DIR}/spica5/api/capi/tests xmlrpc
  - [dle@dc5lp-vinetx076 capi]$ make -C ${HSC_DIR}/spica5/api/capi/tests PROJECT_PREFIX=spica5 CC_OPTIONS="-g3 -O0" DEBUG=1 -j 3

Debug the api, test program with the gdb

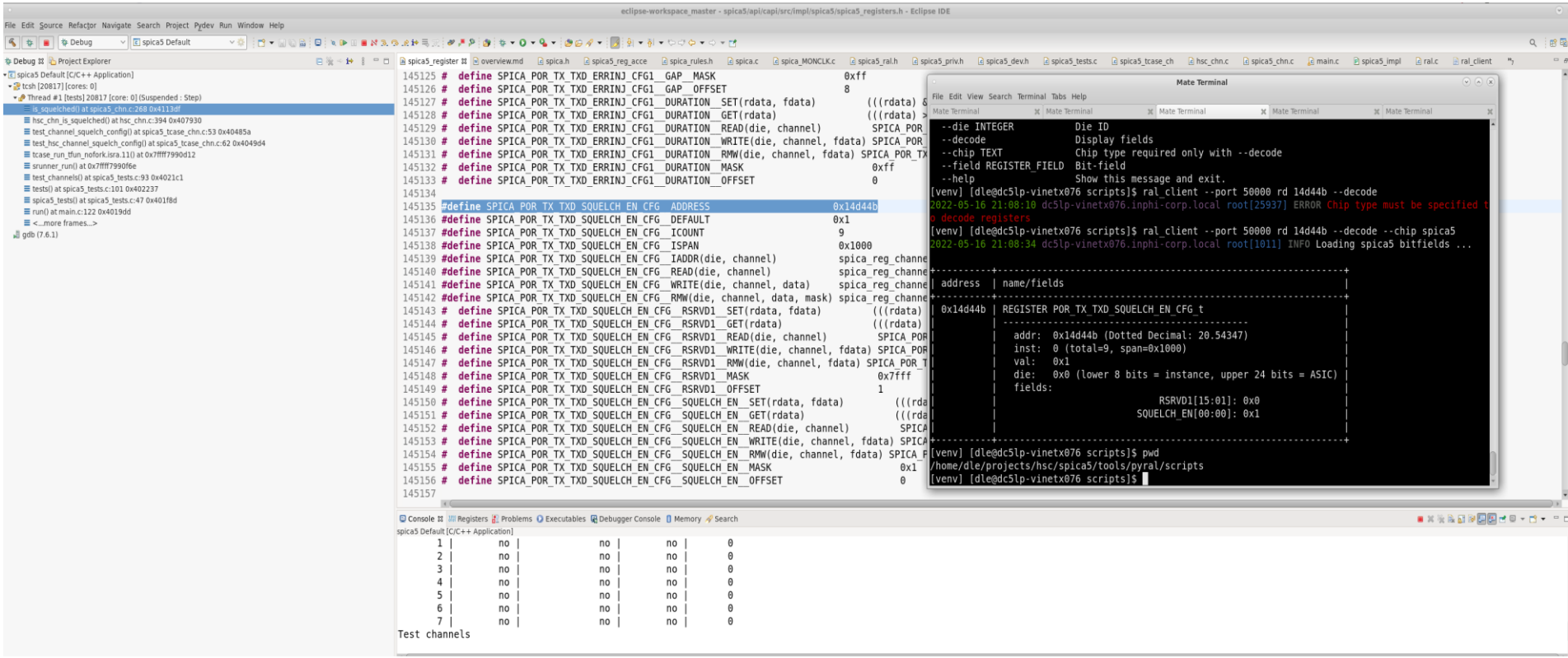




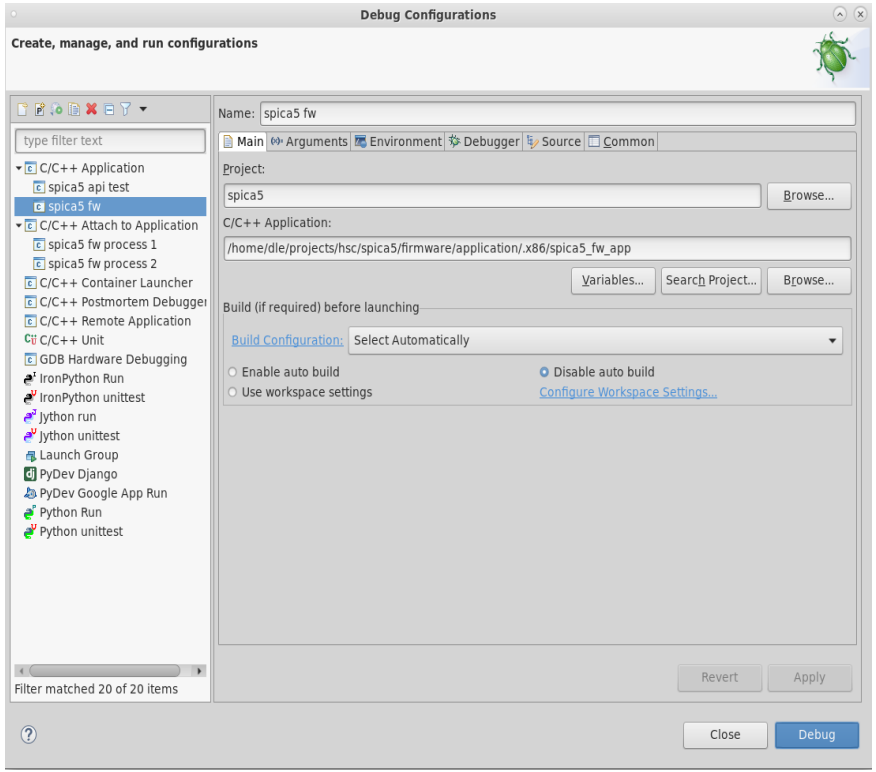
Debug the api, test program with the gdb

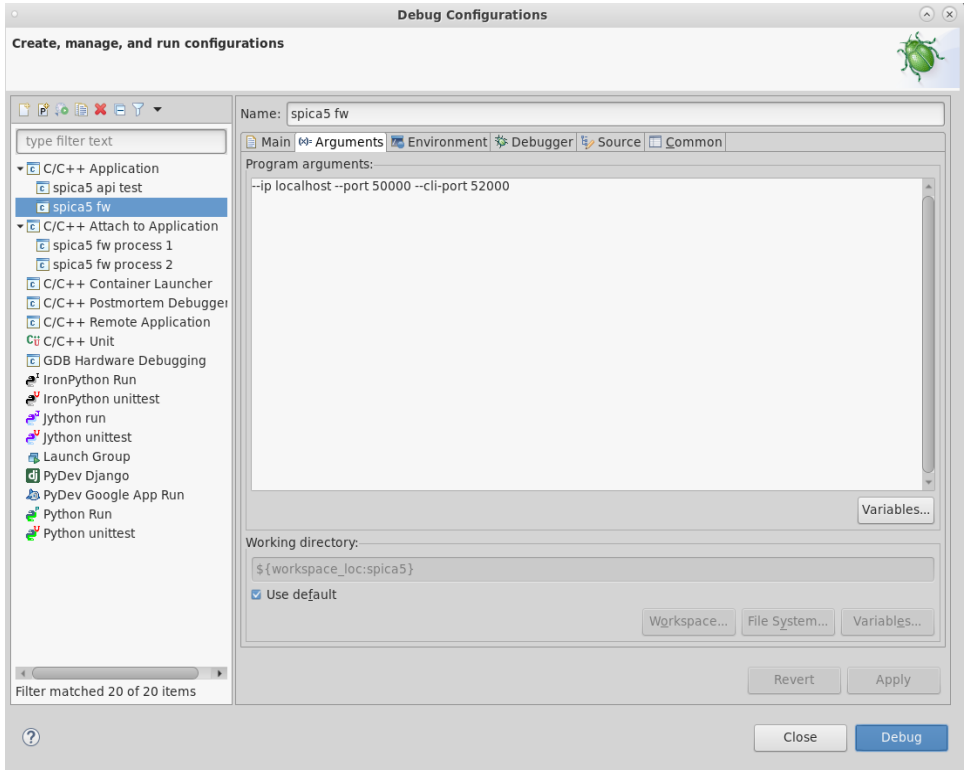


Double check register with the ral_client tool

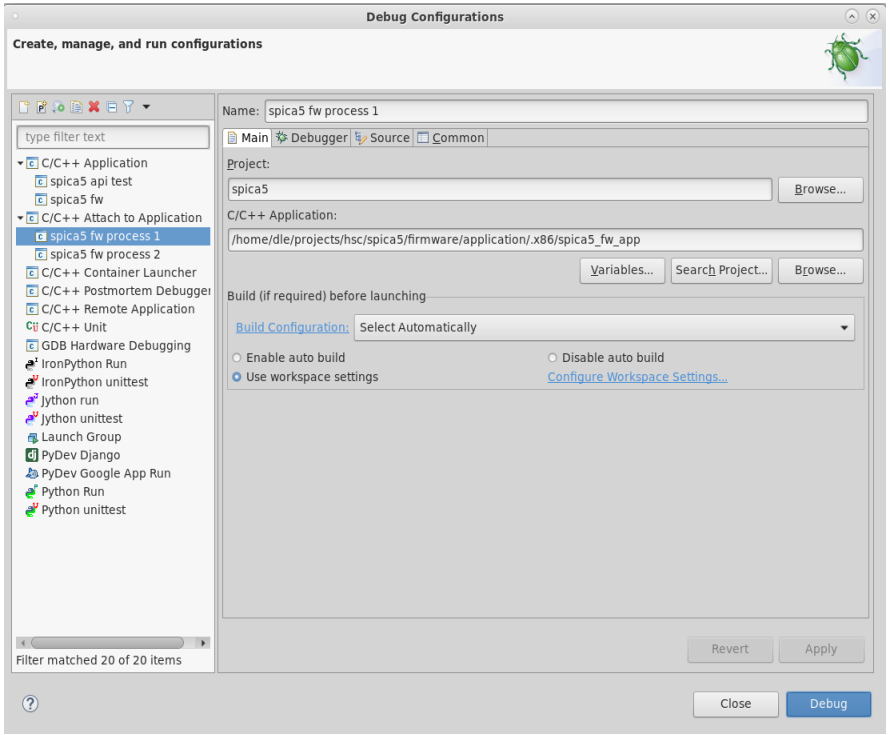


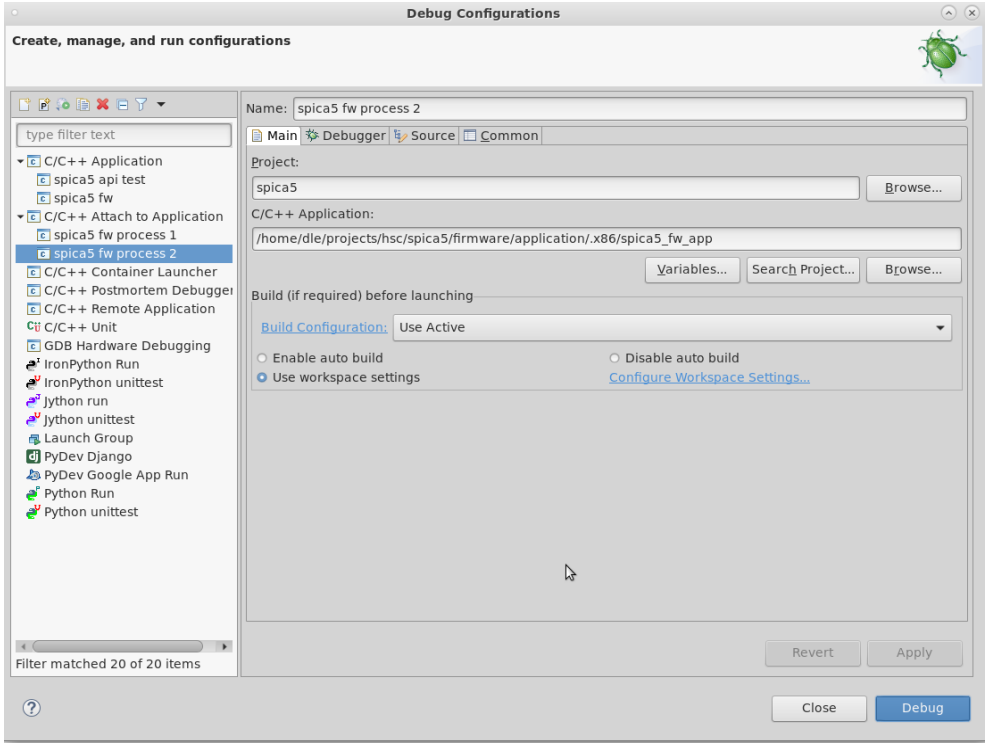
Debug the FW with gdb: setup eclipse for the main process



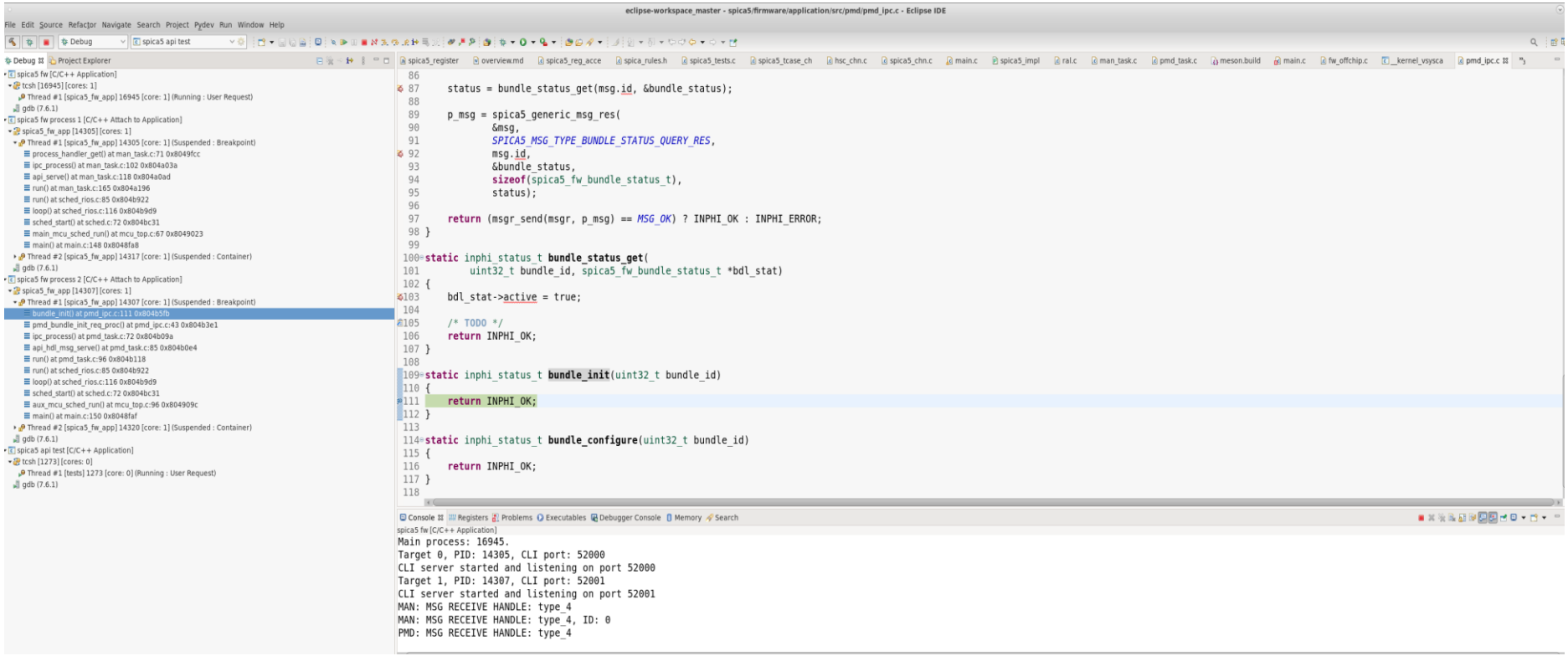


Debug the FW with gdb: setup eclipse for the 2 forked processes (pmd_task, man_task)

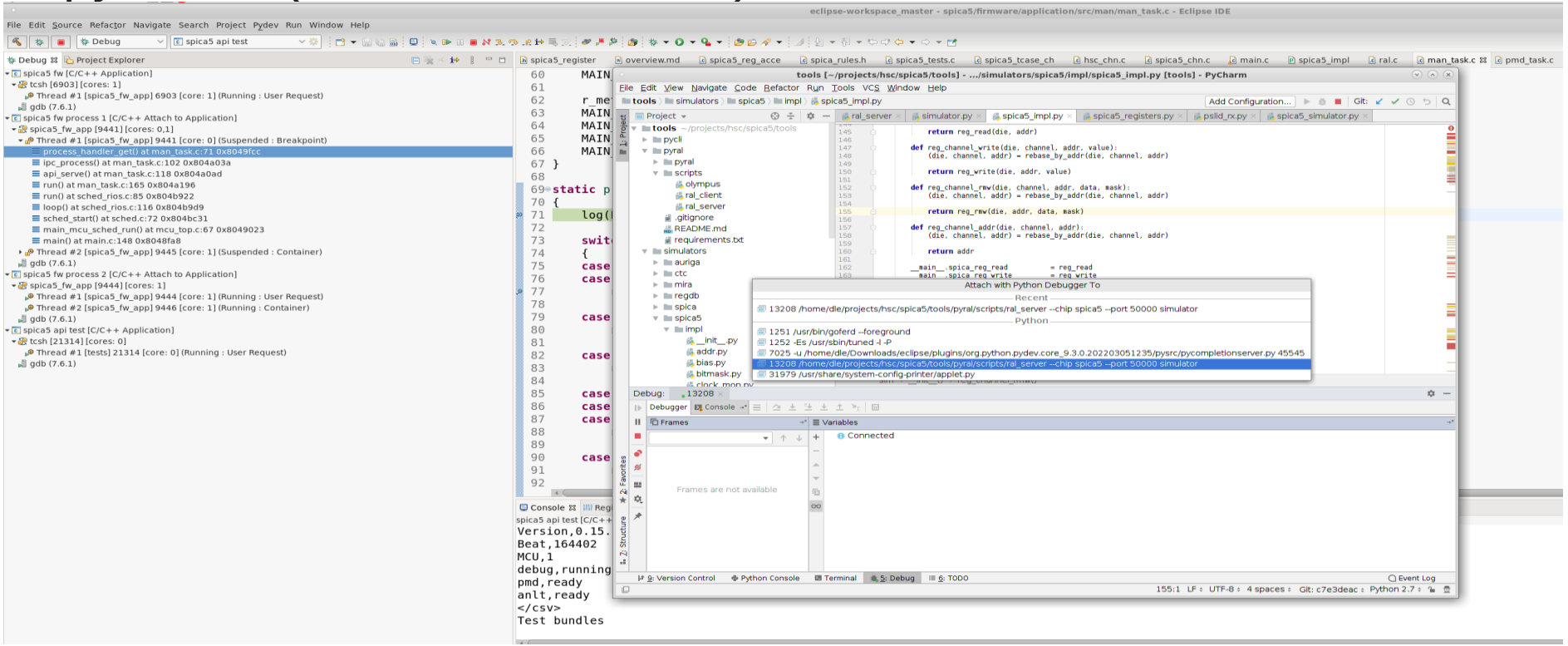




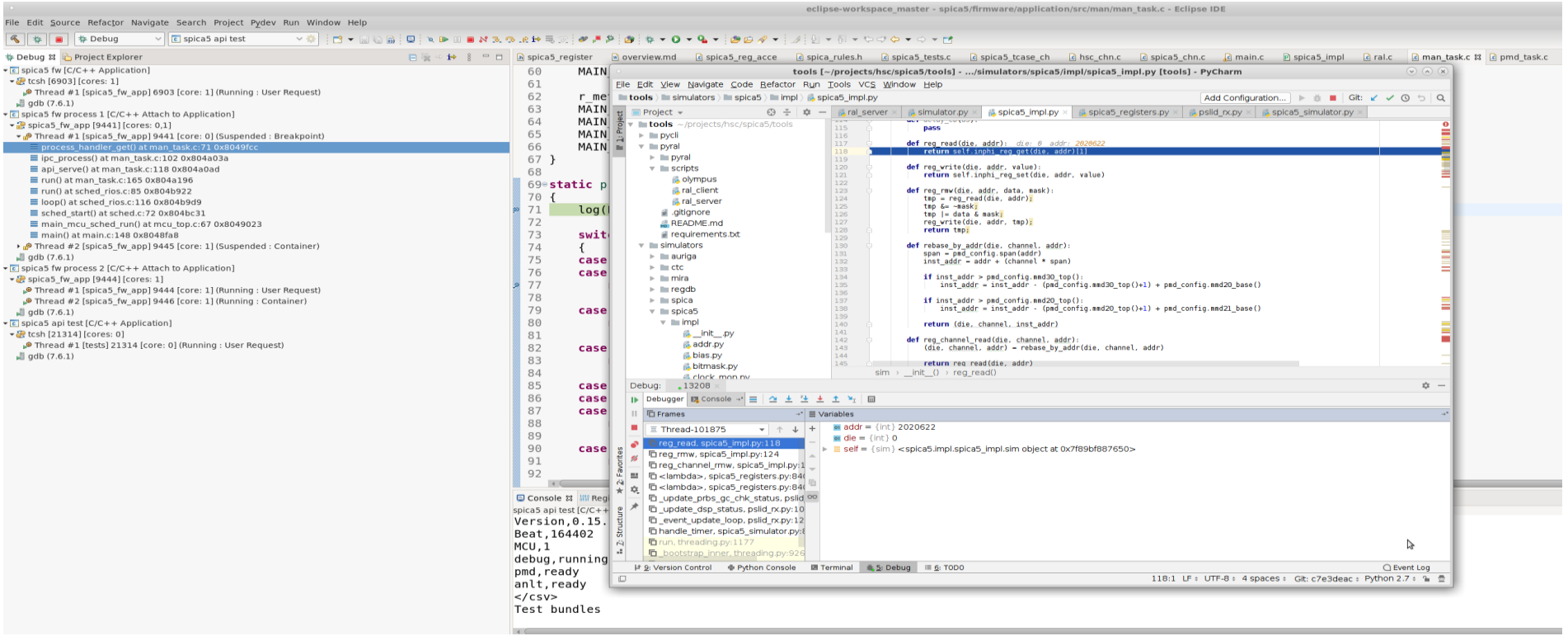
Debug the FW with gdb:
api unit test <-> fw man task process <-> pmd task process



Debug the python simulator with pycharm: run ral_server with the simulator and attach the python process to the pycharm (from the run menu)



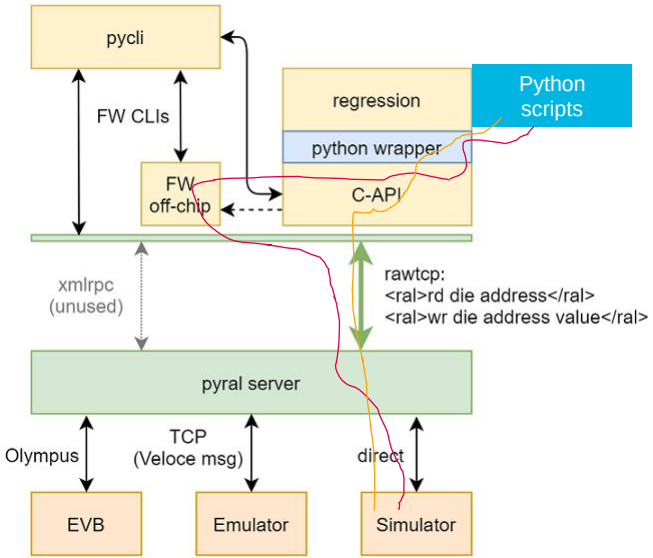
Debug the python simulator with pycharm:
api unit test <-> fw man task process <-> pmd task process <-> simulator



Tools in dc5lp

- Pycharm:
  - /tools/cad/jetbrains/pycharm/2019.13/Linux/bin/pycharm.sh
- Python3:
  - /tools/cad/python3.9.12/bin/python3
  - setenv PATH /tools/cad/python3.9.12/bin/:$PATH
- Create python3 virtual environment (so that you can use pip to install any python packages):
  - [dle@dc5lp-vinetx076]$ python -m venv python3.9_venvok
  - source python3.9_venvok/bin/activate.csh
  - pip install -r ${HSC_DIR}/spica5/tools/pyral/requirements.txt

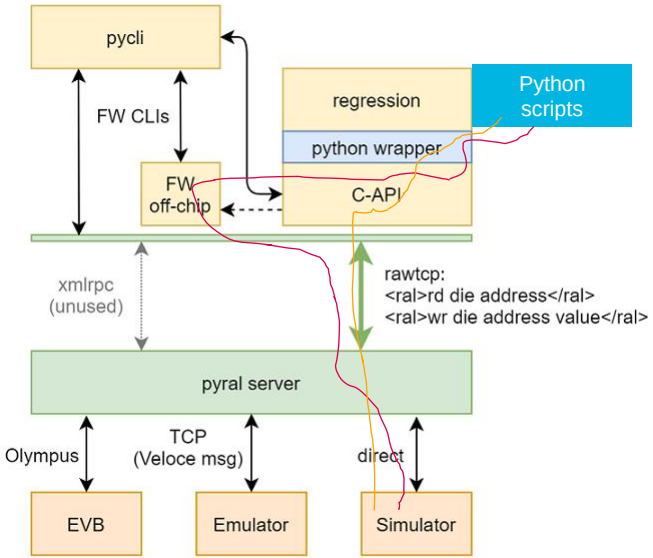
Use python3 with swig wrapper python3 lib to generate Python API and run python3 api sample script with 2 mcus



Ref: overview.md

- Start pyral along with simulator:
  - source ~/python3.9_venvok/bin/activate.csh
  - python ${HSC_DIR}/spica5/tools/pyral/scripts/ral_server --chip spica5 --port 50000 simulator
- Start off-chip FW:
  - [dle@dc5lp-vinetx028 ~]$ ${HSC_DIR}/spica5/firmware/application/.x86/spica5_fw_app --ip localhost --port 50000 --cli-port 52000
- 3 options to make api to generate python lib
  - Generate both universal api and legacy api:
    - [dle@dc5lp-vinetx028 capi]$ make -C ${HSC_DIR}/spica5/api/capi PRODUCTS=spica5 PROJECT_PREFIX=spica5 pyversion=3.9 python
  - Generate python lib for universal api only:
    - [dle@dc5lp-vinetx028 capi]$ make -C ${HSC_DIR}/spica5/api/capi PRODUCTS=spica5 PROJECT_PREFIX=spica5 pyversion=3.9 pythonuapi
  - Generate python lib for legacy api only:
    - [dle@dc5lp-vinetx028 capi]$ make -C ${HSC_DIR}/spica5/api/capi PRODUCTS=spica5 PROJECT_PREFIX=spica5 pyversion=3.9 pythoncapi
- Go to examples/py/spica5 in the python3 venv to run an example python script:
  - cd ${HSC_DIR}/spica5/api/capi/examples/py/spica5
  - setenv PYTHONPATH /home/dle/projects/hsc/spica5/tools/pyral/pyral/:$PYTHONPATH
  - Run universal api example: python odsp_opaq_api_sample.py
  - Run legacy api example: python odsp_api_sample.py

Use python3 with swig wrapper python3 lib to generate Python API and run python3 api sample script with 1 cmu framework



Ref: overview.md

- Start pyral along with simulator:
  - source ~/python3.9_venvok/bin/activate.csh
  - python ${HSC_DIR}/spica5/tools/pyral/scripts/ral_server --chip spica5 --port 50000 simulator
- Start off-chip FW:
  - [dle@dc5lp-vinetx028 ~]$ ${HSC_DIR}/spica5/firmware/application/.x86/spica5_fw_app --ip localhost --port 50000 --cli-port 52000 --mcu-id 0
- 3 options to make api to generate python lib
  - Generate both universal api and legacy api:
    - [dle@dc5lp-vinetx028 capi]$ make -C ${HSC_DIR}/spica5/api/capi PRODUCTS=spica5 PROJECT_PREFIX=spica5 pyversion=3.9 python
  - Generate python lib for universal api only:
    - [dle@dc5lp-vinetx028 capi]$ make -C ${HSC_DIR}/spica5/api/capi PRODUCTS=spica5 PROJECT_PREFIX=spica5 pyversion=3.9 pythonuapi
  - Generate python lib for legacy api only:
    - [dle@dc5lp-vinetx028 capi]$ make -C ${HSC_DIR}/spica5/api/capi PRODUCTS=spica5 PROJECT_PREFIX=spica5 pyversion=3.9 pythoncapi
- Go to examples/py/spica5 in the python3 venv to run an example python script:
  - cd ${HSC_DIR}/spica5/api/capi/examples/py/spica5
  - setenv PYTHONPATH /home/dle/projects/hsc/spica5/tools/pyral/pyral/:$PYTHONPATH
  - Run universal api example: python odsp_opaq_api_sample_1mcu.py
  - Run legacy api example: python odsp_api_sample.py

Essential technology, done right™


MARVELL™