|  |  |
| --- | --- |
| **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.

# Creating new feature

## check branch

### .gitmodule

lib 디렉토리는 변경사항이 없으므로 develop을 사용하고, 나머지 submodule은 해당 브랜치를 사용한다는 가정.



### branch

apps: 주요 작업이 해당 브랜치에서 이루어지므로 명확히 브랜치 이동이 이루어져 있어야 한다.

lib, qml, service: 서브모듈 링크를 따라가도록 해쉬를 가리키도록 되어 있다. 만약 GUI 담당자가 작업을 해서 업데이트가 되면 CMake가 돌 때, 자동으로 업데이트되도록 하는 효과가 있다.



## check directory and files



## check CMakeLists.txt





  


lib 디렉토리 때문에 아래 부분도 수정  


tmAnalyzer\_SOURCE 부분에 적어도 하나의 파일을 주석처리 하지 않아야 Run CMake가 이상없이 진행된다.





## check main.cpp

아래 내용을 알맞게 수정해야 함. 지금은 Run CMake 때문에 임시로 한 것임.



## Check processes.json



## Check devload.sh

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



## Check 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.

## Check SCPI

./services/scpi/config/ 디렉토리에 해당 feature와 action.json, config.json, result.json 파일을 만든다.



# 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???

# Useful git command

## add

git add manifest/processes.json scripts/system/devload.sh

git add dspTM/

## branch

git branch

## checkout

git checkout feature/HA-3653-tm-analyzer

git checkout develop

git checkout origin/feature/HA-3653-tm-analyzer

git checkout 26e67aba

git checkout <file name>

## clean

git clean -f

## commit

git commit -am “Moved dspTM to apps repo”

git commit -a

## merge

git merge --abort

git merge develop

## mergetool

git mergetool

## pull

git pull origin feature/HA-3653-tm-analyzer

git pull

## push

git push

git push origin feature/HA-3653-tm-analyzer

git push origin origin/feature/HA-3653-tm-analyzer

## rebase

git rebase --continue

git rebase --abort

git rebase develop

git rebase --help

git rebase origin/develop

## reset

git reset --hard origin/feature/HA-3653-tm-analyzer

git reset --hard

## rm

git rm -rf dspTM/

git rm dsp5GNR/consts.h

git rm -r dsp5GNR/ -f

## submodule

git submodule update --recursive --remote

git submodule update --recursive --remote -f

git submodule init

git submodule deinit -f --all

## status

git status

# Debug

## gview

SSH에서 해당 명령을 입력하여 코드에서의 printf 메시지를 본다.

|  |
| --- |
| gview -f $(tty) -o gem.base.signalAnalyzerTMBuildIn\_app |

## ps

SSH에서 해당 명령을 입력하여 process가 제대로 도는지 확인한다.

|  |
| --- |
| ps -ax |
| 3625 ? Sl 0:02 /base/bin/signalAnalyzerTMBuildIn\_app |

## qDebug()

logOn.py 파일을 수정한다.

SSH에서 logOn을 한번 입력해 준다. 그러면 /tmp/ 디렉토리에 추가한 5gtm.log 파일이 생긴다.

SSH에서 tail -f /tmp/5gtm.log 명령을 주면 qdebug 메시지가 보인다.



# 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.