|  |  |
| --- | --- |
| **Program / Project Name:** | CA5G |
| **Checklist / Template Completed by:** | TBD |
| **Date Completed:** | Click here to enter a date. |

**Tips for CA5G Development**

Revision: A.0

**Detailed Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Editor(s)** | **Description of change** |
| A.0 | 18-SEP-2019 | Sang-Gu Kang | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1 Environment 4](#_Toc21613849)

[1.1 SSH 4](#_Toc21613850)

[1.2 Manual updated 4](#_Toc21613851)

[1.3 QtCreator Setup 4](#_Toc21613852)

[2 CMake 9](#_Toc21613853)

[2.1 CMakeLists.txt 9](#_Toc21613854)

[3 Option활성화 비활성화 9](#_Toc21613855)

[3.1 Activation 9](#_Toc21613856)

[3.2 Deactivation 9](#_Toc21613857)

[4 Temp 9](#_Toc21613858)

[5 Util 20](#_Toc21613859)

[5.1 doQt 20](#_Toc21613860)

[5.2 gitk 20](#_Toc21613861)

[6 Project 21](#_Toc21613862)

# Environment

## SSH

Port: 22

User: root

Password: SiG2018

## Manual updated

\\ds\apdata$\SEO\RFTest\PLM\Public\Technical Publications

## QtCreator Setup

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명  
스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

스크린샷이(가) 표시된 사진

자동 생성된 설명

### Build 설정

스크린샷, 노트북, 컴퓨터, 실내이(가) 표시된 사진

자동 생성된 설명

### Run(Deploy) 설정

스크린샷, 컴퓨터, 노트북이(가) 표시된 사진

자동 생성된 설명

* Build 결과물을 장비에 write하기 위해 필요한 설정이다.
* Add Deploy Step버튼을 눌러 Custom Process Step을 선택한다.
* Command: /usr/bin/rsync
* Arguments: -av ./install/\* --rsh="sshpass -p SiG2018 ssh -l root" [root@10.82.26.22:/](mailto:root@10.82.26.22:/)
* Run configuration: (on Remote Device)로 되어 있는 것 선택
* Alternate executable on device: 에서 Use this command instead 체크박스 선택한 후 service.sh입력

### Trouble shooting

* 설정값이 제대로 되지 않는다면(Projects>Build에 Build directory가 제대로 잡히지 않을 때) Build directory 안의 내용을 모두 지우고, 커맨드 라인 상에서 build-ca5g-CA5G-Default 디렉토리를 날려버리고 다시 CMake Run을 실행한다.

# CMake

## CMakeLists.txt

~works/ca5g/ca5g/apps/CMakeLists.txt  
-> add\_subdirectory(signalAnalyzer5GNR)

~works/ca5g/ca5g/apps/signalAnalyzer5GNR/CMakeLists.txt  
-> add\_subdirectory (signalAnalyzer5GNRBuildIn)  
-> add\_subdirectory (signalAnalyzer5GNRUtil)

# Option활성화 비활성화

SSH 접속 후 직접 입력. 타 부서 및 문서 노출 절대 안됨.

## Activation

enable\_options.sh F001O

## Deactivation

disable\_options.sh F001O

# Branch

## Branch changing flow

브랜치를 변경한 경우에는 부모 프로젝트와 자식 프로젝트 간 submodule을 초기화 시켜주어야 한다.

* Git clone  
  shlim@s:~/workspace/practice$ git clone ssh://git@cosgit1.ds.jdsu.net:7999/ca5g/ca5g.git
* Checkout  
  shlim@s:~/workspace/practice/ca5g$ git checkout feature/HA-3653-tm-analyzer
* Submodule deinit  
  
* Submodule init  
  
* doQt 실행  
  open CMakeLists.txt
* Run CMake  
  Build > Run CMake
* Build, Run 설정 체크  
  -. 좌측 Projects 버튼 선택  
  -. Build 항목 선택 후 1.3.1 Build 설정 참조  
  -. Run 항목 선택 후 1.3.2 Run(Deploy)설정 참조
* Build All  
  Build > Build All
* Deploy All  
  Build > Deploy All
* Run
* Warning message  
  Warning: The unit file, source configuration file or drop-ins of scpi.service changed on disk. Run 'systemctl daemon-reload' to reload units.와 같은 warning 메시지가 발생하면 장비에 systemctl daemon-reload 명령을 타이핑 해 주거나 장비를 다시 시작하면 된다.

## Branch commit

하위 디렉토리의 변경된 내용을 먼저 commit 하고 상위 ca5g 디렉토리로 와서 다시 commit한다???

# submodule

.gitmodules에 정보가 있음.

|  |
| --- |
| [submodule "elf"]  path = elf  url = https://gitlab.ds.jdsu.net/lee68914/elf-separation.git  [submodule "lib"]  path = lib  url = ssh://git@cosgit1.ds.jdsu.net:7999/ca5g/ca5g\_libs.git  branch = feature/HA-3653-tm-analyzer  [submodule "apps"]  path = apps  url = ssh://git@cosgit1.ds.jdsu.net:7999/ca5g/ca5g\_apps.git  branch = feature/HA-3653-tm-analyzer  [submodule "qml"]  path = qml  url = ssh://git@cosgit1.ds.jdsu.net:7999/ca5g/ca5g\_ui.git  branch = feature/HA-3653-tm-analyzer  [submodule "services"]  path = services  url = ssh://git@cosgit1.ds.jdsu.net:7999/ca5g/services.git  branch = feature/HA-3653-tm-analyzer |

ca5g 디렉토리에서 branch를 변경하더라도 ca5g\_apps 디렉토리에는 branch 변경이 이루어지지 않는다. 따라서 branch를 일치시켜 놓고 작업하는 것이 편하다.

# Make New Feature flow

## directory 생성

* mkdir signalAnalyzerTM
* ~workspace/practice/ca5g/apps/ signalAnalyzerTM /
* mkdir signalAnalyzerTMBuildIn
* ~workspace/practice/ca5g/apps/ signalAnalyzerTM/ signalAnalyzerTMBuildIn /
* mkdir signalAnalyzerTMUtil
* ~workspace/practice/ca5g/apps/ signalAnalyzerTM/ signalAnalyzerTMUtil /

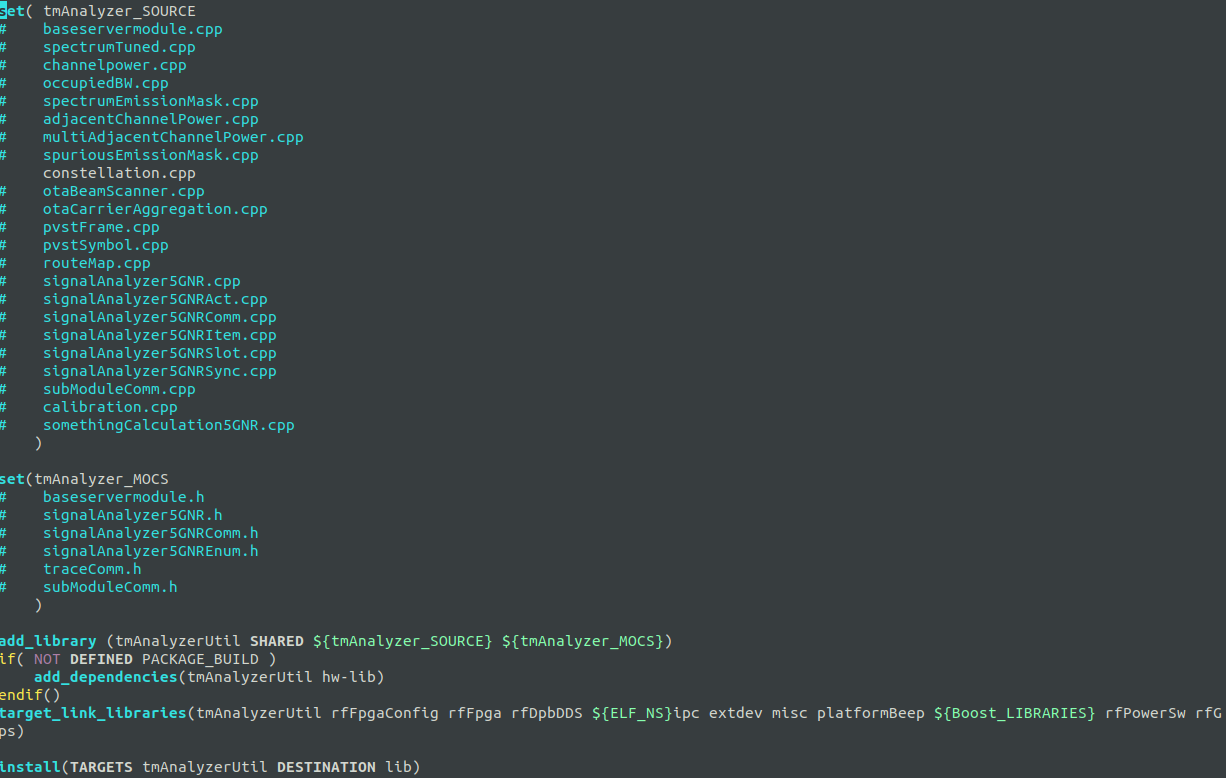
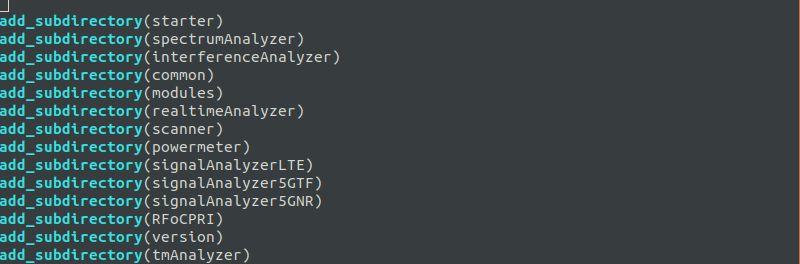
## CMakeLists.txt 파일 생성

* ~workspace/practice/ca5g/apps/signalAnalyzerTM/CMakeLists.txt

|  |
| --- |
| add\_subdirectory (signalAnalyzerTMBuildIn)  add\_subdirectory (signalAnalyzerTMUtil) |

* ~workspace/practice/ca5g/apps/signalAnalyzerTM/signalAnalyzerTMBuildIn/CMakeLists.txt

|  |
| --- |
| include\_directories (${PROJECT\_SOURCE\_DIR}/signalAnalyzerTM/signalAnalyzerTMUtil)  link\_directories (${PROJECT\_SOURCE\_DIR}/signalAnalyzerTM/signalAnalyzerTMUtil)  file( GLOB signalAnalyzerTMBuildIn\_SRCS \*.cpp )  add\_executable( signalAnalyzerTMBuildIn\_app  ${signalAnalyzerTMBuildIn\_SRCS}  )  target\_link\_libraries(signalAnalyzerTMBuildIn\_app  ${DBUS\_SERVER}  ${CRASHLOG\_LIB}  ${ELF\_NS}ipc  ${ELF\_NS}dbglog  ${ELF\_NS}utils  signalAnalyzerTMUtil  chstd  rfParam  rfAudio  rfUtils  rfTraceUtil  rfSA  dsp5GNR  misc  extdev  rfCalibrations  rfDnc  solSysInfo  rfFpga  rfDelay  rfIoExpand  rfPciUdev  rfLog  rfDpbJson  platformInventory  rfDnc18G  rfSystemInfo  platformClock  rfTemp  rfDncRevInfo  rfPci  rfDspCoef  rfI2c  rfFlashCalFileOp  rfOption  rfMmap  rfFlash  rfFlashGlobal  rfSpi  rfFlashJsonOp  ${Boost\_LIBRARIES}  )  #Helps get include paths right  qt5\_use\_modules( signalAnalyzerTMBuildIn\_app DBus Qml Gui Test Positioning SerialPort)  install(TARGETS signalAnalyzerTMBuildIn\_app DESTINATION base/bin) |

* ~workspace/practice/ca5g/apps/signalAnalyzerTM/signalAnalyzerTMBuildIn/main.cpp  
  아래 내용을 알맞게 수정해야 함. 지금은 Run CMake 때문에 임시로 한 것임.  
  
* ~workspace/practice/ca5g/apps/signalAnalyzerTM/signalAnalyzerTMUtil/CMakeLists.txt  
  tmAnalyzer\_SOURCE 부분에 적어도 하나의 파일을 주석처리 하지 않아야 Run CMake가 이상없이 진행된다.  
  
* ~workspace/practice/ca5g/apps/CMakeLists.txt  
  Run CMake를 실행한 후 맨 마지막 줄에 새로운 tmAnalyzer가 추가되었는지 확인하고 추가되지 않았다면 추가한다.  
  
* Manifest/process.json파일도 확인한다.

|  |
| --- |
| "proc\_signalAnalyzerTMBuildIn\_app": {  "tag": "signalAnalyzerTMBuildIn\_app",  "command": "/base/bin/signalAnalyzerTMBuildIn\_app",  "label": "Tm Analyzer Build In App",  "exclusive": "false",  "haltAction": ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:halt:"  } |

## Run CMake

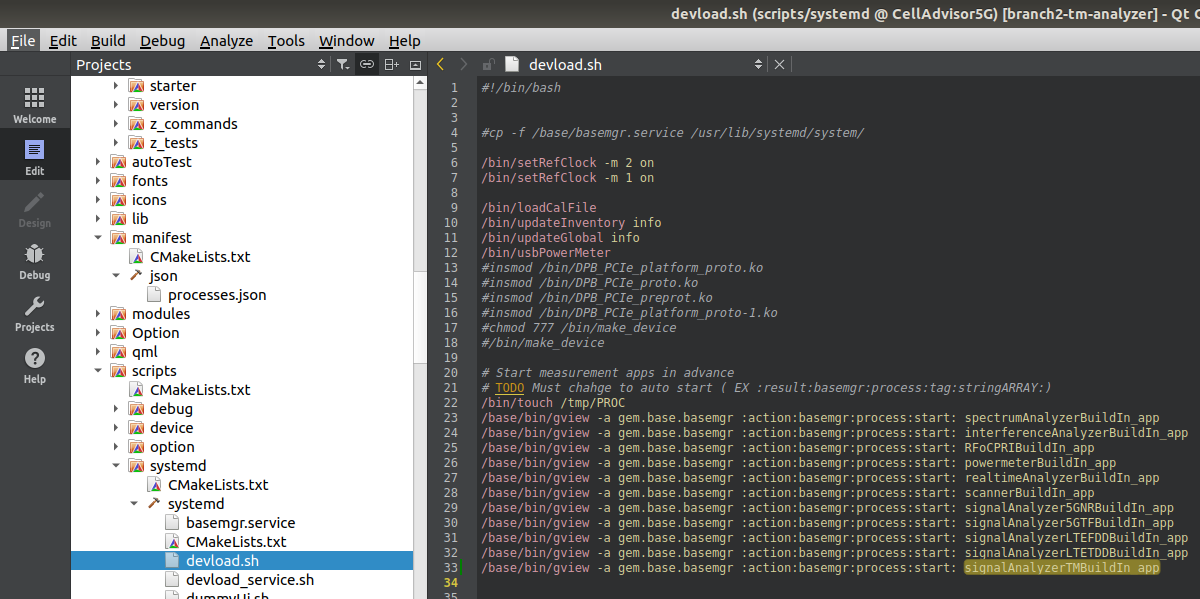
## SCPI

./services/scpi/config/ 디렉토리에 해당 feature와 config.json 파일을 만든다.



## devload.sh

./scripts/systemd/devload.sh를 수정해 주면 start가 동작한다.



# Temp

주요 작업은 /base/

장비의 저장 장치는 /user/cellAdvisor5G/Internal

Save/Load등 할 때 임시 파일 위치 /tmp

/tmp/logOn

Journalct기

X86에서 시작할 때는 start.sh

/base

/base/bin

service.sh or service.sh start는 켤 때, service.sh stop은 끌 때

/base/bin/qmljdsu

/base/manifest process.json 새로 measure를 만들면 이곳에 등록

Confluence에 UI와 협의한 Item들을 등록해야 함(서로 일치하게 하기 위해서)



Debug할 때는 gview나 dbus-monitor를 사용함. Gview는 개발할 때 유용하고, dbus-monitor는 UI와 App 중 문제가 어디 있는지 진단할 때 유용.








































Kit(CA5G)에서 실행할 수 있도록






# Util

## doQt

Qt 실행할 때는 doQt 명령으로 실행.

## gitk

git log와 유사함.

## subl

notepad와 비슷한 tool.

## gitg

## dbus-monitor

## gview

# Project

DSP library는 ca5g\_libs에 위치한다.

JIRA issue resolve하면 QA 팀으로 assign할 것.

# Source Analysis

## apps

### signalAnalyzer5GNR

## lib

### dsp5GNR

#### rfDSP5GNRModule.cpp

-. measure해서 data를 만든다.