

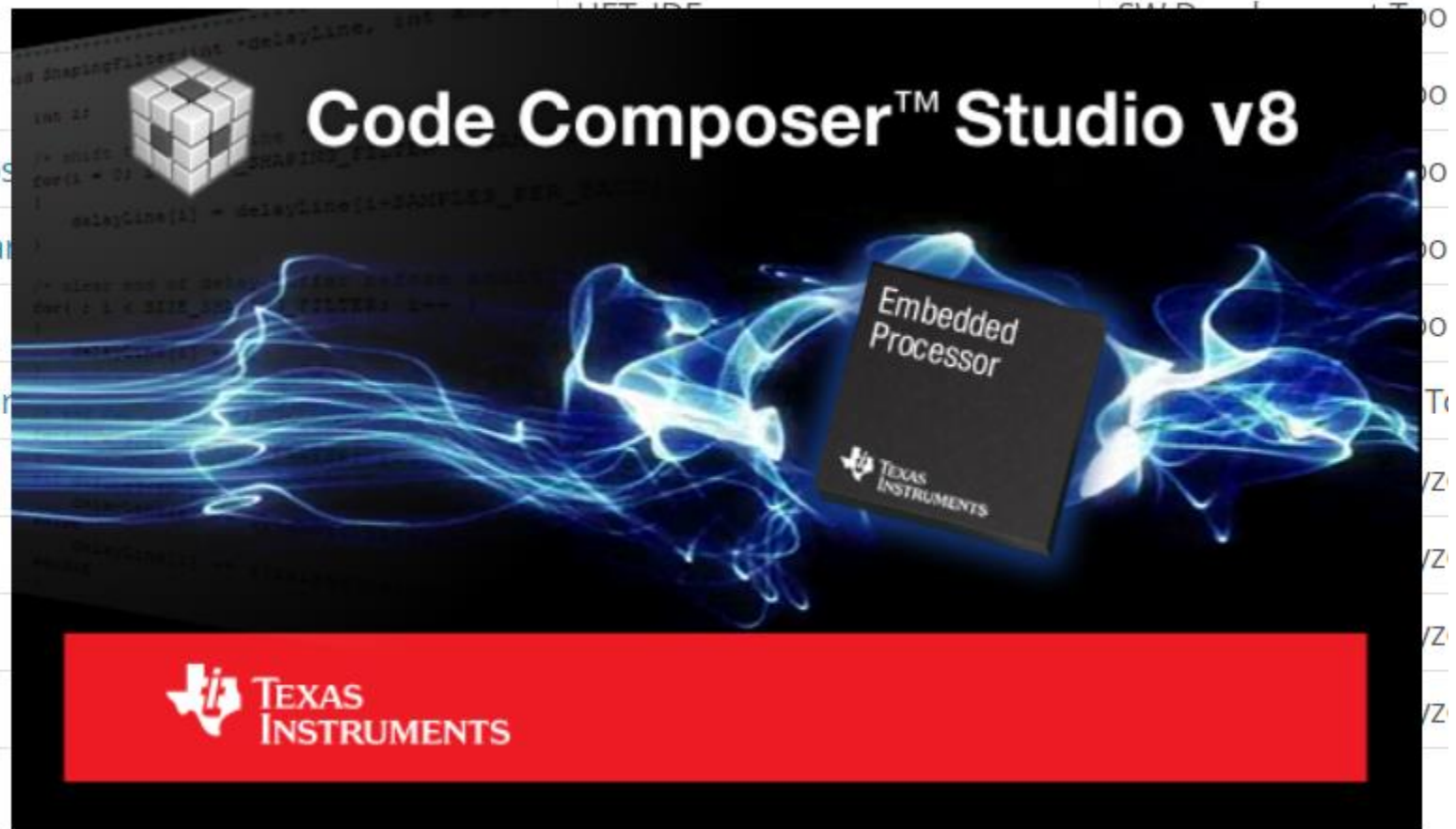
# TI MCU, DSP 및 Xilinx FPGA 프로그래밍 전문가 과정

Innova Lee(이상훈)  
[gcccompil3r@gmail.com](mailto:gcccompil3r@gmail.com)

# **How to Commit Project on GitHub**

이 문서를 만들기 위해 점퍼선과 여러가지를 챙겨왔는데 가만 생각해보니까 빵판이 없다.  
그래서 보드만 있어도 테스트 할 수 있는 것이 무엇일까 생각해보니 HET(High-End Timer) 였다.  
이를 가지고 프로젝트 전체를 git 에 commit 하는 방법을 알아보도록 한다.  
필자가 작성한 윈도우에서 git 설치하기, CCS 설치, HalCoGen 설치 등등이 진행되어 있어야 본 작업의 수행이 가능하다.

이제 CCS 를 키고 git 에 프로젝트 전체를 올려보도록 하자!

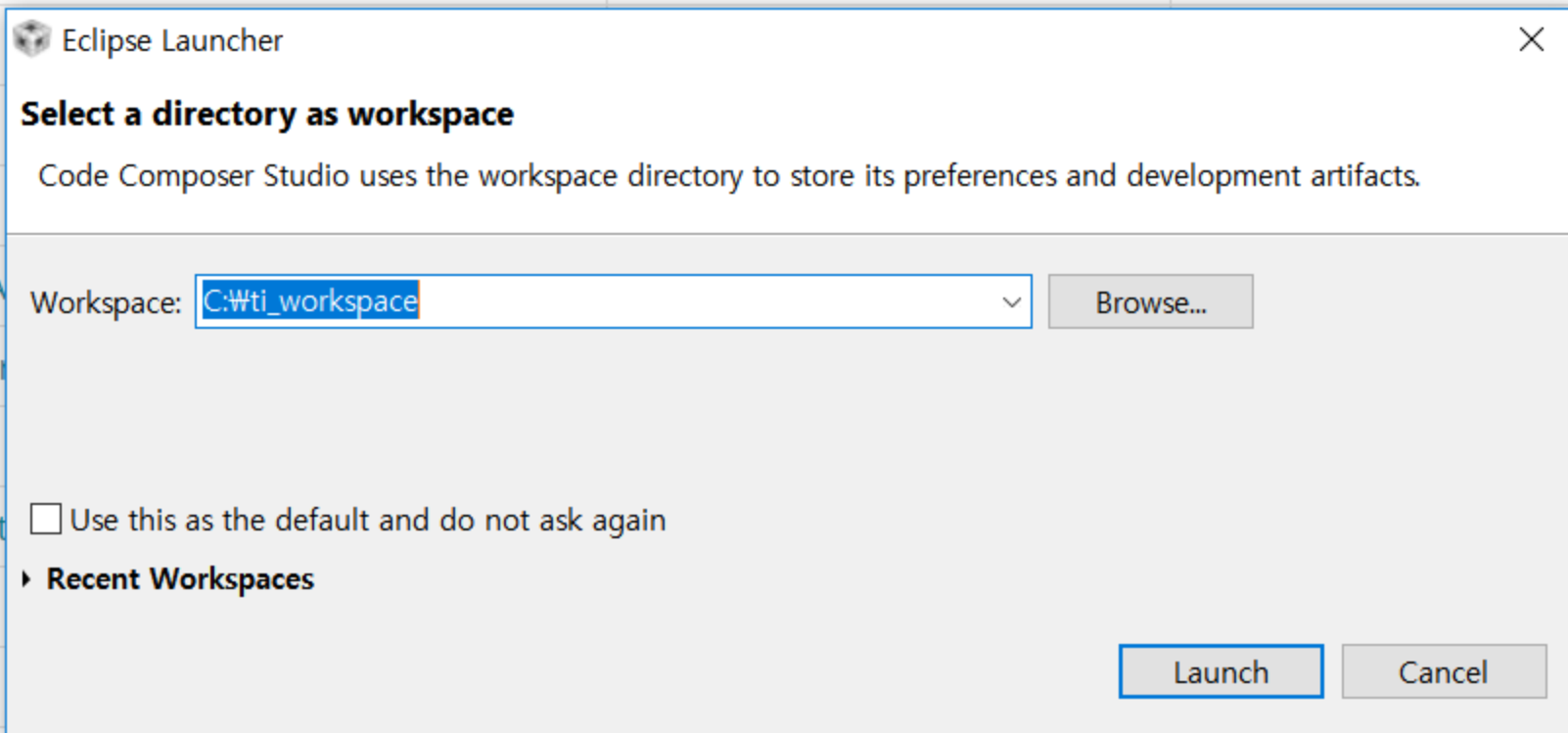


적절하게 workspace 를 지정해주도록 한다.

s MCUs

HALCOGEN

SW Development Tools, IDEs, Co



lware A

stic Lib

(CU), Sit

ers

ti\_workspace - Code Composer Studio

File Edit View Navigate Project Run Scripts Window Help



Getting Started



Resource  
Explorer  
(Examples & Docs)

Browse examples, training  
documentation



Would you like to  
(Recommended for E



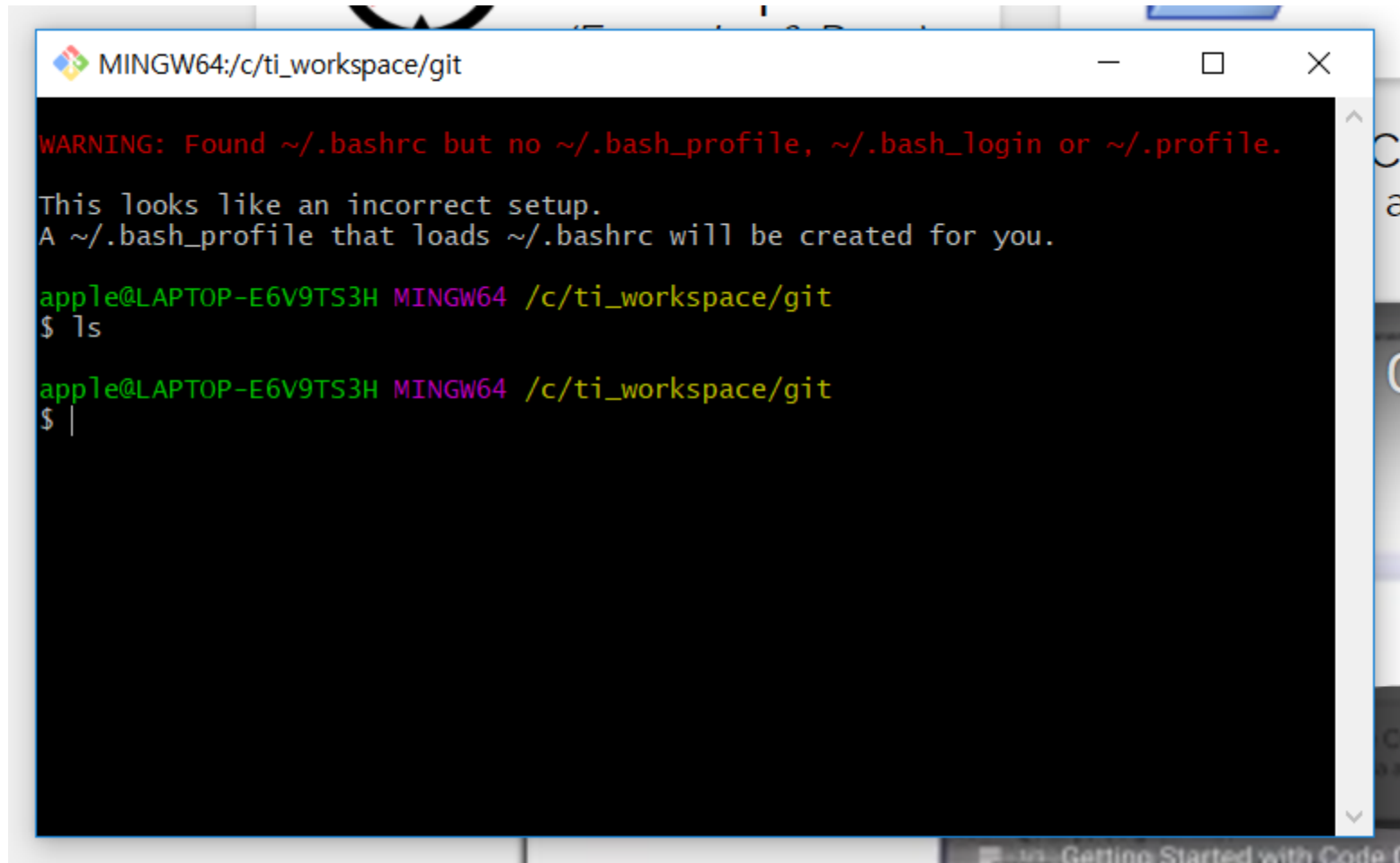
1/1

Getting Started with



Resource  
Explorer  
(Examples & Docs)

이 상태에서 Windows 용 git 을 띄운다.



```
MINGW64:/c/ti_workspace/git

WARNING: Found ~/.bashrc but no ~/.bash_profile, ~/.bash_login or ~/.profile.
This looks like an incorrect setup.
A ~/.bash_profile that loads ~/.bashrc will be created for you.

apple@LAPTOP-E6V9TS3H MINGW64 /c/ti_workspace/git
$ ls

apple@LAPTOP-E6V9TS3H MINGW64 /c/ti_workspace/git
$ |
```

우리의 git 을 ti\_workspace/git 디렉토리 아래에 clone 하도록 한다.  
이때 clone 하는 repository 는 fork 한 repository 여야 한다.

어쨌든 clone 하는데 분량이 엄청나게 많아서 느릴 것이다.  
요즘 뽕세서 힘들텐데 느긋하게 티 타임이라도 가지길 바란다.  
(근대 내가 제일 힘든거 았 ?!)

아래와 같이 fork 한 repository 로 가보자!

The screenshot shows a GitHub repository page for 'link180 / Autonomous RC Car for Baby'. The repository is forked from 'KOITT2/'. The page includes navigation tabs for Code, Pull requests (0), Projects (0), Wiki, Insights, and Settings. Below the repository name, there are statistics: 13 commits, 1 branch, and 0 releases. A pink progress bar is visible. The 'Branch: master' dropdown and 'New pull request' button are present. A message states: 'This branch is 1 commit ahead, 482 commits behind KOITT2:master.' Below this, a table lists the repository's contents:

link180 Test	
circuit	Add Experiment & Circuit
doc	Add Experiment
experiment	change the file name proper
test	Test
.gitignore	Initial commit
LICENSE	Initial commit

업데이트를 해줘야 하는데 먼저 해당 사용자 설정을 하고 repository 를 다운로드 받도록 한다.

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj (master)
$ git config --global user.name "silen[redacted]"
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj (master)
$ git config --global user.email "gcccompil3r@gmail.com"
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj (master)
$ |
```

아이디 적으면됨

메일 주소

그리고 아래와 같이 clone 을 받도록 한다.

```
$ git clone https://github.com/link180/[redacted].git
Cloning into 'RC_Car'...
remote: Counting objects: 1508, done.
remote: Total 1508 (delta 0), reused 0 (delta 0), pack-reused 1508
Receiving objects: 100% (1508/1508), 89.67 MiB | 1.76 MiB/s, done.
Resolving deltas: 100% (194/194), done.
Checking out files: 100% (1399/1399), done.
```

업데이트가 전혀 안되어 있는 것을 볼 수 있다.

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/test
$ cd RC_Car/

apple@LAPTOP-E6V9TS3H MINGW64 /c/test/[redacted] (master)
$ ls
circuit/  doc/  experiment/  LICENSE  README.md  test/

apple@LAPTOP-E6V9TS3H MINGW64 /c/test/[redacted] (master)
$ |
```



현재 프로젝트의 원격 저장소가 어디 있는지 보여주는 명령어를 입력해본다.

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/test/ [redacted] (master)
$ git remote -v
origin https://github.com/link180/RC_Car.git (fetch)
origin https://github.com/link180/RC_Car.git (push)
```

우리의 원격 프로젝트가 어디에 있는지 새롭게 기록해준다.  
당근 빠따로 우리의 본 프로젝트 주소를 여기에 걸어줘야 한다.

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/test/ [redacted] (master)
$ git remote add upstream https://github.com/KOITT2/RC_Car.git

apple@LAPTOP-E6V9TS3H MINGW64 /c/test/ [redacted] (master)
$ git remote -v
origin https://github.com/link180/RC_Car.git (fetch)
origin https://github.com/link180/RC_Car.git (push)
upstream https://github.com/KOITT2/RC_Car.git (fetch)
upstream https://github.com/KOITT2/RC_Car.git (push)
```

단순히 업데이트만 가져온 것 뿐이므로 적용은 아직 진행되지 않았다.

```
$ git fetch upstream
remote: Counting objects: 14393, done.
remote: Compressing objects: 100% (98/98), done.
remote: Total 14393 (delta 272), reused 364 (delta 271), pack-reused 14022
Receiving objects: 100% (14393/14393), 298.46 MiB | 1.01 MiB/s, done.
Resolving deltas: 100% (1583/1583), completed with 25 local objects.
From https://github.com/KOITT2/RC_Car
 * [new branch]      master      -> upstream/master

apple@LAPTOP-E6V9TS3H MINGW64 /c/test/ [redacted] (master)
$ ls
circuit/  doc/  experiment/  LICENSE  README.md  test/
```

아래와 같이 업데이트 정보인 upstream 을 master 에 적용하면 된다.

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/test/ (master)
$ git merge upstream/master
```

메시지를 기록하라고 나오는데 여기에는 그냥 update 를 쓰고 저장하고 나가면 된다.  
사용법은 리눅스의 vi 와 동일하므로 문제 없으리라고 본다.

```
Merge remote-tracking branch 'upstream/master'

# Please enter a commit message to explain why this merge is necessary,
# especially if it merges an updated upstream into a topic branch.
#
# Lines starting with '#' will be ignored, and an empty message aborts
# the commit.
```

이후 아래와 같이 본격적인 업데이트 적용이 시작된다.

```
352\263\204.txt"
create mode 100644 "real_test/\355\225\230\353\223\234\354\
353\266\204\354\204\235.txt"
create mode 100644 "real_test/\355\225\255\352\263\265\354\
355\224\204\353\241\234\352\267\270\353\236\230\353\260\215
create mode 100644 test/180730.txt
create mode 100644 test/8E-ESP8266__Interface_UART__EN_v0.1
create mode 100644 test/ESP8266 - AT Command Reference.pdf
create mode 100644 test/ESP8266_Connect_WiFi.pdf
create mode 100644 test/ESP8266_Specifications_English.pdf
delete mode 100644 test/test.txt
create mode 100644 "test/test/1\354\243\274\354\260\250\DS
3\234 \352\270\260\353\260\230\354\235\230 \353\204\244\355\
355\201\254 \355\224\204\353\241\234\352\267\270\353\236\230
243\274\354\260\250).pdf"
create mode 100644 "test/test/1\354\243\274\354\260\250\SP
3\240.pdf"
create mode 100644 "test/test/1\354\243\274\354\260\250\wi
23\210 \355\216\214\354\233\250\354\226\264 \354\227\205\352
\354\235\264\353\223\234.pdf"
```

이제 아래와 같이 업데이트가 적용된 모습을 볼 수 있다.

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/test/ (master)
$ ls
circuit/      dsp_proj/    LICENSE      past_present/ real_test/
cur_present/  experiment/  mcu_proj/    pcb/           test/
doc/          fpga_proj/  member_profile/ README.md
```

그리고 mv 명령어를 사용하여 test 디렉토리 이름을 work 로 변경하도록 한다.  
애초에 변경할 필요 없이 work 로 만들어서 작업해도 상관 없다.  
필자는 이름이 같은 것이 있어서 백업이 필요하여 test 에서 work 로 만든 것이다.

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/ (master)
$ ls
circuit/      dsp_proj/    LICENSE      past_present/ real_test/
cur_present/  experiment/  mcu_proj/    pcb/           test/
doc/          fpga_proj/  member_profile/ README.md
```

이제부터 MCU 에 대해 작업한 내용들은 mcu\_proj 에 배치하고  
FPGA 에 작업한 내용들은 fpga\_proj 에 배치하고  
DSP 에 작업한 내용들은 dsp\_proj 에 배치하도록 한다.

현재 작업은 프로젝트를 올리는 방법을 몰라서 MPU9250 이 압축 파일로 올라가 있다.  
이를 올리는 방법을 알아보기 위해 HET 로 PWM 제어하던 것을 여기에 올려보도록 한다.  
여기에는 우리 git 의 기형적인 문제를 해결하고자 하는 부분도 존재한다.  
해외에서 볼 때 코드 작업은 별로 없고 커밋량만 이상하게 높네 ? 라고 오해를 할 수 있기 때문이다.

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/ [redacted] (master)
$ ls
circuit/      dsp_proj/    LICENSE      past_present/  real_test/
cur_present/  experiment/  mcu_proj/    pcb/           test/
doc/          fpga_proj/   member_profile/  README.md

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/ [redacted] (master)
$ cd mcu_proj/

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/ [redacted]/mcu_proj (master)
$ ls
__MPU9250.zip  prepare.txt

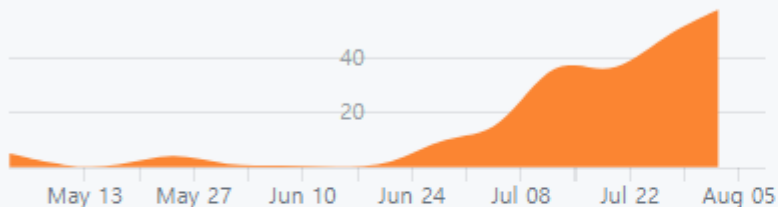
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/ [redacted]/mcu_proj (master)
$ |
```



silenc3502

#1

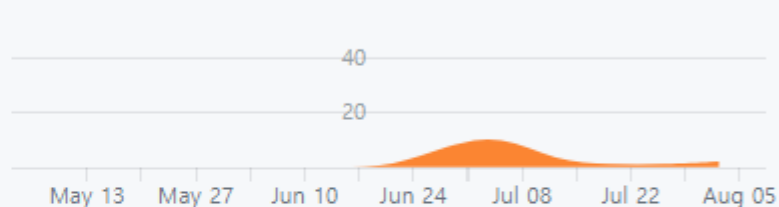
214 commits 5,607,989 ++ 197,182 --



ahnsangjae

#2

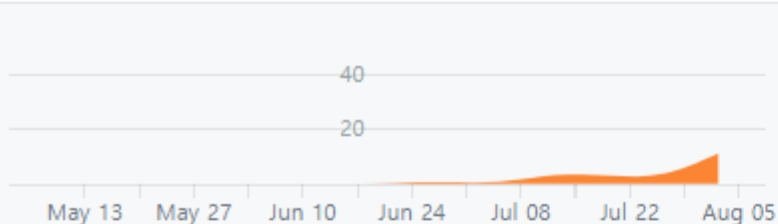
26 commits 2 ++ 3 --



hanbulkr

#3

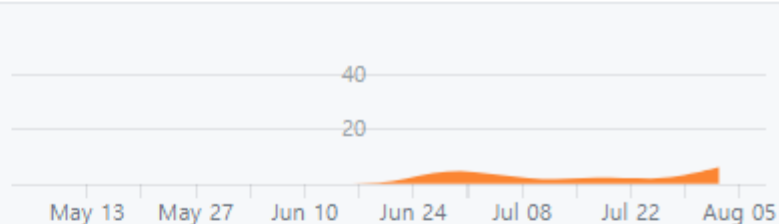
21 commits 505 ++ 0 --



HowardKIM2

#4

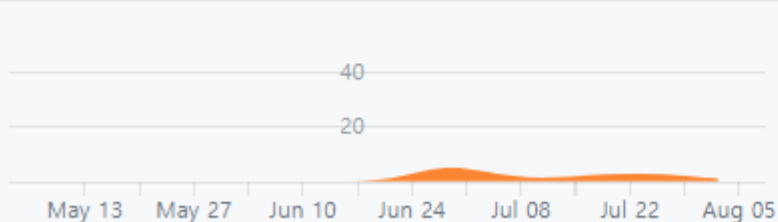
20 commits 1 ++ 0 --



glgltlids

#5

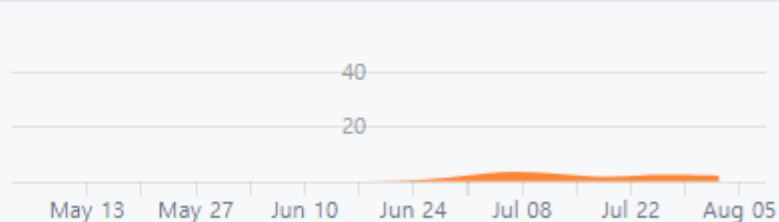
17 commits 4 ++ 1 --



HyunwooParkk

#6

14 commits 0 ++ 0 --





moonhanna

8 commits 0 ++ 0 --

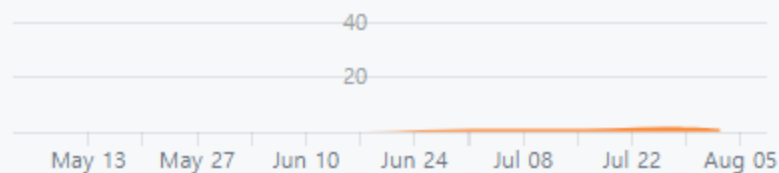
#7



sue100012

7 commits 0 ++ 0 --

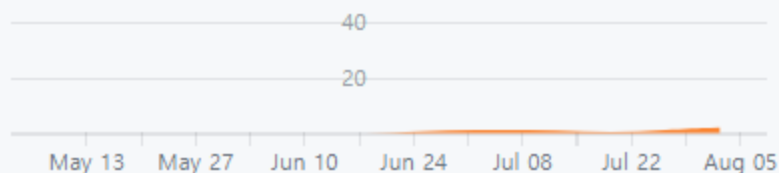
#8



SungHwan-Jang

6 commits 0 ++ 0 --

#9



jamiech41

6 commits 0 ++ 0 --

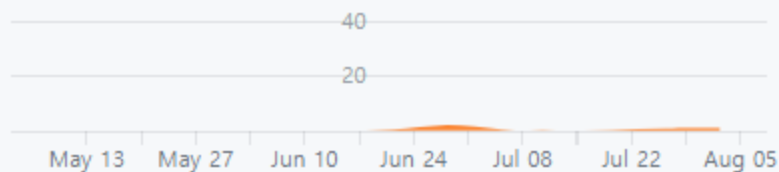
#10



LeeHoseong

5 commits 0 ++ 1 --

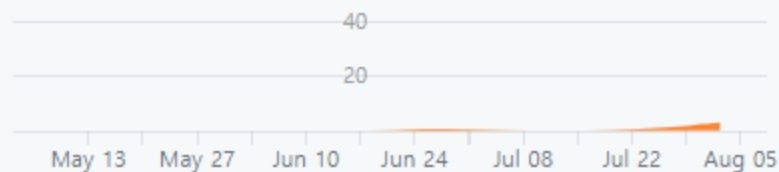
#11



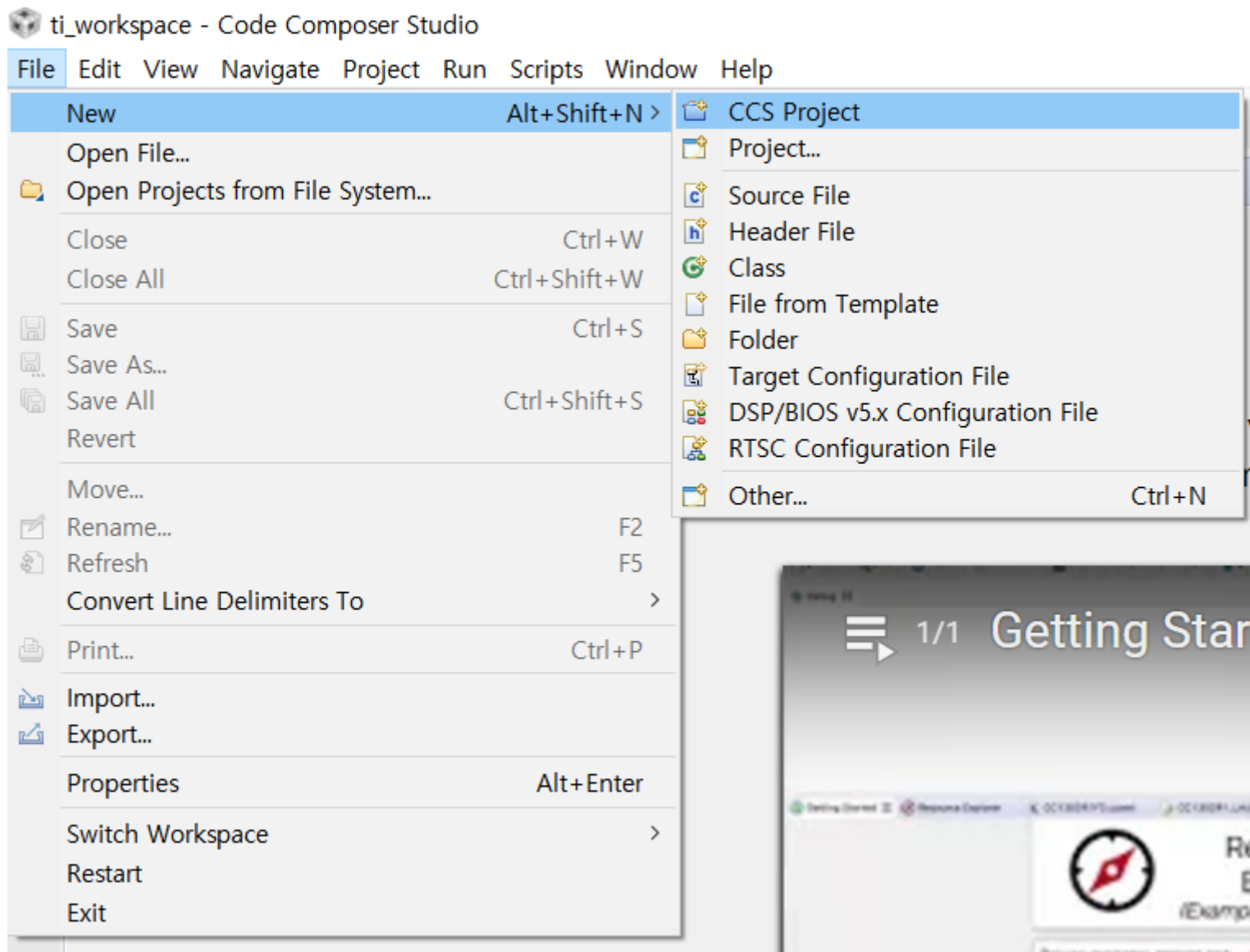
fstopdg

5 commits 0 ++ 0 --

#12



이제 프로젝트를 만들어보자!



JTAG 연결이 잘 되는지 확인을 수행하기 위해 Verify 를 눌러야 한다.


New CCS Project

**CCS Project**

① Project name must be specified

Target:

Connection:

 Cortex R [ARM]

Project name:

☐ Use default location

Location:

Compiler version:

▶ Tool-chain



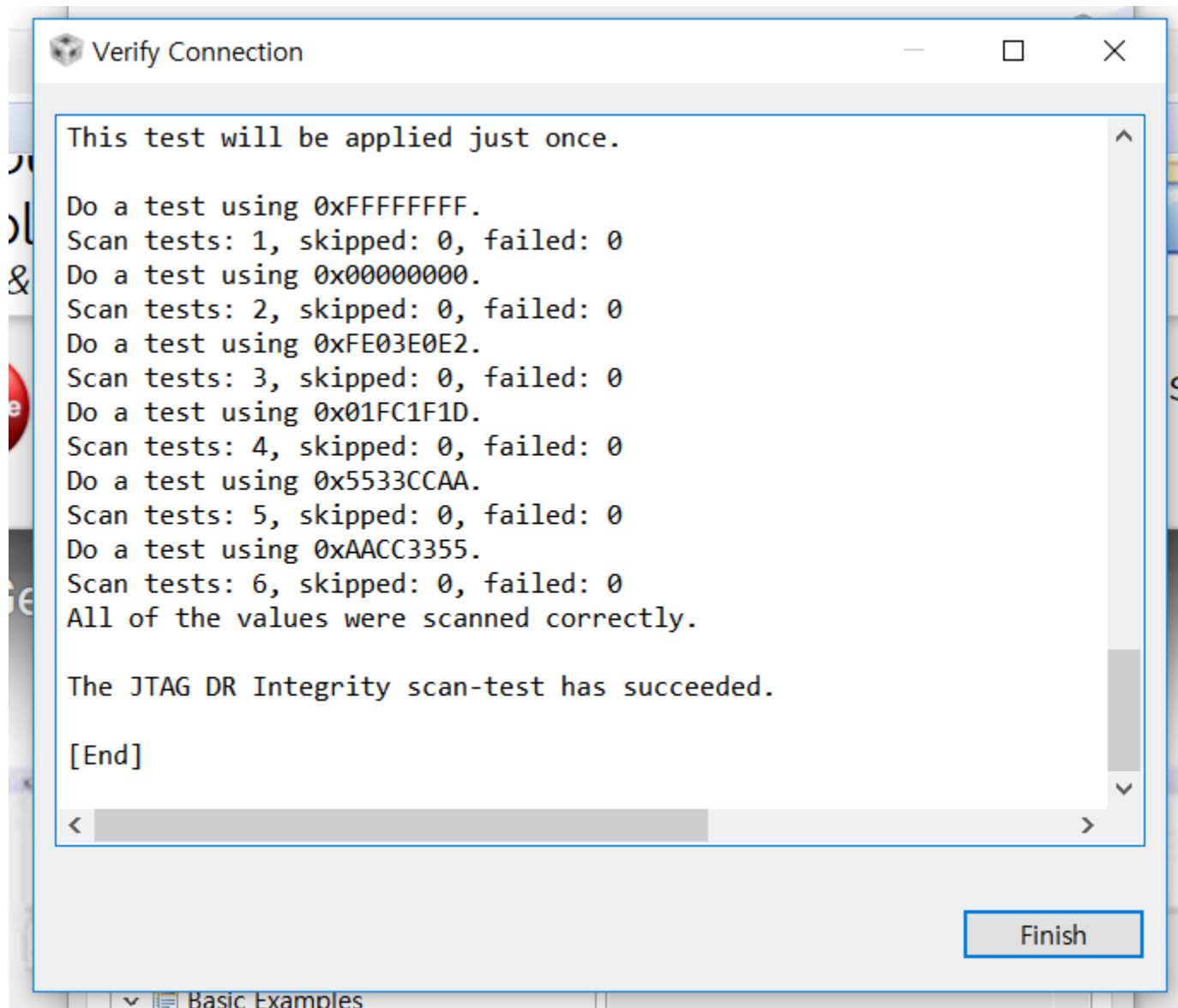
USB 만 연결하면 전원 공급이 안정적이지 못해  
Verify 가 정상적으로 수행되지 않는다.



전원을 안정적으로 제공해주면  
불빛이 이렇게 나오게 된다.

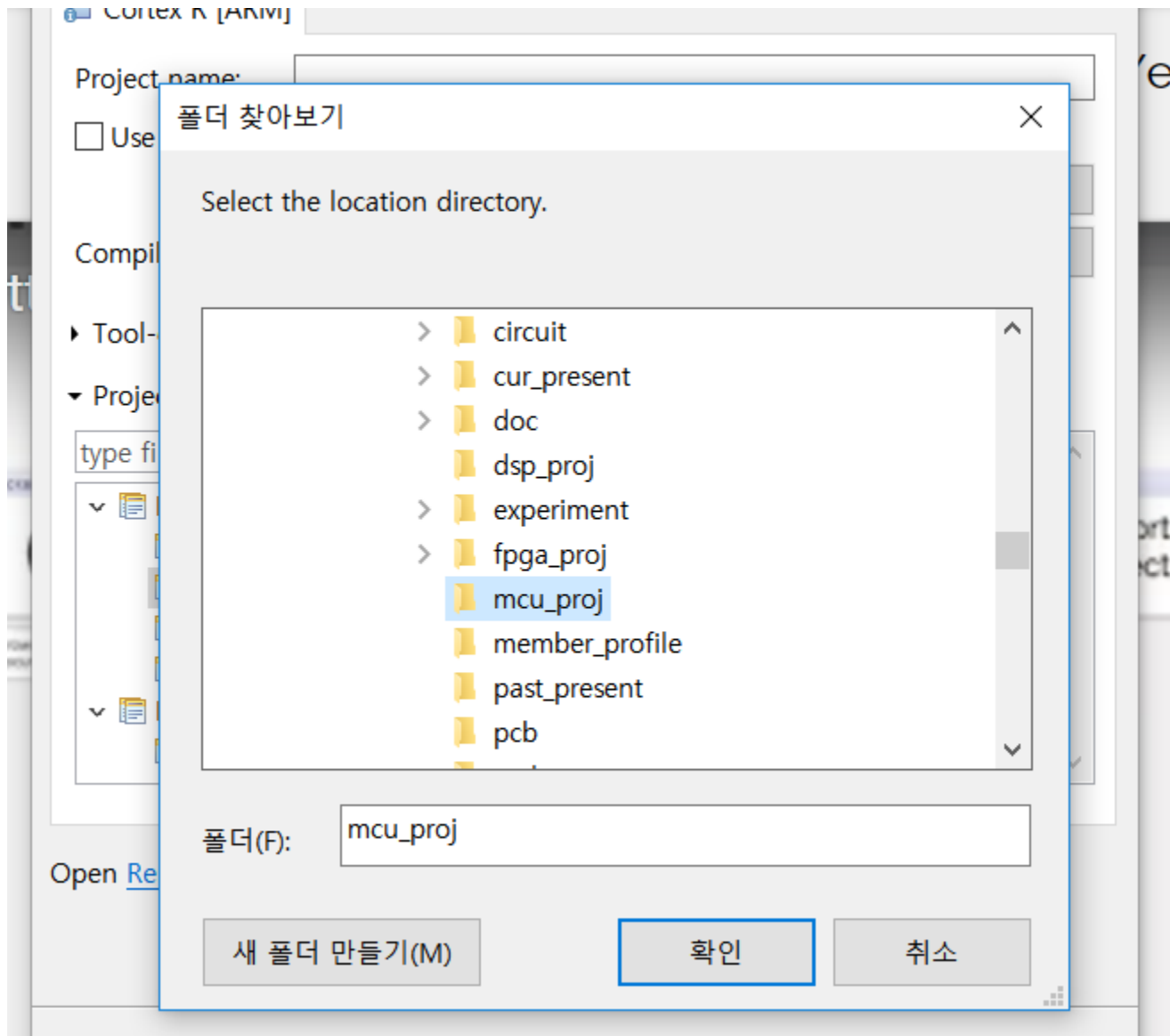


보드가 잘 동작함을 알 수 있다.



이제 프로젝트 위치를 잡도록 한다.


앞서서 git clone 하여 받았던 위치에 가보면 mcu\_proj 디렉토리를 찾을 수 있을 것이다.



경로가 길다고 뭐라고 하는데 아래와 같이 경로를 짧게 만들어준다.  
가려진 부분은 우리의 git repository 에 해당한다.

Target: Unclassified Devices TMS570LC43xx

Connection: Texas Instruments XDS100v2 USB Debug Probe Verify...

 Cortex R [ARM]

Project name:


☐ Use default location


Location:  Browse...


Compiler version: TI v18.1.2.LTS More...


▶ Tool-chain


▼ Project templates and examples


▼  Empty Projects


 Empty Project

 Empty Project (with main.c)

 Empty Assembly-only Project

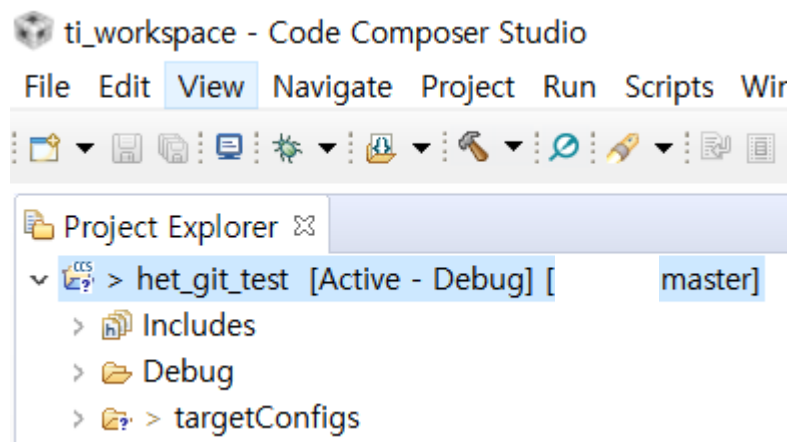
 Empty RTSC Project

▼  Basic Examples

 Hello World

Creates an empty project initialized for the selected device.

그럼 아래와 같이 프로젝트가 만들어진다.



아래와 같이 git 저장소쪽에 프로젝트가 생성된 것을 볼 수 있다.

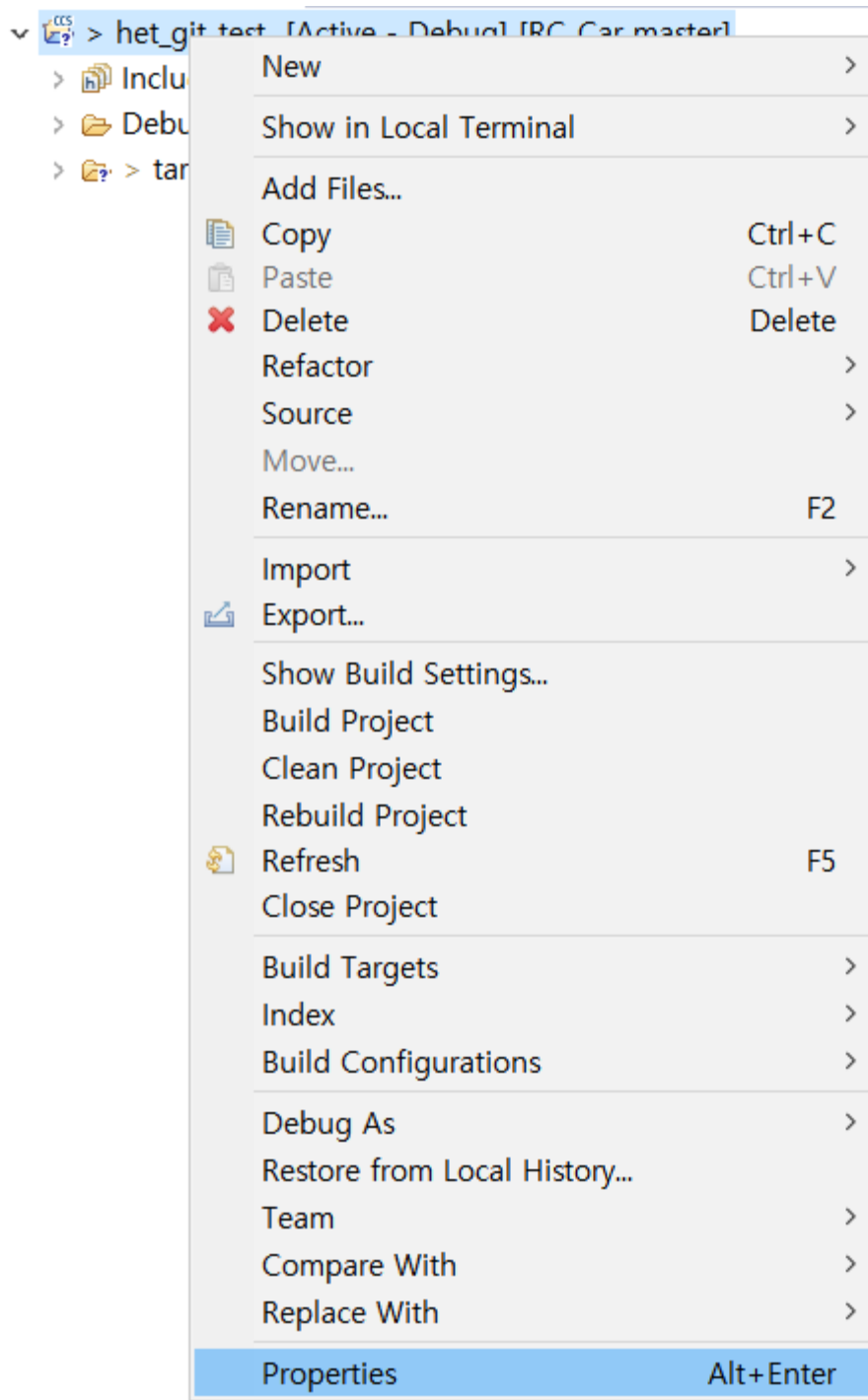
```
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj (master)
$ ls ..
circuit/      doc/      experiment/ LICENSE  member_profile/ pcb/      real_test/
cur_present/ dsp_proj/ fpga_proj/ mcu_proj/ past_present/  README.md test/

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj (master)
$ ls
__MPU9250.zip  het_git_test/  prepare.txt

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj (master)
$ ls het_git_test/
Debug/  targetConfigs/

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj (master)
$
```

속성을 눌러서 경로를 파악한다.



type filter text

- Resource
- General
- Build
  - ARM Compiler
    - Processor Options
    - Optimization
    - Include Options
    - ULP Advisor
    - Predefined Symbols
  - Advanced Options
  - ARM Linker
    - ARM Hex Utility [Disable]
- Debug
- Git

## Resource

Path: /het\_git\_test

Type: Project

Location: C:\work\mcu\_proj\het\_git\_test

Last modified: 2018년 8월 11일 오후 1:17:29

### Text file encoding

☒ Inherited from container (MS949)

☐ Other: MS949

☐ Store the encoding of derived resources separately

### New text file line delimiter

☒ Inherited from container (Windows)

☐ Other: Windows

경로를 복사해둔다.

Restore Defaults

Apply

[Show advanced settings](#)

Apply and Close

Cancel



평소대로 HalCoGen 을 연동하도록 한다.

**HC** HAL Code Generator - [Start Page]

File Edit View Tools Window Help

Start Page

My.TI TI Home Microcontrollers

**TEXAS INSTRUMENTS** INNOVATE. CREATE. MAKE THE DIFFERENCE.™

**HALCoGen** **HALCoGen: 04.07.00 - Released 7.July.2017**  
[HALCoGen Release Notes](#)

**Important Hercules Safety MCU Links:**  
Hercules product web pages provide access to device data sheets, technical reference manuals, videos, software downloads/updates, and online ordering of evaluation and development kits.

**[HALCoGen Wiki Page](#)**

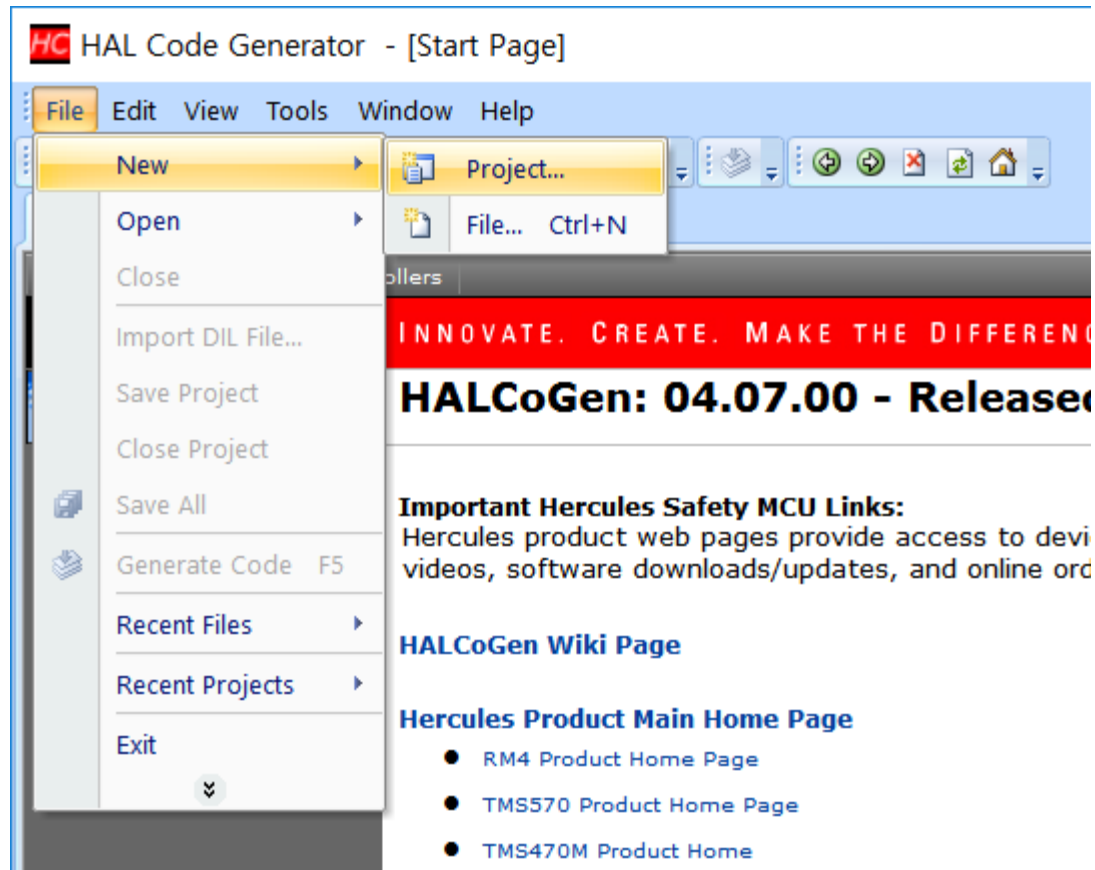
**[Hercules Product Main Home Page](#)**

- [RM4 Product Home Page](#)
- [TMS570 Product Home Page](#)
- [TMS470M Product Home Page](#)

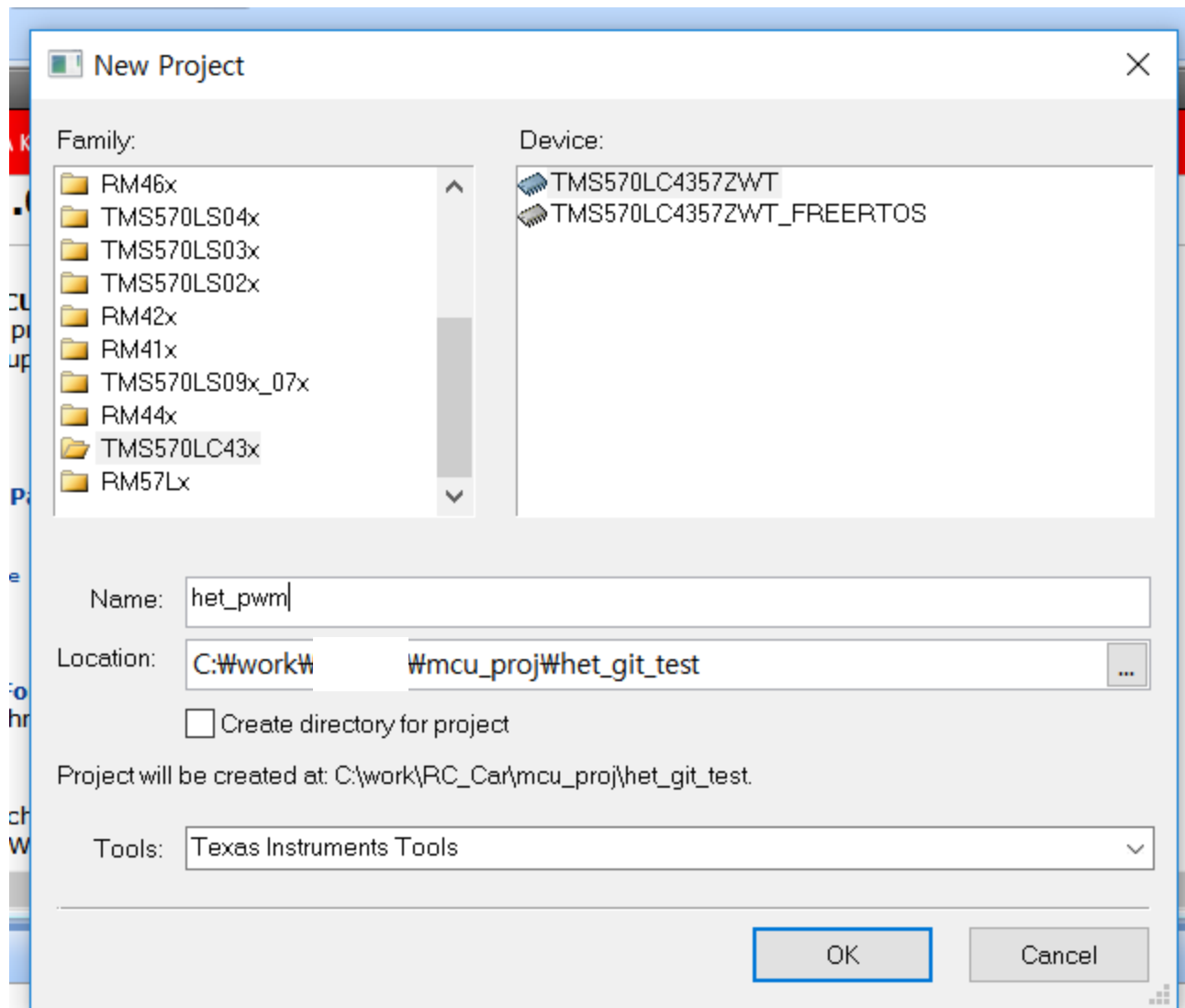
**[Hercules Technical Support Forum](#)**  
Search for topics or ask technical questions about all Hercules MCUs - RM4, TMS570 and TMS470M.

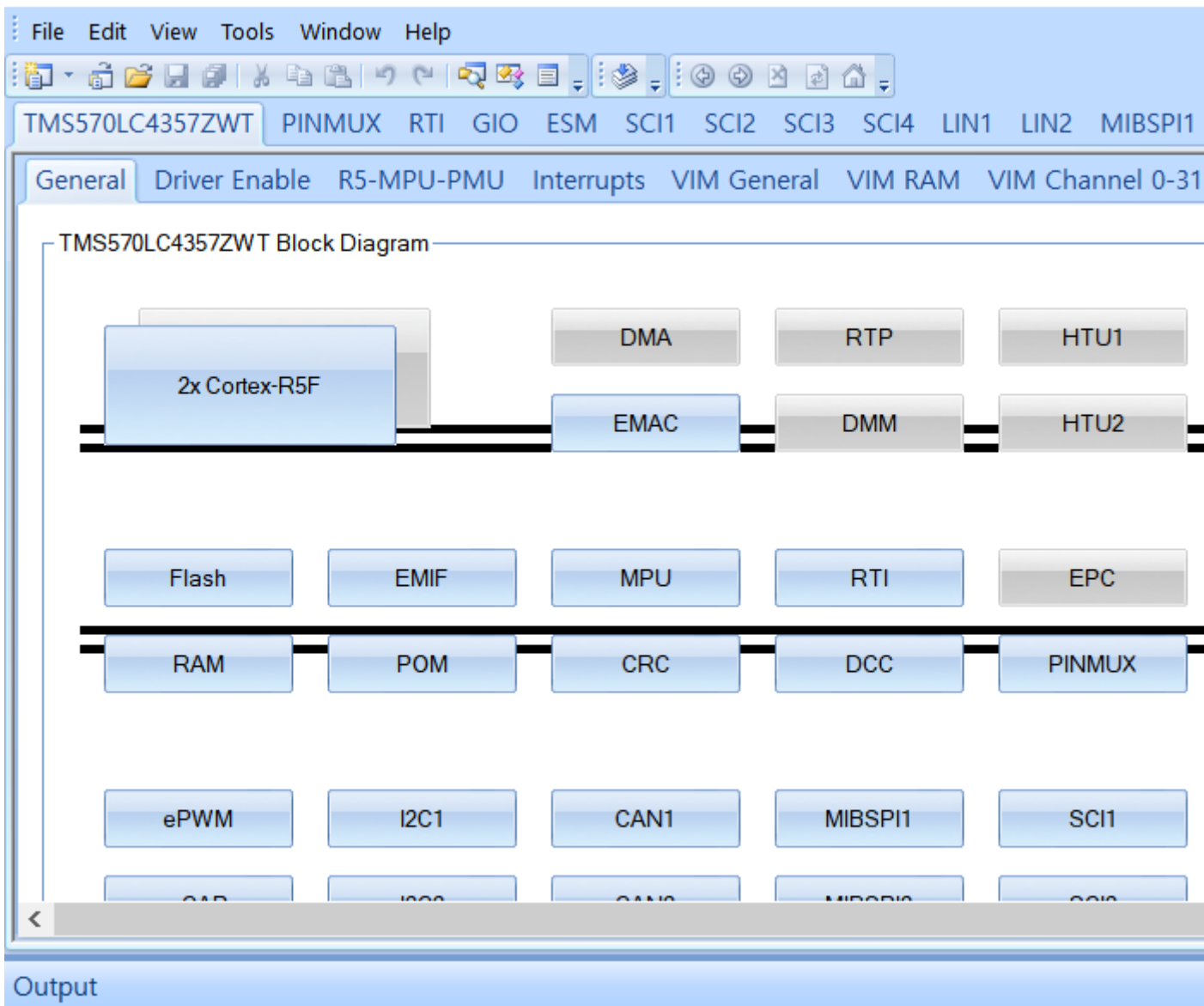
**[Hercules MCU Wiki Site](#)**  
Download development kit schematics, software examples, training videos and information much more on the Hercules WIKI pages.

프로젝트를 설정한다.



RTOS 든 Bare Metal 이던 적절하게 선택하도록 한다.

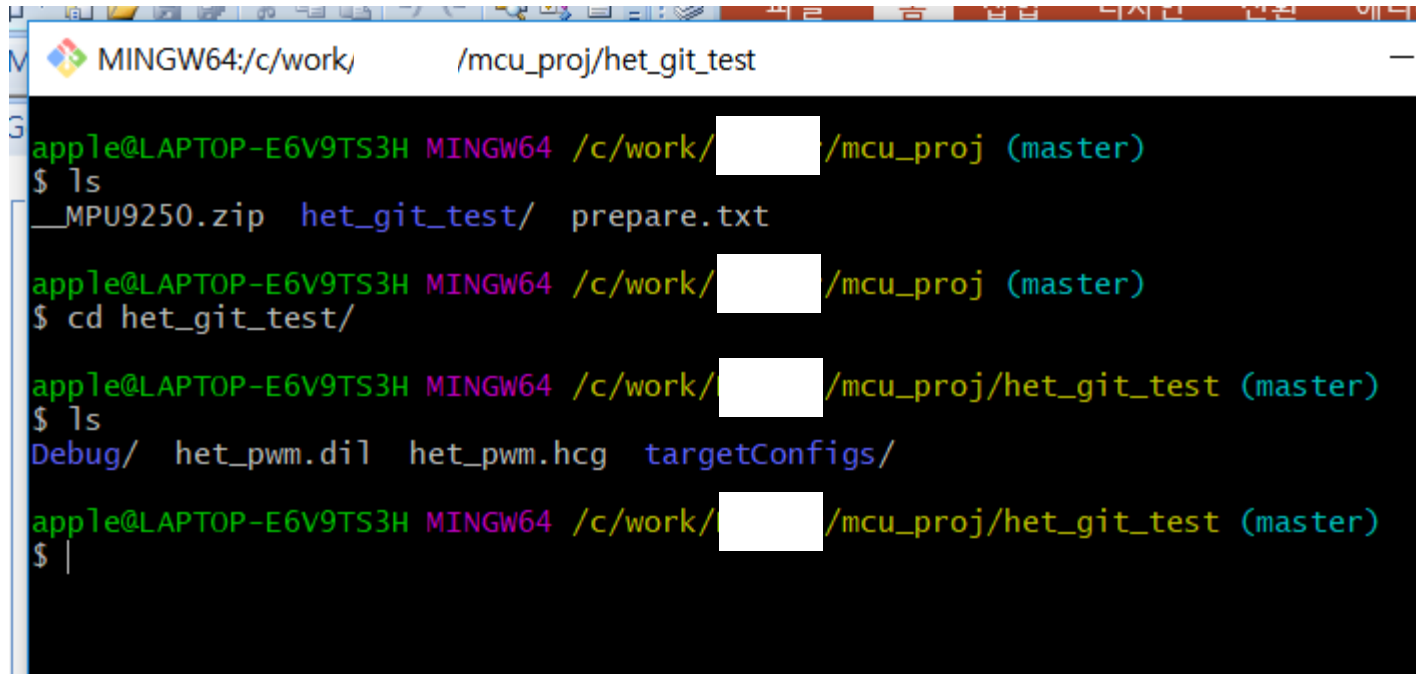




## Output

Loading: FEE: 'FEEv000.xml'  
 Loading: AJSM: 'AJSMv000.xml'  
 Load complete

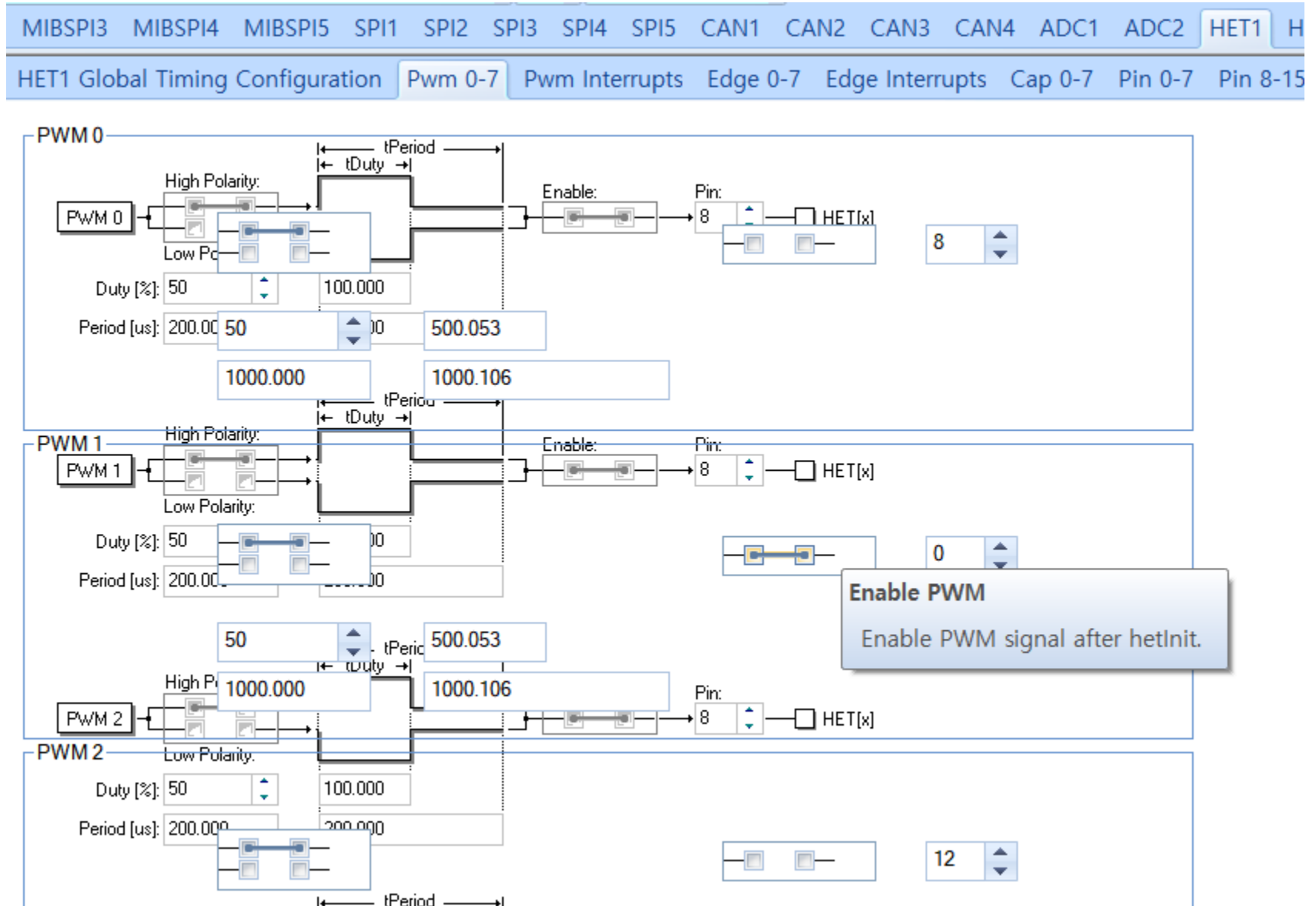
아래 부분을 보면 het\_pwm.hcg 라는 것이 보일 것이다.  
이것이 HalCoGen 프로젝트에 해당한다.



```
MINGW64:/c/work/ /mcu_proj/het_git_test  
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj (master)  
$ ls  
__MPU9250.zip  het_git_test/  prepare.txt  
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj (master)  
$ cd het_git_test/  
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj/het_git_test (master)  
$ ls  
Debug/  het_pwm.dil  het_pwm.hcg  targetConfigs/  
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj/het_git_test (master)  
$ |
```

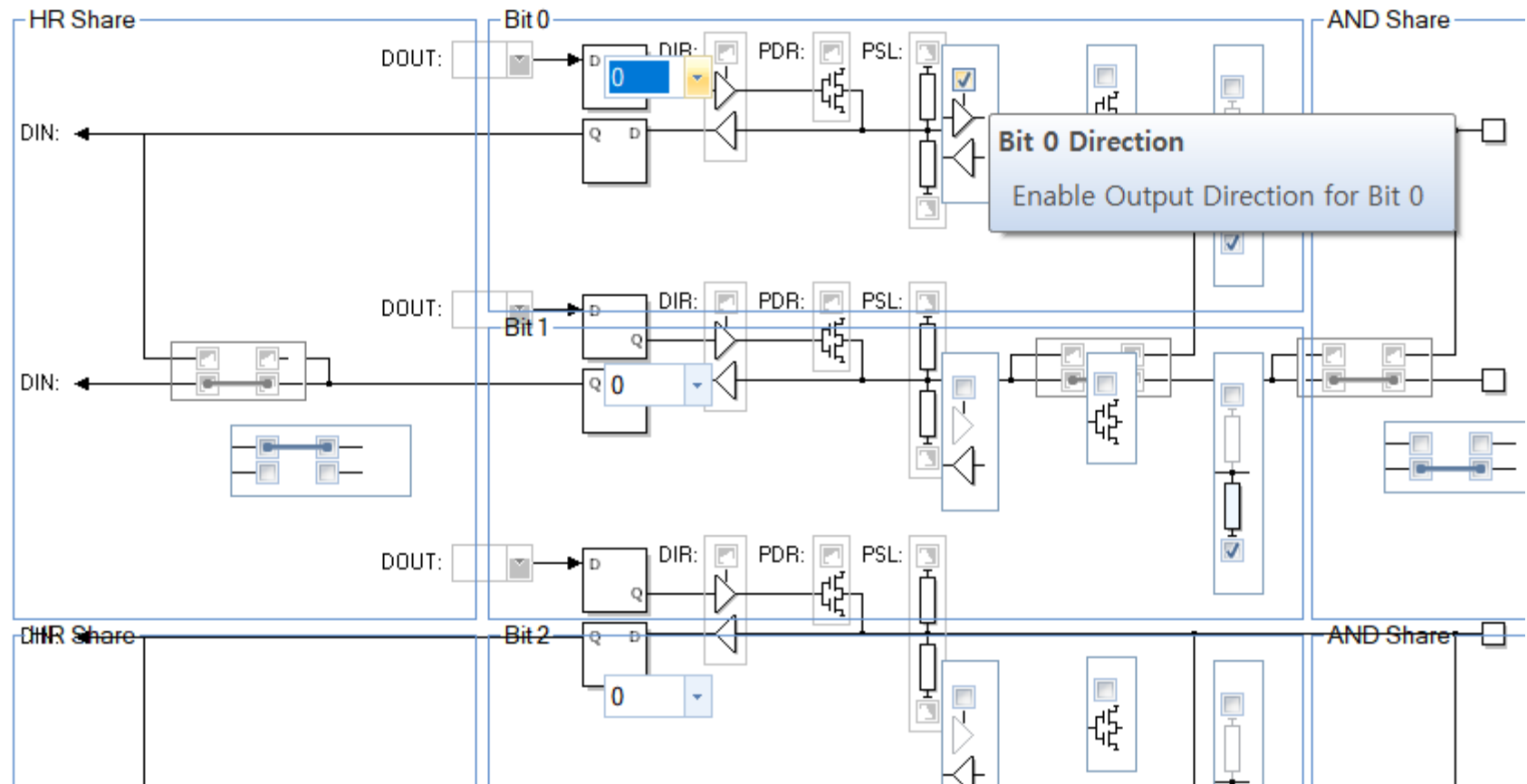
☐ Enable RTI driver☐ Mark/Unmark all drivers☐ Enable GIO driver \*\*☐ Enable SCI drivers☐ Enable SCI3 driver \*\*☐ Enable SCI4 driver \*\*☐ Enable LIN drivers☐ Enable LIN1 driver \*\* /☐ Enable SCI1 driver \*\*☐ Enable LIN2 driver \*\* /☐ Enable SCI2 driver \*\*☐ Enable MIBSPI drivers☐ Enable MIBSPI1 driver \*\*☐ Enable SPI1 driver \*\*☐ Enable MIBSPI2 driver \*\*☐ Enable SPI2 driver \*\*☐ Enable MIBSPI3 driver \*\*☐ Enable SPI3 driver \*\*☐ Enable MIBSPI4 driver \*\*☐ Enable SPI4 driver \*\*☐ Enable MIBSPI5 driver \*\*☐ Enable SPI5 driver \*\*☐ Enable CAN drivers☐ Enable CAN1 driver☐ Enable CAN2 driver☐ Enable CAN3 driver☐ Enable CAN4 driver \*\*☐ Enable ADC drivers☐ Enable ADC1 driver \*\*☐ Enable ADC2 driver \*\*☒ Enable HET drivers☒ Enable HET1 driver \*\*☐ Enable HET2 driver \*\*

HET PWM 을 인가해준다.

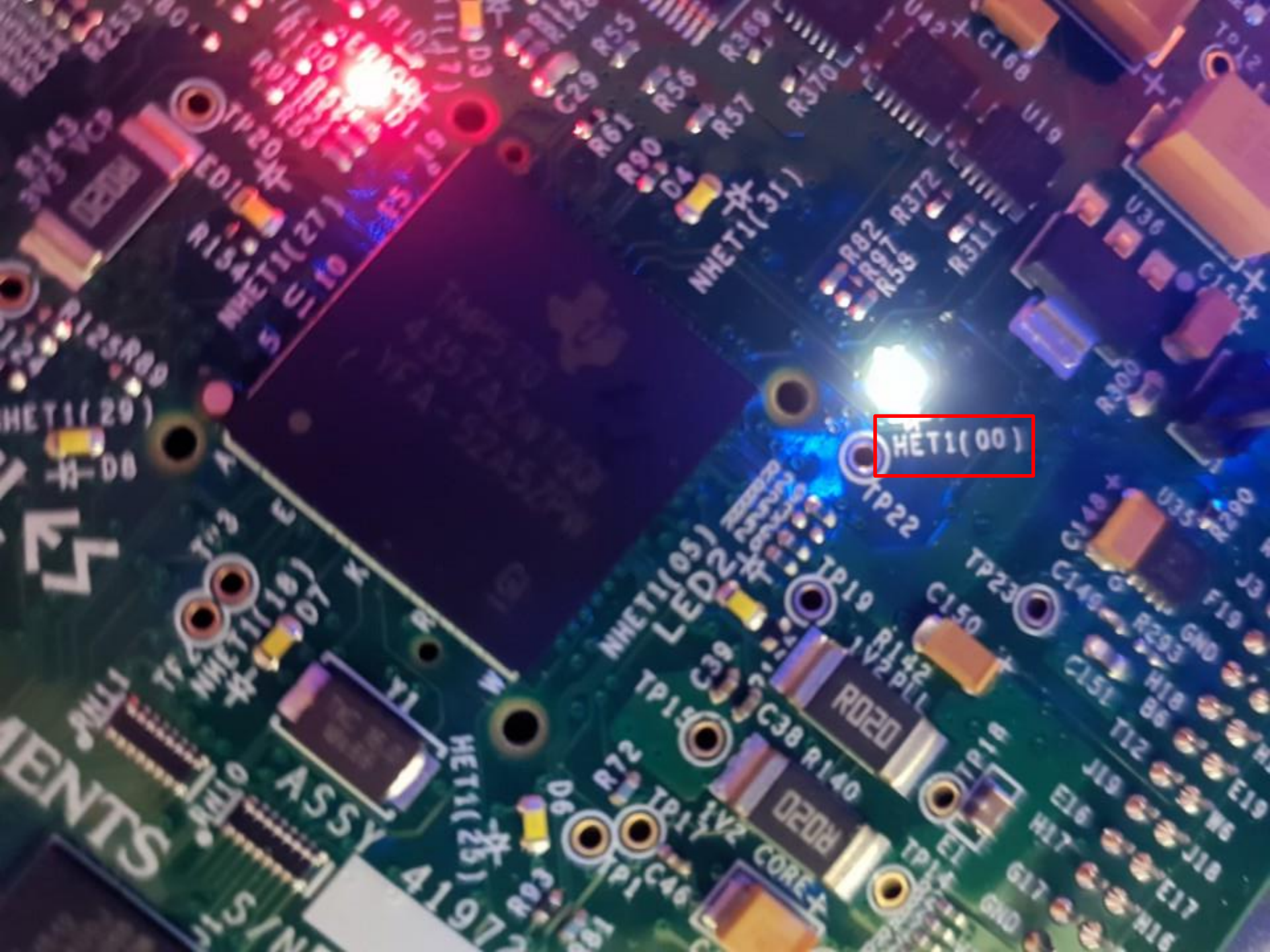


MIBSPI3 MIBSPI4 MIBSPI5 SPI1 SPI2 SPI3 SPI4 SPI5 CAN1 CAN2 CAN3 CAN4 ADC1 ADC2 HET1 H

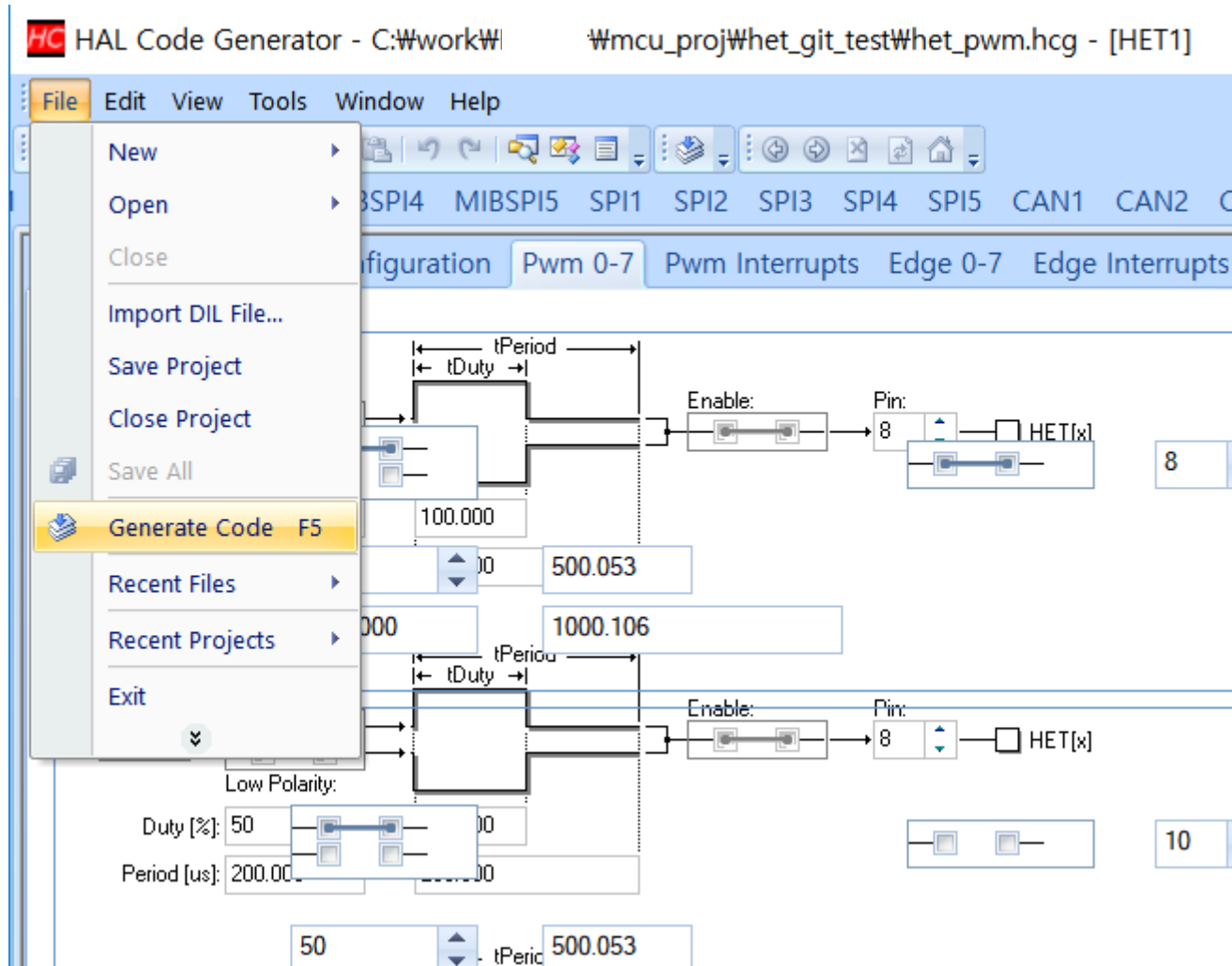
HET1 Global Timing Configuration Pwm 0-7 Pwm Interrupts Edge 0-7 Edge Interrupts Cap 0-7 Pin 0-7 Pin 8-15







HalCoGen 에서 작업할 내용들을 모두 설정하고 **Generate Code** 하면 코드가 만들어진다.



Project Explorer

CCS > het\_git\_test [Active - Debug] [master]

- > Includes
- > Debug
- > include
- > source
  - > HL\_epc.c
  - > HL\_errata.c
  - > HL\_esm.c
  - > HL\_het.c
  - > HL\_nmpu.c
  - > HL\_notification.c
  - > HL\_pinmux.c
  - > HL\_sys\_core.asm
  - > HL\_sys\_dma.c
  - > HL\_sys\_intvecs.asm
  - > HL\_sys\_link.cmd
  - > HL\_sys\_main.c
  - > HL\_sys\_mpu.asm
  - > HL\_sys\_pcr.c
  - > HL\_sys\_phantom.c
  - > HL\_sys\_pmm.c
  - > HL\_sys\_pmu.asm
  - > HL\_sys\_startup.c
  - > HL\_sys\_vim.c
  - > HL\_system.c
- > targetConfigs
  - het\_pwm.dil
  - het\_pwm.hcg

CCS 로 돌아가 보면 이렇게 코드들이 보인다.

Getting Started

Problems 0 items

아래와 같이 코드들이 추가된 모습을 볼 수 있다.

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[REDACTED]/mcu_proj/het_git_test (master)
$ ls
Debug/  het_pwm.dil  het_pwm.hcg  include/  source/  targetConfigs/

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[REDACTED]/mcu_proj/het_git_test (master)
$ ls include/
Device_header.h      HL_gio.h            HL_reg_can.h        HL_reg_l2ramw.h     HL_rti.h
Device_TMS570LC43.h  HL_hal_stdtypes.h   HL_reg_ccmr5.h       HL_reg_lin.h        HL_rtp.h
Device_types.h       HL_het.h            HL_reg_crc.h         HL_reg_mibspi.h     HL_sci.h
fee_interface.h      HL_htu.h            HL_reg_dcc.h         HL_reg_nmpu.h       HL_spi.h
HL_adc.h             HL_hw_emac.h        HL_reg_dma.h         HL_reg_pbist.h      HL_sys_common.h
HL_can.h             HL_hw_emac_ctrl.h   HL_reg_dmm.h         HL_reg_pcr.h        HL_sys_core.h
HL_crc.h             HL_hw_mdio.h        HL_reg_ecap.h        HL_reg_pinmux.h     HL_sys_dma.h
HL_dcc.h             HL_hw_reg_access.h  HL_reg_efc.h         HL_reg_pmm.h        HL_sys_mpu.h
HL_dmm.h             HL_i2c.h            HL_reg_emif.h        HL_reg_pom.h        HL_sys_pcr.h
HL_ecap.h            HL_lin.h            HL_reg_epc.h         HL_reg_rti.h        HL_sys_pmm.h
HL_emac.h            HL_mdio.h           HL_reg_eqep.h        HL_reg_rtp.h        HL_sys_pmu.h
HL_emac_phyConfig.h  HL_mibspi.h         HL_reg_esm.h         HL_reg_sci.h        HL_sys_vim.h
HL_emif.h            HL_nmpu.h           HL_reg_etpwm.h       HL_reg_scm.h        HL_system.h
HL_epc.h             HL_phy_dp83640.h    HL_reg_flash.h       HL_reg_sdcmmr.h     MemMap.h
HL_eqep.h            HL_phy_tlk111.h     HL_reg_gio.h         HL_reg_spi.h        std_nhet.h
HL_errata.h          HL_pinmux.h         HL_reg_het.h         HL_reg_stc.h        ti_fee.h
HL_esm.h             HL_pom.h            HL_reg_htu.h         HL_reg_system.h     ti_fee_cfg.h
HL_etpwm.h          HL_reg_adc.h        HL_reg_i2c.h         HL_reg_vim.h        ti_fee_types.h

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[REDACTED]/mcu_proj/het_git_test (master)
$ |
```

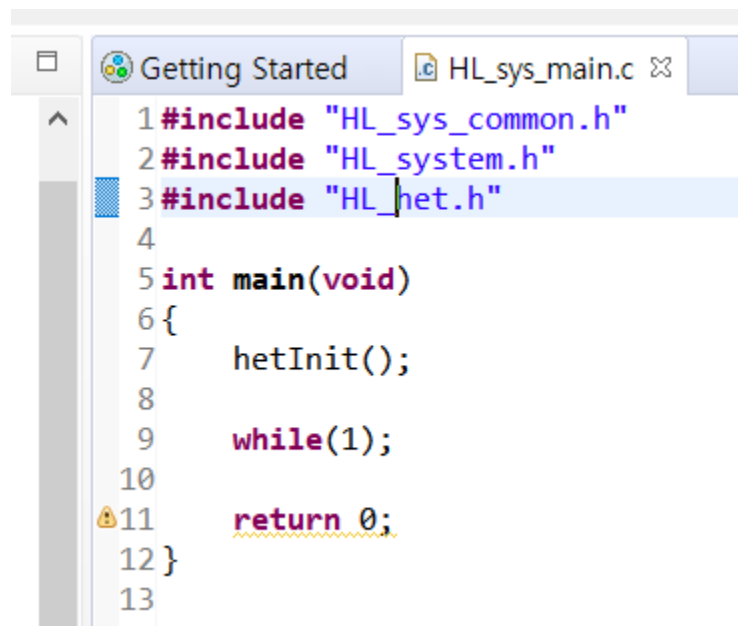
소스 코드들도 잘 보이는 것을 확인 할 수 있다.

MINGW64:/c/work, /mcu\_proj/het\_git\_test

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj/het_git_test (master)
$ ls
Debug/  het_pwm.dil  het_pwm.hcg  include/  source/  targetConfigs/

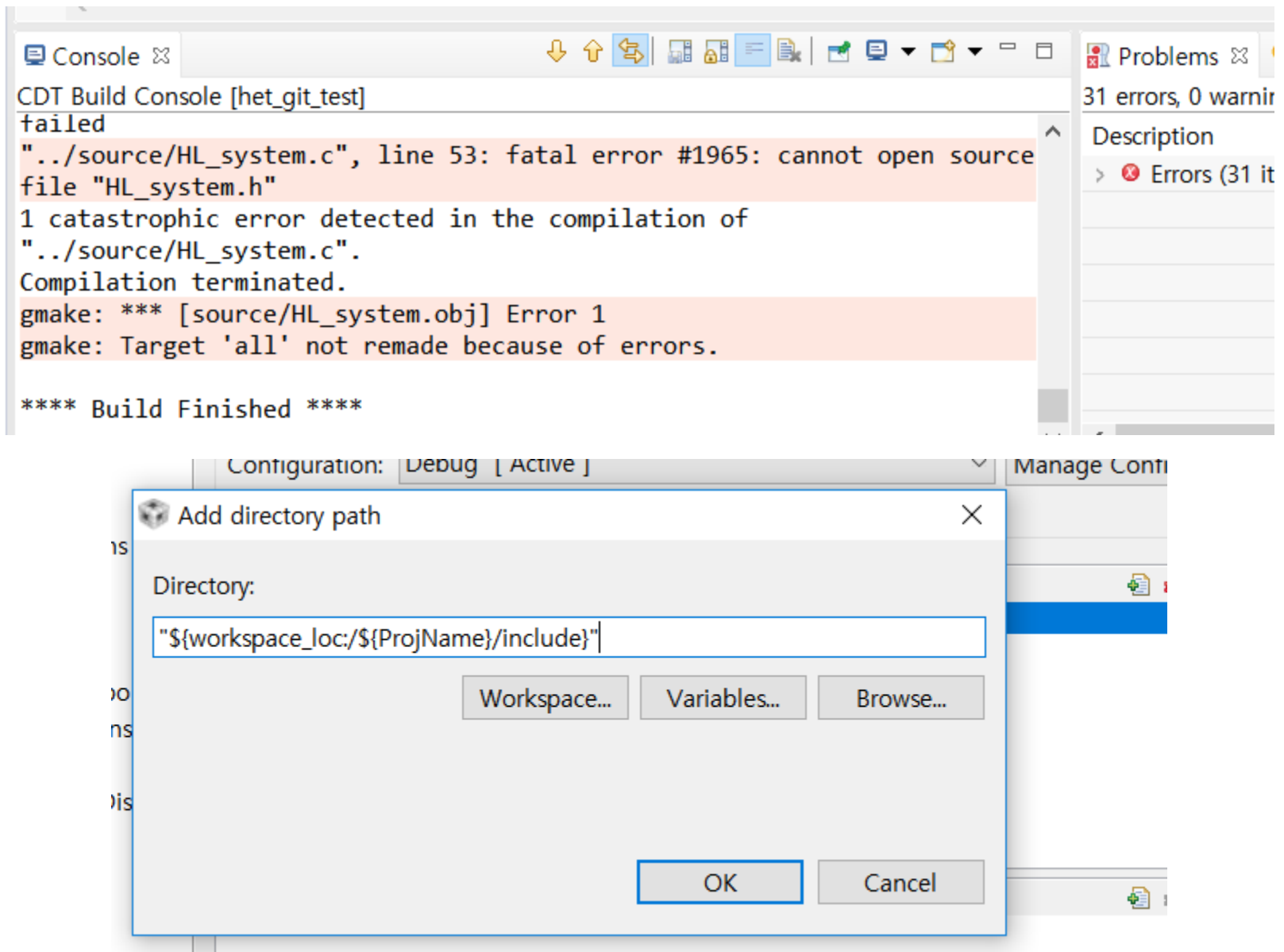
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj/het_git_test (master)
$ ls source/
HL_epc.c      HL_nmpu.c      HL_sys_dma.c      HL_sys_mpu.asm    HL_sys_pmu.asm
HL_errata.c   HL_notification.c  HL_sys_intvecs.asm  HL_sys_pcr.c      HL_sys_startup.c
HL_esm.c      HL_pinmux.c     HL_sys_link.cmd    HL_sys_phantom.c  HL_sys_vim.c
HL_het.c      HL_sys_core.asm  HL_sys_main.c      HL_sys_pmm.c      HL_system.c

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj/het_git_test (master)
$ |
```



```
Getting Started  HL_sys_main.c
1 #include "HL_sys_common.h"
2 #include "HL_system.h"
3 #include "HL_het.h"
4
5 int main(void)
6 {
7     hetInit();
8
9     while(1);
10
11     return 0;
12 }
13
```

에러가 31 개나 뜨는데 환경 설정을 아직 안해줘서 그런데 추가해주도록 한다.





## Properties for het\_git\_test

type filter text

- › Resource
- General
- ✓ Build
  - ✓ ARM Compiler
    - Processor Options
    - Optimization
    - Include Options**
    - ULP Advisor
    - Predefined Symbols
  - › Advanced Options
  - › ARM Linker
    - ARM Hex Utility [Disable]
- Debug
- Git

### Include Options



Configuration: Debug [ Active ]

Manage Configurations...

Add dir to #include search path (--include\_path, -I)



`${PROJECT_ROOT}` ...

`"${workspace_loc}/${ProjName}/include"` ...

`${CG_TOOL_ROOT}/include` ...

Specify a preinclude file (--preinclude)



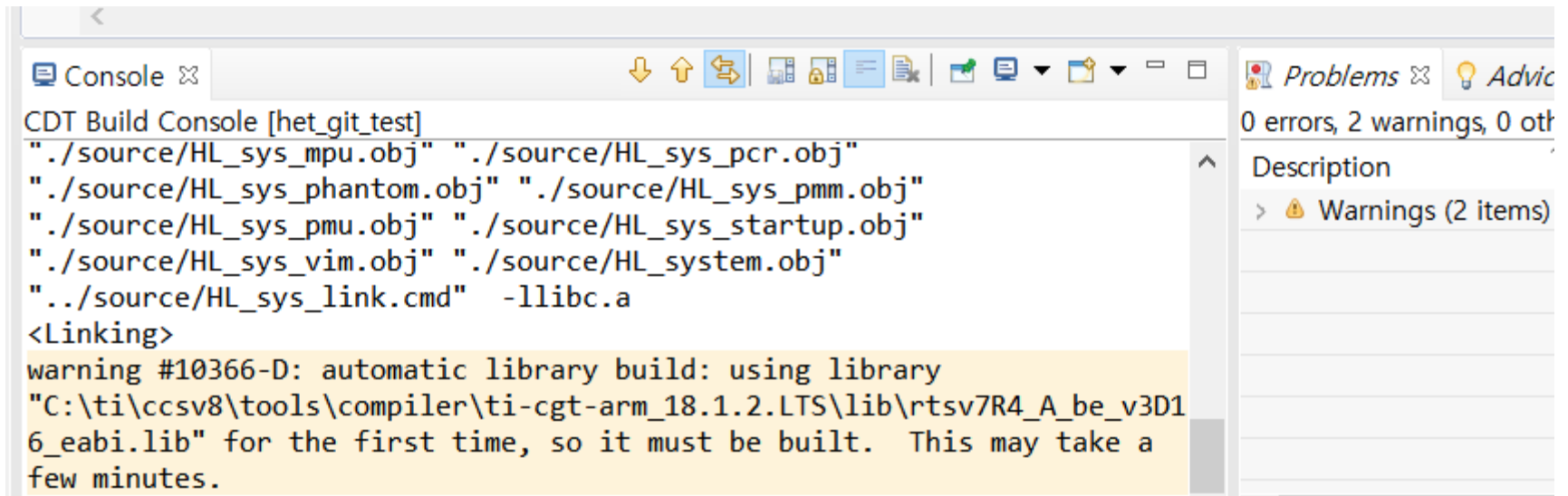
[Show advanced settings](#)

Apply and Close

Cancel

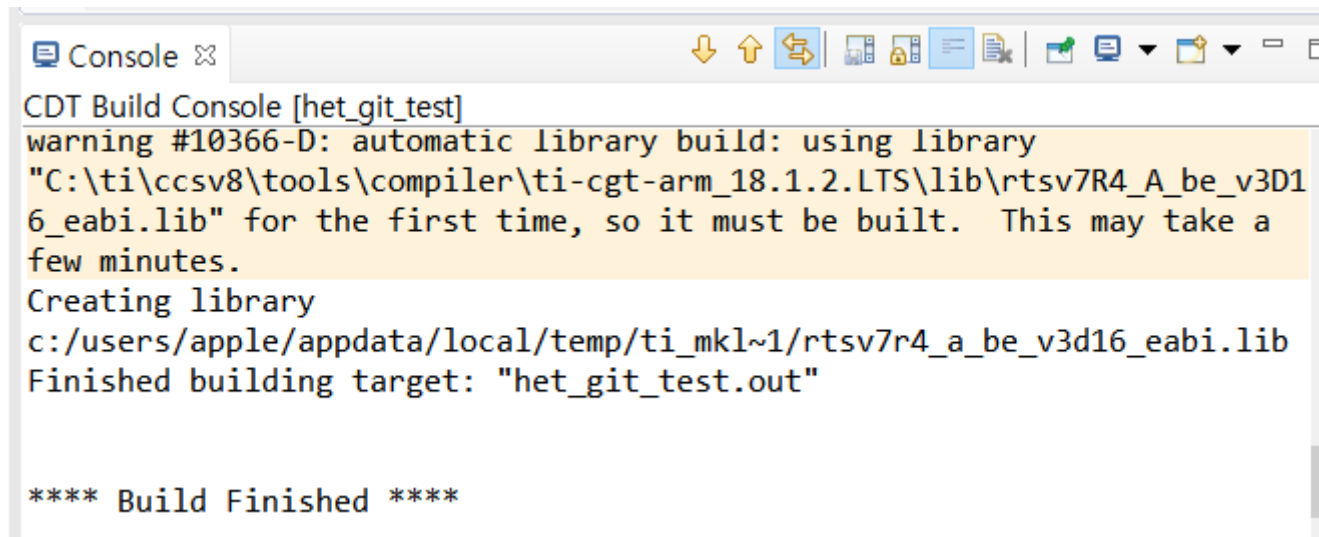


컴파일이 잘 진행되는 것을 확인하도록 한다.



```
CDT Build Console [het_git_test]
"./source/HL_sys_mpu.obj" "./source/HL_sys_pcr.obj"
"./source/HL_sys_phantom.obj" "./source/HL_sys_pmm.obj"
"./source/HL_sys_pmu.obj" "./source/HL_sys_startup.obj"
"./source/HL_sys_vim.obj" "./source/HL_system.obj"
"../source/HL_sys_link.cmd" -llibc.a
<Linking>
warning #10366-D: automatic library build: using library
"C:\ti\ccsv8\tools\compiler\ti-cgt-arm_18.1.2.LTS\lib\rtsv7R4_A_be_v3D1
6_eabi.lib" for the first time, so it must be built. This may take a
few minutes.
```

빌드가 완료되었으면 다시 git bash 를 실행하도록 한다.



```
CDT Build Console [het_git_test]
warning #10366-D: automatic library build: using library
"C:\ti\ccsv8\tools\compiler\ti-cgt-arm_18.1.2.LTS\lib\rtsv7R4_A_be_v3D1
6_eabi.lib" for the first time, so it must be built. This may take a
few minutes.
Creating library
c:/users/apple/appdata/local/temp/ti_mkl~1/rtsv7r4_a_be_v3d16_eabi.lib
Finished building target: "het_git_test.out"

**** Build Finished ****
```



여기서 .out 을 가지고 있는 것이 실행 파일인데 알다시피 용량이 크고 무겁다.  
어쨌든 실행이 잘 되는지 먼저 확인하도록 한다.



```
MINGW64:/c/work/ /mcu_proj/het_git_test/Debug

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/ /mcu_proj/het_git_test (master)
$ ls
Debug/  het_pwm.dil  het_pwm.hcg  include/  source/  targetConfigs/

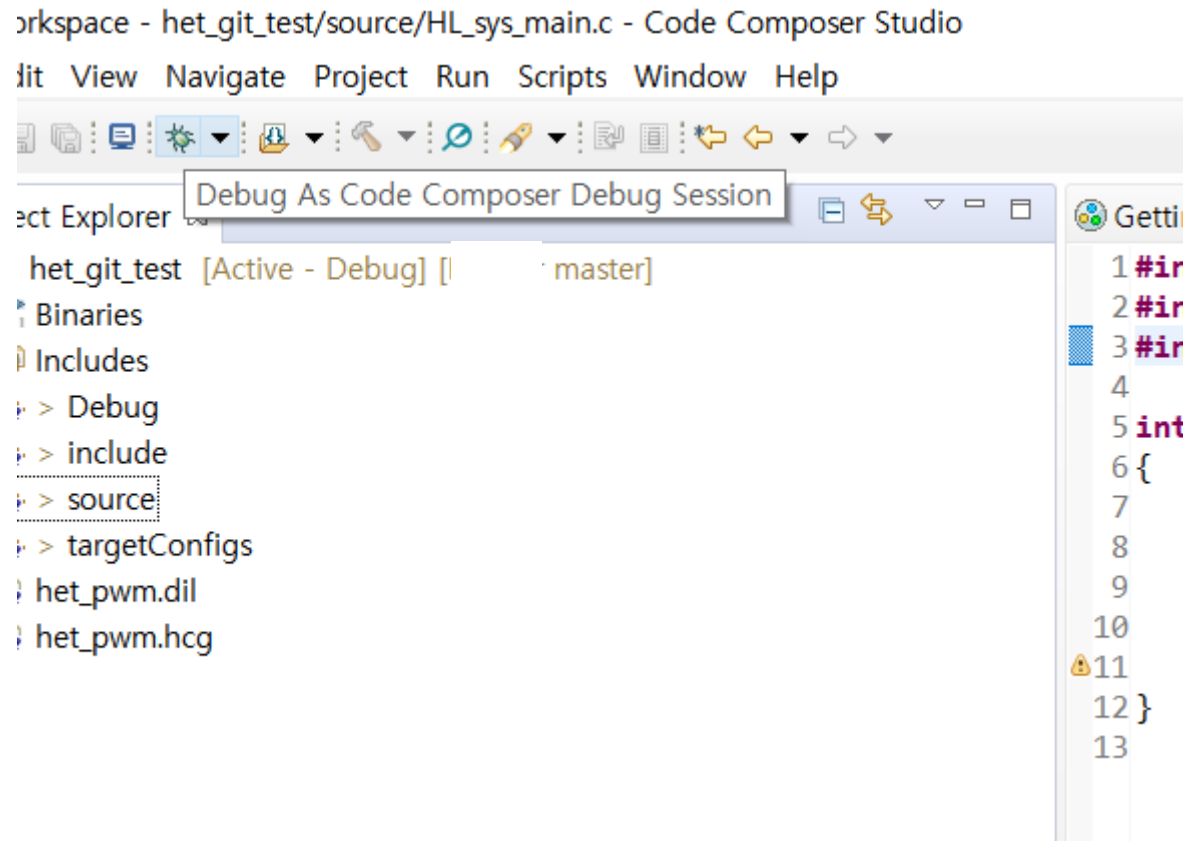
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/ /mcu_proj/het_git_test (master)
$ cd Debug/

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/ /mcu_proj/het_git_test/Debug (master)
$ ls
ccs0bjs.opt      het_git_test.out      makefile    source/
het_git_test.map  het_git_test_linkInfo.xml  objects.mk  sources.mk

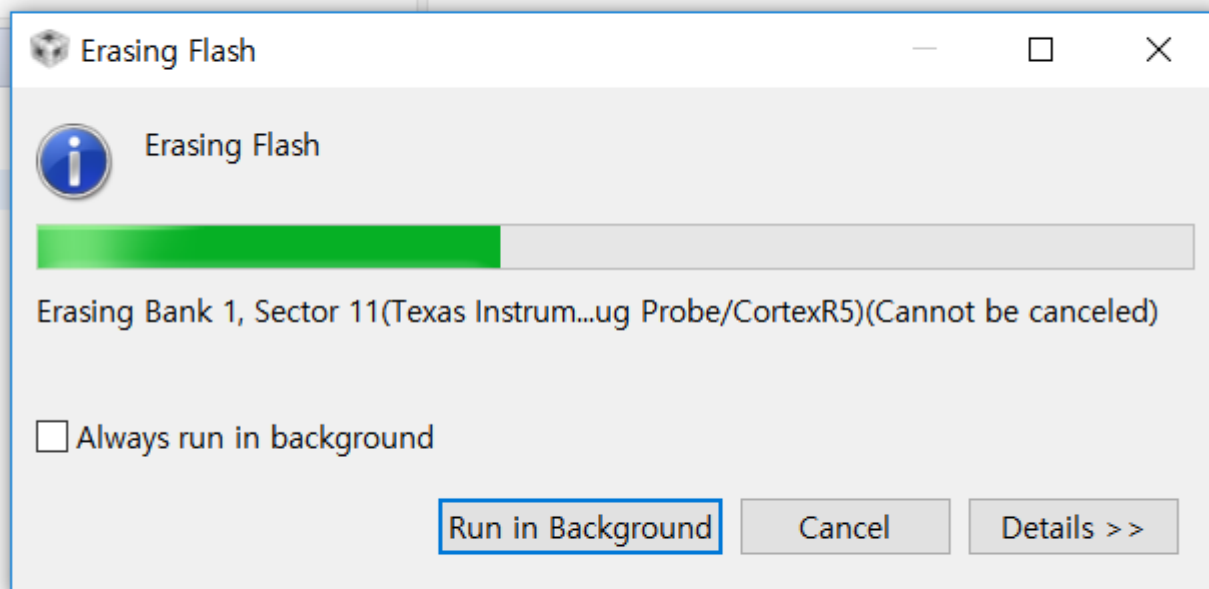
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/ /mcu_proj/het_git_test/Debug (master)
$ file het_git_test.out
het_git_test.out: ELF 32-bit MSB executable, ARM, EABI5 version 1 (SYSV), statically linked, with debug_info, not stripped

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/ /mcu_proj/het_git_test/Debug (master)
$ |
```

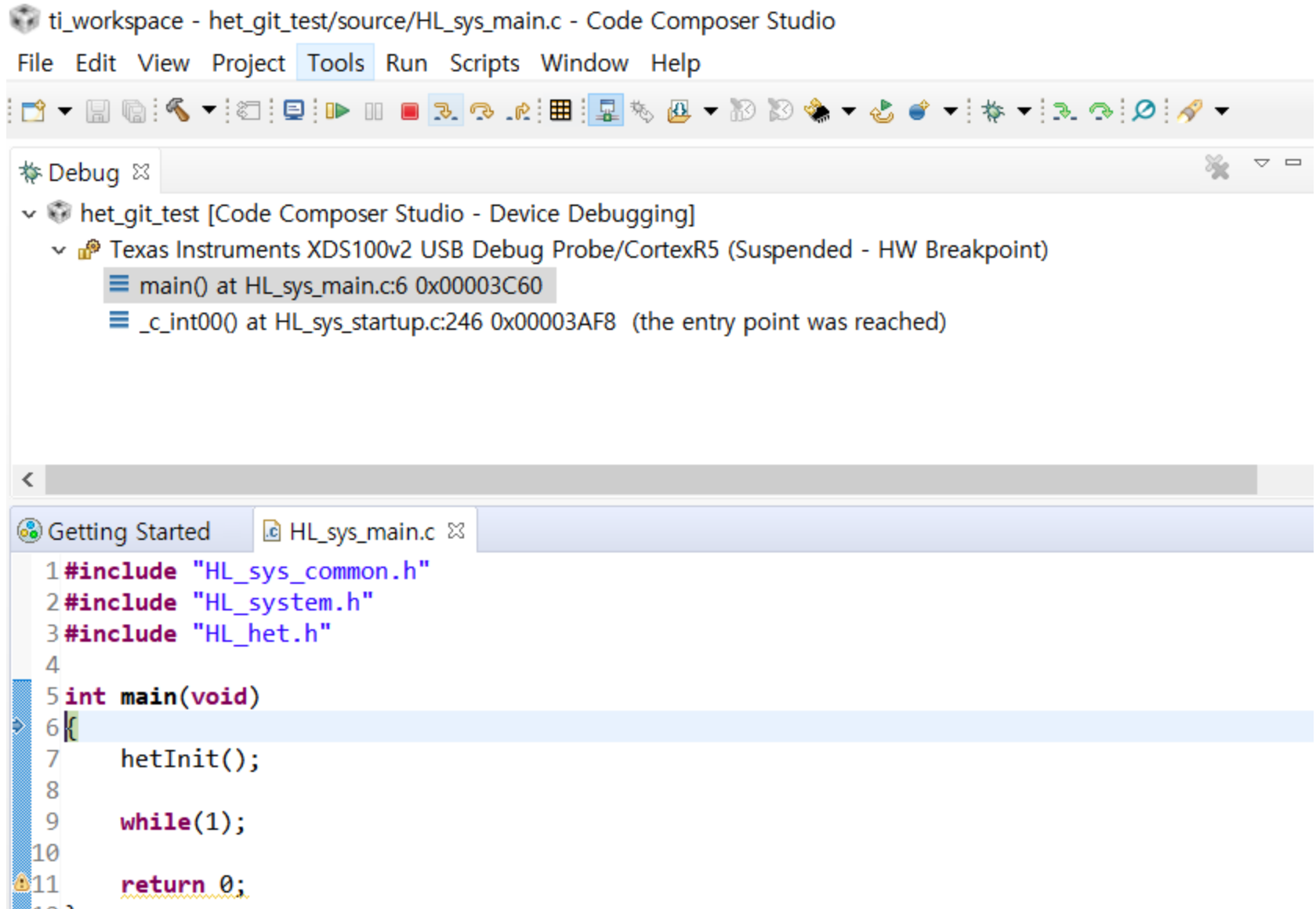
별레 버튼을 누르면 디버깅이 시작된다.



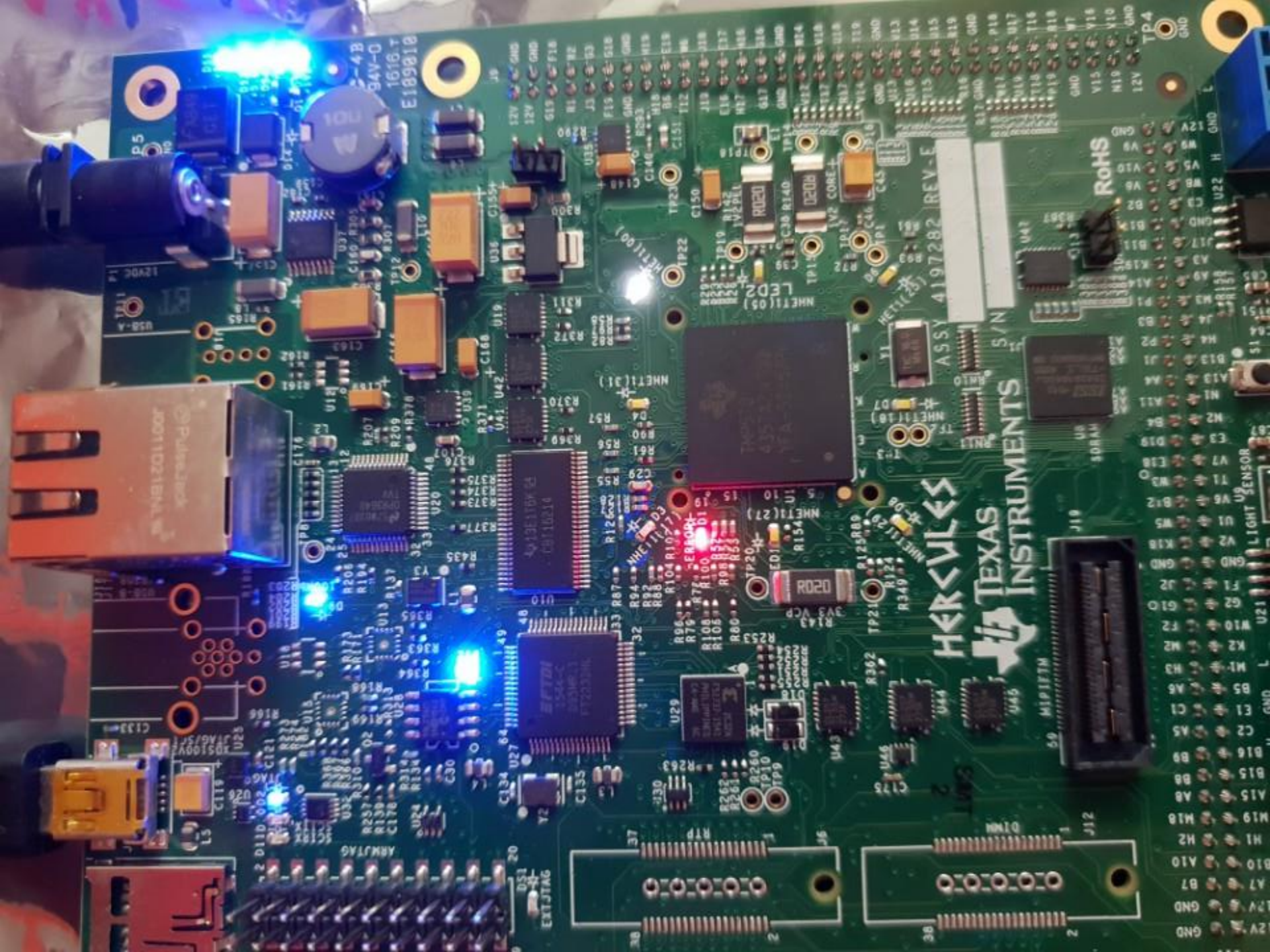
디버깅을 할 때 자동으로 Flash 가 된다.



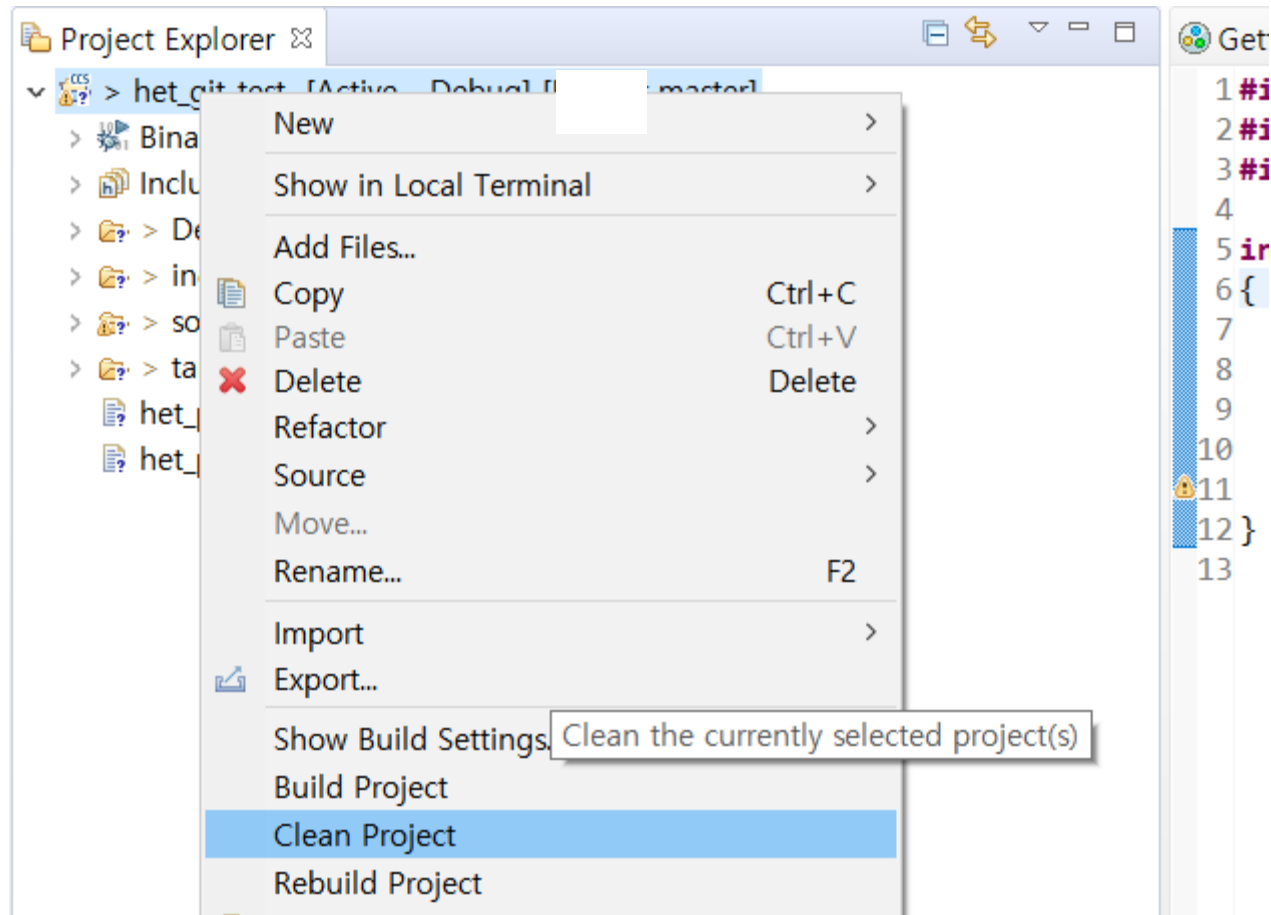
화살표(플레이) 버튼을 누르면 프로그램이 구동되기 시작한다.



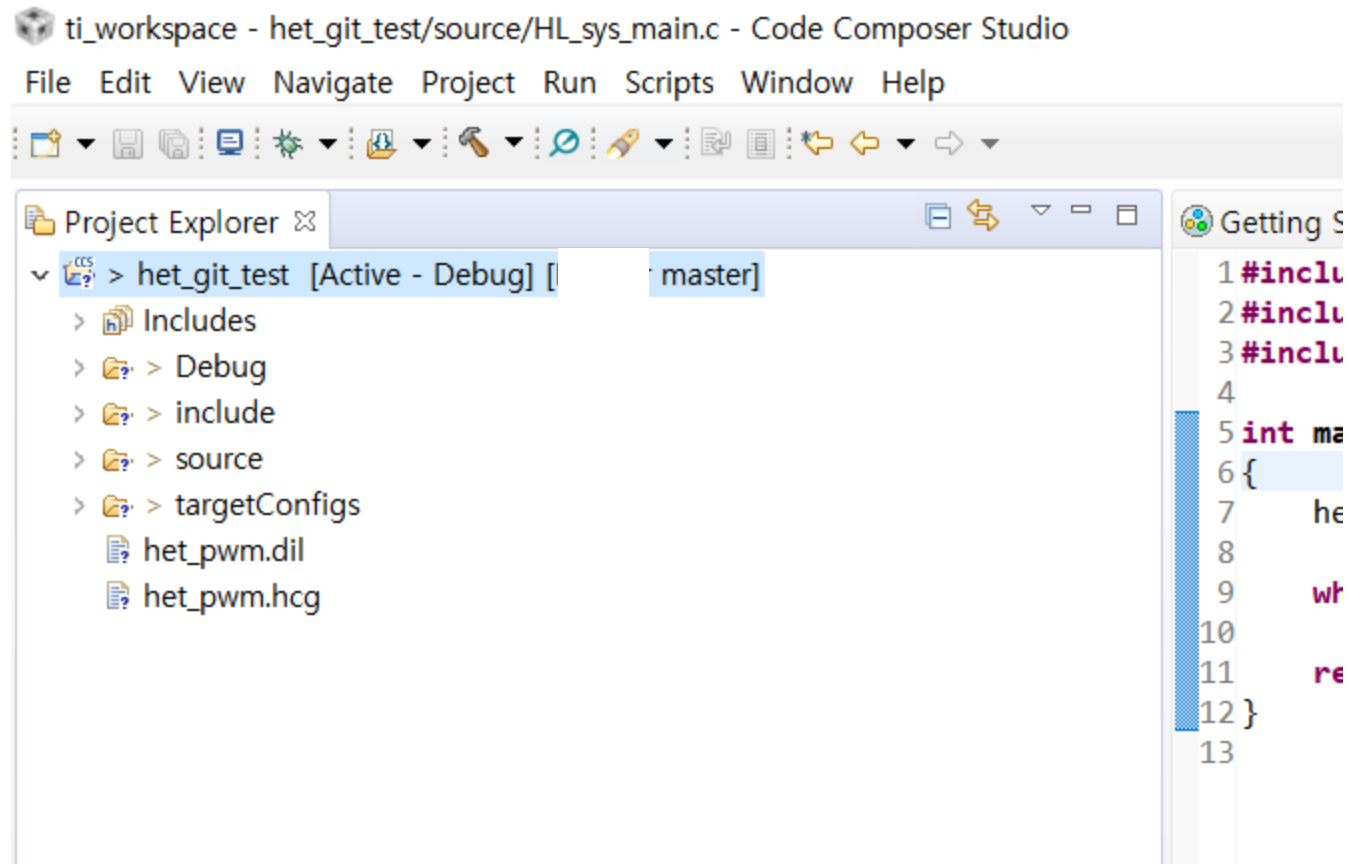




이제 구동을 확인 했으니 Clean Project 를 눌러서 용량이 무거운 것들을 싹 다 지워준다.



아래와 같이 Binary 관련 항목들이 모두 사라진 것을 볼 수 있다.



git status 를 통해 갱신할 것들이 있는지 살펴본다.

```
$ ls
__MPU9250.zip  het_git_test/  prepare.txt

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[REDACTED]/mcu_proj (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   ../fpga_proj/xlnx-4.0/include/uapi/linux/netfilter/xt_CONNMARK.h
        modified:   ../fpga_proj/xlnx-4.0/include/uapi/linux/netfilter/xt_DSCP.h
        modified:   ../fpga_proj/xlnx-4.0/include/uapi/linux/netfilter/xt_MARK.h
        modified:   ../fpga_proj/xlnx-4.0/include/uapi/linux/netfilter/xt_RATEEST.h
        modified:   ../fpga_proj/xlnx-4.0/include/uapi/linux/netfilter/xt_TCPMSS.h
        modified:   ../fpga_proj/xlnx-4.0/include/uapi/linux/netfilter_ipv4/ipt_ECN.h
        modified:   ../fpga_proj/xlnx-4.0/include/uapi/linux/netfilter_ipv4/ipt_TTL.h
        modified:   ../fpga_proj/xlnx-4.0/include/uapi/linux/netfilter_ipv6/ip6t_HL.h
        modified:   ../fpga_proj/xlnx-4.0/net/netfilter/xt_DSCP.c
        modified:   ../fpga_proj/xlnx-4.0/net/netfilter/xt_HL.c
        modified:   ../fpga_proj/xlnx-4.0/net/netfilter/xt_RATEEST.c
        modified:   ../fpga_proj/xlnx-4.0/net/netfilter/xt_TCPMSS.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        het_git_test/

no changes added to commit (use "git add" and/or "git commit -a")
```

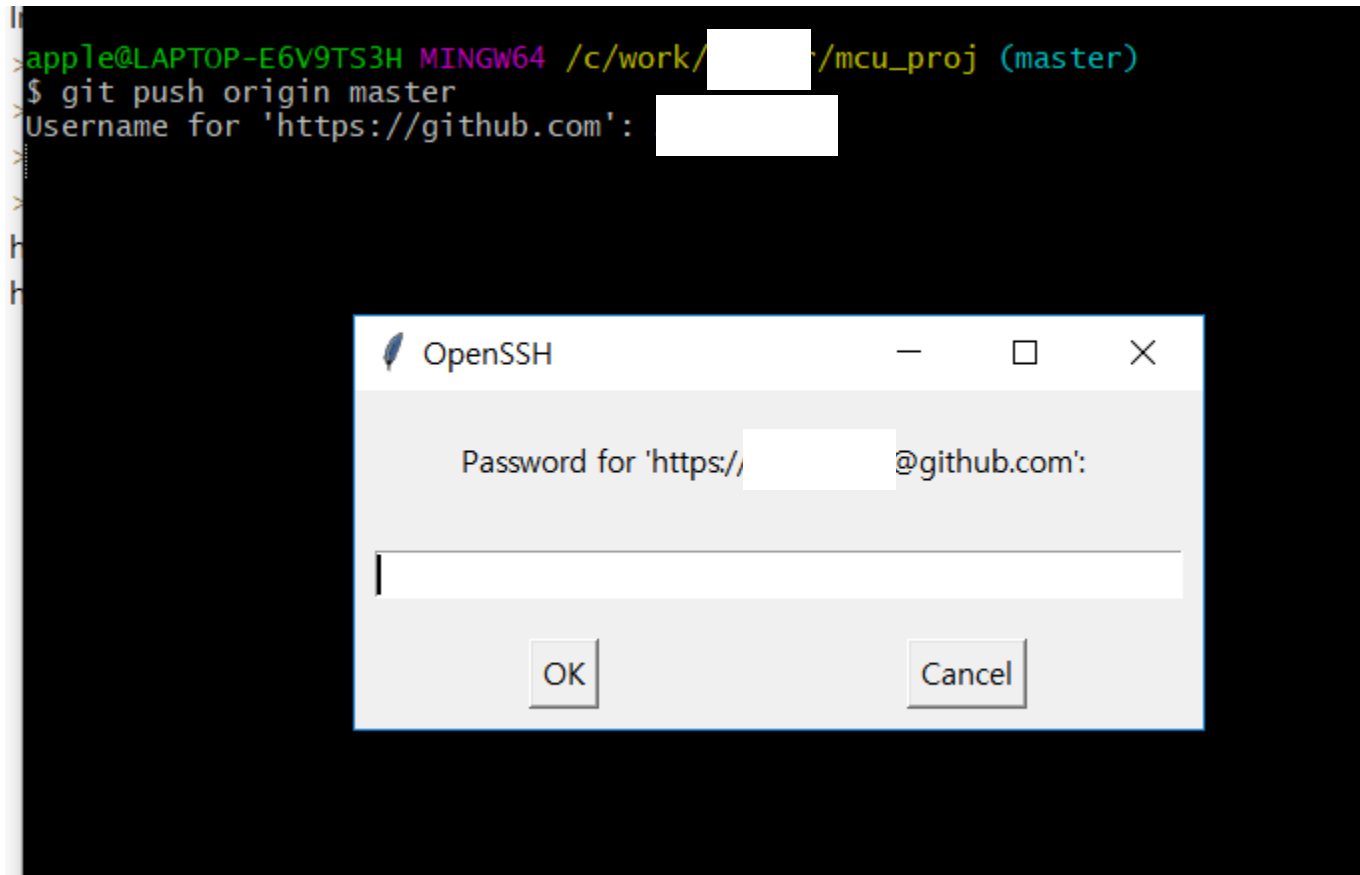


그리고 추가할 녀석을 add 하고 commit 해준다.

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj (master)
$ git add het_git_test
warning: LF will be replaced by CRLF in mcu_proj/het_git_test/.ccsproject.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in mcu_proj/het_git_test/Debug/.gitignore.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in mcu_proj/het_git_test/Debug/makefile.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in mcu_proj/het_git_test/Debug/sources.mk.
The file will have its original line endings in your working directory.

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[redacted]/mcu_proj (master)
$ git commit -am "MCU Proj Commit Test"
[master 3ee3263] MCU Proj Commit Test
126 files changed, 51318 insertions(+)
create mode 100644 mcu_proj/het_git_test/.ccsproject
create mode 100644 mcu_proj/het_git_test/.cproject
create mode 100644 mcu_proj/het_git_test/.launches/het_git_test.launch
create mode 100644 mcu_proj/het_git_test/.project
create mode 100644 mcu_proj/het_git_test/.settings/org.eclipse.cdt.codan.core.prefs
create mode 100644 mcu_proj/het_git_test/.settings/org.eclipse.cdt.debug.core.prefs
create mode 100644 mcu_proj/het_git_test/.settings/org.eclipse.core.resources.prefs
```

비밀 번호 입력하고 OK 한다.



그동안 Update 를 안해 왔다가 오랜만에 업데이트를 하는 것이라면  
엄청나게 많은 분량의 업데이트가 있을 것이므로 역시 잠깐 티 타임을 가지면 된다.

그렇게 오래 걸리지 않을 수도 있다.  
또한 처음만 오래 걸리고 이후부터는 금방 금방 진행 된다는 것도 상기해두자!

```
apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[REDACTED]/mcu_proj (master)
> $ git push origin master
Username for 'https://github.com': [REDACTED]
> Enumerating objects: 136, done.
> Counting objects: 100% (136/136), done.
> Delta compression using up to 4 threads.
> Compressing objects: 100% (132/132), done.
> Writing objects: 100% (134/134), 294.31 KiB | 4.33 MiB/s, done.
> Total 134 (delta 43), reused 0 (delta 0)
remote: Resolving deltas: 100% (43/43), completed with 1 local object.
To https://github.com/KOITT2/[REDACTED].git
  1b51c6d..3ee3263  master -> master

apple@LAPTOP-E6V9TS3H MINGW64 /c/work/[REDACTED]/mcu_proj (master)
$ git status
On branch master
Your branch is up to date with 'origin/master'.
```

이후 fork 한 본인의 repository 에 가보면 업데이트가 적용된 것을 볼 수 있을 것이다.  
추가적으로 mcu\_proj 에 추가한 프로젝트 내용도 같이 올라가 있는 것을 볼 수 있는데  
새로 추가한 내용이니 Pull Request 를 숙제 제출할 때처럼 하면 된다 - 아무걸로나 해도 상관없다.

Branch: master ▾ **New pull request** Create new file Upload files Find file Clone or download ▾

This branch is 3 commits ahead of KOITT2:master. **Pull request** Compare

link180 MCU Proj Test Commit Latest commit 5e62010 16 minutes ago

circuit	Merge pull request #89 from HyunwooParkk/master	4 days ago
cur_present	Merge pull request #99 from ynjw375812/master	15 hours ago
doc	TI AM5728 OpenCL Architecture	12 hours ago
dsp_proj	modify serv	2 days ago
experiment	Ardu on Windows	5 hours ago
fpga_proj	FPGA Linux Kernel	8 days ago
mcu_proj	MCU Proj Test Commit	16 minutes ago
member_profile	Add files via upload	3 days ago
past_present	rename past present & add cur present	a month ago
pcb	Setting Doc	a month ago
real_test	Update 디바이스드라이버구현환경구축.txt	20 days ago
test	Merge remote-tracking branch 'upstream/master'	37 minutes ago
.gitignore	Initial commit	3 months ago
LICENSE	Initial commit	3 months ago
README.md	Update README.md	5 hours ago

# Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare](#)



base fork: KOITT2/

base: master ▼



head fork: link180/

compare: master ▼

✓ **Able to merge.** These branches can be automatically merged.



**Create pull request**

Discuss and review the changes in this comparison with others.

🔗 3 commits

📄 131 files changed

💬 0 commit commen



Commits on Jul 01, 2018



link180

Test



Commits on Aug 11, 2018



link180

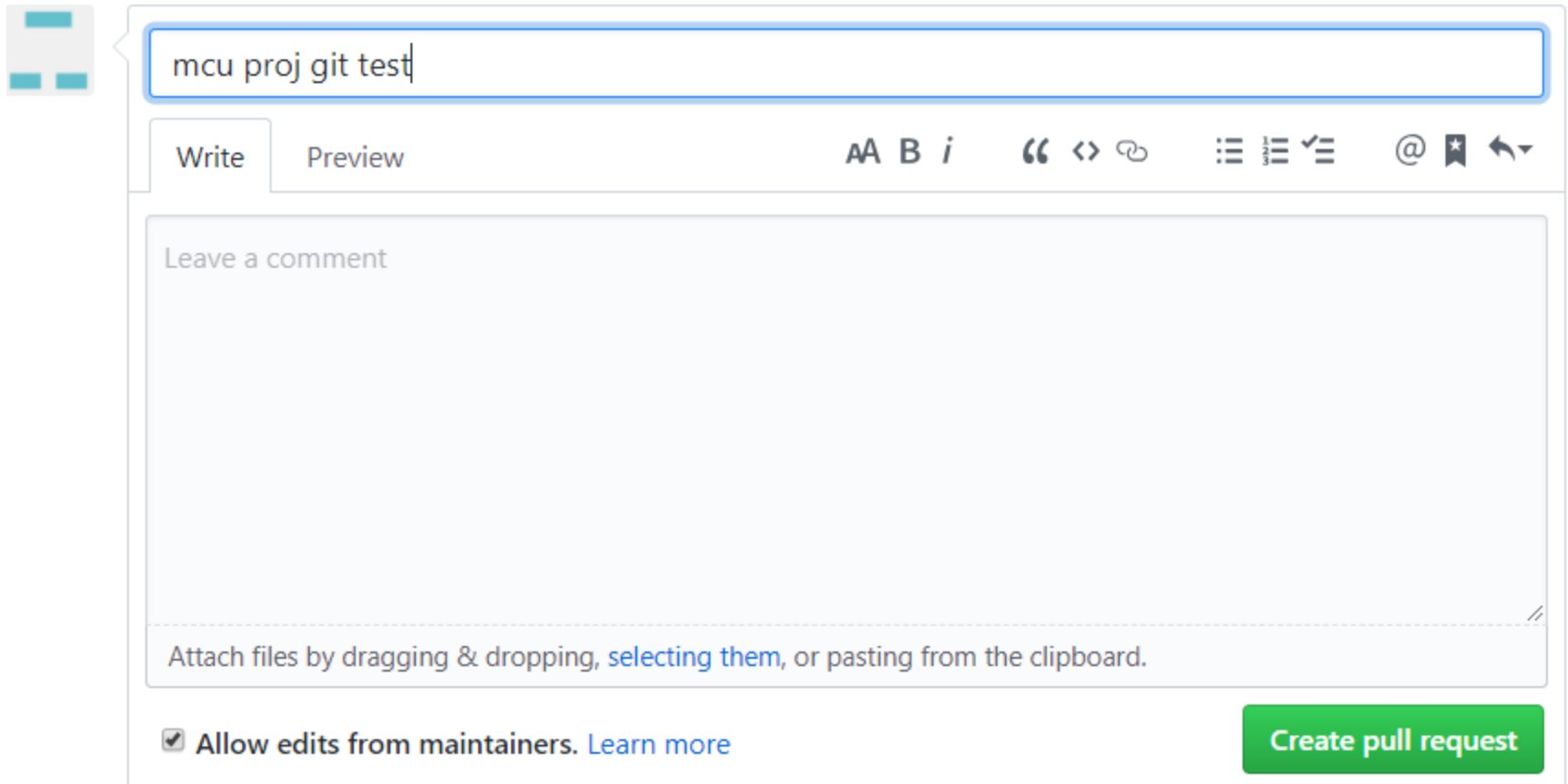
Merge remote-tracking branch 'upstream/master' ...



link180

MCU Proj Test Commit

필자는 test 라 test 라고 적었지만  
실제로 작업 내용을 올릴때는 test 가 아니라 왜 커밋 하는지 이유를 간략히 기술하도록!



The image shows a GitHub pull request comment form. At the top left is a small icon of a computer monitor. Below it is a text input field containing the text "mcu proj git test". Below the input field are two tabs: "Write" (selected) and "Preview". To the right of the tabs is a rich text editor toolbar with icons for bold (AA), italic (i), bulleted list, numbered list, link, unlink, quote, code, and other formatting options. Below the toolbar is a large text area for writing the comment, with the placeholder text "Leave a comment". At the bottom of the text area is a dashed line and the text "Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard." Below the text area is a checkbox labeled "Allow edits from maintainers. [Learn more](#)". To the right of the checkbox is a green button labeled "Create pull request".

mcu proj git test

Write Preview

AA B i “ < > 🔗 ☰ ☷ ✓ @ 📌 ↩

Leave a comment

Attach files by dragging & dropping, [selecting them](#), or pasting from the clipboard.

☒ Allow edits from maintainers. [Learn more](#)

Create pull request

아래와 같이 우리의 본 프로젝트 부분에 Pull Requests 가 들어와 있는 것을 볼 수 있다.

The screenshot shows a GitHub interface for a repository named 'KOITT2'. The 'Pull requests' tab is selected, showing a pull request titled 'mcu proj git test #103'. The pull request is from 'link180:master' to 'KOITT2:master' and contains 3 commits. Below the pull request details, there is a comment from 'link180' stating 'link180 commented just now' with 'No description provided.' Below the comment, a commit history is shown for 'link180' on '1 Jul', listing three commits: 'Test', 'Merge remote-tracking branch 'upstream/master'', and 'MCU Proj Test Commit'.

KOITT2 /

<> Code 0 Issues 0 Pull requests 1 Projects 0 Wiki

## mcu proj git test #103

Open link180 wants to merge 3 commits into KOITT2:master from link180:master

Conversation 0 Commits 3 Checks 0 Files changed 131

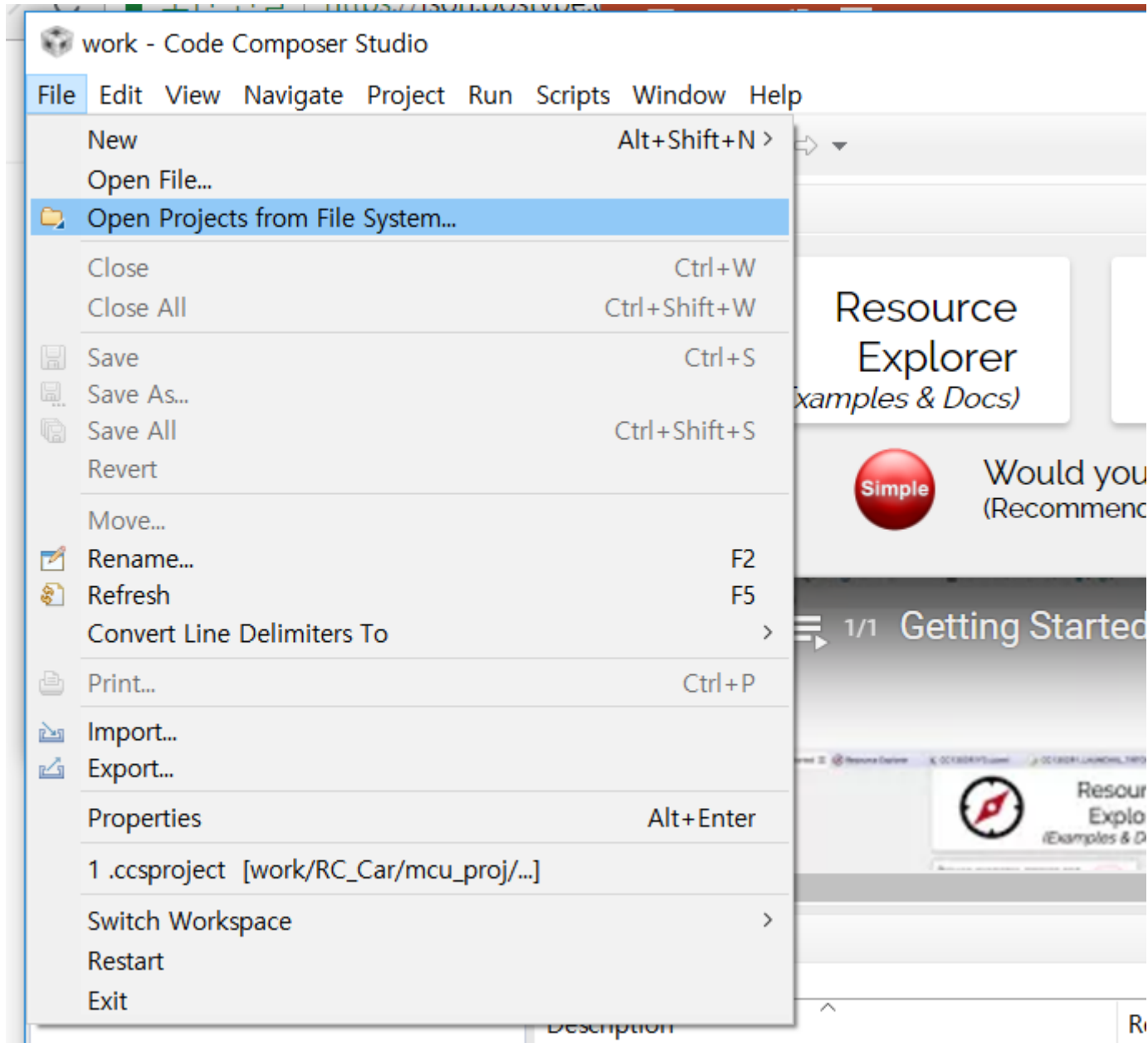
link180 commented just now

*No description provided.*

link180 added some commits on 1 Jul

- Test
- Merge remote-tracking branch 'upstream/master' ...
- MCU Proj Test Commit

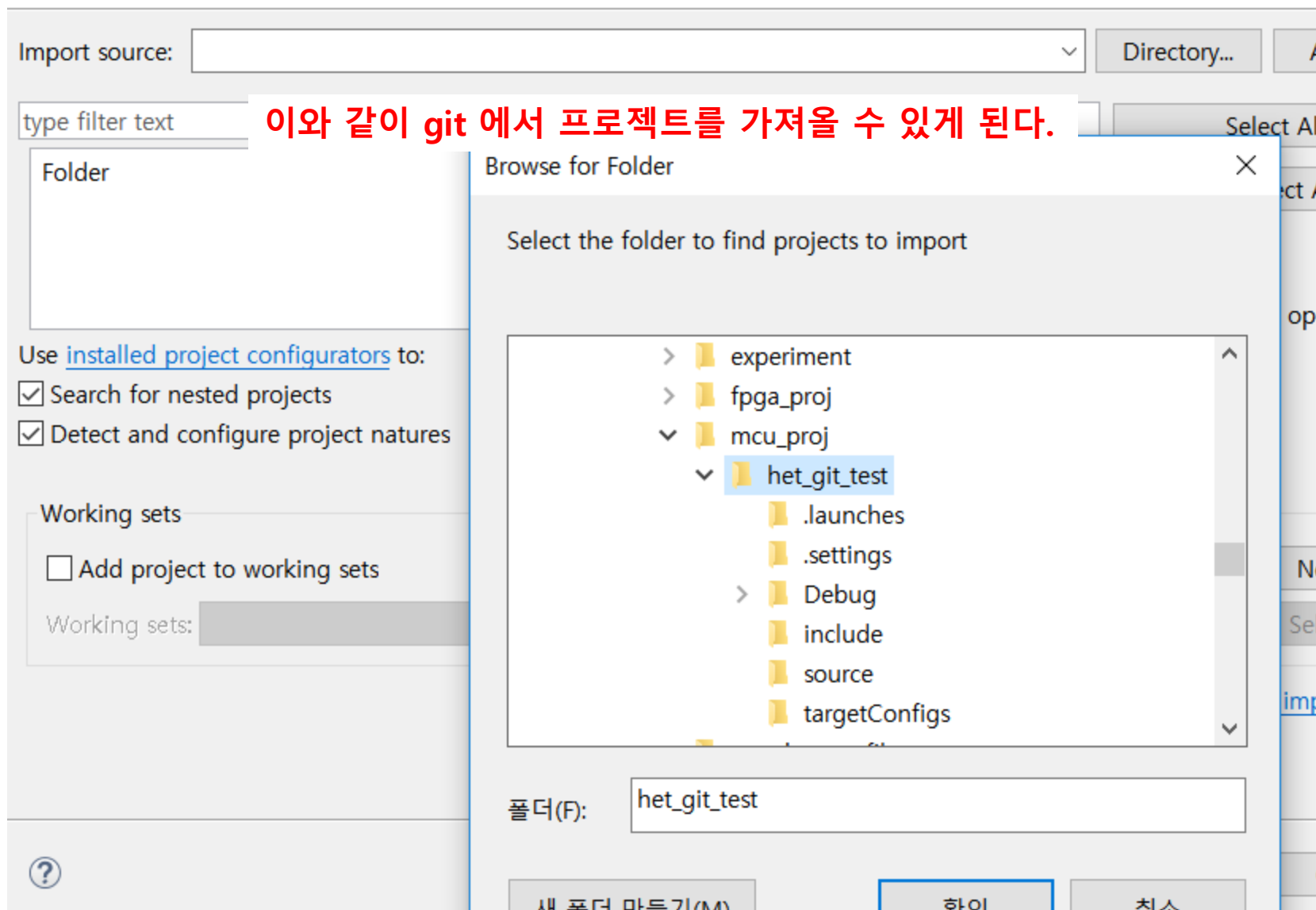
이후에 아래와 같이 해보도록 한다.



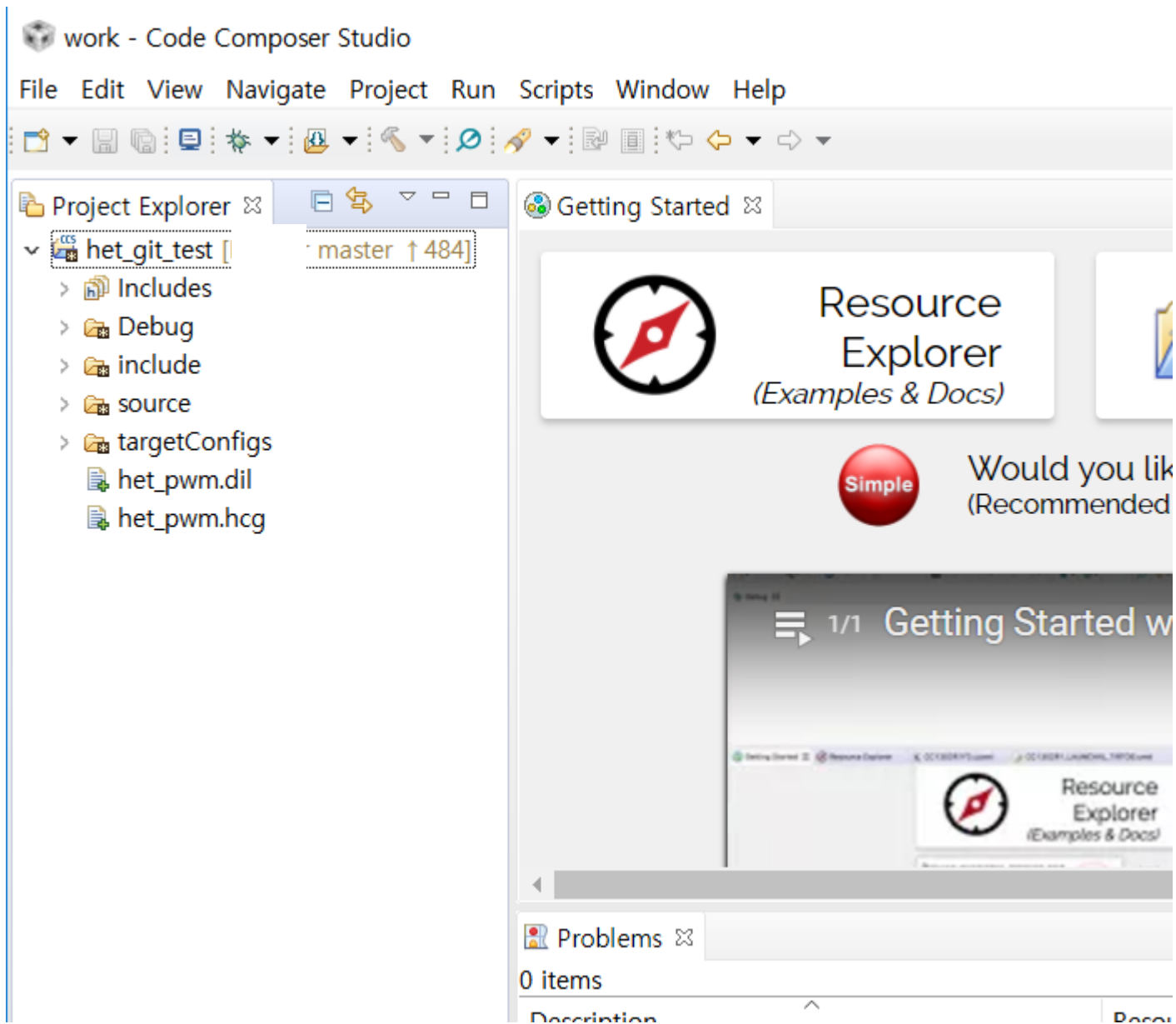


## Import Projects from File System or Archive

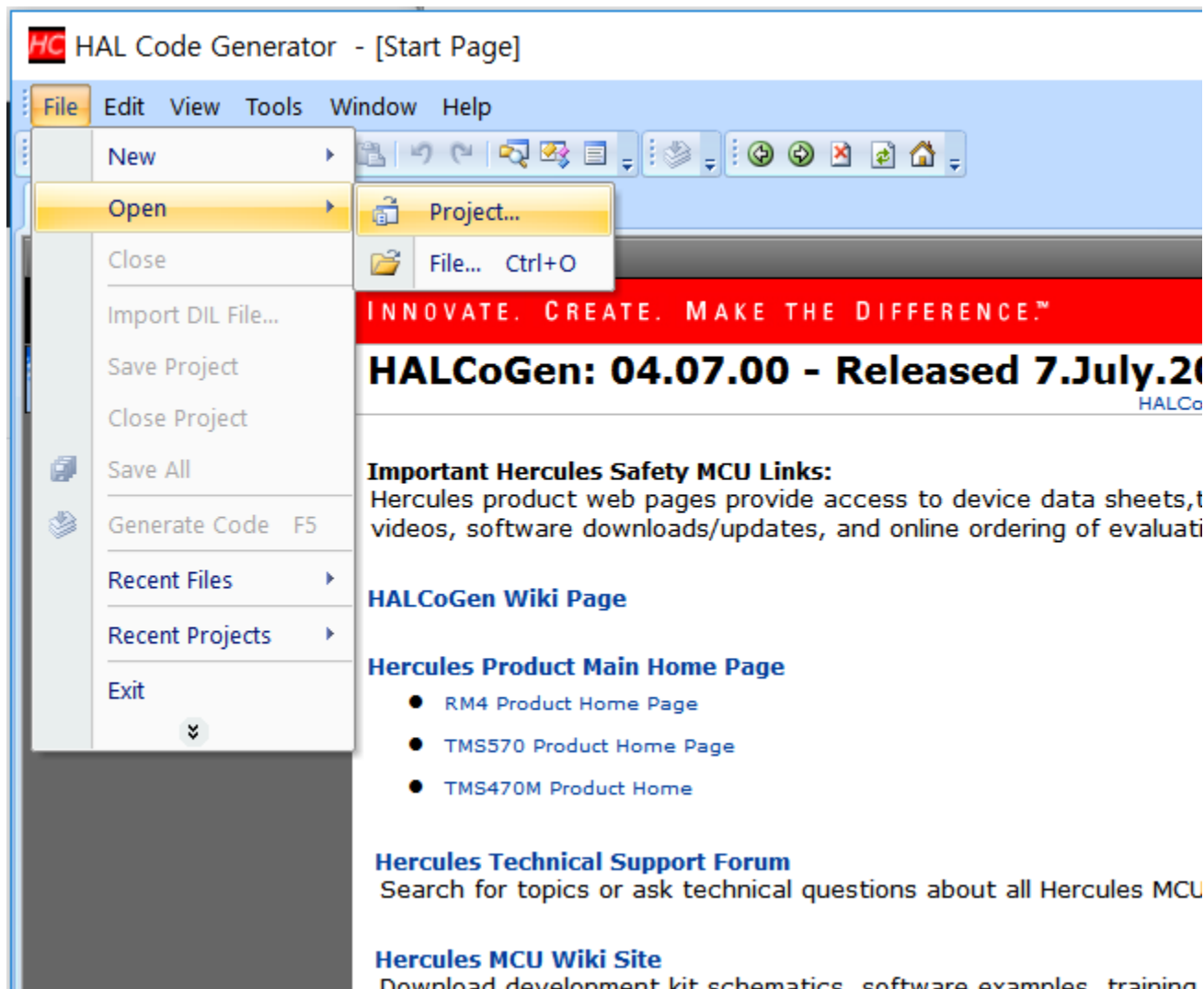
This wizard analyzes the content of your folder or archive file to find projects and import them in the IDE.



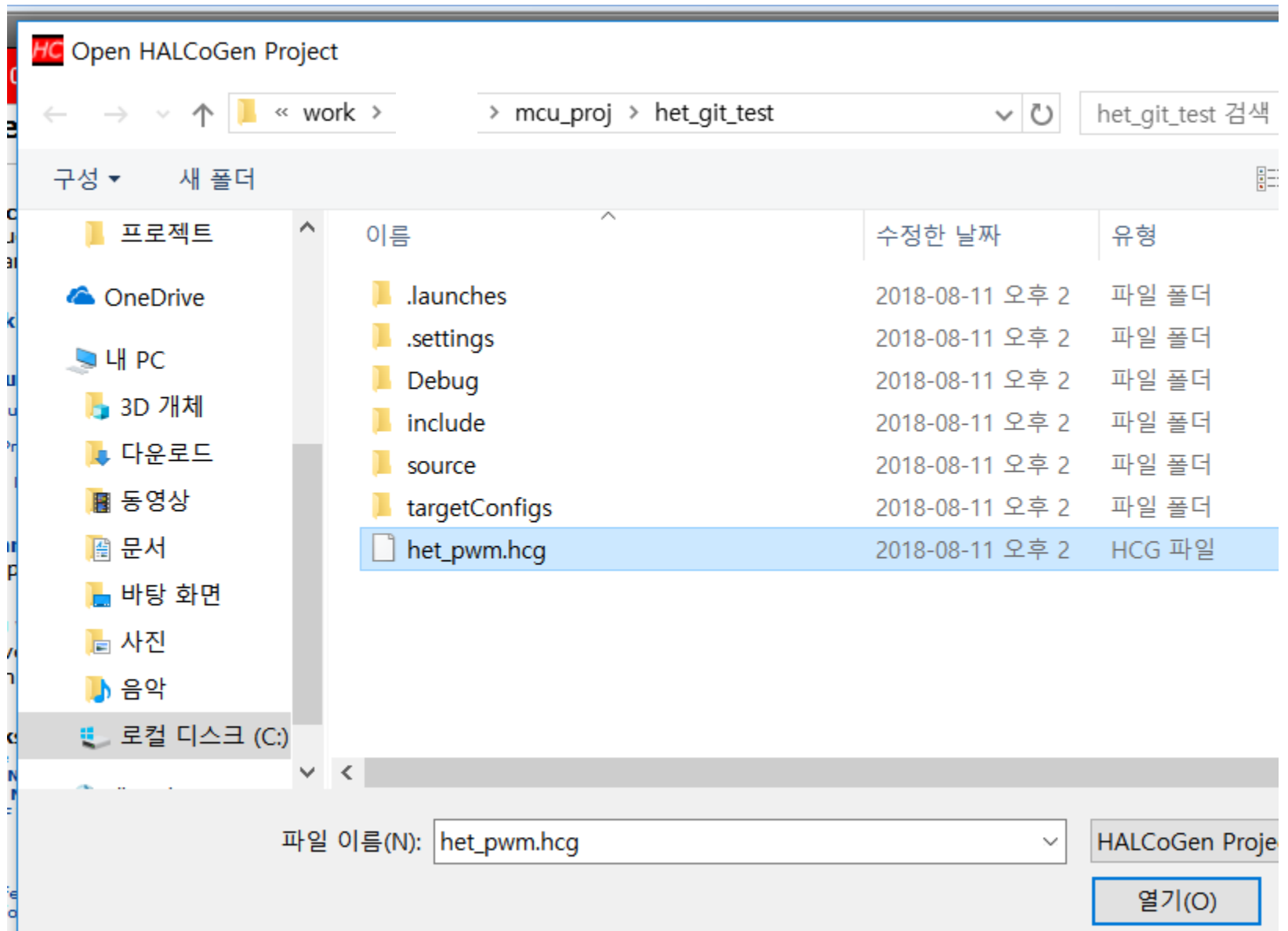
프로젝트가 잘 불러왔음을 확인 할 수 있다.



유사한 것을 HalCoGen 에서도 할 수 있다.



git repository 위치에 가서 mcu\_proj 에 het\_git\_test 에 가보면 het\_pwm.hcg 가 있는데 이를 열면 HalCoGen 구성을 고스란히 볼 수 있다.



아래와 같이 잘 Load 되는 것을 볼 수 있다.

HC HAL Code Generator - C:\work\W\mcu\_proj\Whet\_git\_test\Whet\_pwm.hcg - [TMS570LC4357ZWT]

File Edit View Tools Window Help

TMS570LC4357ZWT PINMUX RTI GIO ESM SCI1 SCI2 SCI3 SCI4 LIN1 LIN2 MIBSPI1 MIBSPI2

General Driver Enable R5-MPU-PMU Interrupts VIM General VIM RAM VIM Channel 0-31 VIM Channel 32-63

TMS570LC4357ZWT Block Diagram

The block diagram illustrates the internal architecture of the TMS570LC4357ZWT microcontroller. It features a central '2x Cortex-R5F' block. To its right are 'DMA', 'RTP', and 'HTU1'. Below these are 'EMAC', 'DMM', and 'HTU2'. Further right are 'Flash', 'EMIF', 'MPU', 'RTI', and 'EPC'. At the bottom, there are 'RAM', 'POM', 'CRC', 'DCC', and 'PINMUX'. The components are interconnected via a system bus, represented by thick black lines.

Output

```
Loading: FEE: 'FEEv000.xml'
Loading: AJSM: 'AJSMv000.xml'
Load complete
```