Real-Time Operating System (Day 1 Lab)

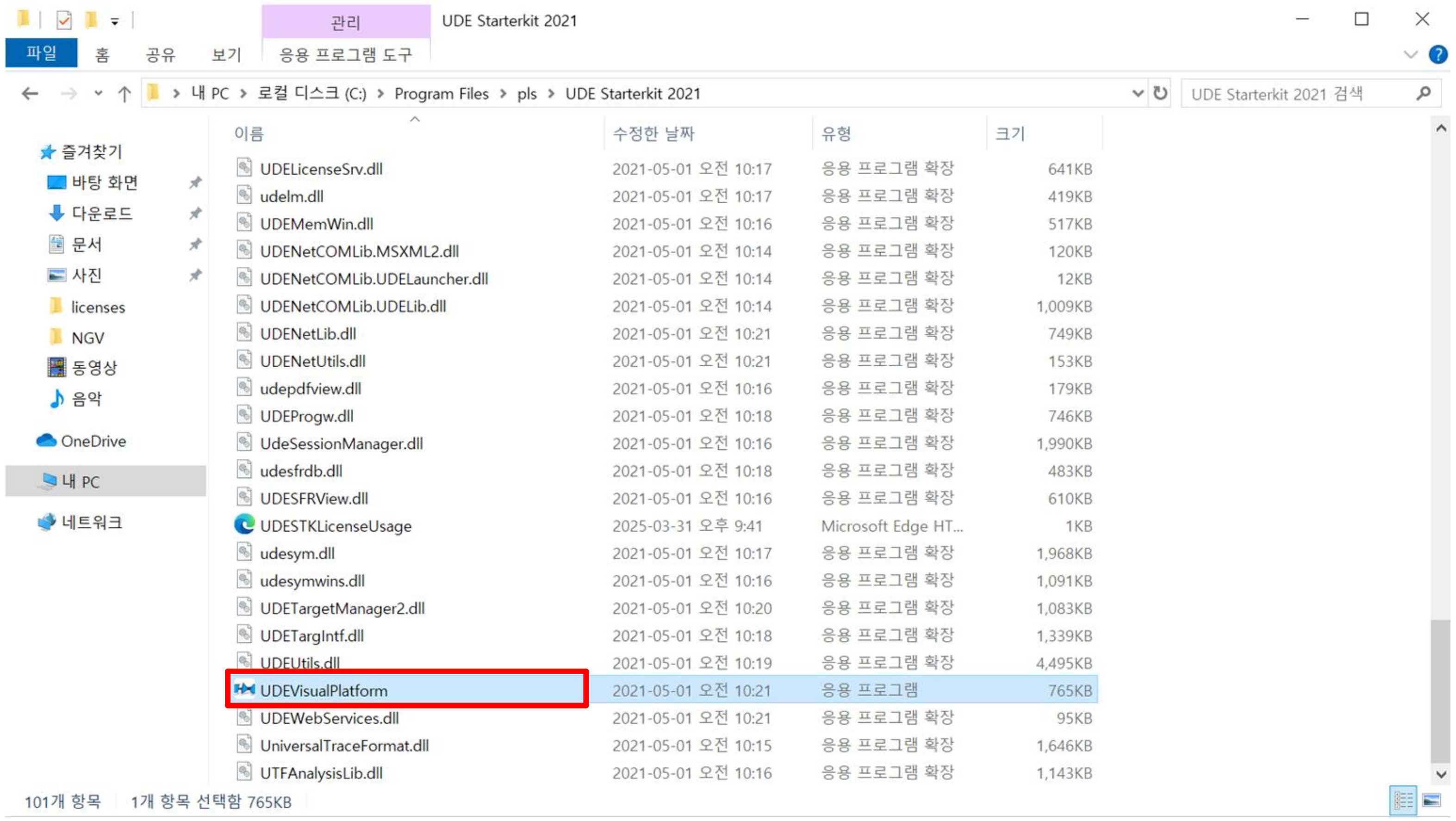
Jong-Chan Kim

Graduate School of Automotive Engineering

KMU 국민대학교 KOOKMIN UNIVERSITY

UDE 디버거 실행

- C:□Program Files□pls□UDE Starterkit 2021 경로에 있는 UDEVisualPlatform 바로가기 생성 후 실행

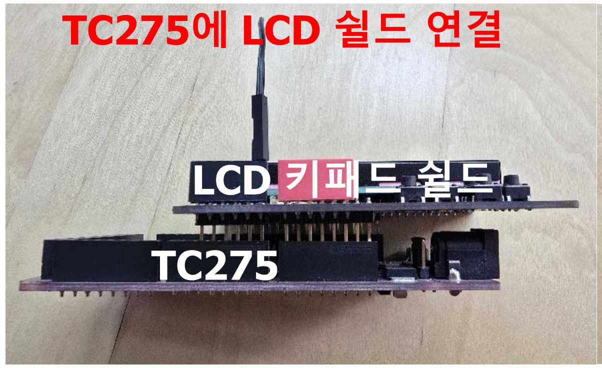


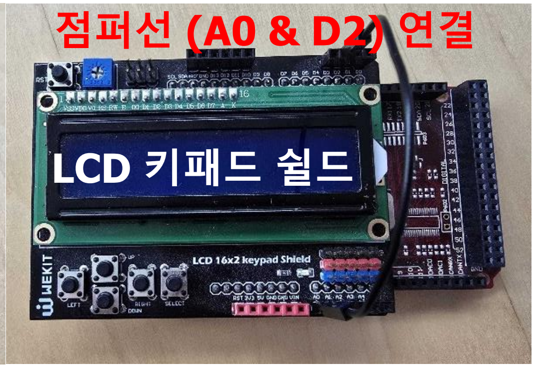
이름	수정한 날짜	유형	크기
UDELicenseSrv.dll	2021-05-01 오전 10:17	응용 프로그램 확장	641KB
udelm.dll	2021-05-01 오전 10:17	응용 프로그램 확장	419KB
UDEMemWin.dll	2021-05-01 오전 10:16	응용 프로그램 확장	517KB
UDENetCOMLib.MSXML2.dll	2021-05-01 오전 10:14	응용 프로그램 확장	120KB
UDENetCOMLib.UDELauncher.dll	2021-05-01 오전 10:14	응용 프로그램 확장	12KB
UDENetCOMLib.UDELib.dll	2021-05-01 오전 10:14	응용 프로그램 확장	1,009KB
UDENetLib.dll	2021-05-01 오전 10:21	응용 프로그램 확장	749KB
UDENetUtils.dll	2021-05-01 오전 10:21	응용 프로그램 확장	153KB
udepdfview.dll	2021-05-01 오전 10:16	응용 프로그램 확장	179KB
UDEProgw.dll	2021-05-01 오전 10:18	응용 프로그램 확장	746KB
UdeSessionManager.dll	2021-05-01 오전 10:16	응용 프로그램 확장	1,990KB
udesfrdb.dll	2021-05-01 오전 10:18	응용 프로그램 확장	483KB
UDESFRView.dll	2021-05-01 오전 10:16	응용 프로그램 확장	610KB
UDESTKLicenseUsage	2025-03-31 오후 9:41	Microsoft Edge HT...	1KB
udesym.dll	2021-05-01 오전 10:17	응용 프로그램 확장	1,968KB
udesymwins.dll	2021-05-01 오전 10:16	응용 프로그램 확장	1,091KB
UDETargetManager2.dll	2021-05-01 오전 10:20	응용 프로그램 확장	1,083KB
UDETargIntf.dll	2021-05-01 오전 10:18	응용 프로그램 확장	1,339KB
UDEUtils.dll	2021-05-01 오전 10:19	응용 프로그램 확장	4,495KB
UDEVisualPlatform	2021-05-01 오전 10:21	응용 프로그램	765KB
UDEWebServices.dll	2021-05-01 오전 10:21	응용 프로그램 확장	95KB
UniversalTraceFormat.dll	2021-05-01 오전 10:15	응용 프로그램 확장	1,646KB
UTFAnalysisLib.dll	2021-05-01 오전 10:16	응용 프로그램 확장	1,143KB

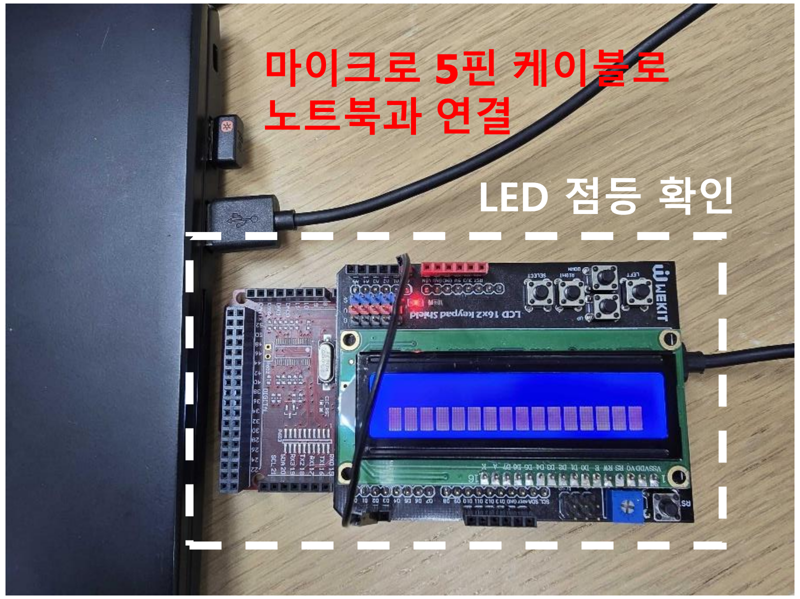
101개 항목 1개 항목 선택함 765KB

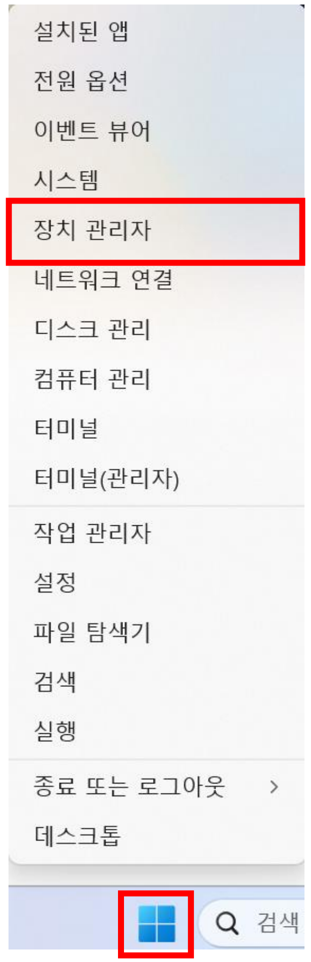
프로젝트 빌드

- TC275 연결 후 포트 번호 확인(장치 관리자 > 포트)

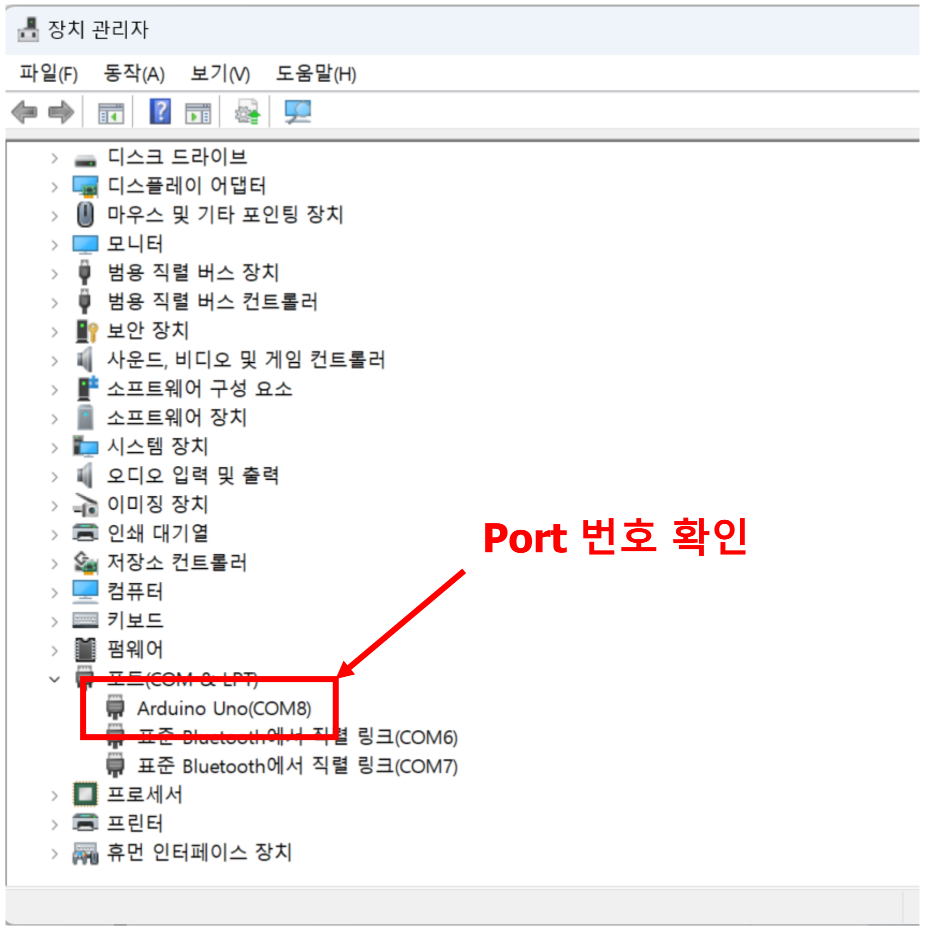








우클릭



장치 관리자

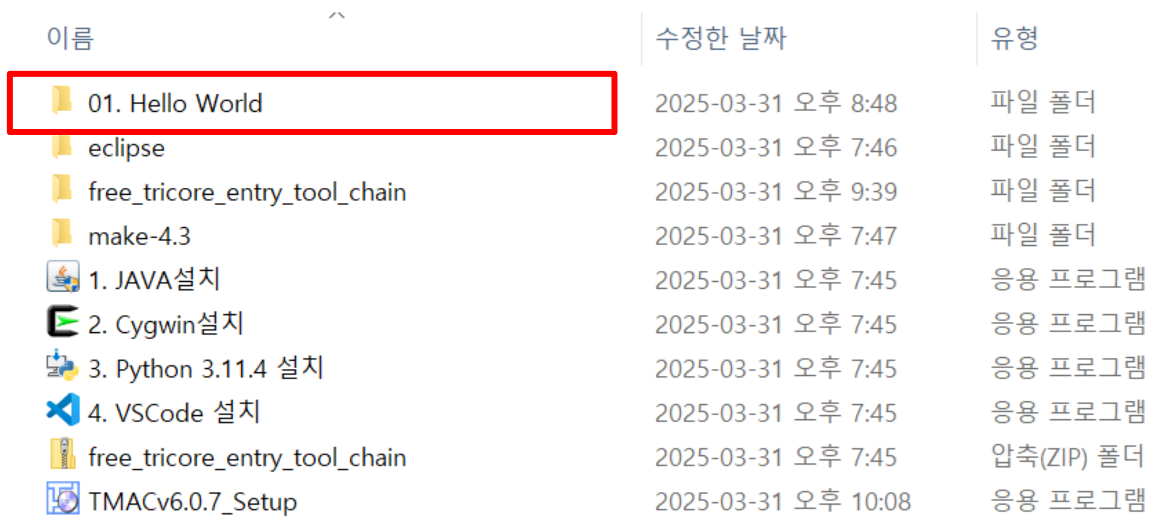
파일(F) 동작(A) 보기(V) 도움말(H)

- 디스크 드라이브
- 디스플레이 어댑터
- 마우스 및 기타 포인팅 장치
- 모니터
- 범용 직렬 버스 장치
- 범용 직렬 버스 컨트롤러
- 보안 장치
- 사운드, 비디오 및 게임 컨트롤러
- 소프트웨어 구성 요소
- 소프트웨어 장치
- 시스템 장치
- 오디오 입력 및 출력
- 이미징 장치
- 인쇄 대기열
- 저장소 컨트롤러
- 컴퓨터
- 키보드
- 펌웨어
- 포트(COM & LPT)
  - Arduino Uno(COM8)
  - 표준 Bluetooth에서 직렬 링크(COM6)
  - 표준 Bluetooth에서 직렬 링크(COM7)
- 프로세서
- 프린터
- 휴먼 인터페이스 장치

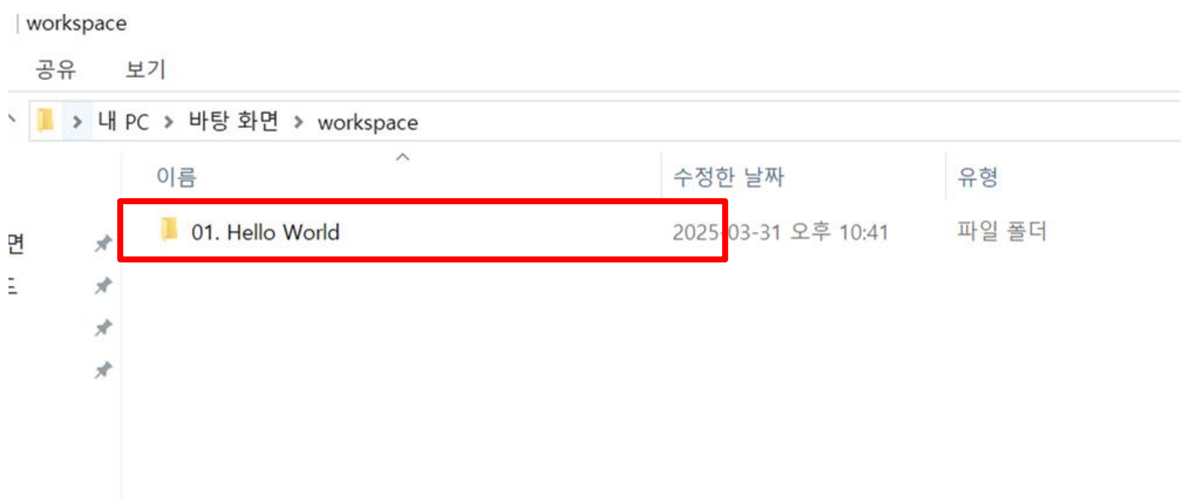
Port 번호 확인

프로젝트 빌드

- NGV 폴더 안의 '01. Hello World' 폴더 복사
- 바탕화면에 workspace 생성 후 붙여넣기



이름	수정한 날짜	유형
01. Hello World	2025-03-31 오후 8:48	파일 폴더
eclipse	2025-03-31 오후 7:46	파일 폴더
free_tricore_entry_tool_chain	2025-03-31 오후 9:39	파일 폴더
make-4.3	2025-03-31 오후 7:47	파일 폴더
1. JAVA설치	2025-03-31 오후 7:45	응용 프로그램
2. Cygwin설치	2025-03-31 오후 7:45	응용 프로그램
3. Python 3.11.4 설치	2025-03-31 오후 7:45	응용 프로그램
4. VSCode 설치	2025-03-31 오후 7:45	응용 프로그램
free_tricore_entry_tool_chain	2025-03-31 오후 7:45	압축(ZIP) 폴더
TMACv6.0.7_Setup	2025-03-31 오후 10:08	응용 프로그램



workspace

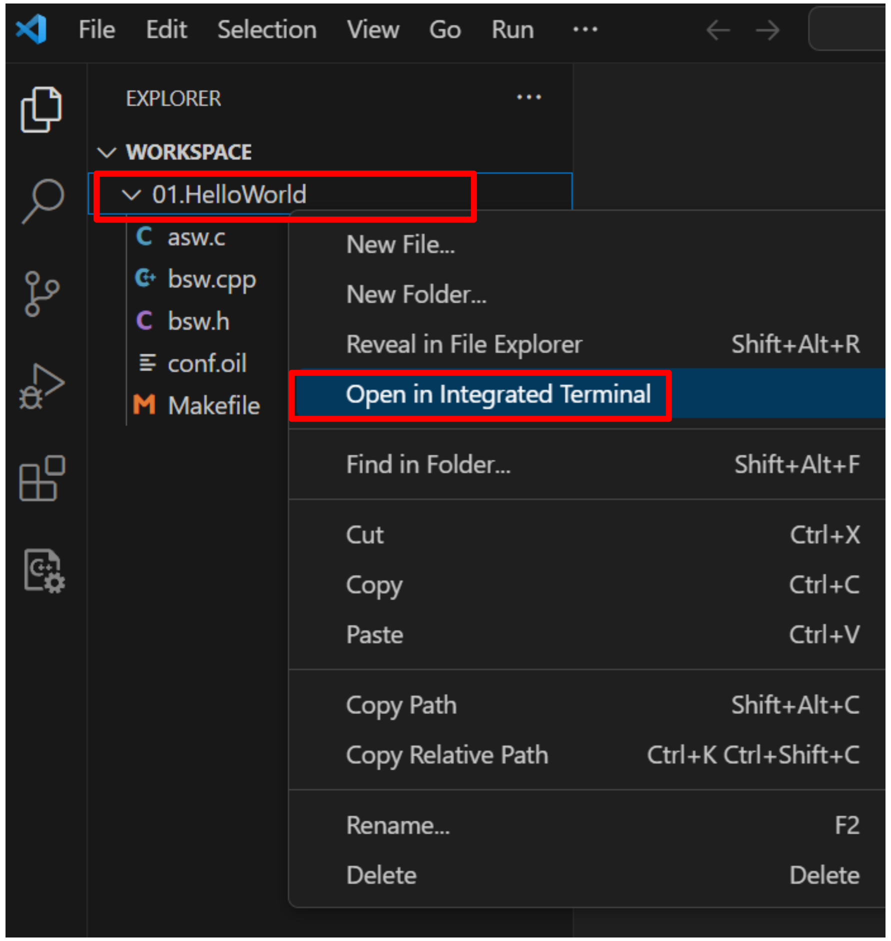
공유 보기

내 PC > 바탕 화면 > workspace

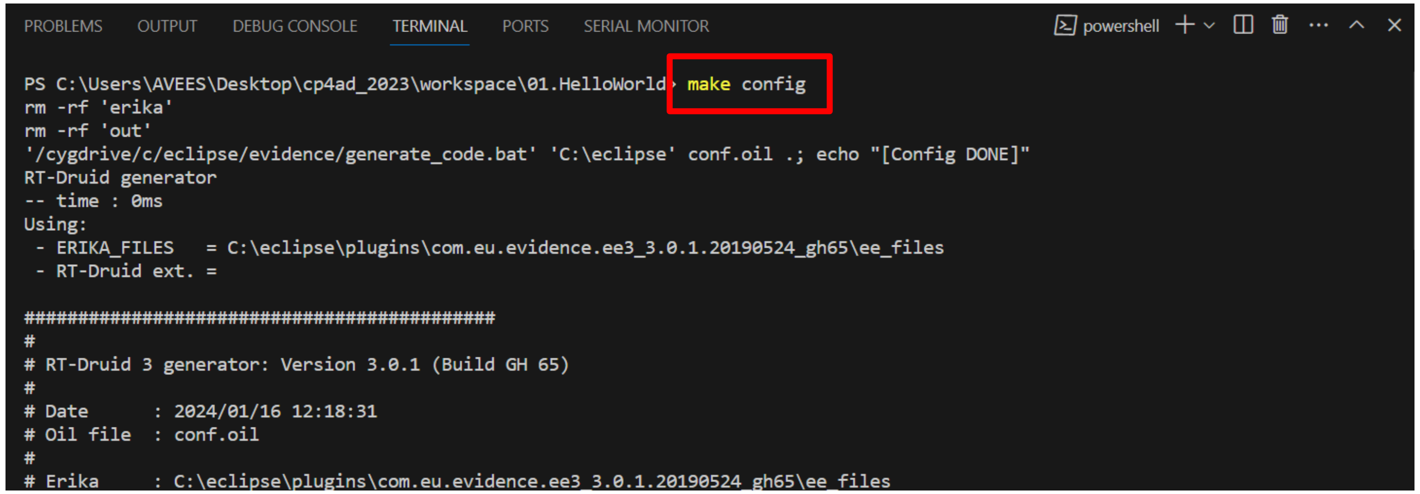
이름	수정한 날짜	유형
01. Hello World	2025-03-31 오후 10:41	파일 폴더

프로젝트 빌드

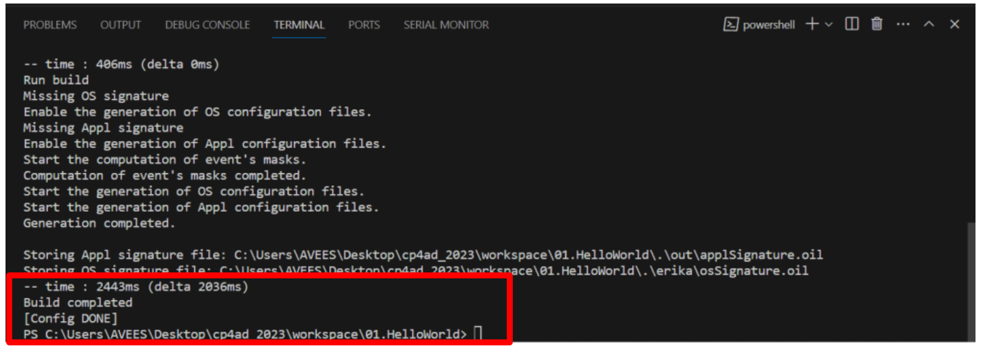
- '01.Hello World' 폴더 우클릭 후
  'Open in Integrated Terminal' 실행



- Terminal에서 make config 실행



```
PS C:\Users\AVEES\Desktop\cp4ad_2023\workspace\01.HelloWorld> make config
rm -rf 'erika'
rm -rf 'out'
'/cygdrive/c/eclipse/evidence/generate_code.bat' 'C:\eclipse' conf.oil .; echo "[Config DONE]"
RT-Druid generator
-- time : 0ms
Using:
 - ERIKA_FILES   = C:\eclipse\plugins\com.eu.evidence.ee3_3.0.1.20190524_gh65\ee_files
 - RT-Druid ext. =

############################################
#
# RT-Druid 3 generator: Version 3.0.1 (Build GH 65)
#
# Date      : 2024/01/16 12:18:31
# Oil file  : conf.oil
#
# Erika     : C:\eclipse\plugins\com.eu.evidence.ee3_3.0.1.20190524_gh65\ee_files
```

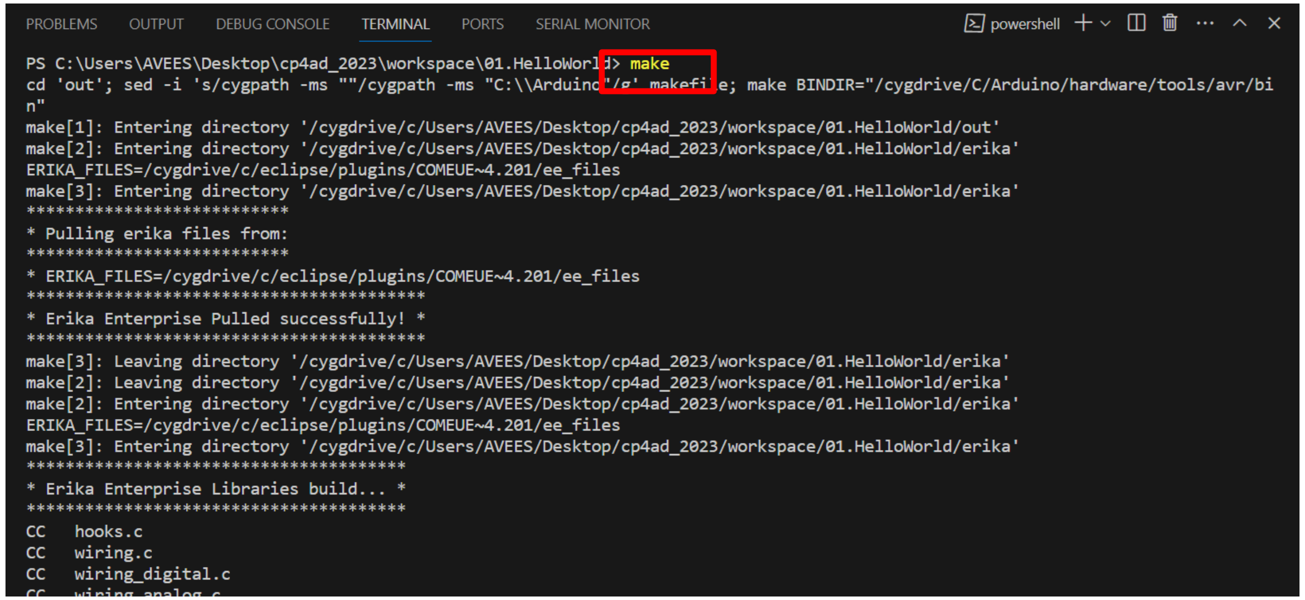


```
-- time : 406ms (delta 0ms)
Run build
Missing OS signature
Enable the generation of OS configuration files.
Missing Appl signature
Enable the generation of Appl configuration files.
Start the computation of event's masks.
Computation of event's masks completed.
Start the generation of OS configuration files.
Start the generation of Appl configuration files.
Generation completed.

Storing Appl signature file: C:\Users\AVEES\Desktop\cp4ad_2023\workspace\01.HelloWorld\.\out\applSignature.oil
Storing OS signature file: C:\Users\AVEES\Desktop\cp4ad_2023\workspace\01.HelloWorld\.\erika\osSignature.oil
-- time : 2443ms (delta 2036ms)
Build completed
[Config DONE]
PS C:\Users\AVEES\Desktop\cp4ad_2023\workspace\01.HelloWorld>
```

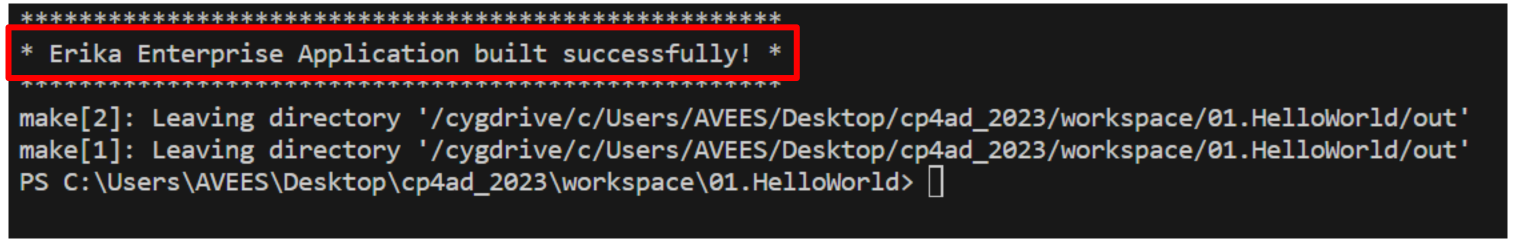
"Build Completed"
출력 확인 (빌드 완료)

프로젝트 빌드

- Terminal에서 make 실행



```
PS C:\Users\AVEES\Desktop\cp4ad_2023\workspace\01.HelloWorld> make
cd 'out'; sed -i 's/cygpath -ms ""/cygpath -ms "C:\\Arduino'/g' makefile; make BINDIR="/cygdrive/C/Arduino/hardware/tools/avr/bi
n"
make[1]: Entering directory '/cygdrive/c/Users/AVEES/Desktop/cp4ad_2023/workspace/01.HelloWorld/out'
make[2]: Entering directory '/cygdrive/c/Users/AVEES/Desktop/cp4ad_2023/workspace/01.HelloWorld/erika'
ERIKA_FILES=/cygdrive/c/eclipse/plugins/COMEUE~4.201/ee_files
make[3]: Entering directory '/cygdrive/c/Users/AVEES/Desktop/cp4ad_2023/workspace/01.HelloWorld/erika'
***************************
* Pulling erika files from:
***************************
* ERIKA_FILES=/cygdrive/c/eclipse/plugins/COMEUE~4.201/ee_files
*****************************************
* Erika Enterprise Pulled successfully! *
*****************************************
make[3]: Leaving directory '/cygdrive/c/Users/AVEES/Desktop/cp4ad_2023/workspace/01.HelloWorld/erika'
make[2]: Leaving directory '/cygdrive/c/Users/AVEES/Desktop/cp4ad_2023/workspace/01.HelloWorld/erika'
make[2]: Entering directory '/cygdrive/c/Users/AVEES/Desktop/cp4ad_2023/workspace/01.HelloWorld/erika'
ERIKA_FILES=/cygdrive/c/eclipse/plugins/COMEUE~4.201/ee_files
make[3]: Entering directory '/cygdrive/c/Users/AVEES/Desktop/cp4ad_2023/workspace/01.HelloWorld/erika'
***************************************
* Erika Enterprise Libraries build... *
***************************************
CC   hooks.c
CC   wiring.c
CC   wiring_digital.c
```

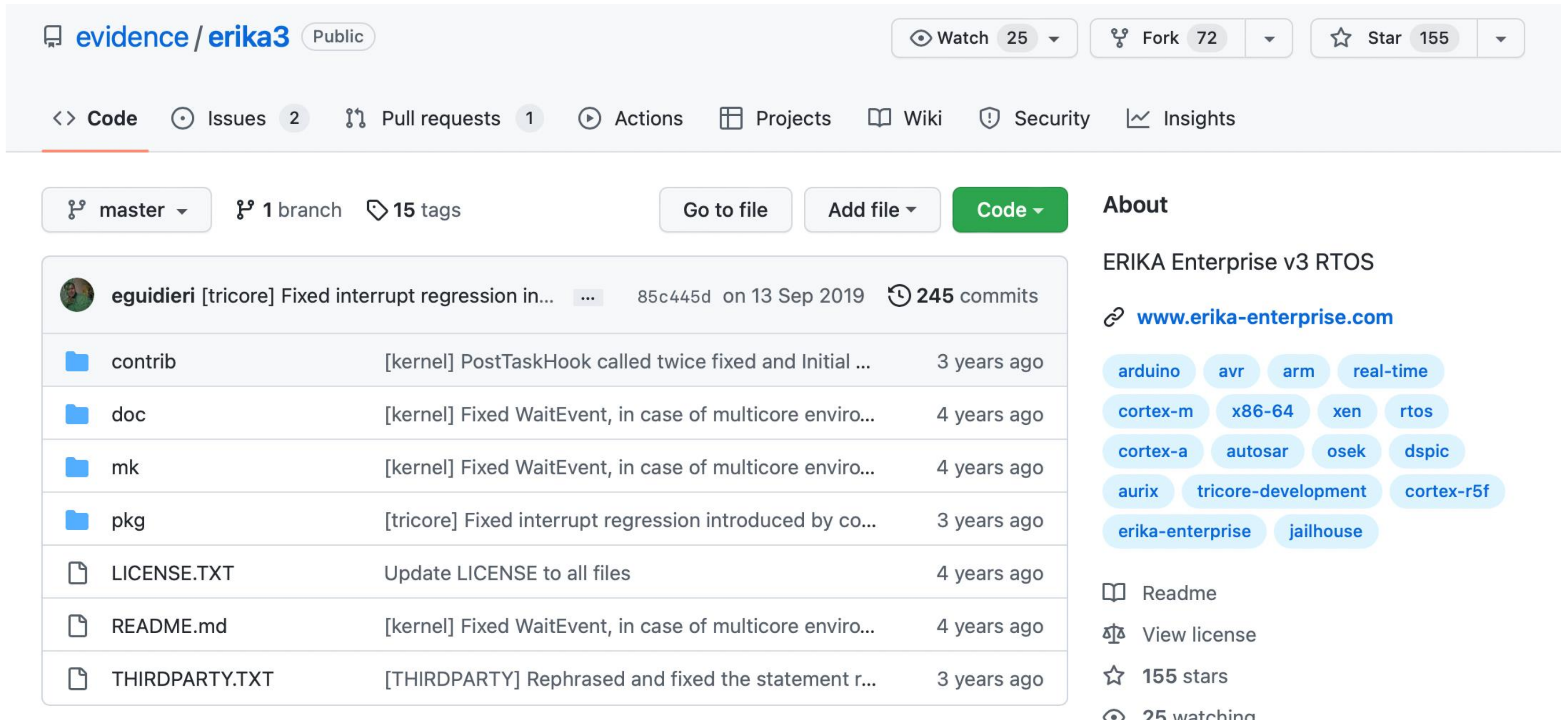


```
**********************************************************
* Erika Enterprise Application built successfully! *
**********************************************************
make[2]: Leaving directory '/cygdrive/c/Users/AVEES/Desktop/cp4ad_2023/workspace/01.HelloWorld/out'
make[1]: Leaving directory '/cygdrive/c/Users/AVEES/Desktop/cp4ad_2023/workspace/01.HelloWorld/out'
PS C:\Users\AVEES\Desktop\cp4ad_2023\workspace\01.HelloWorld>
```

"Built successfully"
출력 확인 (make 완료)

Erika Enterprise

- 이탈리아 EVIDENCE에서 개발된 오픈소스 OSEK/VDX RTOS
- 듀얼 라이센스 정책 (오픈소스 라이센스 + 상용 라이센스)
- RTOS 연구 및 교육에 널리 사용
- GitHub 리포



evidence / erika3 Public

Watch 25 | Fork 72 | Star 155

Code | Issues 2 | Pull requests 1 | Actions | Projects | Wiki | Security | Insights

master | 1 branch | 15 tags | Go to file | Add file | Code

eguidieri [tricore] Fixed interrupt regression in... 85c445d on 13 Sep 2019 245 commits

contrib	[kernel] PostTaskHook called twice fixed and Initial ...	3 years ago
doc	[kernel] Fixed WaitEvent, in case of multicore enviro...	4 years ago
mk	[kernel] Fixed WaitEvent, in case of multicore enviro...	4 years ago
pkg	[tricore] Fixed interrupt regression introduced by co...	3 years ago
LICENSE.TXT	Update LICENSE to all files	4 years ago
README.md	[kernel] Fixed WaitEvent, in case of multicore enviro...	4 years ago
THIRDPARTY.TXT	[THIRDPARTY] Rephrased and fixed the statement r...	3 years ago

About

ERIKA Enterprise v3 RTOS

www.erika-enterprise.com

arduino, avr, arm, real-time, cortex-m, x86-64, xen, rtos, cortex-a, autosar, osek, dspic, aurix, tricore-development, cortex-r5f, erika-enterprise, jailhouse

Readme

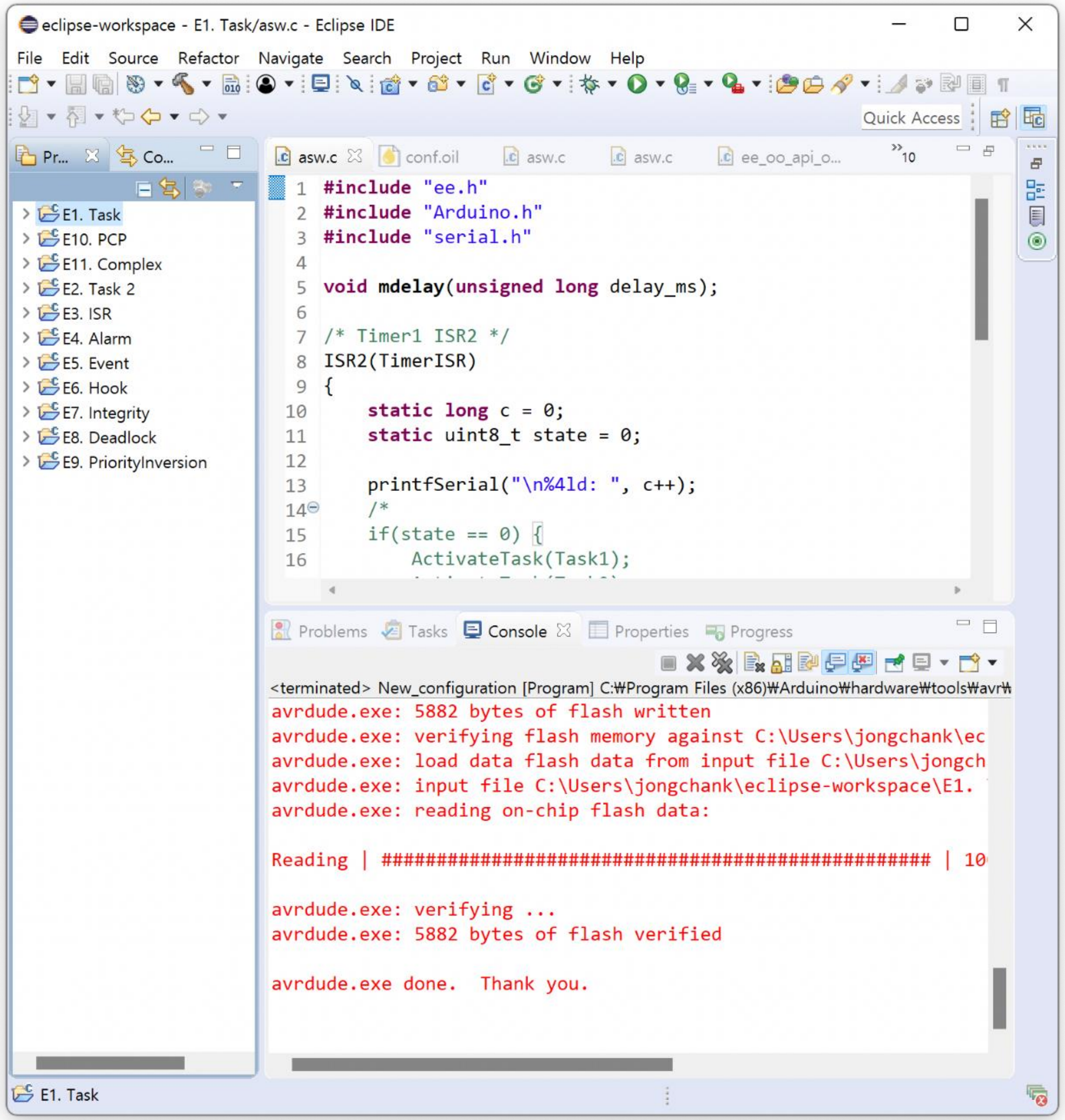
View license

155 stars

25 watching

Eclipse 기반 IDE (old)

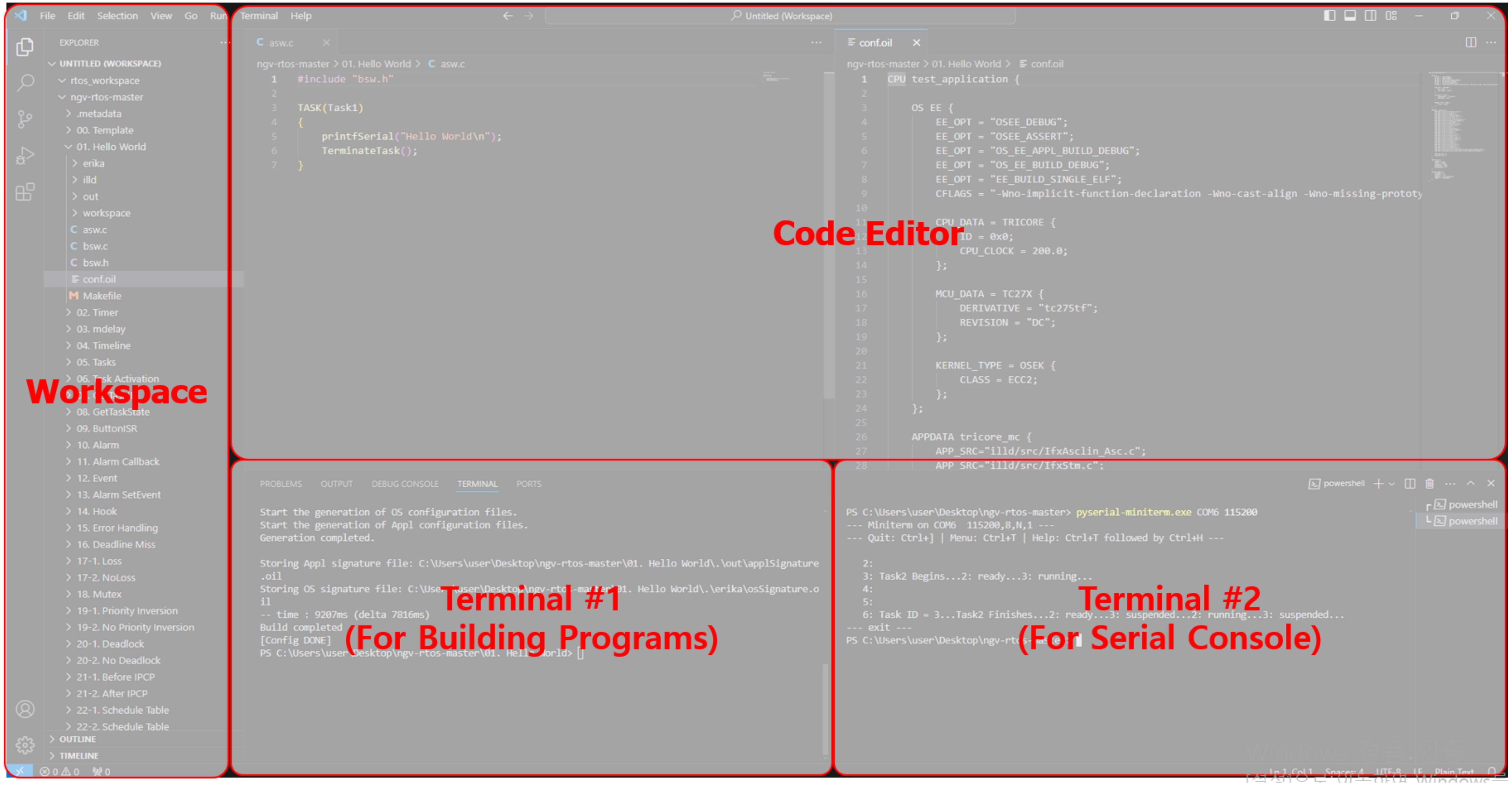
- 프로젝트 생성
- OIL 파일, C/C++ 파일 편집
- 프로젝트 빌드
- 실행파일 다운로드
- …



```
#include "ee.h"
#include "Arduino.h"
#include "serial.h"

void mdelay(unsigned long delay_ms);

/* Timer1 ISR2 */
ISR2(TimerISR)
{
    static long c = 0;
    static uint8_t state = 0;

    printfSerial("\n%4ld: ", c++);
    /*
    if(state == 0) {
        ActivateTask(Task1);
```

```
<terminated> New_configuration [Program] C:₩Program Files (x86)₩Arduino₩hardware₩tools₩avr₩
avrdude.exe: 5882 bytes of flash written
avrdude.exe: verifying flash memory against C:\Users\jongchank\ec
avrdude.exe: load data flash data from input file C:\Users\jongch
avrdude.exe: input file C:\Users\jongchank\eclipse-workspace\E1.
avrdude.exe: reading on-chip flash data:

Reading | ################################################## | 10

avrdude.exe: verifying ...
avrdude.exe: 5882 bytes of flash verified

avrdude.exe done.  Thank you.
```

Visual Studio Code 기반 개발환경 (new)



```c
#include "bsw.h"

TASK(Task1)
{
    printfSerial("Hello World\n");
    TerminateTask();
}
```

```
CPU test_application {

    OS EE {
        EE_OPT = "OSEE_DEBUG";
        EE_OPT = "OSEE_ASSERT";
        EE_OPT = "OS_EE_APPL_BUILD_DEBUG";
        EE_OPT = "OS_EE_BUILD_DEBUG";
        EE_OPT = "EE_BUILD_SINGLE_ELF";
        CFLAGS = "-Wno-implicit-function-declaration -Wno-cast-align -Wno-missing-prototy

        CPU_DATA = TRICORE {
            ID = 0x0;
            CPU_CLOCK = 200.0;
        };

        MCU_DATA = TC27X {
            DERIVATIVE = "tc275tf";
            REVISION = "DC";
        };

        KERNEL_TYPE = OSEK {
            CLASS = ECC2;
        };
    };

    APPDATA tricore_mc {
        APP_SRC="illd/src/IfxAsclin_Asc.c";
        APP_SRC="illd/src/IfxStm.c";
```

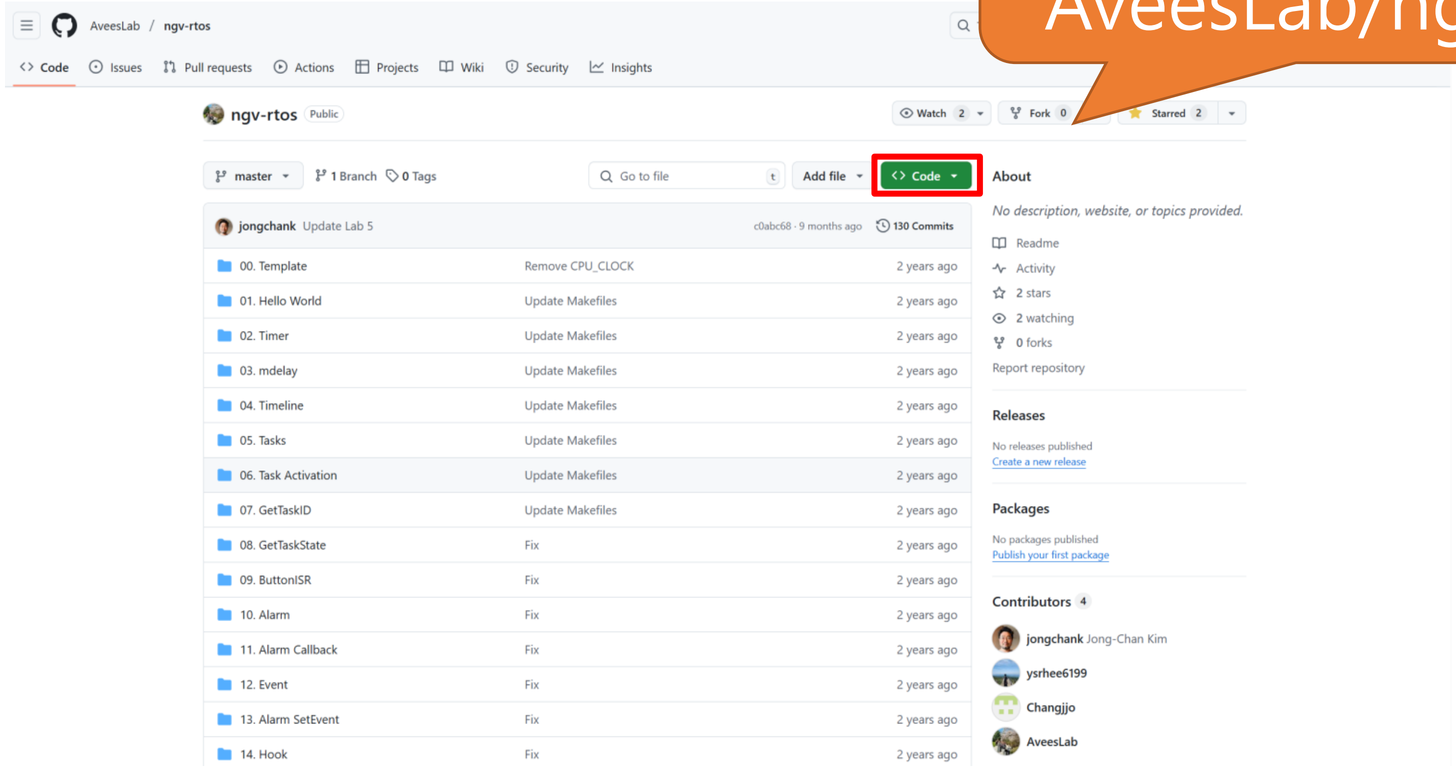
```
Start the generation of OS configuration files.
Start the generation of Appl configuration files.
Generation completed.

Storing Appl signature file: C:\Users\user\Desktop\ngv-rtos-master\01. Hello World\.\out\applSignature
.oil
Storing OS signature file: C:\Users\user\Desktop\ngv-rtos-master\01. Hello World\.\erika\osSignature.o
il
-- time : 9207ms (delta 7816ms)
Build completed
[Config DONE]
PS C:\Users\user\Desktop\ngv-rtos-master\01. Hello World> 
```

```
PS C:\Users\user\Desktop\ngv-rtos-master> pyserial-miniterm.exe COM6 115200
--- Miniterm on COM6  115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

  2:
  3: Task2 Begins...2: ready...3: running...
  4:
  5:
  6: Task ID = 3...Task2 Finishes...2: ready...3: suspended...2: running...3: suspended...
--- exit ---
PS C:\Users\user\Desktop\ngv-rtos-master> 
```

workspace

- Directory Structure
  - `workspace`
    - `rtos_workspace`: working directory
    - `ngv-rtos-master`: example source files

https://github.com/AveesLab/ngv-rtos



AveesLab / ngv-rtos

Code · Issues · Pull requests · Actions · Projects · Wiki · Security · Insights

ngv-rtos Public

Watch 2 · Fork 0 · Starred 2

master · 1 Branch · 0 Tags · Go to file · Add file · Code

jongchank Update Lab 5 · c0abc68 · 9 months ago · 130 Commits

00. Template	Remove CPU_CLOCK	2 years ago
01. Hello World	Update Makefiles	2 years ago
02. Timer	Update Makefiles	2 years ago
03. mdelay	Update Makefiles	2 years ago
04. Timeline	Update Makefiles	2 years ago
05. Tasks	Update Makefiles	2 years ago
06. Task Activation	Update Makefiles	2 years ago
07. GetTaskID	Update Makefiles	2 years ago
08. GetTaskState	Fix	2 years ago
09. ButtonISR	Fix	2 years ago
10. Alarm	Fix	2 years ago
11. Alarm Callback	Fix	2 years ago
12. Event	Fix	2 years ago
13. Alarm SetEvent	Fix	2 years ago
14. Hook	Fix	2 years ago

About

No description, website, or topics provided.

Readme
Activity
2 stars
2 watching
0 forks
Report repository

Releases

No releases published
Create a new release

Packages

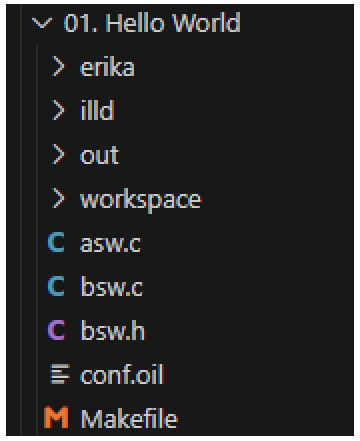
No packages published
Publish your first package

Contributors 4

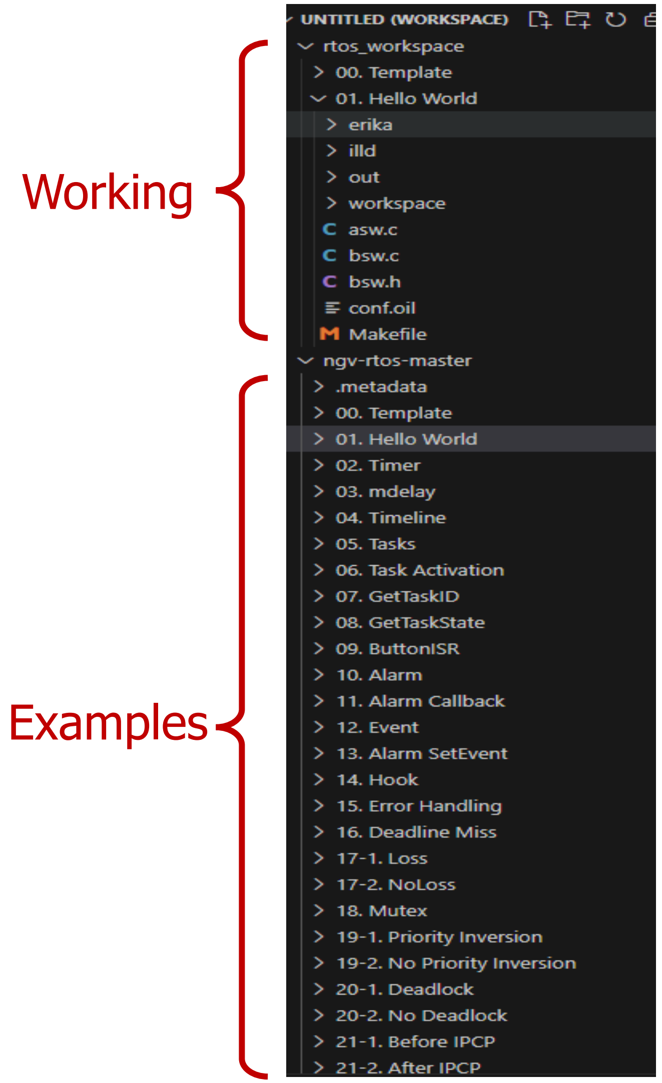
jongchank Jong-Chan Kim
ysrhee6199
Changjjo
AveesLab

Visual Studio Code 기반 개발환경 (new)

- Directory Structure
  - `rtos_workspace`: working directory
  - `ngv-rtos-master`: example source files
- Example Projects Directory Structure
  - `illd/`: Infineon Low Level Driver
  - `asw.c`: Application SW code
  - `bsw.c`: Basic SW code
  - `bsw.h`: Basic SW header
  - `conf.oil`: OIL configuration file
  - `Makefile`: Top-level Makefile



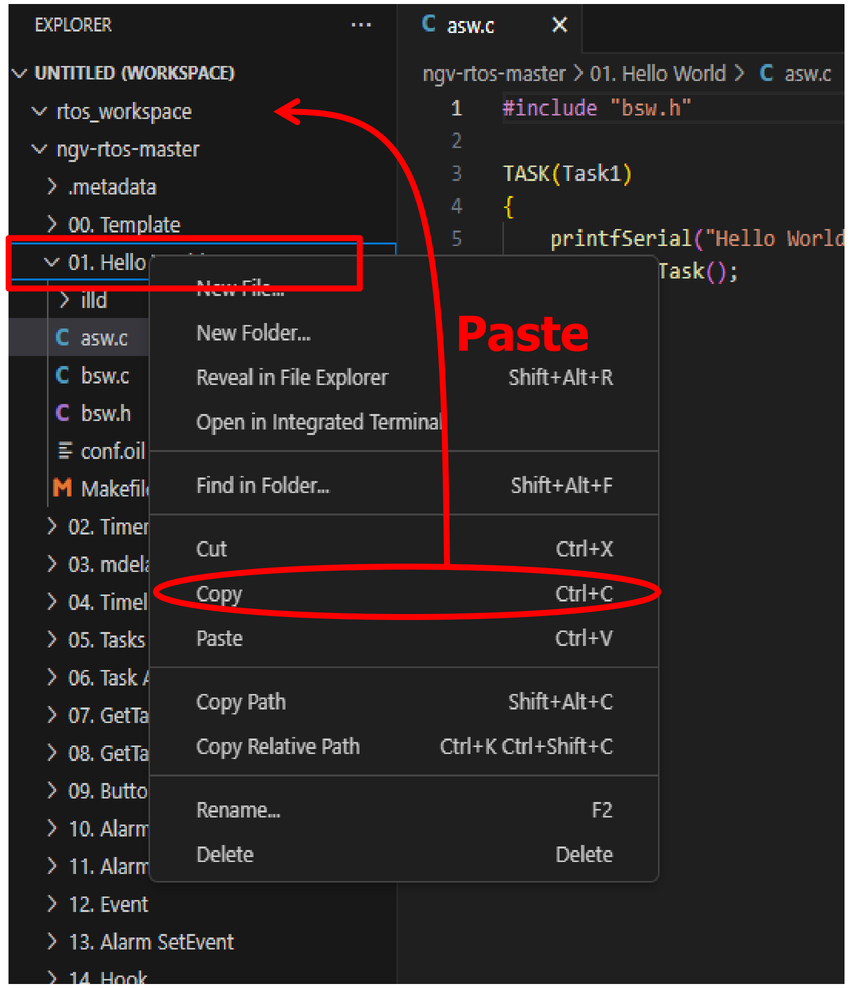
```
⌄ 01. Hello World
  > erika
  > illd
  > out
  > workspace
  C asw.c
  C bsw.c
  C bsw.h
  ≡ conf.oil
  M Makefile
```



```
UNTITLED (WORKSPACE)
⌄ rtos_workspace
  > 00. Template
  ⌄ 01. Hello World
    > erika
    > illd
    > out
    > workspace
    C asw.c
    C bsw.c
    C bsw.h
    ≡ conf.oil
    M Makefile
⌄ ngv-rtos-master
  > .metadata
  > 00. Template
  > 01. Hello World
  > 02. Timer
  > 03. mdelay
  > 04. Timeline
  > 05. Tasks
  > 06. Task Activation
  > 07. GetTaskID
  > 08. GetTaskState
  > 09. ButtonISR
  > 10. Alarm
  > 11. Alarm Callback
  > 12. Event
  > 13. Alarm SetEvent
  > 14. Hook
  > 15. Error Handling
  > 16. Deadline Miss
  > 17-1. Loss
  > 17-2. NoLoss
  > 18. Mutex
  > 19-1. Priority Inversion
  > 19-2. No Priority Inversion
  > 20-1. Deadlock
  > 20-2. No Deadlock
  > 21-1. Before IPCP
  > 21-2. After IPCP
```

Workflow

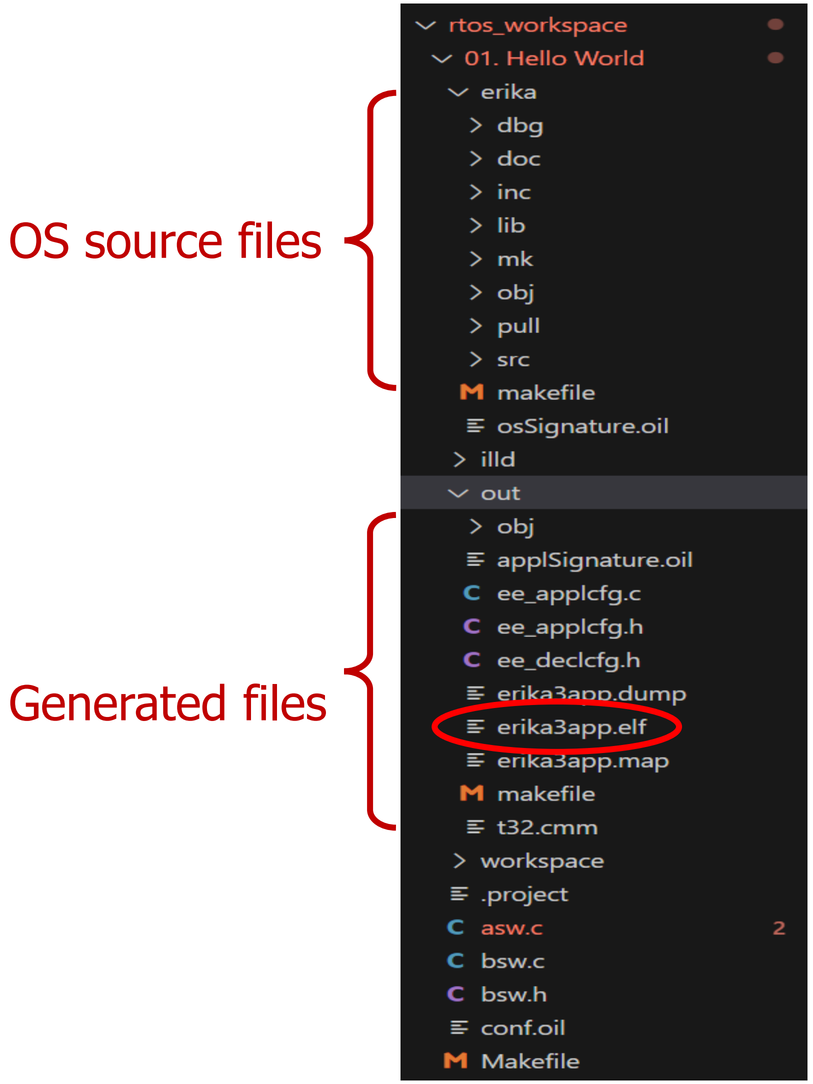
- Basic Workflow
  1) Copy a project (e.g., 01. Hello World) from to working directory
  2) Edit source files (.C or .H) and OIL files
  3) Build
- Build Process (in Terminal #1)
  1) `cd 01. Hello World\`
  2) `make config`
     - Generate OS kernel files from conf.oil
  3) `make`
     - Generate an executable file



```
#include "bsw.h"

TASK(Task1)
{
    printfSerial("Hello World
                        Task();
```

Generated Files

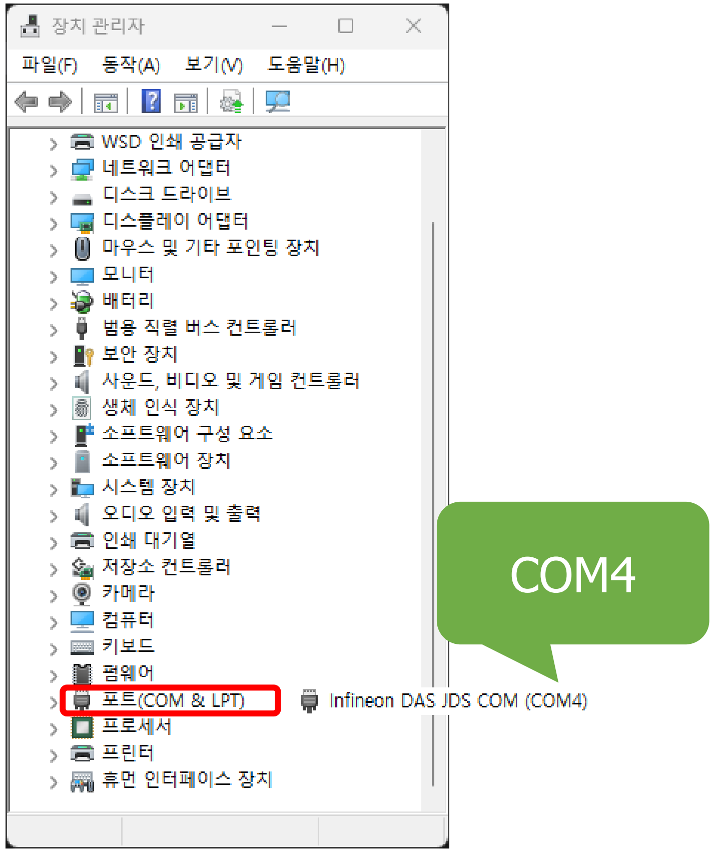
- erika/
  - OS source files
- out/
  - Generated files from the OIL file
  - Object files
  - Executable (ELF) file (`erika3app.elf`)



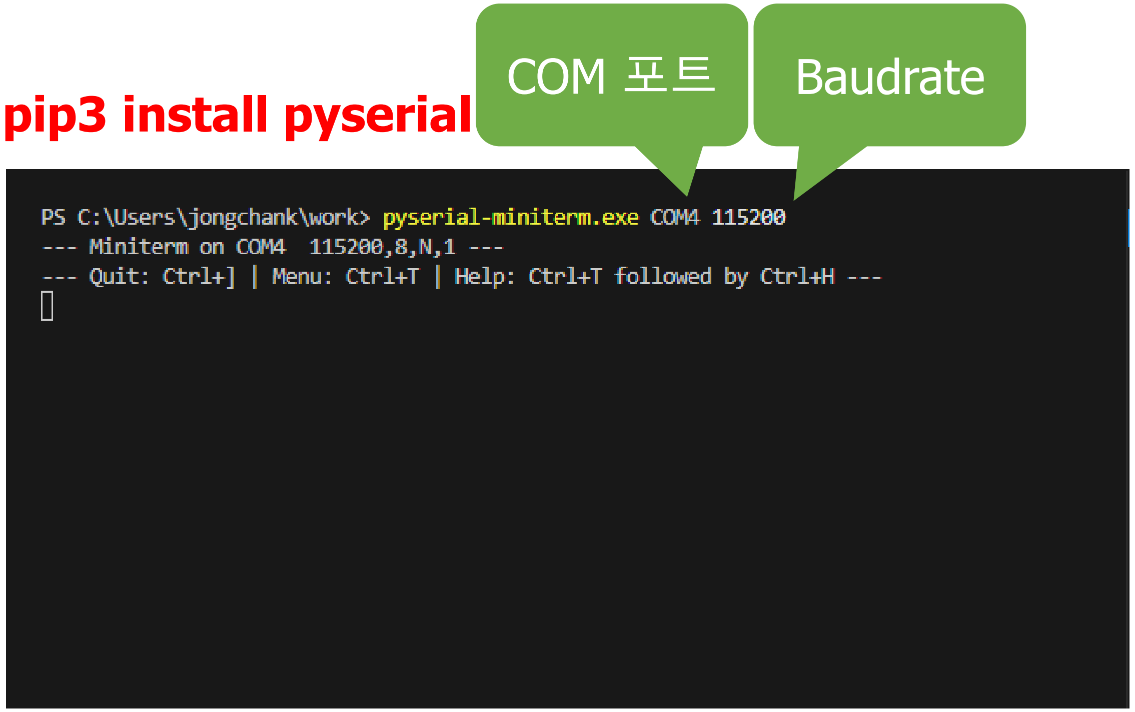
```
rtos_workspace
  01. Hello World
    erika
      > dbg
      > doc
      > inc
      > lib
      > mk
      > obj
      > pull
      > src
      makefile
      osSignature.oil
    > illd
    out
      > obj
      applSignature.oil
      ee_applcfg.c
      ee_applcfg.h
      ee_declcfg.h
      erika3app.dump
      erika3app.elf
      erika3app.map
      makefile
      t32.cmm
    > workspace
    .project
    asw.c                2
    bsw.c
    bsw.h
    conf.oil
    Makefile
```

Serial Console

- 장치관리자에서 COM 포트 확인 (e.g., COM4)
- pyserial-miniterm 이용하여 시리얼 콘솔 시작 (Terminal #2)



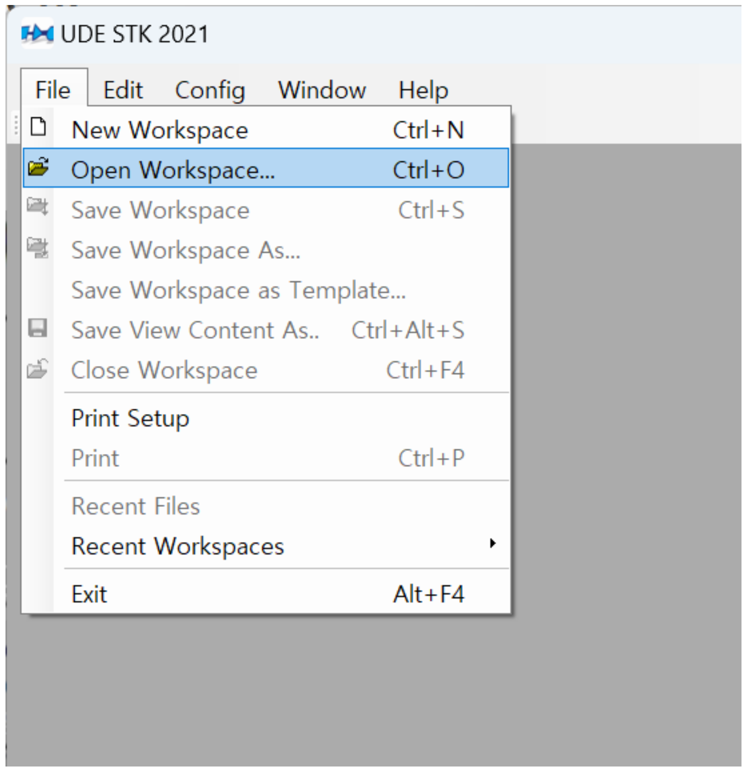
pip3 install pyserial

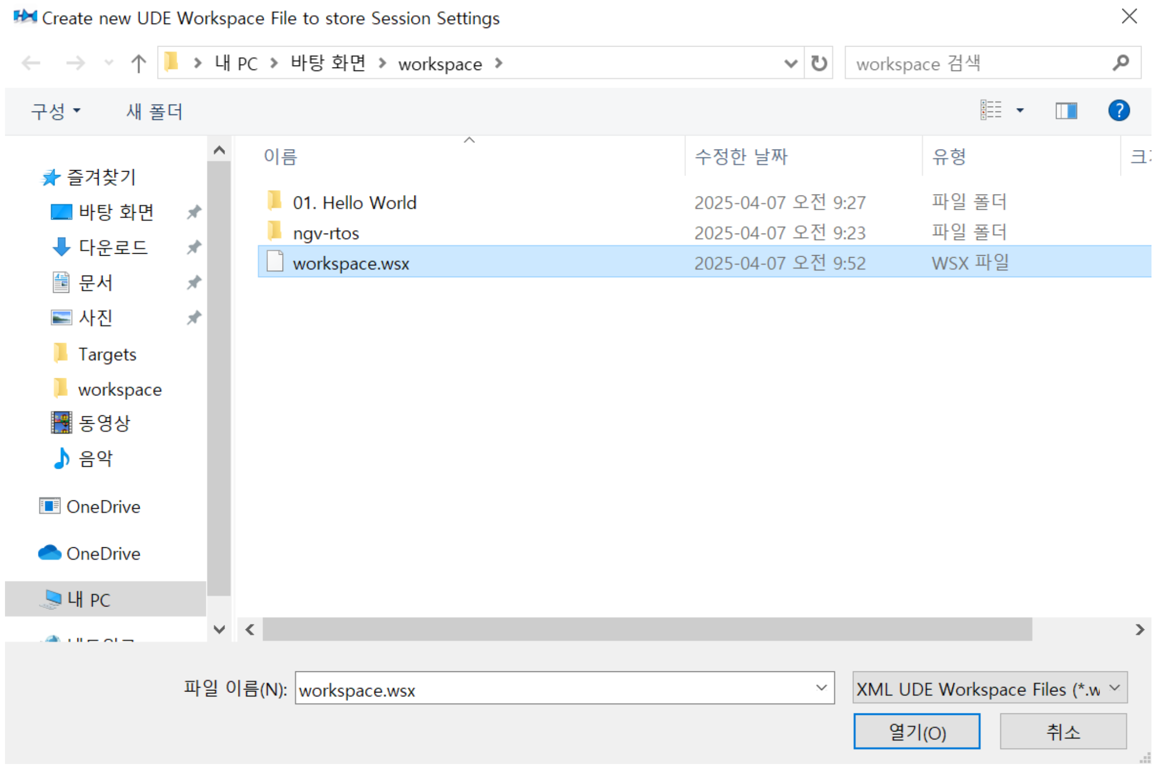


```
PS C:\Users\jongchank\work> pyserial-miniterm.exe COM4 115200
--- Miniterm on COM4  115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---
```

New Workspace

- File → New Workspace
- 바탕화면 > workspace > workspace.wsx 생성





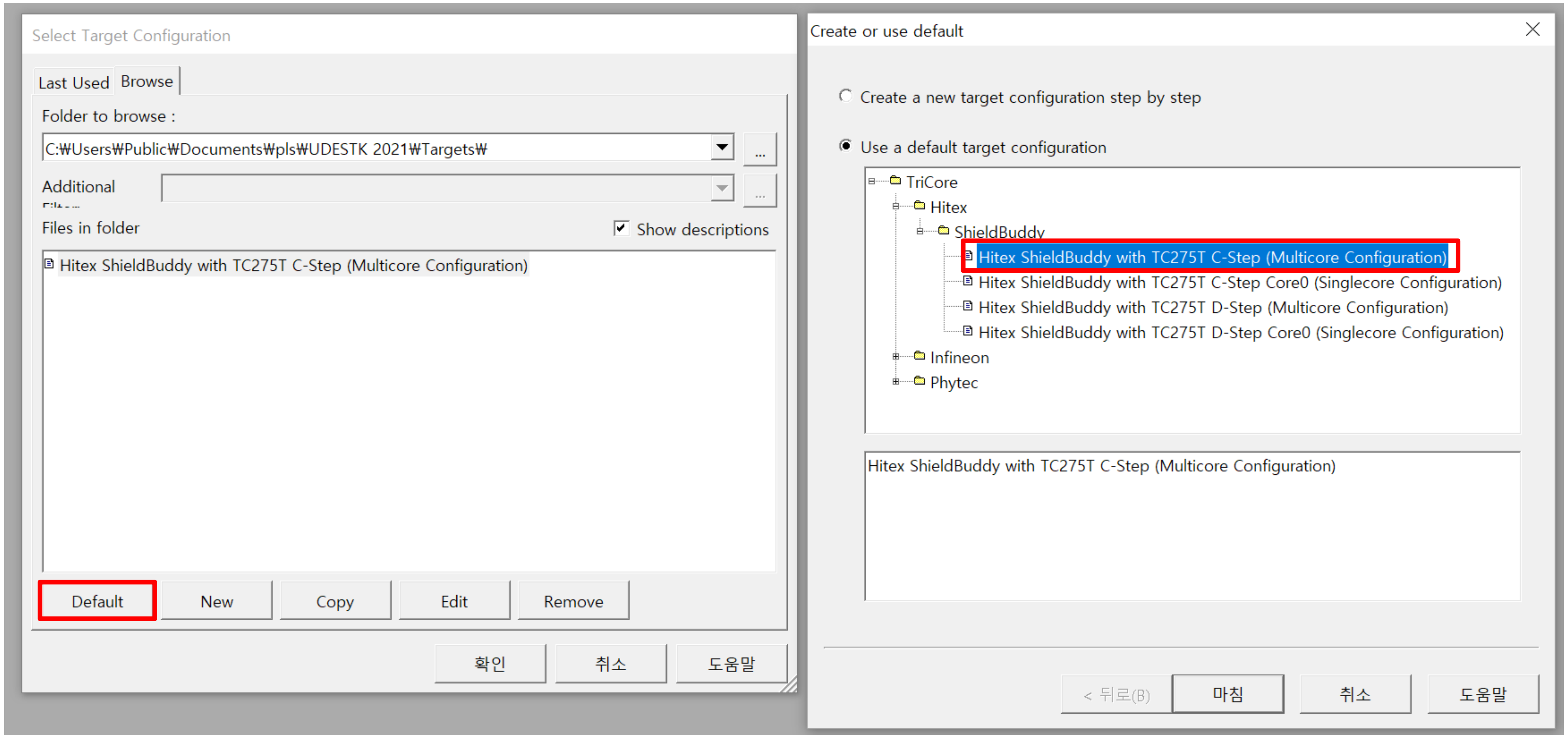
이름	수정한 날짜	유형
01. Hello World	2025-04-07 오전 9:27	파일 폴더
ngv-rtos	2025-04-07 오전 9:23	파일 폴더
workspace.wsx	2025-04-07 오전 9:52	WSX 파일

파일 이름(N): workspace.wsx

XML UDE Workspace Files (*.w

New Workspace

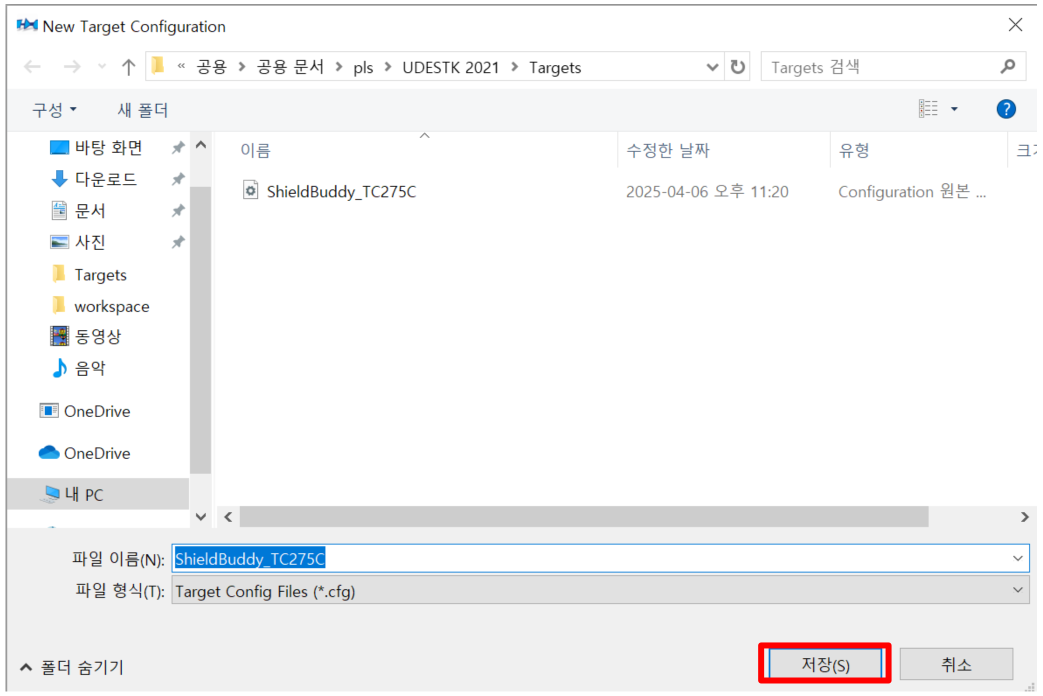
- Select Target Configuration > Default
- TriCore > HiTex > ShieldBuddy > … with TC275T C-step (Multicore…)

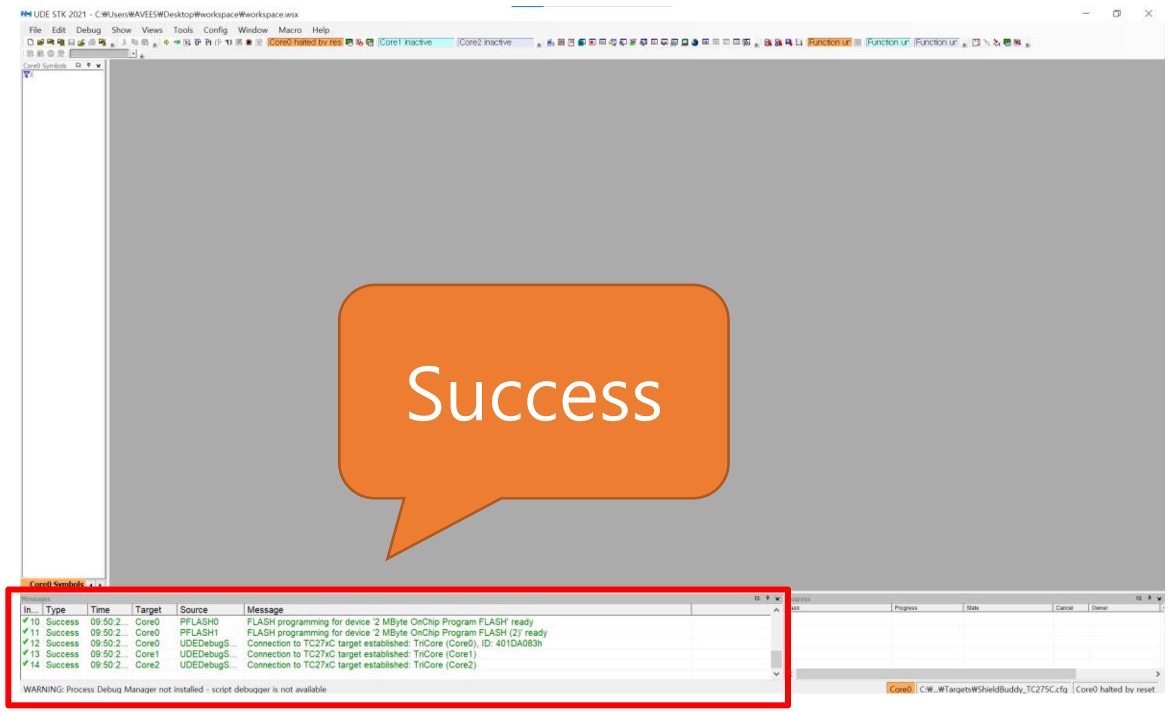


Select Target Configuration
Last Used
Browse
Folder to browse :
C:₩Users₩Public₩Documents₩pls₩UDESTK 2021₩Targets₩
Additional Filter
Files in folder
✓ Show descriptions
Hitex ShieldBuddy with TC275T C-Step (Multicore Configuration)
Default
New
Copy
Edit
Remove
확인
취소
도움말
Create or use default
Create a new target configuration step by step
● Use a default target configuration
TriCore
Hitex
ShieldBuddy
Hitex ShieldBuddy with TC275T C-Step (Multicore Configuration)
Hitex ShieldBuddy with TC275T C-Step Core0 (Singlecore Configuration)
Hitex ShieldBuddy with TC275T D-Step (Multicore Configuration)
Hitex ShieldBuddy with TC275T D-Step Core0 (Singlecore Configuration)
Infineon
Phytec
Hitex ShieldBuddy with TC275T C-Step (Multicore Configuration)
< 뒤로(B)
마침
취소
도움말

New Workspace

- ShiedlBuddy_TC275C.cfg 저장
- 연결 실패시, … with TC275T D-step (Multicore…)으로 다시 진행

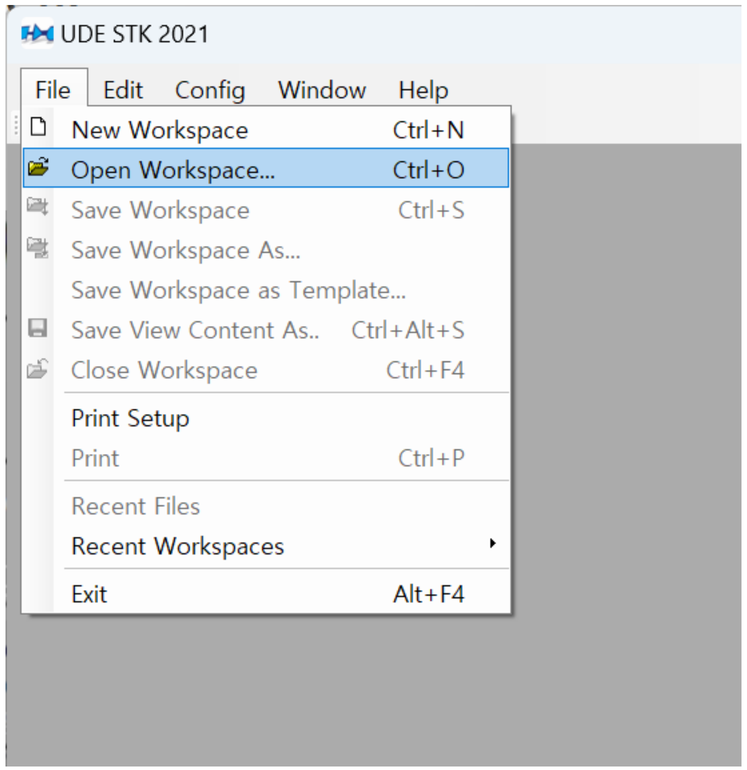


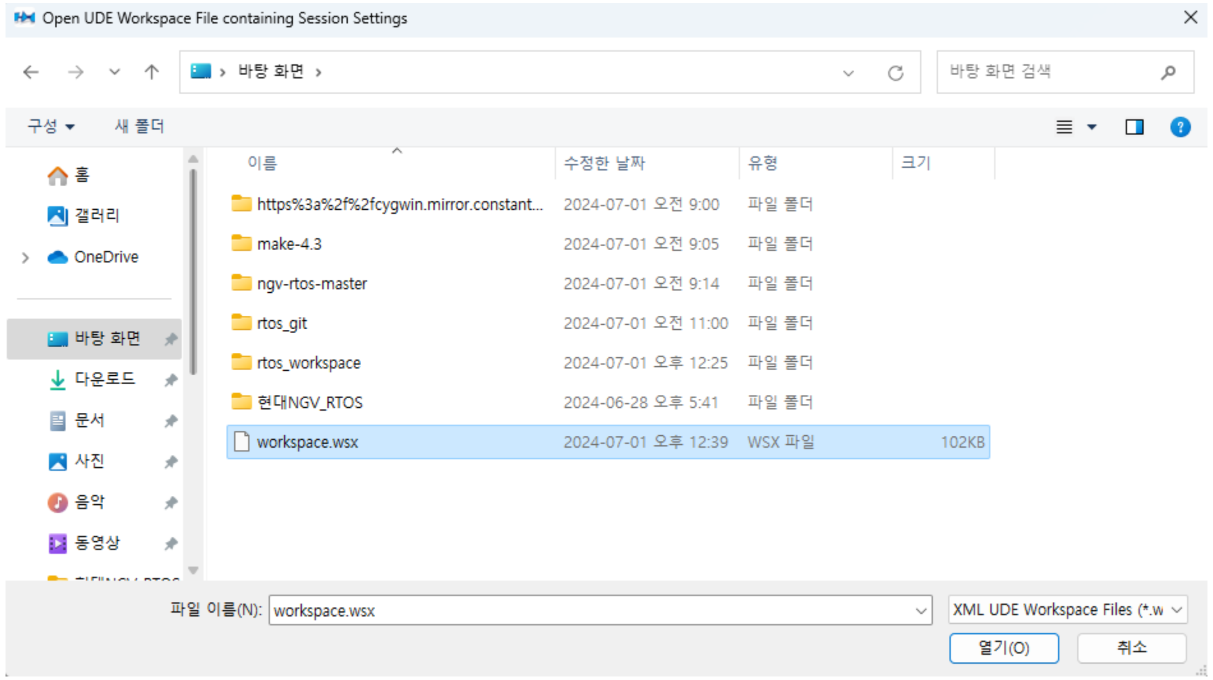

UDE STK 2021 - C:₩Users₩AVEES₩Desktop₩workspace₩workspace.wsx
File Edit Debug Show Views Tools Config Window Macro Help
Core0 Symbols
Success

```
In... Type     Time       Target  Source        Message
10    Success  09:50:2... Core0   PFLASH0       FLASH programming for device '2 MByte OnChip Program FLASH' ready
11    Success  09:50:2... Core0   PFLASH1       FLASH programming for device '2 MByte OnChip Program FLASH (2)' ready
12    Success  09:50:2... Core0   UDEDebugS...  Connection to TC27xC target established: TriCore (Core0), ID: 401DA083h
13    Success  09:50:2... Core1   UDEDebugS...  Connection to TC27xC target established: TriCore (Core1)
14    Success  09:50:2... Core2   UDEDebugS...  Connection to TC27xC target established: TriCore (Core2)
WARNING: Process Debug Manager not installed - script debugger is not available
```

Program Flashing by UDE #1

- File → Open Workspace
- 바탕화면의 workspace.wsx 열기



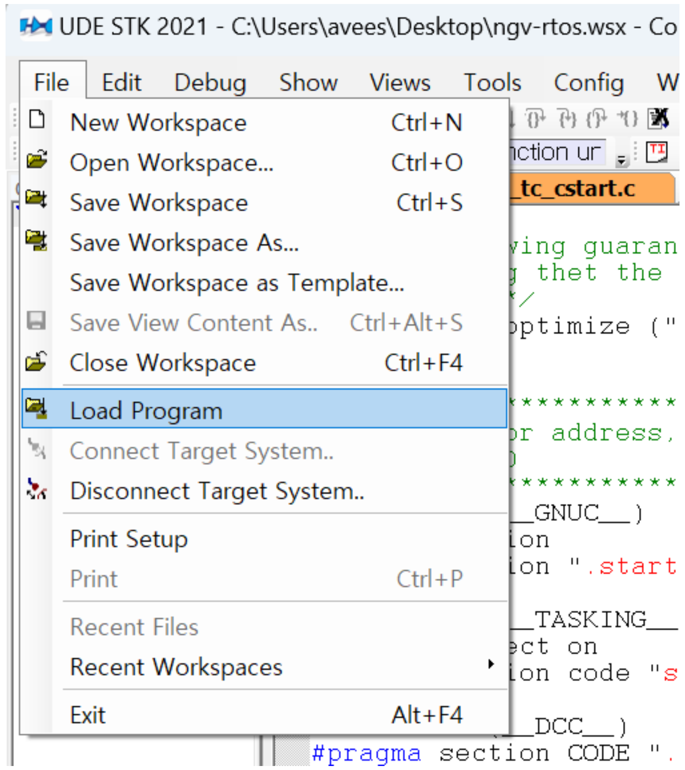


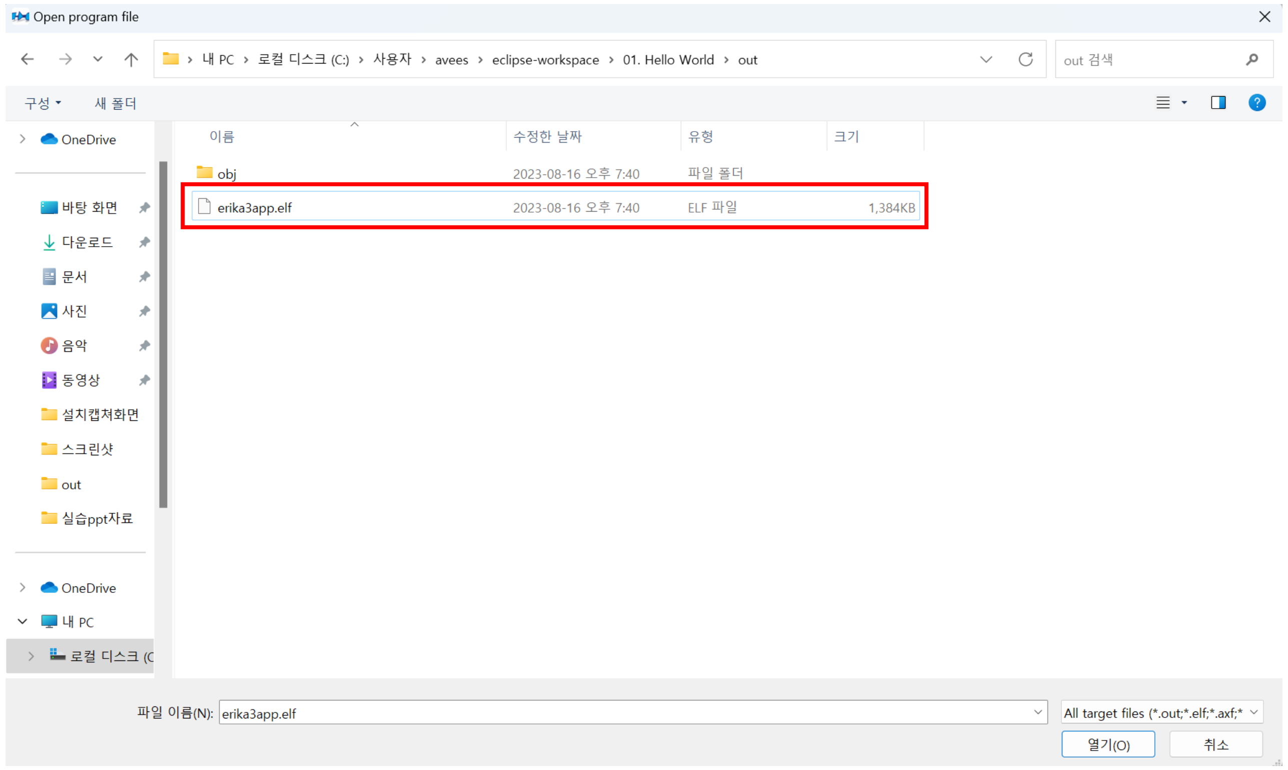
이름	수정한 날짜	유형	크기
https%3a%2f%2fcygwin.mirror.constant...	2024-07-01 오전 9:00	파일 폴더	
make-4.3	2024-07-01 오전 9:05	파일 폴더	
ngv-rtos-master	2024-07-01 오전 9:14	파일 폴더	
rtos_git	2024-07-01 오전 11:00	파일 폴더	
rtos_workspace	2024-07-01 오후 12:25	파일 폴더	
현대NGV_RTOS	2024-06-28 오후 5:41	파일 폴더	
workspace.wsx	2024-07-01 오후 12:39	WSX 파일	102KB

파일 이름(N): workspace.wsx

Program Flashing by UDE #2

- File → Load Program
- 프로젝트 폴더의 `out/erika3app.elf` 열기





이름	수정한 날짜	유형	크기
obj	2023-08-16 오후 7:40	파일 폴더	
erika3app.elf	2023-08-16 오후 7:40	ELF 파일	1,384KB

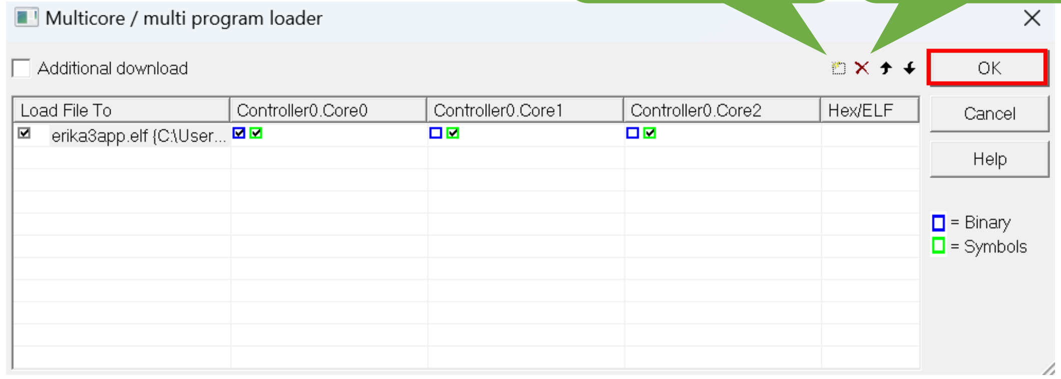
파일 이름(N): erika3app.elf

Program Flashing by UDE #3

새 파일 열기

기존 파일 해제

- Step1



Multicore / multi program loader

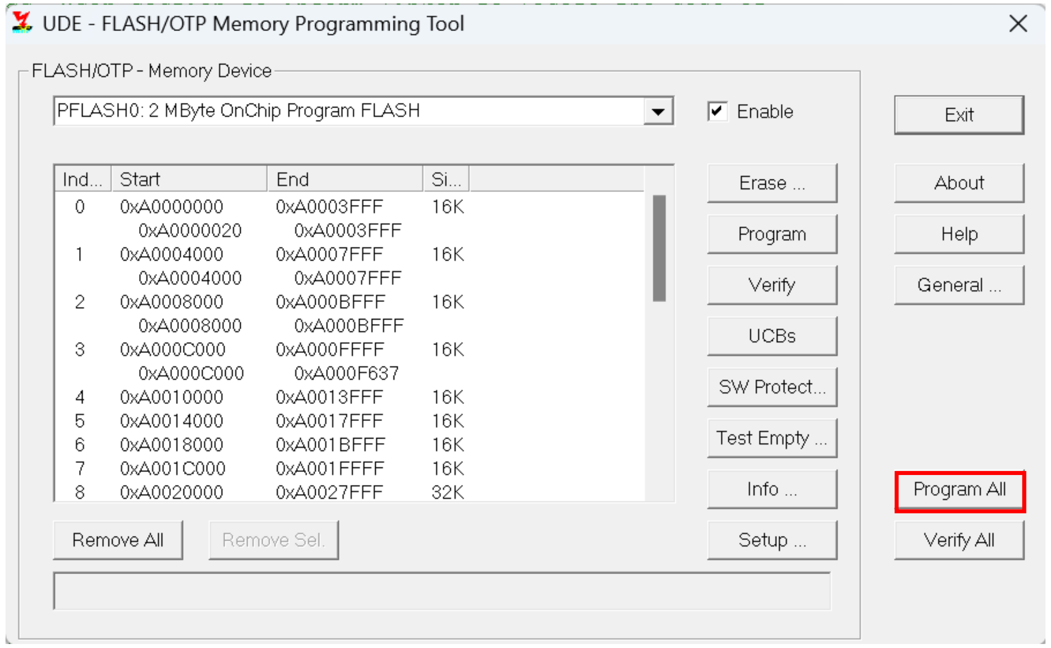
☐ Additional download

OK
Cancel
Help

Load File To	Controller0.Core0	Controller0.Core1	Controller0.Core2	Hex/ELF
☑ erika3app.elf {C:\User...	🟦 ☑	🟦 ☑	🟦 ☑	

🟦 = Binary
🟩 = Symbols

- Step2



UDE - FLASH/OTP Memory Programming Tool

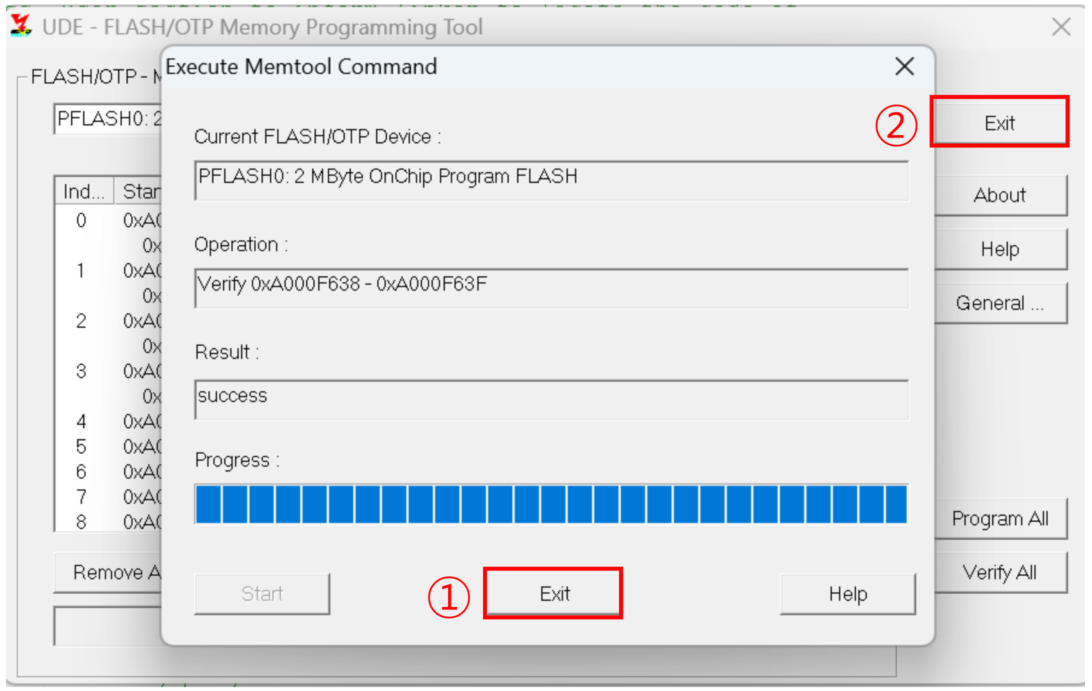
FLASH/OTP - Memory Device

PFLASH0: 2 MByte OnChip Program FLASH

☑ Enable

Ind...	Start	End	Si...
0	0xA0000000	0xA0003FFF	16K
	0xA0000020	0xA0003FFF	
1	0xA0004000	0xA0007FFF	16K
	0xA0004000	0xA0007FFF	
2	0xA0008000	0xA000BFFF	16K
	0xA0008000	0xA000BFFF	
3	0xA000C000	0xA000FFFF	16K
	0xA000C000	0xA000F637	
4	0xA0010000	0xA0013FFF	16K
5	0xA0014000	0xA0017FFF	16K
6	0xA0018000	0xA001BFFF	16K
7	0xA001C000	0xA001FFFF	16K
8	0xA0020000	0xA0027FFF	32K

- Step3



UDE - FLASH/OTP Memory Programming Tool

Execute Memtool Command

Current FLASH/OTP Device :
PFLASH0: 2 MByte OnChip Program FLASH

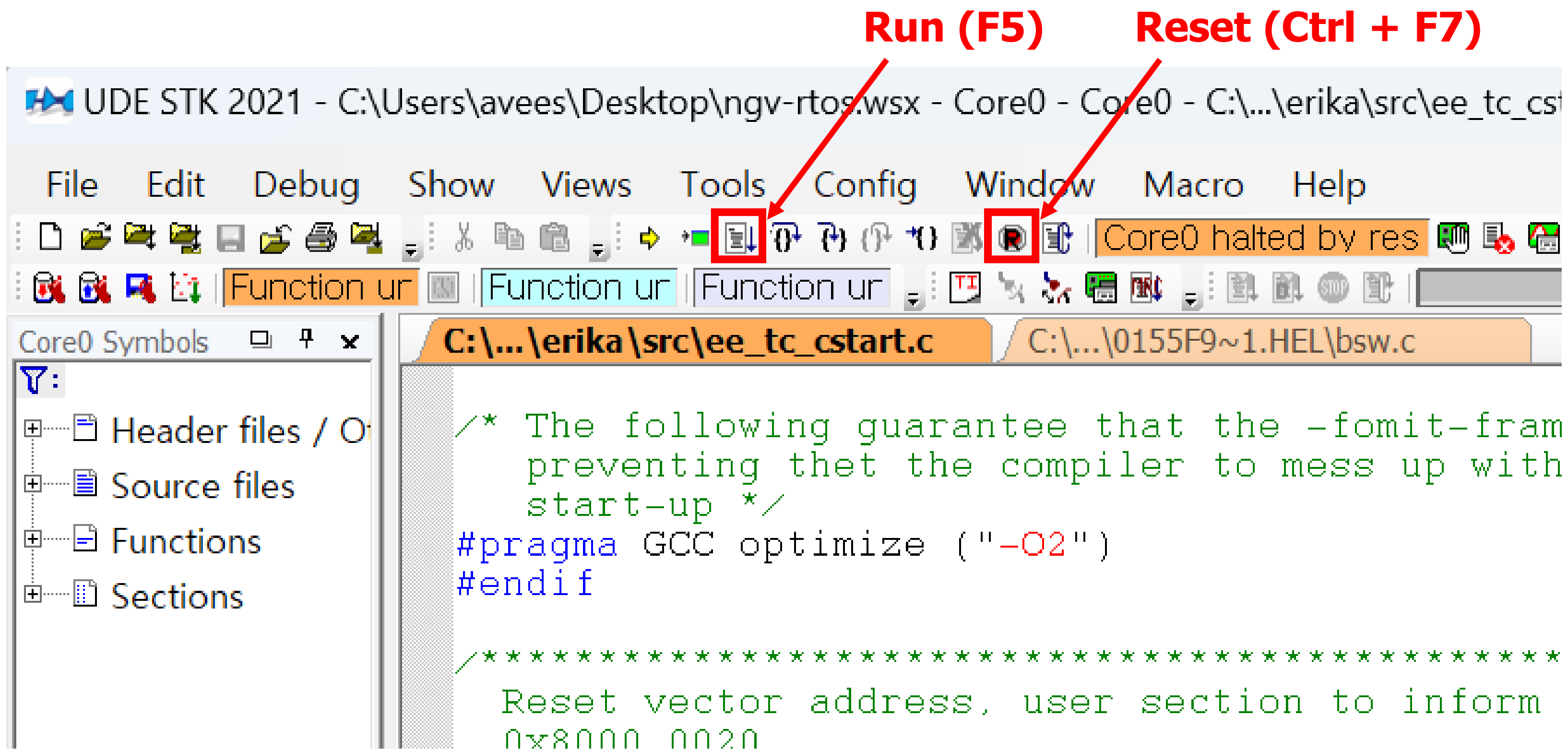
Operation :
Verify 0xA000F638 - 0xA000F63F

Result :
success

Progress :

Program Flashing by UDE #4

- 프로그램 실행 (run)
- 기존 프로그램 실행중일 경우 reset 후 실행


Run (F5)
Reset (Ctrl + F7)
UDE STK 2021 - C:\Users\avees\Desktop\ngv-rtos.wsx - Core0 - Core0 - C:\...\erika\src\ee_tc_cst
File Edit Debug Show Views Tools Config Window Macro Help
Core0 halted by res
Function ur
Function ur
Function ur
Core0 Symbols
Header files / O
Source files
Functions
Sections
C:\...\erika\src\ee_tc_cstart.c
C:\...\0155F9~1.HEL\bsw.c

```
/* The following guarantee that the -fomit-fram
   preventing thet the compiler to mess up with
   start-up */
#pragma GCC optimize ("-O2")
#endif

/*******************************************************
  Reset vector address, user section to inform
  0x8000_0020
```

01. Hello World

- 00. Template 복사
- asw.c에 TASK 추가
- OIL 파일에 TASK 추가
- printfSerial() 함수 사용 (시리얼 콘솔 출력)

Make config는 OIL 변경시에만

```
$ make config
$ make
```

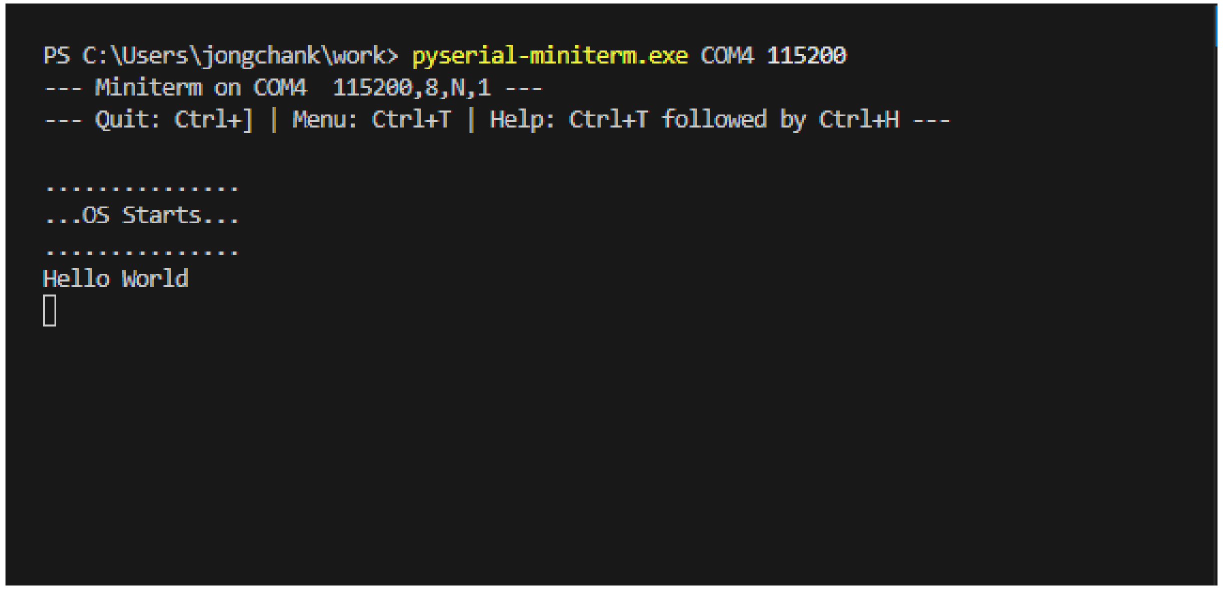
```c
#include "bsw.h"


TASK(Task1)
{
    printfSerial("Hello World\n");

    TerminateTask();
}
```

```
TASK Task1 {
    PRIORITY = 1;
    STACK = SHARED;
    SCHEDULE = FULL;
    AUTOSTART = TRUE;
    ACTIVATION = 1;
};
```

자동 시작

01. Hello World

- OS 시작 후
- Hello World 출력



```
PS C:\Users\jongchank\work> pyserial-miniterm.exe COM4 115200
--- Miniterm on COM4  115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

...............
...OS Starts...
...............
Hello World
```

02. Timer

- C 파일에 ISR2로 TimerISR 추가
- OIL 파일에 TimerISR 추가
  - Category 2

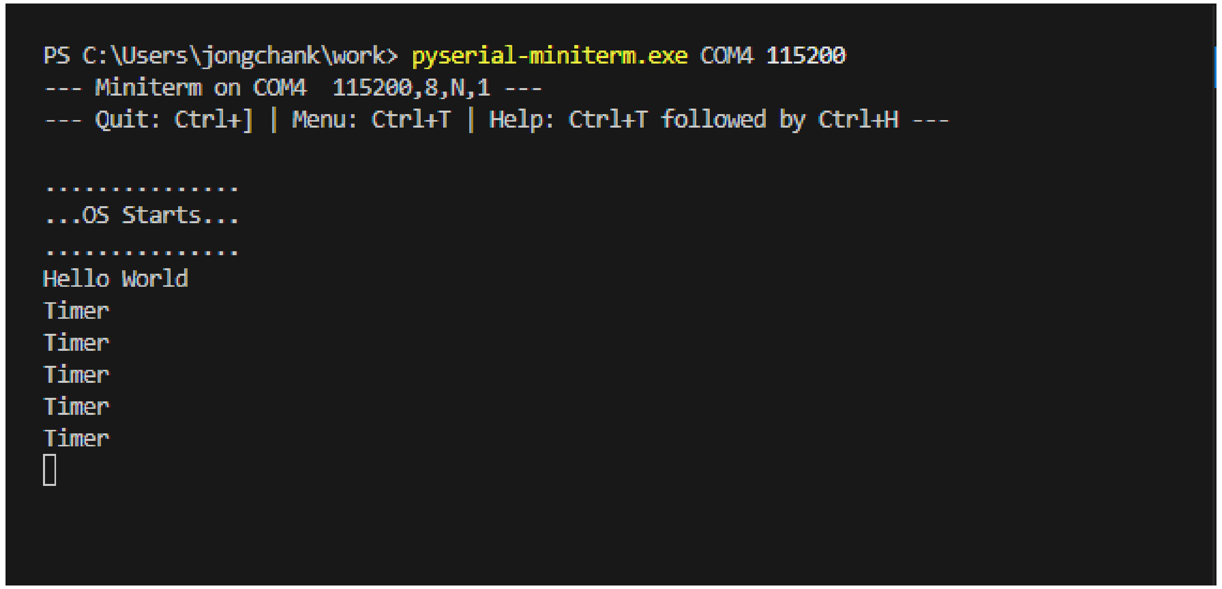
1초 뒤
interrupt 등록

```c
ISR2(TimerISR)
{
    osEE_tc_stm_set_sr0_next_match(1000000U);
    printfSerial("Timer\n");
}
```

```
ISR TimerISR {
    CATEGORY = 2;
    SOURCE = "STM0SR0";
    PRIORITY = 2;
};
```

02. Timer

- Hello World 출력후 Timer 반복

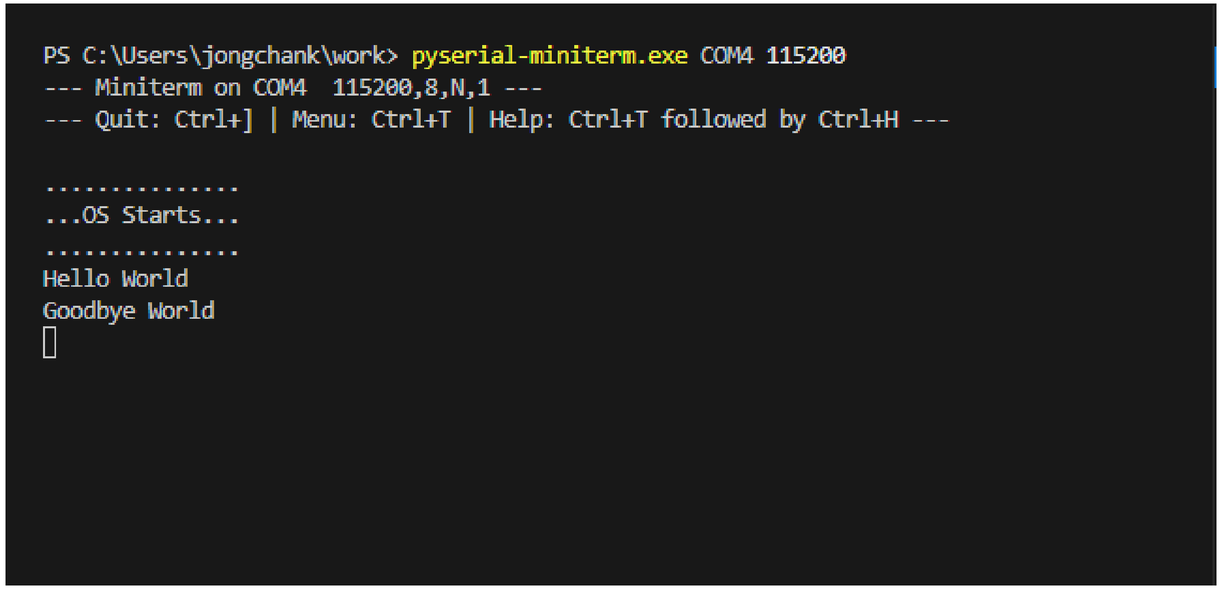


```
PS C:\Users\jongchank\work> pyserial-miniterm.exe COM4 115200
--- Miniterm on COM4  115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

...............
...OS Starts...
...............
Hello World
Timer
Timer
Timer
Timer
Timer
```

03. mdelay

- mdelay 함수 이용 3초 실행시간

```
TASK(Task1)
{
    printfSerial("Hello World\n");

    mdelay(3000);

    printfSerial("Goodbye World\n");
    TerminateTask();
}
```

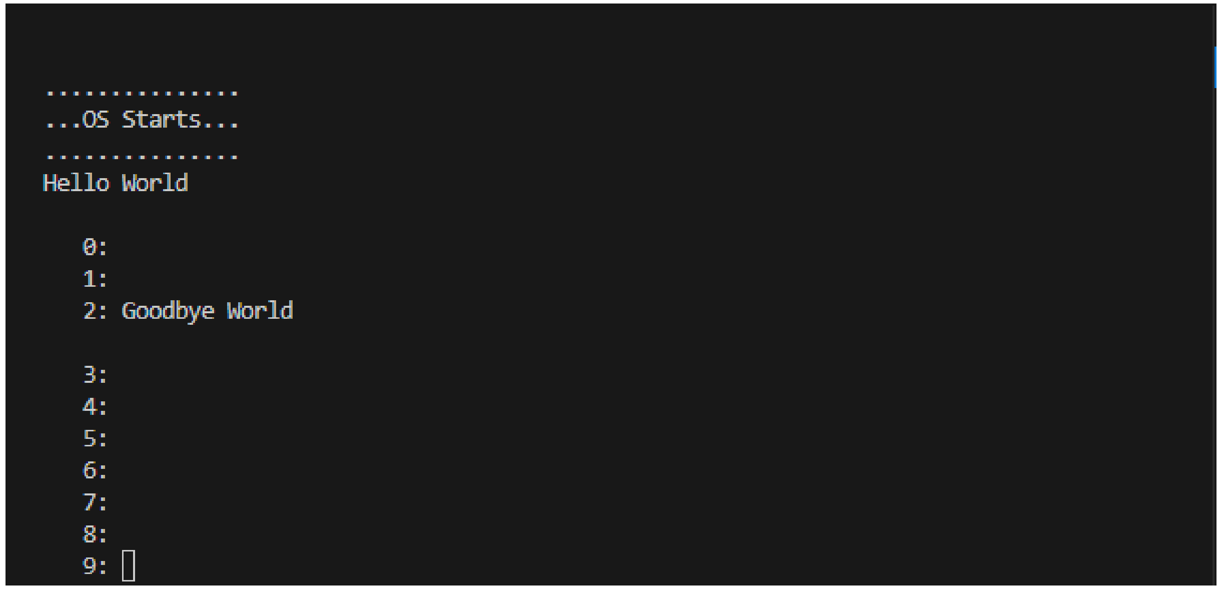


```
PS C:\Users\jongchank\work> pyserial-miniterm.exe COM4 115200
--- Miniterm on COM4  115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

...............
...OS Starts...
...............
Hello World
Goodbye World
```

04. Timeline

- TimerISR 이용 초단위 Timeline 출력

```
ISR2(TimerISR)
{
    static long c = 0;
    osEE_tc_stm_set_sr0_next_match(1000000U);
    printfSerial("\n%4ld: ", c++);
}
```



```
...............
...OS Starts...
...............
Hello World

   0:
   1:
   2: Goodbye World

   3:
   4:
   5:
   6:
   7:
   8:
   9: 
```

05. Tasks

```
TASK(Task1)
{
    printfSerial("Task1 Begins...");
    mdelay(3000);
    printfSerial("Task1 Finishes...");

    TerminateTask();
}

TASK(Task2)
{
    printfSerial("Task2 Begins...");
    mdelay(3000);
    printfSerial("Task2 Finishes...");

    TerminateTask();
}
```

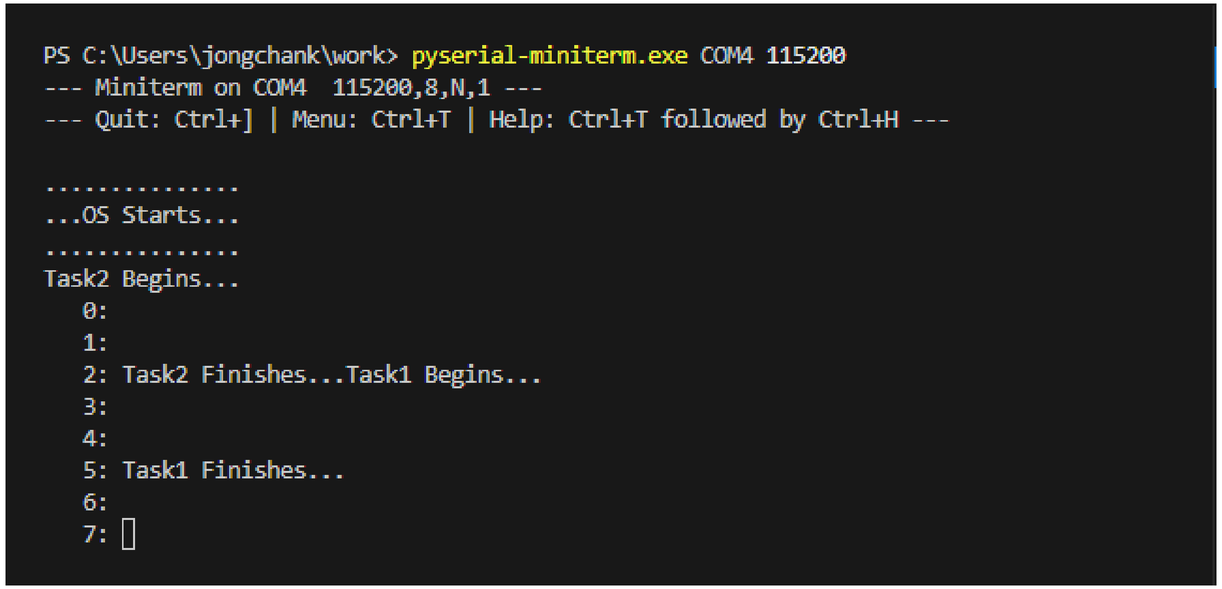
- 우선순위 2의 Task2 추가

클수록 높은 우선순위

```
TASK Task2 {
    PRIORITY = 2;
    STACK = SHARED;
    SCHEDULE = FULL;
    AUTOSTART = TRUE;
    ACTIVATION = 1;
};
```

05. Tasks

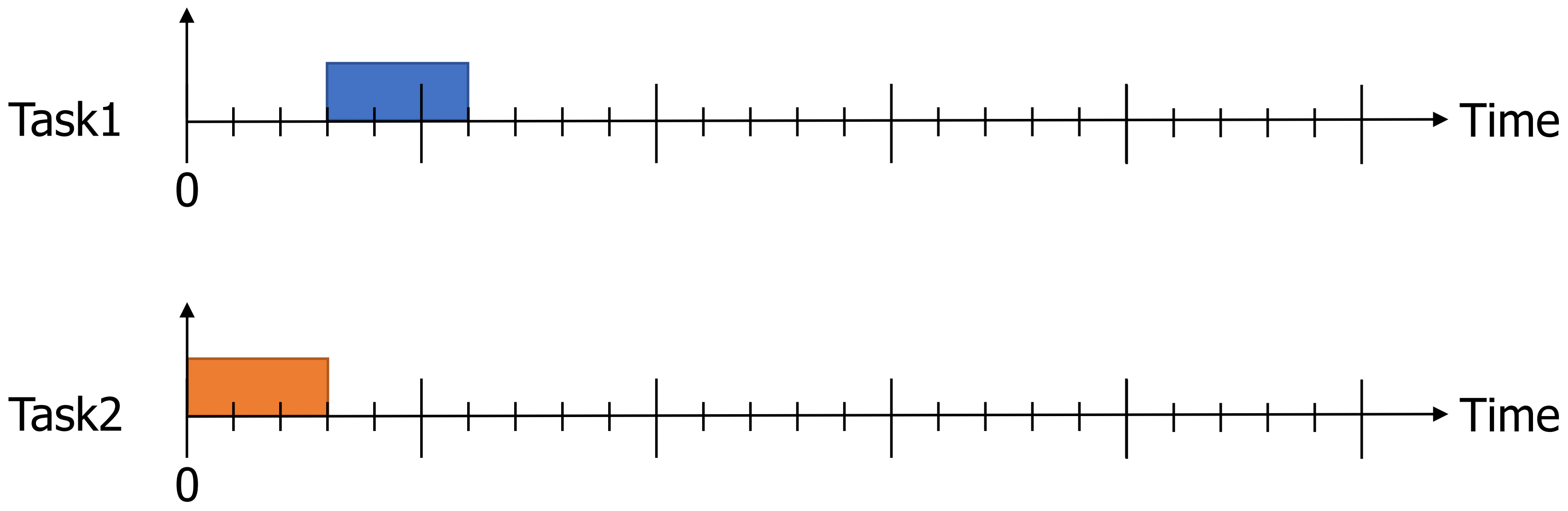
- Task2가 먼저 시작
- Task2 종료 후 Task1 시작



```
PS C:\Users\jongchank\work> pyserial-miniterm.exe COM4 115200
--- Miniterm on COM4  115200,8,N,1 ---
--- Quit: Ctrl+] | Menu: Ctrl+T | Help: Ctrl+T followed by Ctrl+H ---

...............
...OS Starts...
...............
Task2 Begins...
   0:
   1:
   2: Task2 Finishes...Task1 Begins...
   3:
   4:
   5: Task1 Finishes...
   6:
   7: 
```

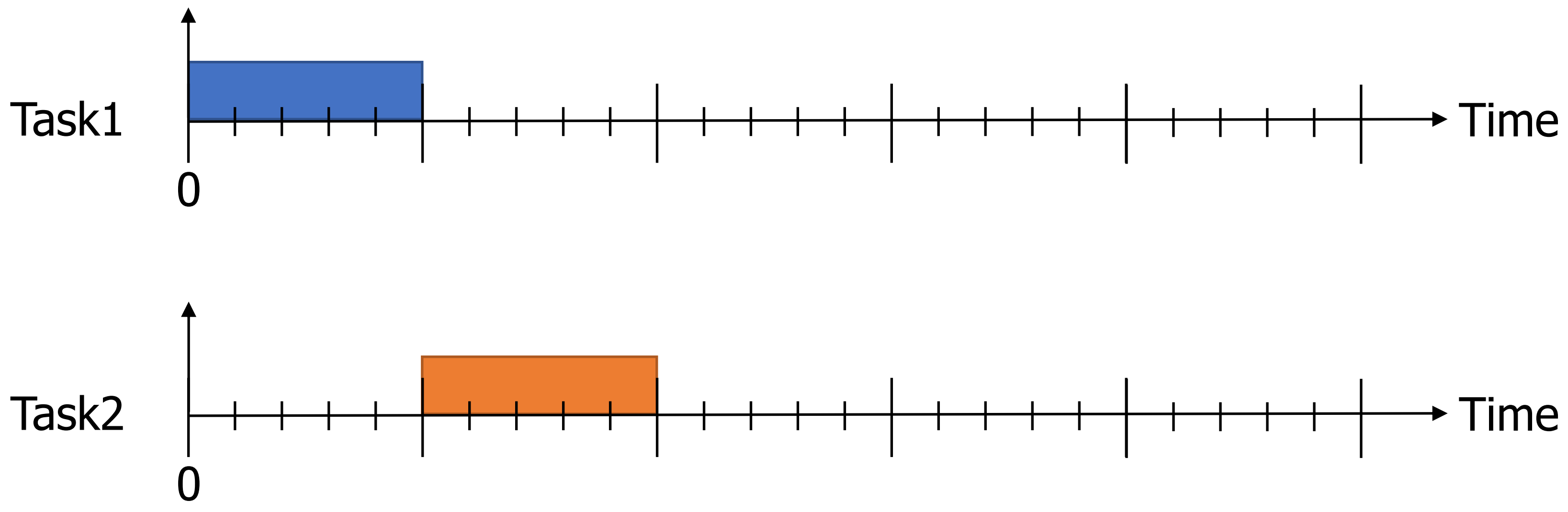
05-1. Tasks

- 실행결과
  - Task1: 낮은 우선순위, 실행 시간 3초
  - Task2: 높은 우선순위, 실행 시간 3초



05-2. Tasks

- [예제] 아래 조건의 Task들을 구현해보기
  - Task1: 높은 우선순위, 실행 시간 5초
  - Task2: 낮은 우선순위, 실행 시간 5초



06-1. Task Activation

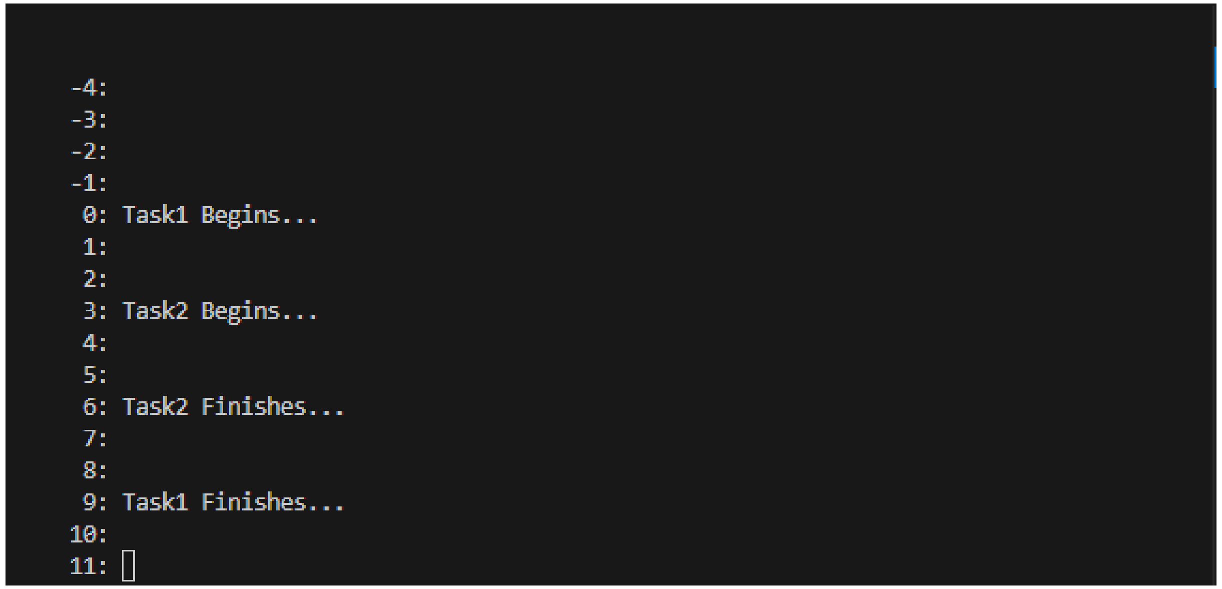
```
ISR2(TimerISR)
{
    static long c = -4;
    osEE_tc_stm_set_sr0_next_match(1000000U);
    if (c == 0)
        ActivateTask(Task1);
    printfSerial("\n%4ld: ", c++);
}

TASK(Task1)
{
    printfSerial("Task1 Begins...");
    mdelay(3000);
    ActivateTask(Task2);
    mdelay(3000);
    printfSerial("Task1 Finishes...");
    TerminateTask();
}
```

```
TASK(Task2)
{
    printfSerial("Task2 Begins...");
    mdelay(3000);
    printfSerial("Task2 Finishes...");
    TerminateTask();
}
```

```
TASK Task1 {
    …
    AUTOSTART = FALSE;
    …
};

TASK Task2 {
    …
    AUTOSTART = FALSE;
    …
};
```

06-1. Task Activation

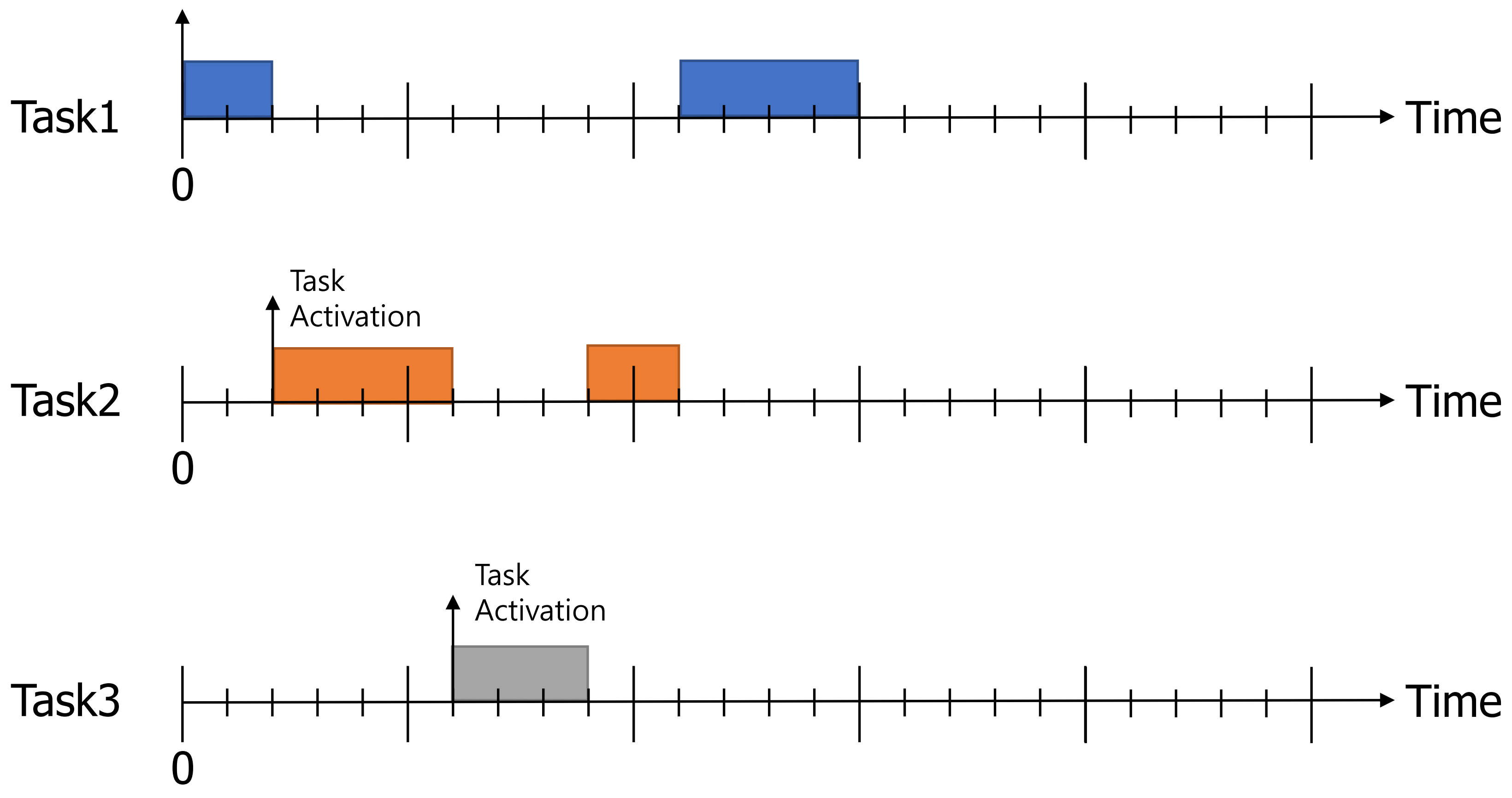
- Timeline -4부터 카운트다운
- Task2의 Task1 선점 확인



```
-4:
-3:
-2:
-1:
 0: Task1 Begins...
 1:
 2:
 3: Task2 Begins...
 4:
 5:
 6: Task2 Finishes...
 7:
 8:
 9: Task1 Finishes...
10:
11:
```

06-2. Task Activation

- [예제] ActivateTask 시점 변경 및 Task3까지 만들어서 연쇄 실행
- 아래 그림의 Task 구현해보기



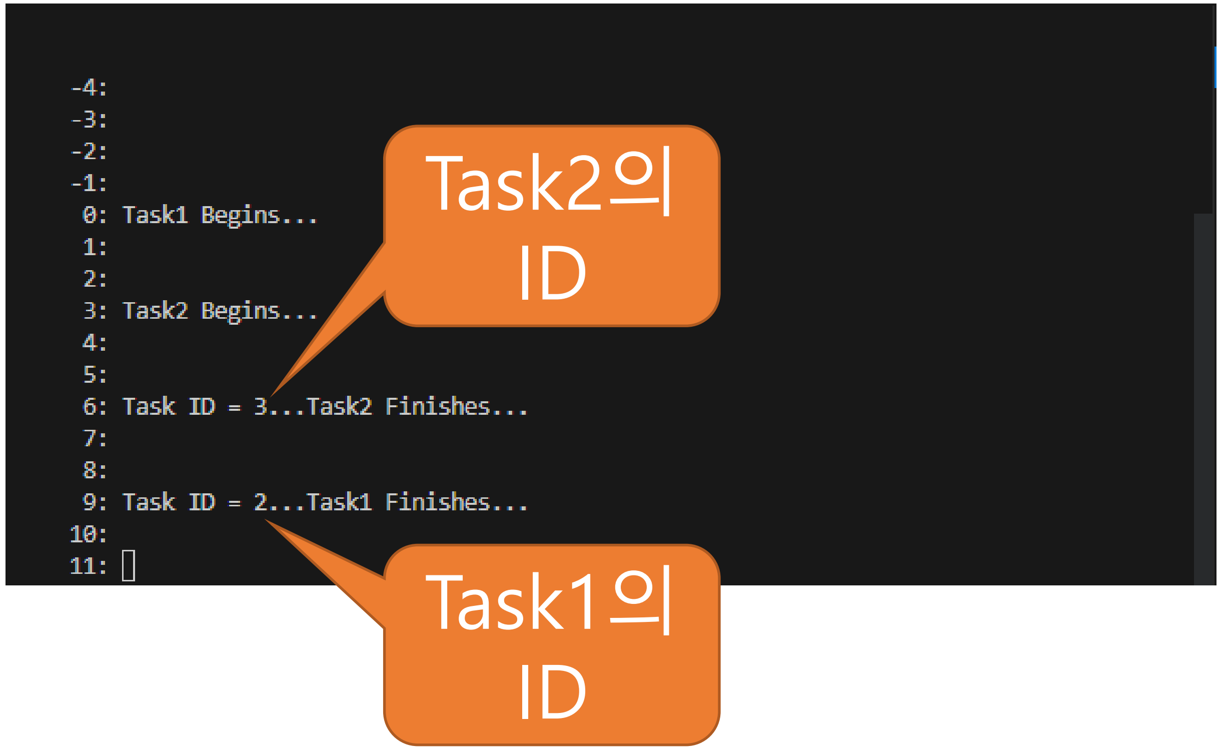
07. GetTaskID

```
TASK(Task1)
{
    TaskType id;
    printfSerial("Task1 Begins...");
    mdelay(3000);
    ActivateTask(Task2);
    mdelay(3000);
    GetTaskID(&id);
    printfSerial("Task ID = %d...", id);
    printfSerial("Task1 Finishes...");
    TerminateTask();
}
```

```
TASK(Task2)
{
    TaskType id;
    printfSerial("Task2 Begins...");
    mdelay(3000);
    GetTaskID(&id);
    printfSerial("Task ID = %d...", id);
    printfSerial("Task2 Finishes...");
    TerminateTask();
}
```

07. GetTaskID

- 자연수 Task ID 확인
- Unique ID일 뿐 정의된 의미 없음



```
-4:
-3:
-2:
-1:
 0: Task1 Begins...
 1:
 2:
 3: Task2 Begins...
 4:
 5:
 6: Task ID = 3...Task2 Finishes...
 7:
 8:
 9: Task ID = 2...Task1 Finishes...
10:
11:
```

08-1. GetTaskState

```
TASK(TaskM)
{
    printState(Task1);
    printState(Task2);

    TerminateTask();
}
```

```
TASK TaskM {
    PRIORITY = 3;
    STACK = SHARED;
    SCHEDULE = FULL;
    AUTOSTART = FALSE;
    ACTIVATION = 1;
};
```

```
void printState(TaskType id) {
    TaskStateType state;

    if (GetTaskState(id, &state) == E_OK) {
        switch (state) {
            case SUSPENDED:
                printfSerial("%d: suspended...", id);
                break;
            case READY:
                printfSerial("%d: ready...", id);
                break;
            case WAITING:
                printfSerial("%d: waiting...", id);
                break;
            case RUNNING:
                printfSerial("%d: running...", id);
                break;
        }
    }
}
```

08-1. GetTaskState

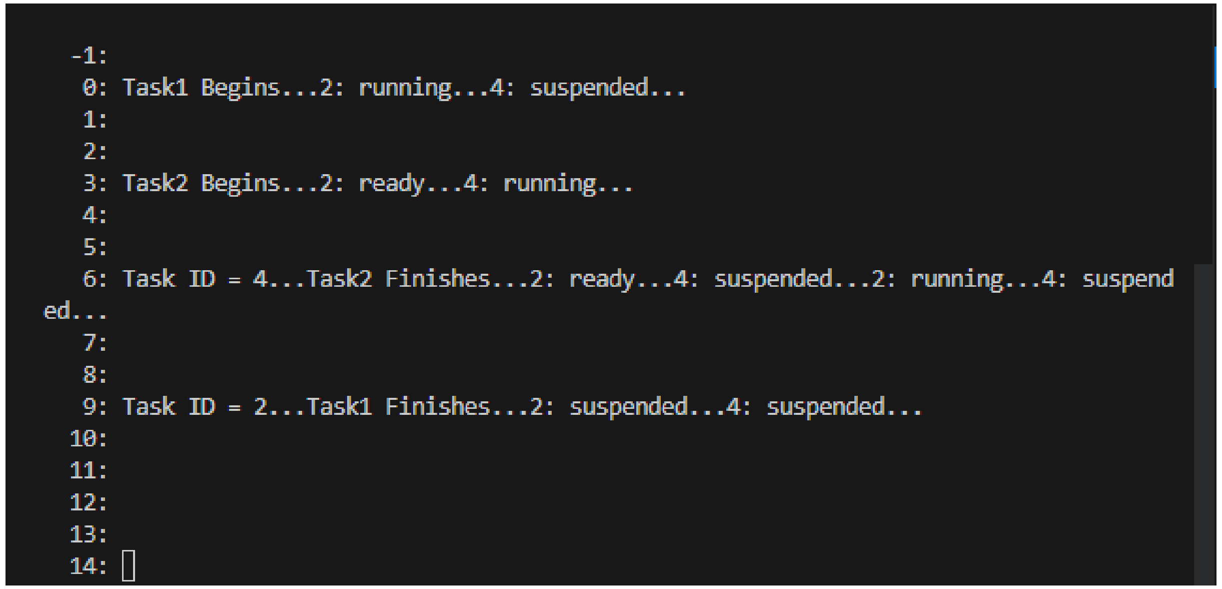
```
TASK(Task1)
{
    TaskType id;
    printfSerial("Task1 Begins...");
    printState(Task1);
    printState(Task2);
    mdelay(3000);
    ActivateTask(Task2);
    printState(Task1);
    printState(Task2);
    mdelay(3000);
    GetTaskID(&id);
    printfSerial("Task ID = %d...", id);
    printfSerial("Task1 Finishes...");
    ChainTask(TaskM);
}
```

```
TASK(Task2)
{
    TaskType id;
    printfSerial("Task2 Begins...");
    printState(Task1);
    printState(Task2);
    mdelay(3000);
    GetTaskID(&id);
    printfSerial("Task ID = %d...", id);
    printfSerial("Task2 Finishes...");
    ChainTask(TaskM);
}
```

08-1. GetTaskState

- Task 상태 변화 관찰

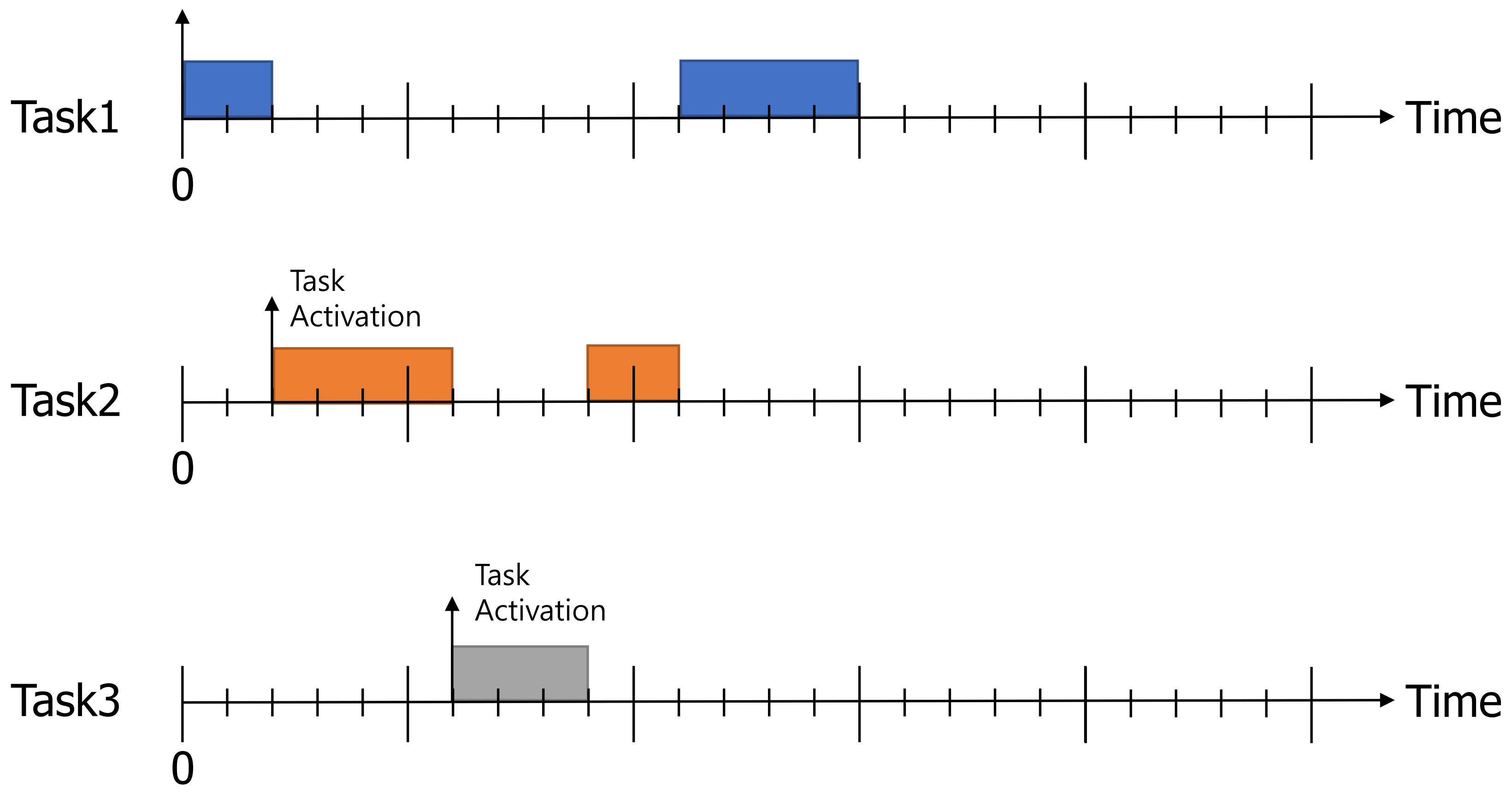
- 우선순위, Activation 패턴 변화의 영향은?



```
  -1:
   0: Task1 Begins...2: running...4: suspended...
   1:
   2:
   3: Task2 Begins...2: ready...4: running...
   4:
   5:
   6: Task ID = 4...Task2 Finishes...2: ready...4: suspended...2: running...4: suspend
ed...
   7:
   8:
   9: Task ID = 2...Task1 Finishes...2: suspended...4: suspended...
  10:
  11:
  12:
  13:
  14:
```

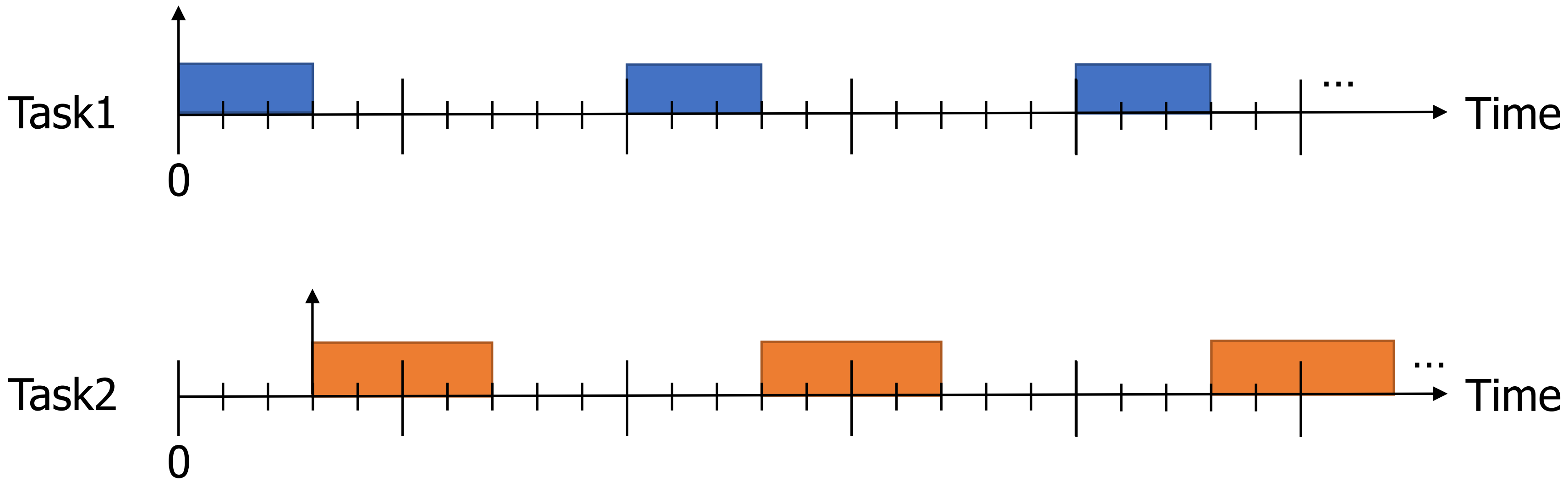
08-2. GetTaskState

- [예제] 06.Task Activation 2번째 실습에서 구현한 Task들 상태 변화 관찰



[예제] Tasks

- 아래 조건의 Task들을 구현해보기
  - Task1: 높은 우선순위, 실행 시간 3초
  - Task2: 낮은 우선순위, 실행 시간 4초
  - AUTOSTART = False로 수정하고 동일하게 구현
  - 10초마다 반복



Questions