|  |  |
| --- | --- |
| **Program / Project Name:** | Street Lamp Auto Sensing System |
| **Checklist / Template Completed by:** | TBD |
| **Date Completed:** | Click here to enter a date. |

**NuTiny-SDK-M051 User Manual**

Revision: A.0

**Detailed Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Rev** | **Date** | **Editor(s)** | **Description of change** |
| A.0 | 21-JUL-2020 | Sang-Gu Kang | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

[1 Overview 7](#_Toc46261626)

[2 NuTiny-SDK-M051 introduction 7](#_Toc46261627)

[2.1 NuTiny-SDK-M051 Jumper Description 8](#_Toc46261628)

[2.1.1 Power Setting 8](#_Toc46261629)

[2.1.2 Debug Connector 8](#_Toc46261630)

[2.1.3 USB Connector 8](#_Toc46261631)

[2.1.4 Extended Connector 8](#_Toc46261632)

[2.1.5 Reset Button 9](#_Toc46261633)

[2.1.6 Power Connector 9](#_Toc46261634)

[2.2 Pin Assignment for Extended Connector 9](#_Toc46261635)

[2.3 NuTiny-SDK-M051 PCB Placement 9](#_Toc46261636)

[3 Keil uVision® IDE에서 NuTiny -SDK-M051을 시작하는 방법 10](#_Toc46261637)

[3.1 Keil uVision® IDE Software Download & Install 10](#_Toc46261638)

[3.2 Nuvoton Nu-Link Driver Download & Install 10](#_Toc46261639)

[3.3 Hardware Setup 10](#_Toc46261640)

[3.4 Smpl\_NuTiny-M051 Example Program 11](#_Toc46261641)

[4 IAR Embedded Workbench에서 NuTiny-SDK-M051을 시작하는 방법 12](#_Toc46261642)

[4.1 IAR Embedded Workbench Software Download &Install 12](#_Toc46261643)

[4.2 Nuvoton Nu-Link Driver Download & Install 13](#_Toc46261644)

[4.3 Hardware Setup 13](#_Toc46261645)

[4.4 Smpl\_NuTiny-M051 Example Program 13](#_Toc46261646)

[5 M052\_TINY-EVB Schematic 14](#_Toc46261647)

[5.1 add file 17](#_Toc46261648)

[5.2 SCPI 명령 작성 17](#_Toc46261649)

[5.3 telnet을 이용한 검증 17](#_Toc46261650)

[5.4 scpi debugging 17](#_Toc46261651)

[6 CMakeLists 18](#_Toc46261652)

[7 Option활성화 비활성화 19](#_Toc46261653)

[7.1 Activation 19](#_Toc46261654)

[7.2 Deactivation 19](#_Toc46261655)

[8 Bitbucket 19](#_Toc46261656)

[8.1 pull request 19](#_Toc46261657)

[9 SW Integration Test 20](#_Toc46261658)

[9.1 Cucumber 20](#_Toc46261659)

[9.1.1 repository 20](#_Toc46261660)

[9.1.2 환경설정 21](#_Toc46261661)

[9.1.3 실행 21](#_Toc46261662)

[9.1.4 결과 21](#_Toc46261663)

[9.1.5 등록 21](#_Toc46261664)

[9.1.6 Debug 22](#_Toc46261665)

[10 Git 25](#_Toc46261666)

[10.1 Branch 25](#_Toc46261667)

[10.1.1 Branch changing flow 25](#_Toc46261668)

[10.1.2 Branch commit 26](#_Toc46261669)

[10.2 Useful git command 26](#_Toc46261670)

[10.2.1 add 26](#_Toc46261671)

[10.2.2 branch 27](#_Toc46261672)

[10.2.3 checkout 27](#_Toc46261673)

[10.2.4 clean 27](#_Toc46261674)

[10.2.5 commit 27](#_Toc46261675)

[10.2.6 merge 27](#_Toc46261676)

[10.2.7 mergetool 27](#_Toc46261677)

[10.2.8 pull 27](#_Toc46261678)

[10.2.9 push 27](#_Toc46261679)

[10.2.10 rebase 28](#_Toc46261680)

[10.2.11 reset 28](#_Toc46261681)

[10.2.12 rm 28](#_Toc46261682)

[10.2.13 submodule 28](#_Toc46261683)

[10.2.14 status 28](#_Toc46261684)

[10.3 SSH Key Gen 28](#_Toc46261685)

[11 Debug 31](#_Toc46261686)

[11.1 gview 31](#_Toc46261687)

[11.2 ps 31](#_Toc46261688)

[11.3 qDebug() 31](#_Toc46261689)

[12 FPGA loading 33](#_Toc46261690)

[12.1 develop 33](#_Toc46261691)

[12.2 Using lfs 35](#_Toc46261692)

[12.3 System Information 36](#_Toc46261693)

[13 Package 37](#_Toc46261694)

[13.1 package for test 37](#_Toc46261695)

[13.2 Nightly\_build 38](#_Toc46261696)

[13.3 Making package 39](#_Toc46261697)

[13.4 Sharing package 42](#_Toc46261698)

[13.5 Reimage 43](#_Toc46261699)

[14 Temp 44](#_Toc46261700)

[15 Util 54](#_Toc46261701)

[15.1 doQt 54](#_Toc46261702)

[15.2 gitk 55](#_Toc46261703)

[15.3 subl 55](#_Toc46261704)

[15.4 gitg 55](#_Toc46261705)

[15.5 dbus-monitor 55](#_Toc46261706)

[15.6 gview 58](#_Toc46261707)

[15.7 Byond Compare 61](#_Toc46261708)

[16 Test 61](#_Toc46261709)

[16.1 Measurement Switching Time 61](#_Toc46261710)

[17 Project 62](#_Toc46261711)

[18 Source Analysis 62](#_Toc46261712)

[18.1 파일 별 기능 구조 62](#_Toc46261713)

[18.1.1 baseservermodule.cpp 62](#_Toc46261714)

[18.1.2 각각의 measure 파일 62](#_Toc46261715)

[18.1.3 slot.cpp 63](#_Toc46261716)

[18.1.4 Review 63](#_Toc46261717)

[18.2 Config, Setup, Mode change 64](#_Toc46261718)

[18.3 Measure Update 64](#_Toc46261719)

[18.4 Measurement ID 64](#_Toc46261720)

[18.4.1 Action Item 64](#_Toc46261721)

[18.4.2 Config Item 64](#_Toc46261722)

[18.4.3 Result Item 64](#_Toc46261723)

[18.4.4 Useful Function 64](#_Toc46261724)

[18.5 Technology 65](#_Toc46261725)

[18.6 Function 65](#_Toc46261726)

# Overview

NuTiny-SDK-M051은 NuMicro M051 ™ 시리즈 –M052 / M054 / M058 / M0516 용 개발 툴입니다. 사용자는 NuTiny-SDK-M051을 사용하여 응용 프로그램을 쉽게 개발하고 확인할 수 있습니다.

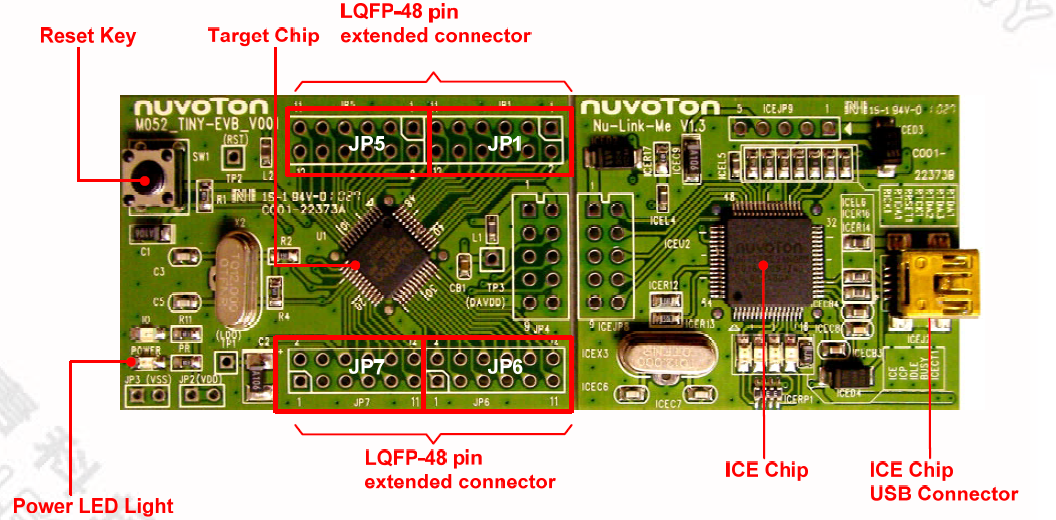
NuTiny-SDK-M051은 2 개의 부분을 포함합니다. 하나는 M052\_Tiny-EVB이고 다른 하나는 Nu-Link-Me입니다. M052\_Tiny-EVB는 평가 보드이며 Nu-Link-Me는 디버그 어댑터입니다. 따라서 사용자는 추가 ICE 장비가 필요하지 않습니다.

# NuTiny-SDK-M051 introduction

NuTiny-SDK-M051은 NuMicro M051 ™ 시리즈를 지원할 수 있습니다. 그림 2-1은 M051 시리즈 용 NuTiny-SDK-M051이며 왼쪽 부분은 M052-TINY-EVB이며 오른쪽 부분은 Nu-Link-Me라는 디버그 어댑터입니다.

M052-TINY-EVB는 다른 개발 보드와 유사합니다. 사용자는 이를 사용하여 실제 동작을 에뮬레이션하는 응용 프로그램을 개발하고 확인할 수 있습니다. 실제로 실제 칩 M0516LAN이 보드에 장착되어 있습니다. 온보드 칩은 M052, M054 및 M058의 기능을 다룹니다. M052\_Tiny-EVB는 사용자 대상 시스템을 설계하기위한 실제 시스템 컨트롤러 일 수 있습니다.

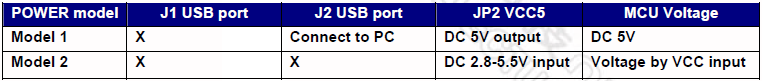
Nu-Link-Me는 디버그 어댑터입니다. Nu-Link-Me 디버그 어댑터는 PC의 USB 포트를 대상 시스템에 직렬 유선 디버그 포트를 통해 연결하고 대상 하드웨어에서 내장 프로그램을 프로그래밍하고 디버깅 할 수 있습니다. Keil 또는 IAR과 함께 Nu-Link-Me 디버그 어댑터를 사용하려면“Nuvoton NuMicro™ IAR ICE driver user manual”또는 “Nuvoton NuMicro™ Keil ICE driver user manual”을 참조하십시오.

  
Figure 2-1 NuTiny-SDK-M051 (Green Color PCB Board)

## NuTiny-SDK-M051 Jumper Description

### Power Setting

* JP2: VCC5 Voltage connecter in M052-TINY-EVB
* J2: USB Port in Nu-Link-Me

  
NOTE: X is Not use.

### Debug Connector

* JP4: Target ICE Connector in M052-TINY-EVB
* JP8: Nuvoton ICE Connector in Nu-Link-Me

### USB Connector

* J1: mini USB Connector in M052-TINY-EVB
* J2: mini USB Connector in Nu-Link-Me

### Extended Connector

* JP1, JP5, JP6 and JP7: Show all of chip pins in M052-TINY-EVB

### Reset Button

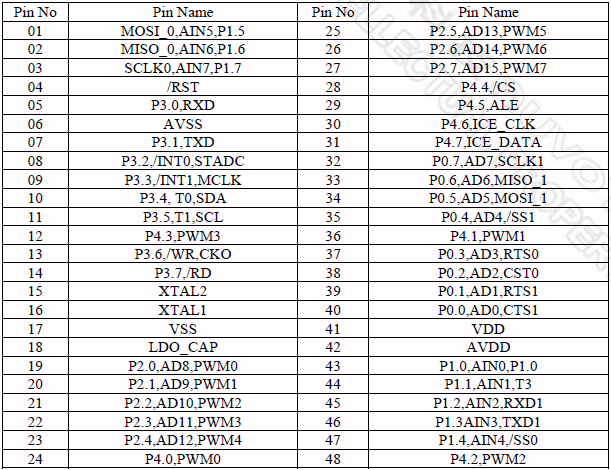
* SW1: Reset button in M052-TINY-EVB

### Power Connector

* JP2: VCC connector in M052-TINY-EVB
* JP3: GND connector in M052-TINY-EVB

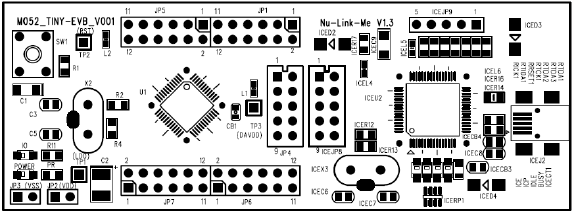
## Pin Assignment for Extended Connector

M052\_TINY-EVB는 M0516LAN 온보드 및 LQFP-48 핀용 확장 커넥터를 제공합니다. 표 2-1은 M0516LAN의 핀 할당입니다.

  
Table 2-1 pin assignment for M0516LAN

## NuTiny-SDK-M051 PCB Placement

NuTiny -SDK-M051 PCB 배치에 대해서는 그림 2-2를 참조 할 수 있습니다.

  
Figure 2-2 NuTiny-SDK-M051 PCB Placement

# Keil uVision® IDE에서 NuTiny -SDK-M051을 시작하는 방법

## Keil uVision® IDE Software Download & Install

Keil uVision® IDE를 다운로드하고 RVMDK를 설치하려면 Keil 회사 웹 사이트 (http://www.keil.com)에 연결하십시오.

## Nuvoton Nu-Link Driver Download & Install

“NuMicro ™ Keil uVision® IDE driver”파일을 다운로드하려면 Nuvoton 회사 NuMicro ™ 웹 사이트 (http://www.nuvoton.com/NuMicro)에 연결하십시오. 자세한 다운로드 과정은 6.1 장을 참조하십시오. 다운로드가 완료되면 파일의 압축을 풀고 "Nu-Link\_Keil\_Driver.exe"를 실행하여 드라이버를 설치하십시오.

## Hardware Setup

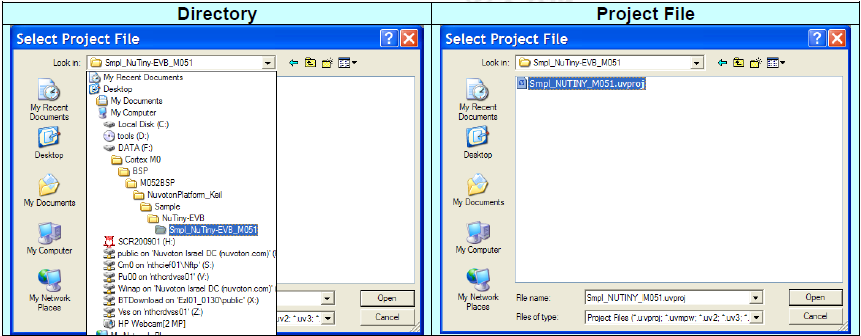
하드웨어 설정은 그림 3-1과 같습니다.

  
Figure 3-1 NuTiny-SDK-M051 Hardware Setup

## Smpl\_NuTiny-M051 Example Program

이 예는 NuTiny-SDK-M051 보드에서 애플리케이션을 쉽게 다운로드하고 디버깅하는 방법을 보여줍니다.

예제는 그림 3-2 목록 디렉토리에서 찾을 수 있습니다.

  
Figure 3-2 Smpl\_NuTiny-M051 Example Directory

이 예제를 사용하려면:

|  |  |
| --- | --- |
| Start uVision | 프로그램을 실행하십시오. |
| Project-Open | Smpl\_NuTiny-M051.uvproj 프로젝트 파일을 여십시오. |
| Project-Build | Smpl\_NuTiny-M051 애플리케이션 컴파일 및 링크 |
| Flash-Download | 온칩 플래시 ROM에 응용 프로그램 프로그래밍 |
| LED는 M052\_TINY-EVB 보드에서 토글됩니다. | |

|  |  |
| --- | --- |
| Start debug mode | 디버거 명령을 사용하여 다음을 수행 할 수 있습니다. |
|  | Single step through code |
|  | Run the application\ |
|  | Review variables in the watch window |
|  | Reset the device to re-run the application |

# IAR Embedded Workbench에서 NuTiny-SDK-M051을 시작하는 방법

## IAR Embedded Workbench Software Download &Install

IAR Embedded Workbench를 다운로드하고 EWARM을 설치하려면 IAR 회사 웹 사이트 (http : // www.iar.com)에 연결하십시오.

## Nuvoton Nu-Link Driver Download & Install

“NuMicro ™ IAR ICE driver user manual”파일을 다운로드하려면 Nuvoton 회사 NuMicro ™ 웹 사이트 (www.nuvoton.com/NuMicro)에 연결하십시오. 자세한 다운로드 흐름은 6.2를 참조하십시오. 다운로드가 완료되면 파일의 압축을 풀고 "Nu-Link\_IAR\_Driver.exe"를 실행하여 드라이버를 설치하십시오.

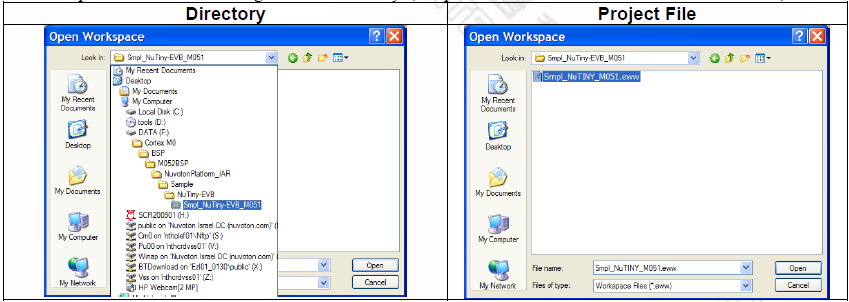
## Hardware Setup

하드웨어 설정은 그림 3-1과 같습니다.

## Smpl\_NuTiny-M051 Example Program

이 예는 NuTiny-SDK-M051 보드에서 애플리케이션을 쉽게 다운로드하고 디버깅하는 방법을 보여줍니다.

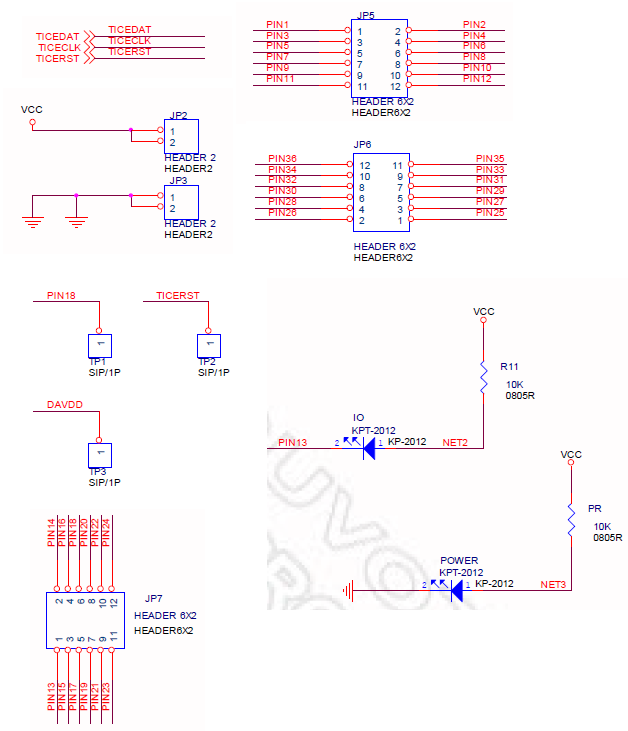
예제는 그림 4-2 목록 디렉토리에서 찾을 수 있습니다. (샘플 코드는 Nuvoton 웹 사이트에서 다운로드 할 수 있습니다)

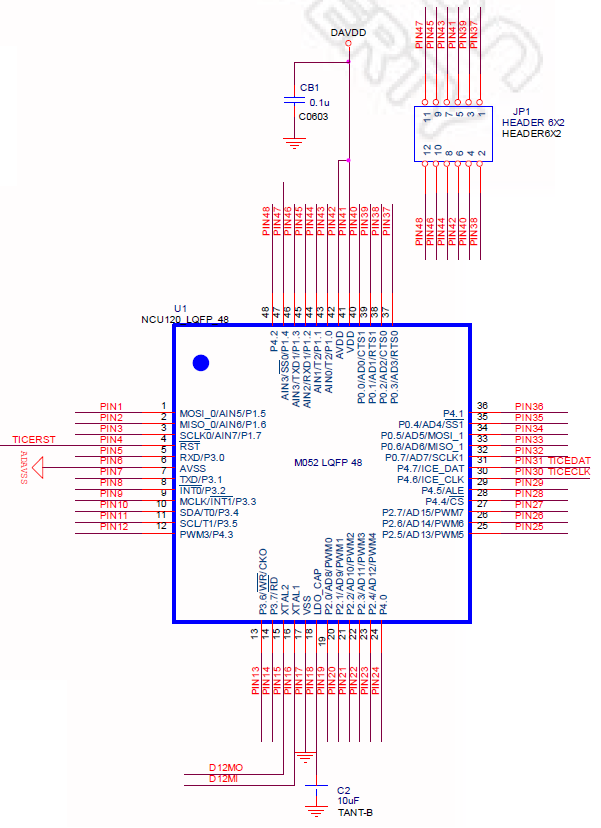
  
Figure 4-2 Smpl\_NuTiny-M051 Example Directory

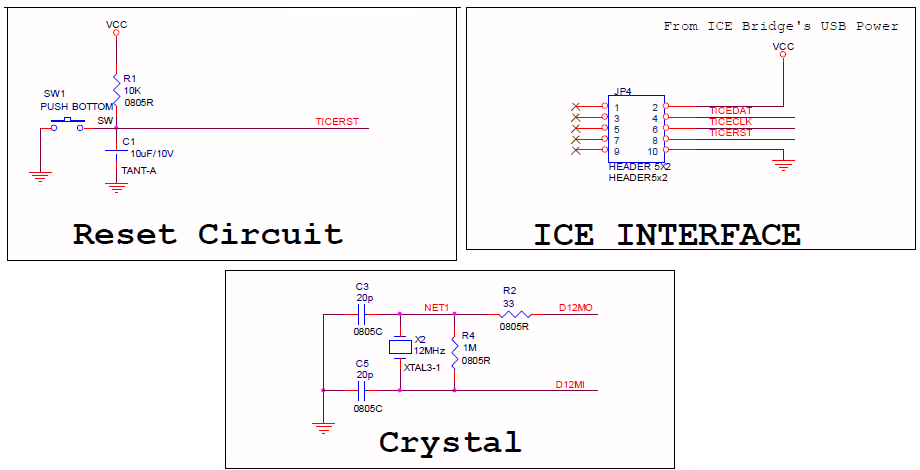
이 예제를 사용하려면:

|  |  |
| --- | --- |
|  | Start IAR Embeded Workbench |
| File-Open-Workspace | Open the Smpl\_NuTiny-M051.eww workspace file |
| Project - Make | Compile and link the Smpl\_NuTiny-M051 application |
| Project-Download and Debug | Program the application into on-chip Flash ROM |
| I / O가 M052\_TINY-EVB 보드에서 토글됩니다. | |

# M052\_TINY-EVB Schematic

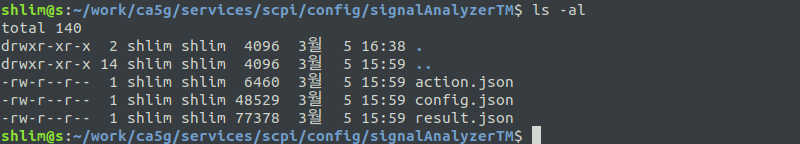






## add file

해당 위치(ca5g/services/scpi/config/.)에 파일을 만들어 추가한다. CMake Run을 하면 Qt에서도 파일을 볼 수 있다.





## SCPI 명령 작성

명령 작성 후 reboot을 해야 적용된다. SCPI 명령 문법에 ‘숫자’나 ‘-‘등의 특수문자는 사용하지 않는다. 단, 부득이하게 순서를 정하기 위해 숫자를 사용하려면 두 자리로 사용한다. 예를 들어, 1을 표현하고자 할때는 01로 표현한다.

## telnet을 이용한 검증

port는 5600

## scpi debugging

SSH에서 scpi 명령을 치면 상태를 볼 수 있다.

|  |
| --- |
| root@tb5800-a0291d:/# scpi  Connnecting remote control service : "com.viavisolutions.platform.RemoteControl"  Connnected remote control service : "com.viavisolutions.platform.RemoteControl"  MMAP Creating gpio DevBlock  MMAP Creating gpioMux DevBlock  MMAP Creating fpgaPci DevBlock  Started Scpi Register execute  "/base/config/scpi/config/realtimeAnalyzer/result.json"  "/base/config/scpi/config/realtimeAnalyzer/action.json"  "/base/config/scpi/config/realtimeAnalyzer/config.json"  SCPI Error2(command has numeric index(#), but elfId does not) : :config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:modeMultiSegmentLineCnfg:  SCPI Error2(command has numeric index(#), but elfId does not) : :config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:numberOfLineCnfg:  SCPI Error1(Duplicated command) : REALtime:MARKer#:TYPE  "/base/config/scpi/config/signalAnalyzer5GNR/result.json"  SCPI Error2(Duplicated command) : NR5G:BAI:PCI?  "/base/config/scpi/config/signalAnalyzer5GNR/action.json" |

개별 scpi 명령의 동작 여부를 알기 위해 다음과 같이 할 수 있다.

|  |
| --- |
| root@tb5800-a0291d:~# scpiClient NRTM:TAEMimo:ANTenna1:TIMoffset? |

# CMakeLists

qml등의 sub module을 원하는 브랜치에서 가져와 사용(컴파일)할 수 있기를 원할 때 아래와 같이 막는다.

|  |
| --- |
| execute\_process(COMMAND git submodule init  WORKING\_DIRECTORY ${CMAKE\_SOURCE\_DIR})  #execute\_process(COMMAND git submodule update --recursive --remote  # WORKING\_DIRECTORY ${CMAKE\_SOURCE\_DIR}) |

그리고 나서 브랜치를 바꾼 후 컴파일하면 된다.

# Option활성화 비활성화

SSH 접속 후 직접 입력. 타 부서 및 문서 노출 절대 안됨. 명령 입력 후 장비 재부팅해야 system info 항목까지 적용됨.

## Activation

enable\_options.sh F001O

## Deactivation

disable\_options.sh F001O

# Bitbucket

## pull request

local에서 push하면 메시지에 remote 주소가 나온다. 링크를 열면 bitbucket에 연결되어 create pull request 화면이 나타난다.

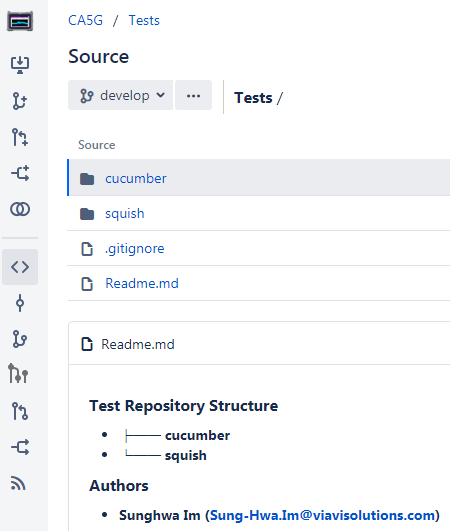
|  |
| --- |
| shlim@s:~/work/ca5g/apps$ **git push**  Counting objects: 27, done.  Delta compression using up to 4 threads.  Compressing objects: 100% (27/27), done.  Writing objects: 100% (27/27), 4.72 KiB | 2.36 MiB/s, done.  Total 27 (delta 21), reused 0 (delta 0)  remote:  remote: Create pull request for feature/nrtm-analyzer:  remote: https://cosgit1.ds.jdsu.net/projects/CA5G/repos/ca5g\_apps/pull-requests?create&sourceBranch=refs/heads/feature/nrtm-analyzer  remote:  To ssh://cosgit1.ds.jdsu.net:7999/ca5g/ca5g\_apps.git  5f8637ff..1fa0d776 feature/nrtm-analyzer -> feature/nrtm-analyzer |

# SW Integration Test

## Cucumber

### repository

Tests repository에서 clone한다.



shlim@s:~/works/tests/cucumber/apps$ 의 위치에 해당 feature file을 작성한다. 기존에 작성되어 있는 디렉토리를 복사해서 이름만 바꿔주면 된다.

|  |
| --- |
| shlim@s:~/works/tests/cucumber/apps/11\_signalAnalyzerTM/features$ ll  total 72  drwxr-xr-x 4 shlim shlim 4096 4월 17 14:23 ./  drwxr-xr-x 3 shlim shlim 4096 4월 17 14:23 ../  -rw-r--r-- 1 shlim shlim 55886 4월 17 14:23 11\_signalAnalyzerTM.feature  drwxr-xr-x 2 shlim shlim 4096 4월 17 14:23 step\_definitions/  drwxr-xr-x 2 shlim shlim 4096 4월 17 14:23 support/ |

### 환경설정

cucumber 디렉토리에 있는 Readme.md 파일을 읽어보면 환경을 설정할 수 있다. app 뿐만 아니라 scpi도 테스트할 수 있는 방법이 있다.

* package install  
  sudo apt-get install cucumber ruby-sshkit ruby-net-ssh -y
* target device ip configure  
  vi apps/script/env.rb 파일에 target ip입력

### 실행

#### TM feature만 실행시키는 방법

* tests/cucumber/apps/11\_signalAnalyzerTM$ 디렉토리로 이동
* 실행 명령 입력  
  cucumber ./features/11\_signalAnalyzerTM.feature

#### 전체 feature를 실행시키는 방법

shlim@s:~/workrelease/tests/tests/cucumber/apps$ run.sh

#### tags 옵션

help를 쳐 보면 사용법을 알 수 있다. cucumber file에 ‘@’ tag가 붙어 있는 부분을 어떻게 처리할 지 결정할 수 있다. 예를 들어 --tags @6G이면 6G 포함, --tags !@6G이면 6G 미포함.

shlim@s:~/workrelease/tests/tests/cucumber/apps$ cucumber -help

### 결과

아래와 같은 명령으로 HTML 파일을 만들어낸다.

cucumber -f html -o tm\_integration\_test\_20200330.html ./features/11\_signalAnalyzerTM.feature

### 등록

bamboo에서 돌아가도록 하려면 all 디렉토리에 심볼릭 링크를 걸어준다.

ln -s <원본파일> <링크파일>

shlim@s:~/workrelease/tests/tests/cucumber/apps/all/features$ ln -s ../../11\_signalAnalyzerTM/features/11\_signalAnalyzerTM.feature 11\_signalAnalyzerTM.feature

삭제하려면 심볼릭 링크를 삭제해 준다.

rm -rf <링크파일>

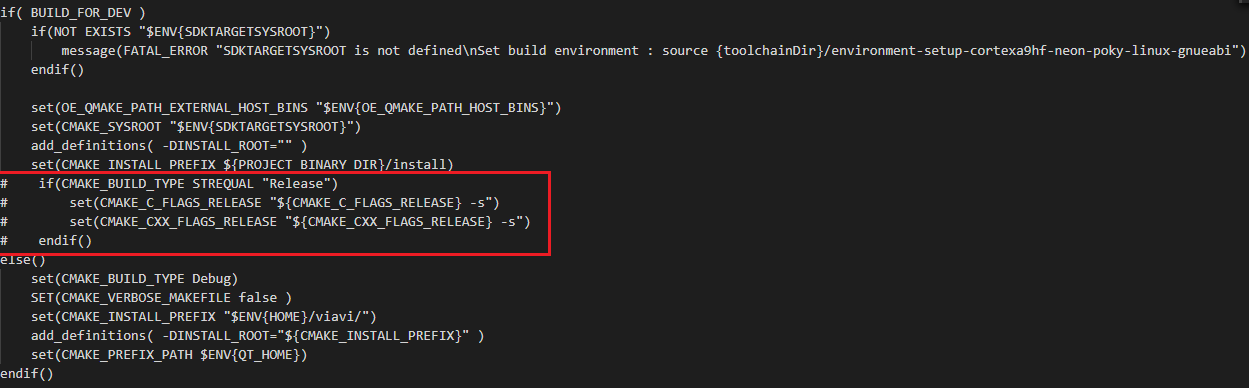
shlim@s:~/workrelease/tests/tests/cucumber/apps/all/features$ rm -rf 11\_signalAnalyzerTM.feature

### Debug

cucumber를 실행시켰는데 crash가 나서 죽는 경우에 사용할 수 있다.

1. gdb 사용을 위해 Qt에서 CMakeLists.txt 파일을 수정한다.

|  |
| --- |
| if(CMAKE\_BUILD\_TYPE STREQUAL "Release")  set(CMAKE\_C\_FLAGS\_RELEASE "${CMAKE\_C\_FLAGS\_RELEASE} -s")  set(CMAKE\_CXX\_FLAGS\_RELEASE "${CMAKE\_CXX\_FLAGS\_RELEASE} -s")  endif() |



2. build해서 실행시킨다.

3. ssh로 접속하여 장비의 해당 feature를 kill한다.

-. gview로 servie name을 확인한다.

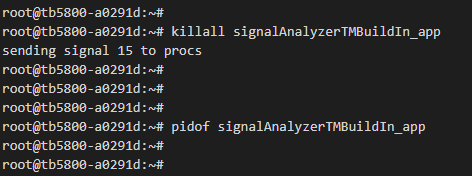


-. ps -ef 혹은 pidof 명령으로 process를 확인한다.

|  |
| --- |
| root@tb5800-a0291d:~# **pidof signalAnalyzerTMBuildIn\_app**  2504 |

-. process가 있으면 kill한다.

|  |
| --- |
| root@tb5800-a0291d:~# **killall signalAnalyzerTMBuildIn\_app**  sending signal 15 to procs  root@tb5800-a0291d:~# **pidof signalAnalyzerTMBuildIn\_app**  root@tb5800-a0291d:~# |



4. gdb를 실행한다.

|  |
| --- |
| root@tb5800-a0291d:~# **gdb signalAnalyzerTMBuildIn\_app**  GNU gdb (GDB) 8.0.1  Copyright (C) 2017 Free Software Foundation, Inc.  License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>  This is free software: you are free to change and redistribute it.  There is NO WARRANTY, to the extent permitted by law. Type "show copying"  and "show warranty" for details.  This GDB was configured as "arm-poky-linux-gnueabi".  Type "show configuration" for configuration details.  For bug reporting instructions, please see:  <http://www.gnu.org/software/gdb/bugs/>.  Find the GDB manual and other documentation resources online at:  <http://www.gnu.org/software/gdb/documentation/>.  For help, type "help".  Type "apropos word" to search for commands related to "word"...  Reading symbols from signalAnalyzerTMBuildIn\_app...(no debugging symbols found)...done.  (gdb) |

|  |
| --- |
| (gdb) **r**  Starting program: /base/bin/signalAnalyzerTMBuildIn\_app  warning: File "/lib/libthread\_db-1.0.so" auto-loading has been declined by your `auto-load safe-path' set to "$debugdir:$datadir/auto-load".  To enable execution of this file add  add-auto-load-safe-path /lib/libthread\_db-1.0.so  line to your configuration file "/home/root/.gdbinit".  To completely disable this security protection add  set auto-load safe-path /  line to your configuration file "/home/root/.gdbinit".  For more information about this security protection see the  "Auto-loading safe path" section in the GDB manual. E.g., run from the shell:  info "(gdb)Auto-loading safe path"  warning: Unable to find libthread\_db matching inferior's thread library, thread debugging will not be available.  [New LWP 14093]  [New LWP 14094]  [ChStd] Version : 1.0.1  이하 생략 |

5. cucumber를 실행한다.

|  |
| --- |
| shlim@s:~/workrelease/tests/tests/cucumber/apps$ ./run.sh |

6. 죽는 경우 gdb 명령(bt)을 사용하여 원인을 분석한다.

7. Qt에서 run 설정을 변경하여 debugging하는 방법도 있음. Working directory부분 추가(문의할 것)

# Git

## 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

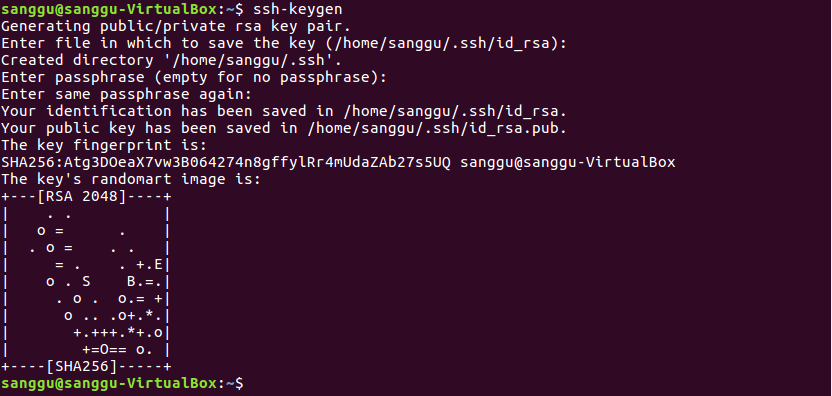
git submodule deinit -f --all

### status

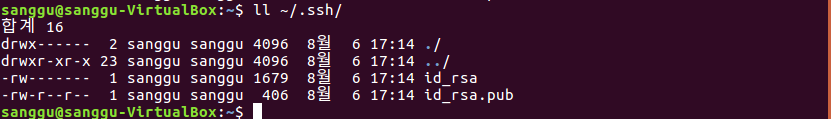
git status

## SSH Key Gen

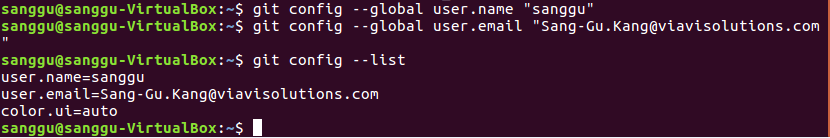
-. Key gen



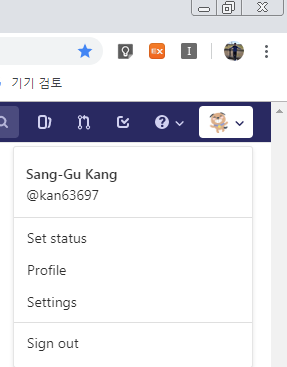
-. Confirm

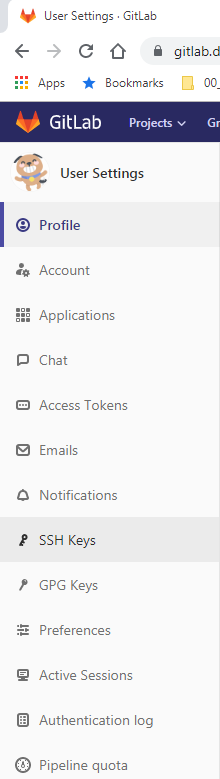


-. Global setup



-. Git Host Add SSH Key





Copy from ~/.ssh/id\_rsa.pub

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDsq/SDJbENre+JeaJK75LxgWWIVd/S6ZTlBziJnfqRKGZxJnIGJqIEI4mIzAmPhYiukaUH/TcZ8lUhwGiE2FpD1mBSkIauMF7/oK0EI67Iun8wyHLgr73/g64SGsnGBIUcqYU/b5ALH2OKm1S6EJpKMFPXSZ2caGq9G7ScoI/08wSNnJg+McdcnYy388vY5EZ1p85L44bzbStZ8DOlGEEAcv5wZ5reO44UfWxHHmlGFTTwbsNHwPLQRR+hJYFGIfQQyBrRQph8FYB96Qoen2mG4fRUmU3n1yk7X4FzFCJE60cuZV9ivD1xe2sVFWUqYJT47RpiB71ZvANbDismVhMt sanggu@sanggu-VirtualBox

Fingerprint: 7a:4b:63:3c:42:f7:e7:ac:8b:4d:fe:92:e0:8d:7d:e0

# Debug

## gview

SSH에서 해당 명령을 입력하여 코드에서의 printf 메시지를 본다.

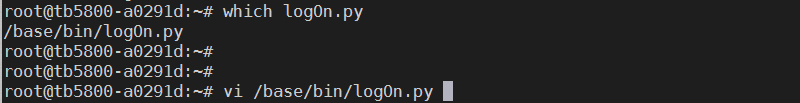
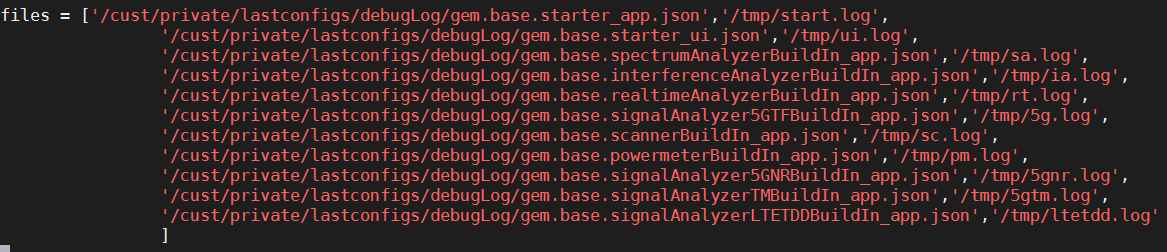
|  |
| --- |
| gview -f $(tty) -o gem.base.signalAnalyzerTMBuildIn\_app |

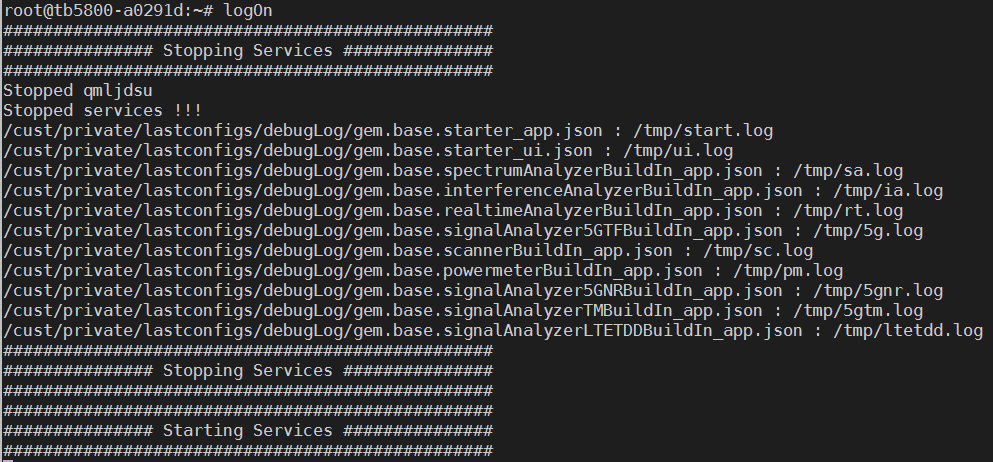
## ps

SSH에서 해당 명령을 입력하여 process가 제대로 도는지 확인한다.

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

## qDebug()

logOn.py 파일 수정(추가)  
  


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


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



# FPGA loading

## develop

* DSP bin 파일을 sftp로 ca5g에 밀어 넣는다. Drag and drop to device.  
    
  -. Using MobaXterm  
  
* protoFlash <bank number> <bin file name>   
  This is for TM(Version: 9333, Bank 8, 5GNR+nrTM+LTE)  
  protoFlash 7 ./9333\_HetNet\_top\_v4.bin  
    
  
* protoFpgaConfig <bank number>  
  protoFpgaConfig 7  
  
* check address and compare with bin file  
    
  0x2457 convert to decimal using calculator  
  0x2457 == 9333(version is same. Ok)

## Using lfs

* lfs가 설치되어 있지 않다면 설치  
  

DSP team에서 제공한 bin 파일의 이름을 변경(DPB~~~.bin)한 후, add-> commit한다.

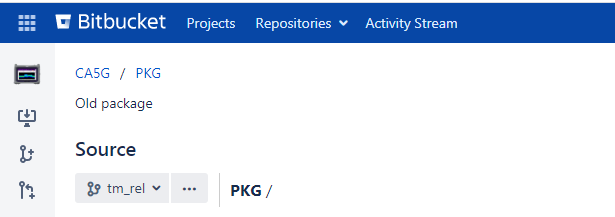
git lfs install한 후 push하고 bitbutket에서 잘 올라갔는지 확인한다. 이때, version파일과 name 파일이 맞는지 확인하고 수정이 필요하다면 수정한 후 add, commit, push한다.

* pull  
  
* bin 파일 commit  
    
    
  
* lfs install  
  
* git push  
  
* check  
  

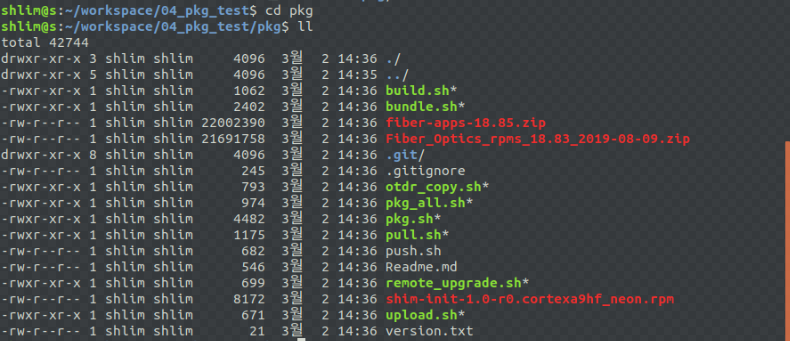
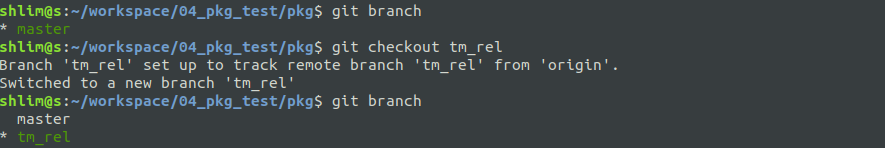
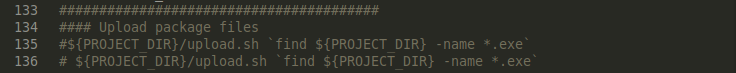
## System Information

# Package

## package for test

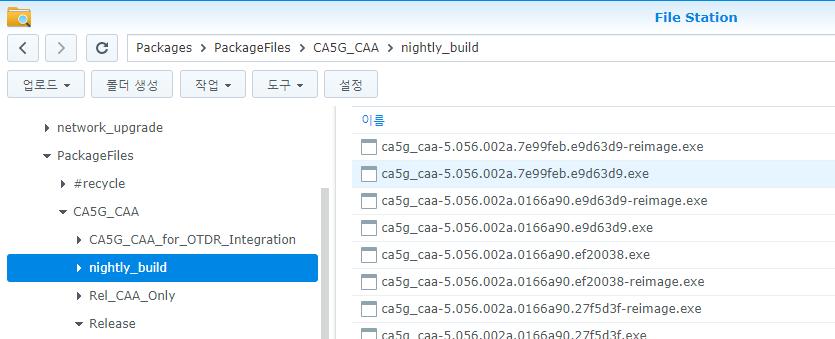


NOTE: 원격 저장소에 테스트용 fw를 위한 브랜치가 PKG에 만들어져 있다고 가정한다. 이 브랜치는 TM analyzer만을 위한 script 등을 포함하고 있고 필요에 따라 수정해서 쓰도록 하기 위해 만들어졌다.

* clone PKG  
  shlim@s:~/work$ git clone ssh://git@cosgit1.ds.jdsu.net:7999/ca5g/pkg.git PKG
* pkg directory  
  
* checkout TM release branch  
  
* version.txt 수정  
  
* pkg.sh 수정  
  REL\_BRANCH(ca5g and caa)  
    
    
  cmake -DBUILD\_PLATFORM  
    
    
  commit, push 주석처리  
    
    
  exe package upload 주석처리  
  
* pkg.sh 스크립트 실행  
  shlim@s:~/workspace/04\_pkg\_test/pkg$ ./pkg.sh  
  …  
  Files read from disk: 1  
  Archive size: 615085639 bytes (587 MiB)  
  Everything is Ok
* exe 파일 확인  
  성공하면 upgrade-exe 디렉토리가 생성되고 그 안에 exe 파일이 만들어진다. 이 파일을 윈도우에서 실행시키면 다운로드 파일이 만들어진다.
* push(나중에 버전이 어디까지 진행되었는지 확인 가능)
* confluence에서 Team>Patch/HotFixRelease에 해당 내용 업데이트

## Nightly\_build

매일 build해서 develop의 image를 저장해 놓는다.



## Making package

-. git lfs pull (???)



-. Check fpga file (???)



-. git clone pkg

mkdir pkg

git clone ssh://git@cosgit1.ds.jdsu.net:7999/ca5g/pkg.git pkg

cd pkg

git checkout tm\_rel

git pull

-. Version.txt 수정



-. Script 실행

./pkg.sh

-. Check release file

Upgrade-exe/\*.exe

-. window에서 실행시키면 파일이 만들어짐



NOTE: pkg.sh 수정

cmake .. -DWITH\_CAA=true -DCMAKE\_C\_FLAGS="-s" -DBUILD\_PLATFORM=ca5g

REL\_BRANCH 수정: REL\_BRANCH=feature/HA-3653-tm-analyzer

caa해당 브랜치 정의 : REL\_BRANCH=release/rel\_055

commit & push 지우기

exe 패키지 업로드 : ${PROJECT\_DIR}/upload.sh

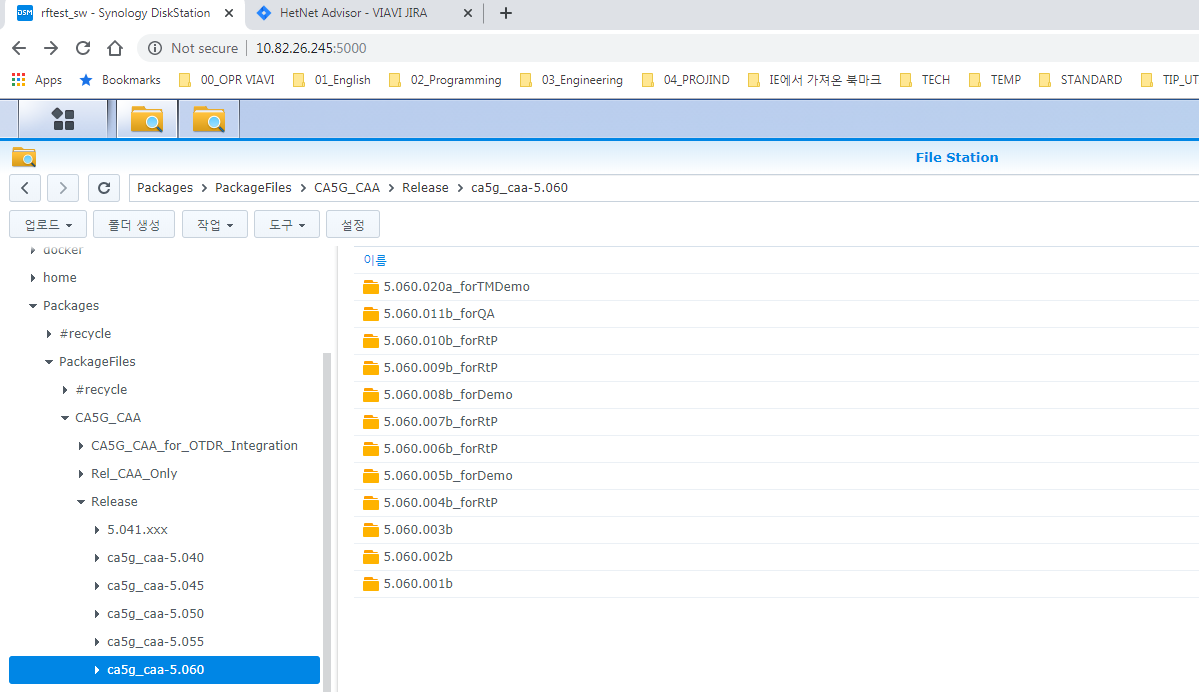






## Sharing package

-. 서버에 올려서 공유한다.



## Reimage

re-image 실행 파일(ca5g\_caa-5.065.010a.ba4edb8.e24cc2e-reimage.exe)을 실행시켜서 USB로 옮긴다.

장비에 USB를 꽂고 reboot한다.

블랙 화면에 “Start Installation”이 나타난다.

완료되면 “Please unplug the USB flash drive to continue”가 나타나는데 이때, USB를 제거하고 기다린다. 수초 후 장비가 재시작하면서 완료된다.

# 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

dbus를 모니터한다.

**root@tb5800-a0291d:~# dbus-monitor**

signal time=1581618907.582107 sender=org.freedesktop.DBus -> destination=:1.693 serial=2 path=/org/freedesktop/DBus; interface=org.freedesktop.DBus; member=NameAcquired  
 string ":1.693"  
signal time=1581618907.582782 sender=org.freedesktop.DBus -> destination=:1.693 serial=4 path=/org/freedesktop/DBus; interface=org.freedesktop.DBus; member=NameLost  
 string ":1.693"  
signal time=1581618907.608141 sender=:1.666 -> destination=(null destination) serial=6770 path=/; interface=jdsu.elf; member=SigResultChangedNotification  
 array [  
 struct {  
 string ":result:spectrumAnalyzerBuildIn\_app:spectrumTuned\_module:traceDataYRslt:doubleARRAY:"  
 variant array [  
 double -50.4281  
 double -49.7094  
 double -48.6156  
 double -48.3969  
 double -48.6976

중간 생략

double 0  
 double 0  
 ]  
 int32 1048579  
 }  
 ]

중간 생략

signal time=1581618909.696288 sender=:1.666 -> destination=(null destination) serial=6791 path=/; interface=jdsu.elf; member=SigResultChangedNotification  
 array [  
 struct {  
 string ":result:spectrumAnalyzerBuildIn\_app:spectrumTuned\_module:traceDataYRslt:doubleARRAY:"  
 variant array [  
 double -60.0726  
 double -54.2523  
 double -51.5258

중간생략

double 0  
 double 0  
^C

**root@tb5800-a0291d:~# dbus-monitor | grep result**string ":result:spectrumAnalyzerBuildIn\_app:spectrumTuned\_module:traceDataYRslt:doubleARRAY:"  
string ":result:spectrumAnalyzerBuildIn\_app:spectrumTuned\_module:traceDataYRslt:doubleARRAY:"  
string ":result:spectrumAnalyzerBuildIn\_app:spectrumTuned\_module:traceDataYRslt:doubleARRAY:"  
string ":result:spectrumAnalyzerBuildIn\_app:spectrumTuned\_module:traceDataYRslt:doubleARRAY:"  
string ":result:spectrumAnalyzerBuildIn\_app:spectrumTuned\_module:traceDataYRslt:doubleARRAY:"  
string ":result:spectrumAnalyzerBuildIn\_app:spectrumTuned\_module:traceDataYRslt:doubleARRAY:"  
string ":result:spectrumAnalyzerBuildIn\_app:spectrumTuned\_module:traceDataYRslt:doubleARRAY:"  
^C

**root@tb5800-a0291d:~# dbus-monitor | grep config**  
string ":config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:stepFrequencyCnfg:"  
string ":config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:displayTypeFrequencyCnfg:"  
string ":config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:centerFrequencyCnfg:"  
string ":config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:centerFrequencyCnfg:"  
string ":config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:stopFrequencyCnfg:"  
string ":config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:startFrequencyCnfg:"  
string ":config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:channelNumberCnfg:"  
string ":config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:resolutionBandwidthCnfg:"  
string ":config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:timeSweepCnfg:"  
string ":config:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:currentMinTimeSweepCnfg:"  
^C

NOTE: config, action의 경우 명령을 먼저 실행한 후 device에 원하는 동작을 시킨다.

**root@tb5800-a0291d:~# dbus-monitor | grep action**  
string ":action:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:stop:"  
string ":action:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:stop:"  
string ":action:spectrumAnalyzerBuildIn\_app:spectrumAnalyzerBuildIn\_reg\_module:stop:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:start:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:start:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:traceClearAllActn:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:start:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:traceClearAllActn:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:traceClearAllActn:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:stop:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:stop:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:stop:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:start:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:start:"  
string ":action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:start:"  
^C

**root@tb5800-a0291d:~# dbus-monitor | grep config -a**

**root@tb5800-a0291d:~# dbus-monitor | grep config -a1**

**root@tb5800-a0291d:~# dbus-monitor | grep config -a2**

**root@tb5800-a0291d:~# dbus-monitor | grep config -a5**

**root@tb5800-a0291d:~# dbus-monitor | grep config -b**

NOTE: 정보의 양을 조절하는 옵션. a means after, b means before

## gview

**root@tb5800-a0291d:~# gview**

Select application by service name:

1. gem.base.basemgr  
2. gem.base.RFoCPRIBuildIn\_app  
3. gem.base.caaHwService  
4. gem.base.cableAntennaAnalyzer\_app  
5. gem.base.cableAntennaAnalyzer\_ui  
6. gem.base.interferenceAnalyzerBuildIn\_app  
7. gem.base.powermeterBuildIn\_app  
8. gem.base.realtimeAnalyzerBuildIn\_app  
9. gem.base.scannerBuildIn\_app  
10. gem.base.signalAnalyzer5GNRBuildIn\_app  
11. gem.base.signalAnalyzer5GTFBuildIn\_app  
12. gem.base.signalAnalyzerLTEFDDBuildIn\_app  
13. gem.base.signalAnalyzerLTETDDBuildIn\_app  
14. gem.base.signalAnalyzerTMBuildIn\_app  
15. gem.base.spectrumAnalyzerBuildIn\_app  
16. gem.base.starter\_app  
17. gem.base.starter\_ui

Enter application choice (1-17, enter 'q' to exit): **14**

Select module:

1. :result:signalAnalyzerTMBuildIn\_app:adjacentChannelPower\_module:manifest:  
2. :result:signalAnalyzerTMBuildIn\_app:bsOutputPower\_module:manifest:  
3. :result:signalAnalyzerTMBuildIn\_app:constellation\_module:manifest:  
4. :result:signalAnalyzerTMBuildIn\_app:multiAdjacentChannelPower\_module:manifest:  
5. :result:signalAnalyzerTMBuildIn\_app:occupiedBW\_module:manifest:  
6. :result:signalAnalyzerTMBuildIn\_app:operatingBandUnwantedEmissions\_module:manifest:  
7. :result:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:manifest:  
8. :result:signalAnalyzerTMBuildIn\_app:timeAlignmentError\_module:manifest:  
9. :result:signalAnalyzerTMBuildIn\_app:transmitOnOffPower\_module:manifest:  
10. :result:signalAnalyzerTMBuildIn\_app:transmitterSpuriousEmissions\_module:manifest:

Enter module choice (1-10, enter 'q' to exit): **7**

Select action for  
service=gem.base.signalAnalyzerTMBuildIn\_app  
module=:result:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:

1. streamstdout  
2. streamstderr  
3. debug log settings  
4. streamlog  
5. dumpconfigs  
6. dumpresults  
7. dumpmanifest  
8. action  
9. setconfig  
10. setresult  
11. getconfig  
12. getresult  
13. showenv  
14. showpid  
15. showcmd  
16. showcwd  
17. raisehome  
18. dumpprocs  
19. Choose another application / module context

Enter action choice (1-19, enter 'q' to exit): **8**

Select ELF action ID:

1. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:applyconfig:  
2. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:autoScaleActn:  
3. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:captureActn:  
4. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:clear\_uservisible\_log:  
5. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:dumpactions:  
6. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:dumpconfigs:  
7. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:dumplog:  
8. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:dumpresults:  
9. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:enableDebugActn:  
10. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:get\_uservisible\_log:  
11. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:halt:  
12. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:loadResultActn:  
13. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:loadconfig:  
14. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:logStartActn:  
15. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:logStopActn:  
16. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:markerAllOffActn:  
17. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:minSearchActn:  
18. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:moveToCenterActn:  
19. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:moveToStartActn:  
20. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:moveToStopActn:  
21. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:nextPeakActn:  
22. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:nextPeakLeftActn:  
23. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:nextPeakRightActn:  
24. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:peakSearchActn:  
25. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:presetActn:  
26. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:presetActnWOMeas:  
27. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:removeconfig:  
28. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:resetMeasureActn:  
29. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:saveConfResultActn:  
30. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:saveIQDataActn:  
31. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:saveIQDataScpiActn:  
32. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:saveResultCsvActn:  
33. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:saveconfig:  
34. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:setCalculateTrace05:  
35. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:setCalculateTrace06:  
36. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:start:  
37. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:startSA:  
38. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:stop:  
39. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:stopSA:  
40. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:streamlog:  
41. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:sweepOnceActn:  
42. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:traceClearAllActn:  
43. :action:signalAnalyzerTMBuildIn\_app:signalAnalyzerTMBuildIn\_reg\_module:updateParameterActn:  
44. Go back to previous menu

Enter ELF action ID choice (1-44, enter 'q' to exit): q

## Byond Compare

ca5g$ **bcompare** apps/signalAnalyzer5GNR/ apps\_old/signalAnalyzer5GNR/

# Test

## Measurement Switching Time

Mode나 Measure 이동에 따른 시간 측정 방법.

confluenc에 기록.  
TEST>Test>Performance>CA5G Mode/Measurement Switching Time

시간 측정  
ssh에서   
logOn 실행  
tail -f /tmp/ui.log | grep Splash 실행

# Project

DSP library는 ca5g\_libs에 위치한다.

JIRA issue resolve하면 QA 팀으로 assign할 것.

# Source Analysis

## 파일 별 기능 구조

### baseservermodule.cpp

* **makeNRTMParameter**(int nMeasureMode,EngTMInput \*inputNRTMData,EngTMOutput \*outputNRTMData)
* **makeNRTMParameterCa**(int nMeasureMode,EngTMInput \*inputNRTMData,EngTMOutput \*outputNRTMData,int nIndex)  
  input parameter를 DSP에 전달.
* **updateParameter**(int nMeasureMode,EngTMInput \*inputNRTMData,EngTMOutput \*outputNRTMData)  
  어플리케이션 자체에서 처리해야 할 일들을 처리???

### 각각의 measure 파일

* updateMeasure(void)  
  makeNRTMParameter(eMEAS\_TAE\_MIMO, inputNRTMData, outputNRTMData); 호출  
  p\_5GTMModule->measureProcess(eMEAS\_TAE\_MIMO, inputNRTMData, outputNRTMData); 호출  
  if(measureProcess가ok) -> updateMeasuredData(outputNRTMData);, updateParameter(eMEAS\_TAE\_MIMO,inputNRTMData,outputNRTMData); 호출  
  checkLimit();호출
* **updateMeasuredData**(EngTMOutput \*outputNRTMData)  
  measureProcess를 실행한 후 그 결과를 GUI에 전달하기 위해 result item 을 set한다.  
  내부적으로 처리해야 하는 기능을 이곳에 넣는다(예, time offset 차이 계산. peak 계산).
* **checkLimit**()  
  pass/fail을 판단하기 위한 기능.

### slot.cpp

* slotUpdateMeasure()  
  일정 시간마다 호출되며 주로 eACTCMD\_UPDATE가 호출되도록 한다. 그 외 auto scale이나 spurious range를 체크한다.

### Review

int signalAnalyzerTM::callActSubmodule(int nMeasMode, int nCmd, int nArg, std::string params)

## Config, Setup, Mode change



## Measure Update



## Measurement ID

### Action Item

ADD\_ACT()

### Config Item

### Result Item

### Useful Function

#### bool signalAnalyzerTM::initItemProperty()

단위 표시(keypad, 등등)

## Technology

Slot number와 sub frame number는 PvsT(Symbol)에 쓰는 입력값이지만 TM에는 필요 없다.

## Function

|  |  |  |  |
| --- | --- | --- | --- |
|  | function | description | remark |
| A |  |  |  |
| B |  |  |  |
| C | checkUpdateSetup | 업데이트 사항을 체크하고 반영할 수 있다. |  |
| D |  |  |  |
| E |  |  |  |
| F |  |  |  |
| G |  |  |  |
| H |  |  |  |
| I |  |  |  |
| J |  |  |  |
| K |  |  |  |
| L |  |  |  |
| M |  |  |  |
| N |  |  |  |
| O |  |  |  |
| P |  |  |  |
| Q |  |  |  |
| R |  |  |  |
| S | sendMsgWarning  syncmeasurementMode  setPropertyMHz | Warning popup  …  주파수 입력창 단위 표시 |  |
| T |  |  |  |
| U |  |  |  |
| V |  |  |  |
| W |  |  |  |
| X |  |  |  |
| Y |  |  |  |
| Z |  |  |  |

|  |  |
| --- | --- |
| double dADCPower = getInsTraceComm(currentMeasureMode())->dADCPower; | signalAnalyzerTMSlot.cpp |
| dADCPower = m\_rfParam->getADCPowerMonitoring(); | signalAnalyzerTMSlot.cpp |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |