Renode GR740 설치 및 디버깅 환경설정

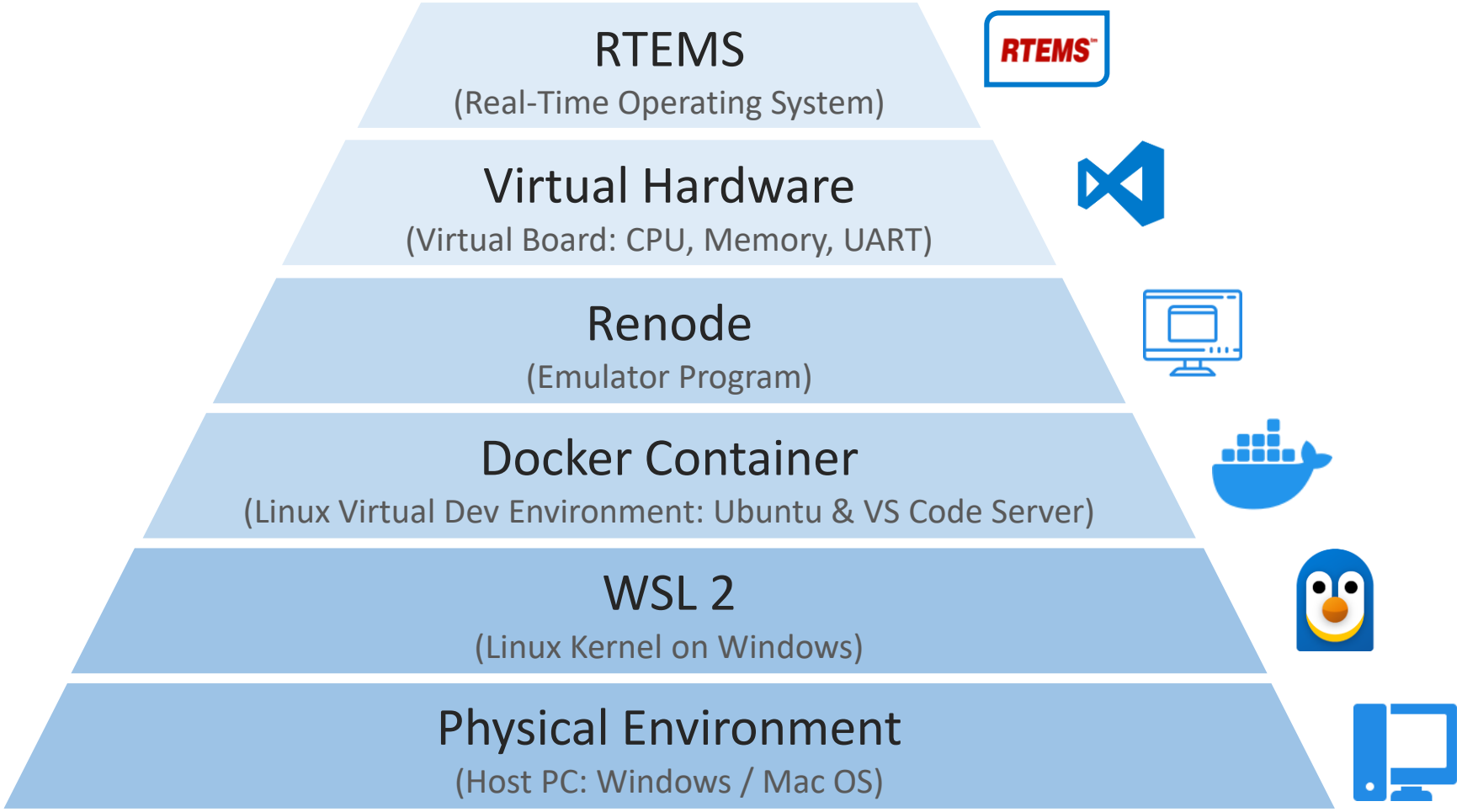
인하대학교 컴퓨터공학과

Intelligent Embedded SW 연구실

12211568 김관

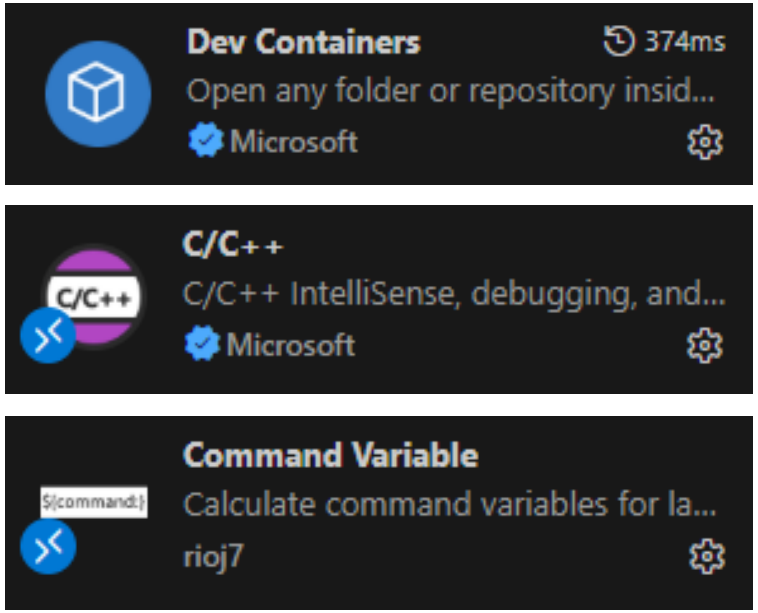
INDEX

Execution Layers



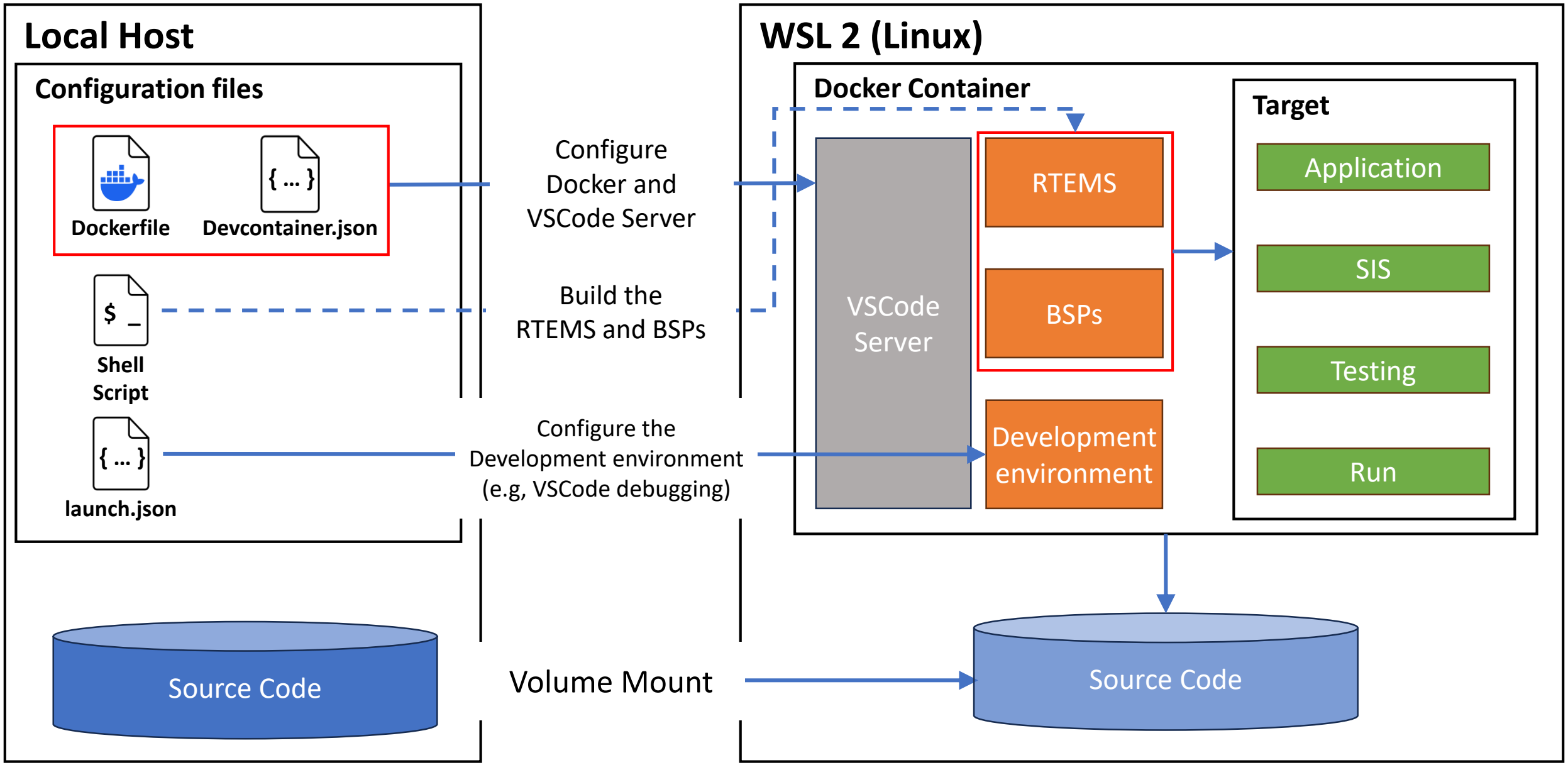
REQUIRED TOOLS

- **System Requirements**
  - 원활한 Dev Container 빌드 및 실행을 위해
    호스트 PC에 최소 20GB 이상의 디스크 여유 공간이 필요
- **Required Tools**
  - **WSL2**
  - **Docker**: Dev Container 실행을 위해 필요
  - **VSCode**: Visual Studio Code
  - **VSCode Extensions**:
    - Dev Containers (Micresoft)
    - C/C++ (Microsoft)
    - Command Variable (rioj7)



Reference: github.com/OBC-SIM/renode-gr740-docker by 박준용

REQUIRED TOOLS



REQUIRED TOOLS

- 관리자 권한으로 CMD 또는 Powershell을 실행

- CMD 또는 PowerShell에 다음 명령어를 입력

```
$ dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart      ….. (1)
$ dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all /norestart                ….. (2)
```

  - (1) WSL 활성화
  - (2) VirtualMachinePlatform 옵션 활성화

- WSL2 설치

```
$ wsl.exe --install                  ….. (1)
$ wsl.exe --update                   ….. (2)
$ wsl --set-default-version 2        ….. (3)
```

Reference: Real-Time Multicore OS RTEMS GR740 Install Guide by 김형호

REQUIRED TOOLS

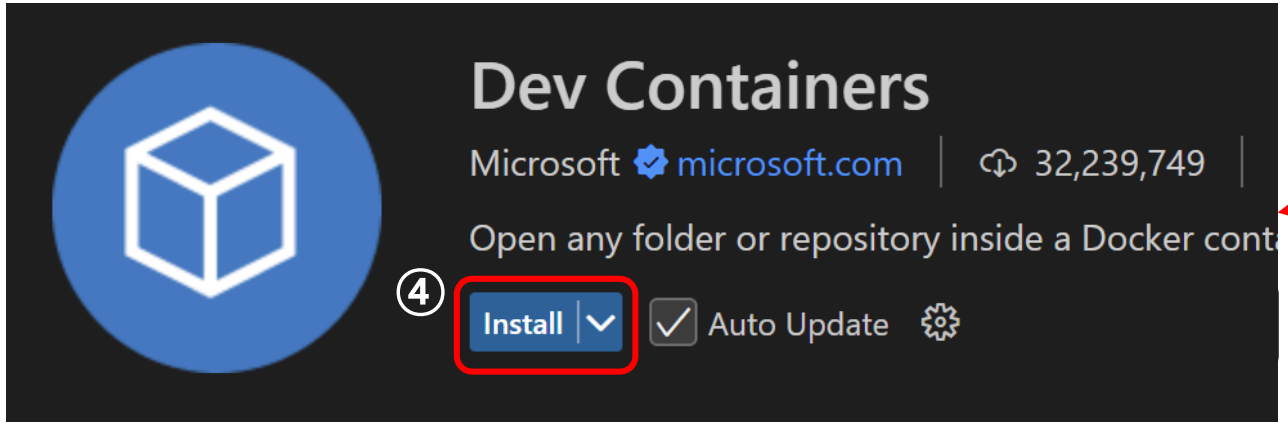
- **Docker Desktop for Windows 다운로드**
  - 공식 링크

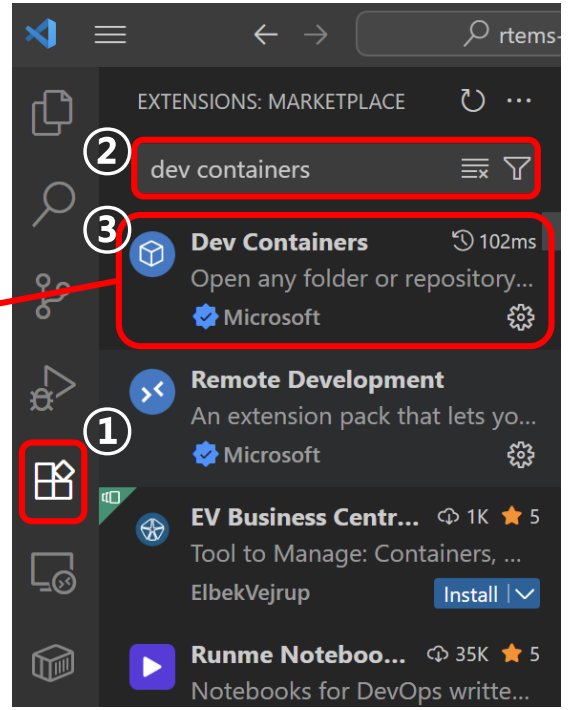
  https://docs.docker.com/desktop/setup/install/windows-install/

- **VSCode 다운로드**
  - 공식 링크

  https://code.visualstudio.com/

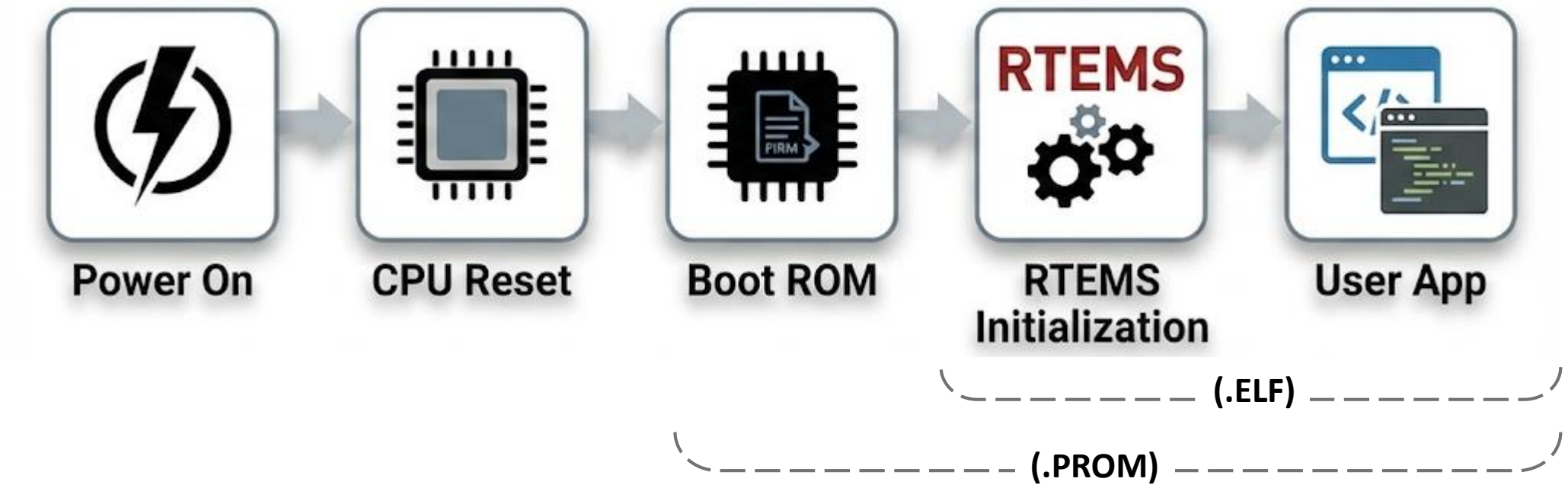
- **Open VSCode and Download extension : Dev Containers**





Reference: Real-Time Multicore OS RTEMS GR740 Install Guide by 김형호

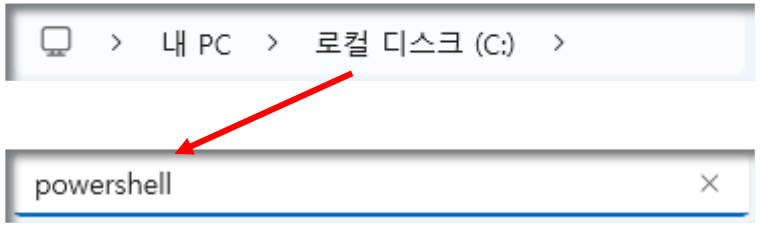
RTEMS Execution flow



Reference: Gemini Nano Banana Pro

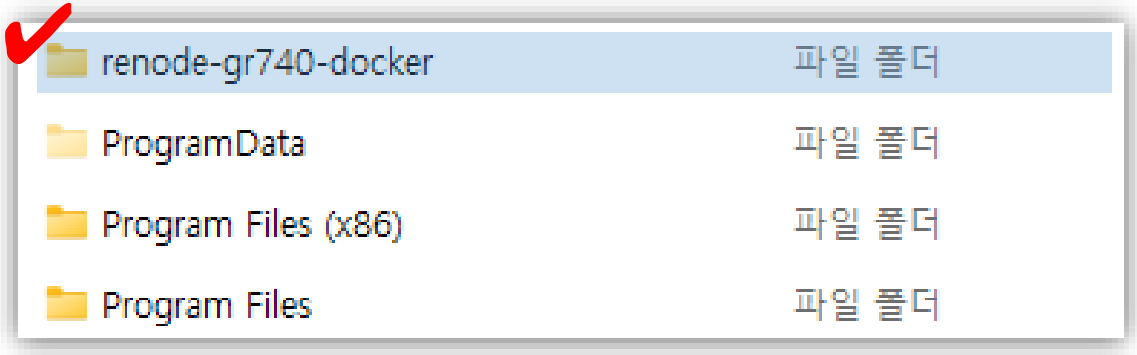
SETUP

- **Renode + GR740 VSCode Debugging Envinronment**
  - 다운로드하고자 하는 위치에서 CMD 또는 Powershell 실행
    - 다만, "User" 경로 제외
      "User" 경로에 있을 시 권한 문제가 발생할 수 있음



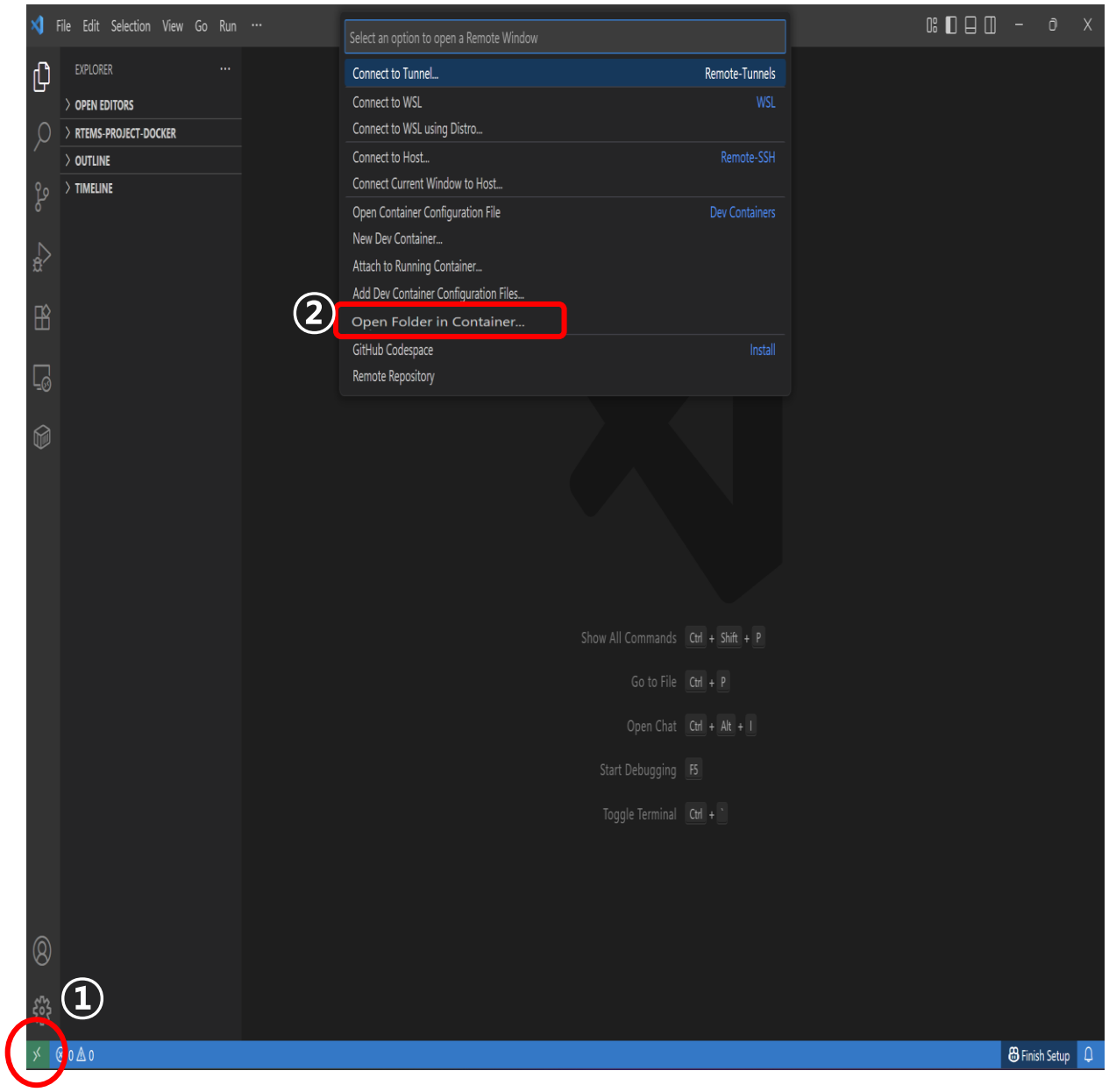
  - CMD 또는 Powershell에 다음 명령어를 입력

```
$ git clone https://github.com/OBC-SIM/renode-gr740-docker.git
```



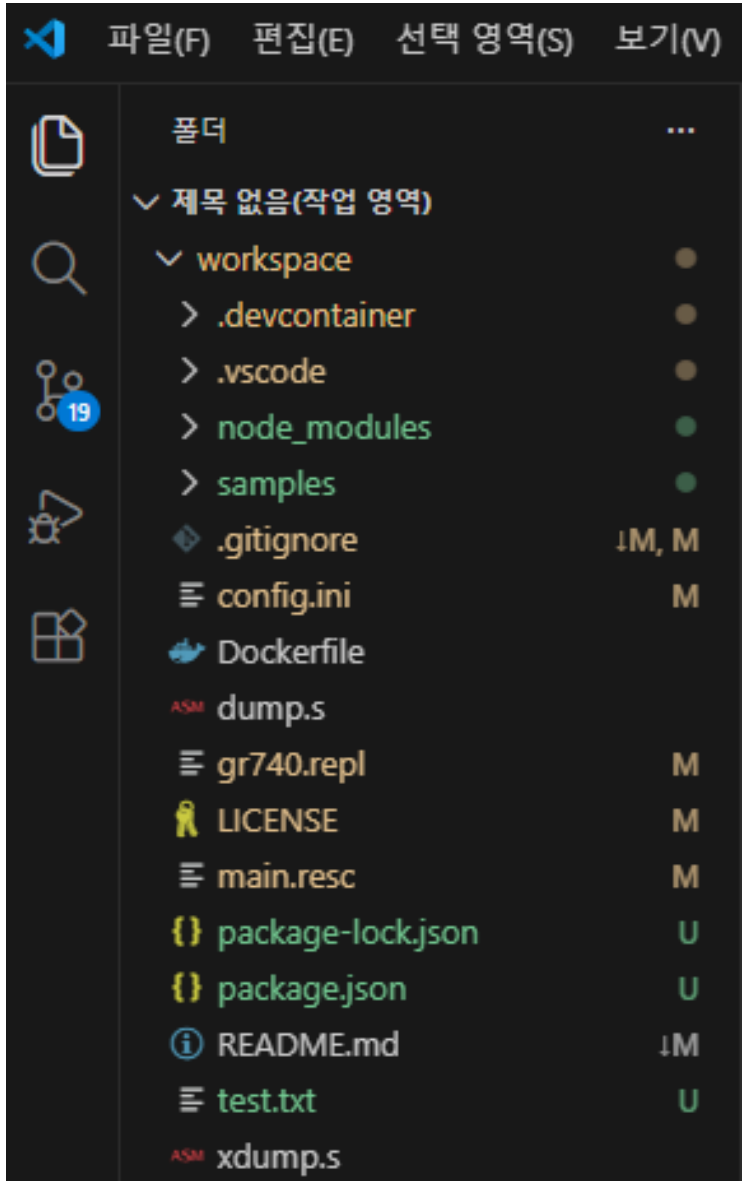
SETUP

- **Docker Container Setup**
  - Docker 실행
  - VS Code 실행
  -  클릭
  - ‘Open folder in Container…’ 선택
  - ‘renode-gr740-docker’ 선택



Reference: Real-Time Multicore OS RTEMS GR740 Install Guide by 김형호

SETUP


파일(F)
편집(E)
선택 영역(S)
보기(V)
폴더
제목 없음(작업 영역)
workspace
.devcontainer
.vscode
node_modules
samples
.gitignore
config.ini
Dockerfile
dump.s
gr740.repl
LICENSE
main.resc
package-lock.json
package.json
README.md
test.txt
xdump.s
19

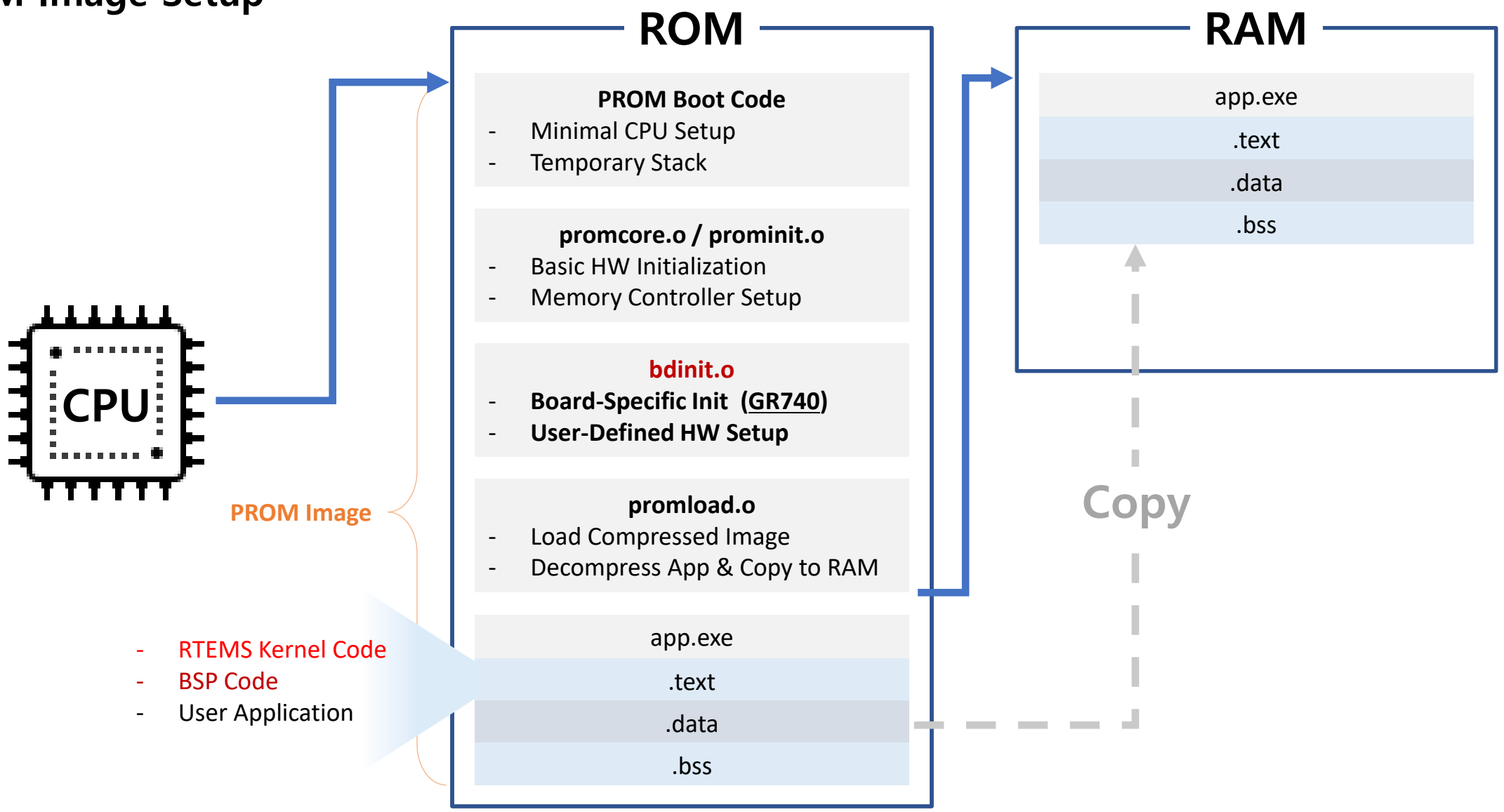
SETUP - BACKGROUND

- **PROM Image Setup**
  - CPU Reset 이후에 **부트 코드**의 시작주소로 이동 **(PC ← Reset Vector)**
  - **부트 코드**
    - **기본적인 하드웨어 초기화**
      - 스택 포인터 설정
      - 캐시/MMU 초기화
      - 메모리 컨트롤러 설정
      - **프로그램 코드를 RAM에 복사**

        ⋮
    - **RAM에 복사된 프로그램의 엔트리 주소로 이동 (PC ← App Entry Address)**

SETUP - BACKGROUND

- **PROM Image Setup**



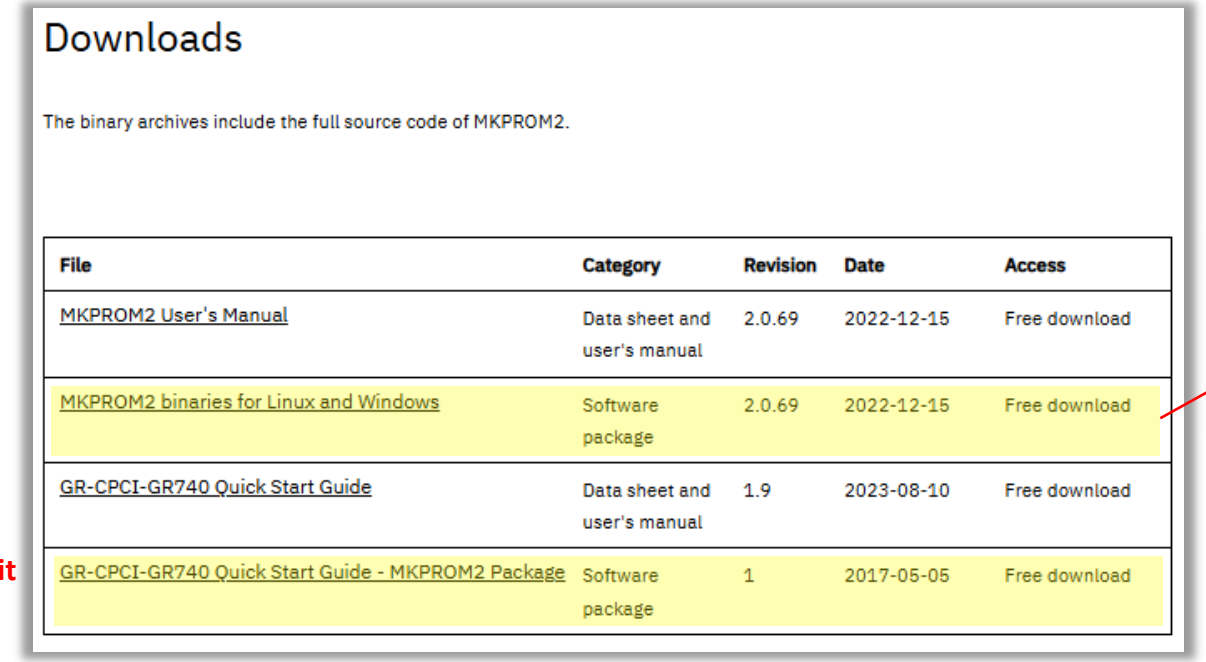
SETUP

- **PROM Image Setup**
  - MKPROM: 실행 파일(.exe, .elf)과 보드 초기화 루틴(bdinit)을 하나의 PROM 이미지(.prom)로 생성하는 툴
  - 공식 링크

https://www.gaisler.com/products/mkprom2

MKPROM 설치

GR740 bdinit 설치



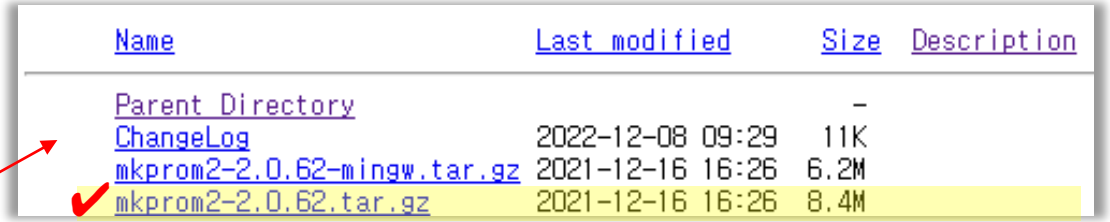
Downloads

The binary archives include the full source code of MKPROM2.

File	Category	Revision	Date	Access
MKPROM2 User's Manual	Data sheet and user's manual	2.0.69	2022-12-15	Free download
MKPROM2 binaries for Linux and Windows	Software package	2.0.69	2022-12-15	Free download
GR-CPCI-GR740 Quick Start Guide	Data sheet and user's manual	1.9	2023-08-10	Free download
GR-CPCI-GR740 Quick Start Guide - MKPROM2 Package	Software package	1	2017-05-05	Free download

Window용(-mingw.tar.gz)이 아닌

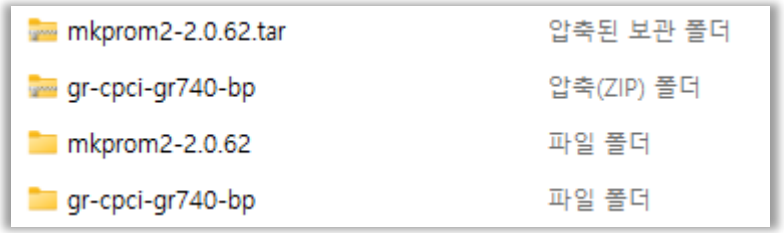
Linux용(.tar.gz)을 다운로드

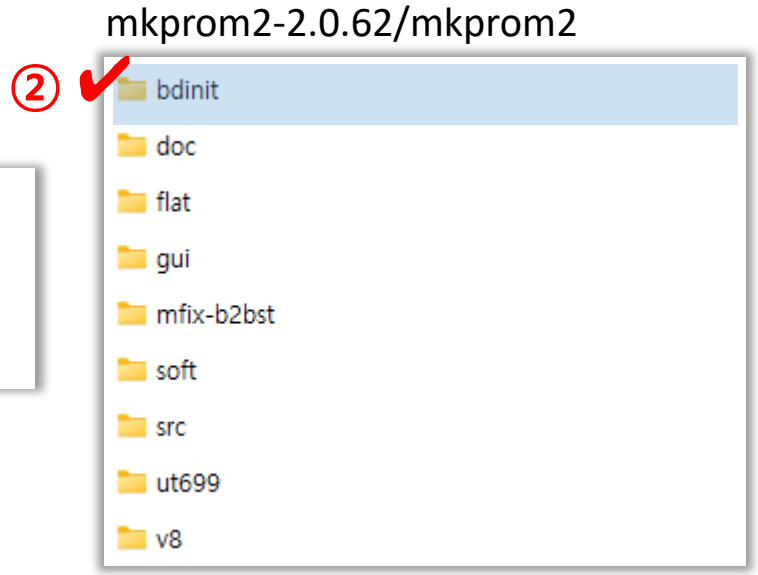


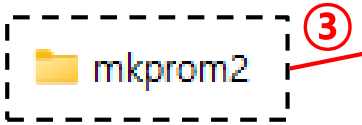
Name	Last modified	Size	Description
Parent Directory		-	
ChangeLog	2022-12-08 09:29	11K	
mkprom2-2.0.62-mingw.tar.gz	2021-12-16 16:26	6.2M	
mkprom2-2.0.62.tar.gz	2021-12-16 16:26	8.4M	

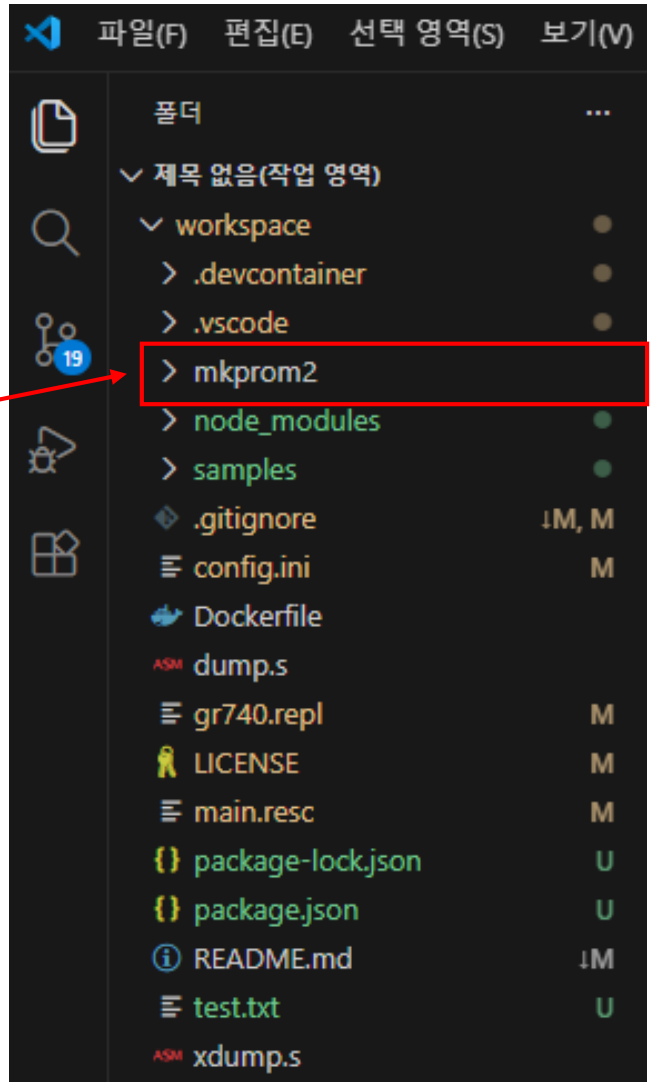
SETUP

- **PROM Image Setup**
  - 압축 해제
  - gr-cpci-gr740-bp 안의 bdinit 폴더를 mkprom2 폴더로 복사
  - mkprom2 폴더를 /workspace로 이동









SETUP

- **PROM Image Setup**
  - 실행 권한 부여
    - VSCode 터미널에 다음 명령어를 입력

```
$ sudo chmod +x /workspace/mkprom2/mkprom2
```

  - mkprom.c 수정 (/workspace/mkprom2/src/mkprom.c)
    - 기본 경로를 현재 mkprom의 위치로 지정

```
63  #ifndef TOOLBASE
64  #define TOOLBASE "/workspace/mkprom2"
65  #endif
```

  - 보드 초기화 루틴(bdinit.o) 빌드 및 현재 폴더로 복사

```
$ cd /workspace/mkprom2/bdinit                    ..... (1)  Change directory to source
$ make -f Makefile.bdinit GCC=sparc-rtems6-gcc    ..... (2)  Compile bdinit.o
```

EMULATE WITH RENODE

- **Build the Hello-World Sample**

```
$ cd /workspace/samples/hello-world      ..... (1)  Change directory to hello-world
$ make                                   ..... (2)  Build the app.exe
```

  - b-gr740 폴더가 생성되고 app.exe 파일이 생성됨

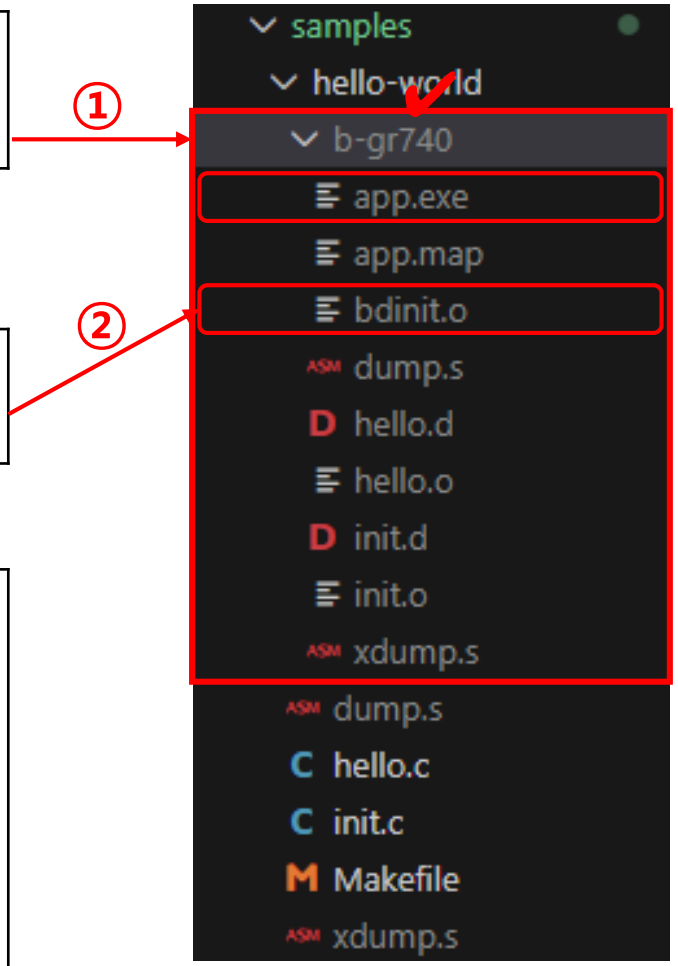
- **Copy bdinit.o to the folder**

```
$ cd /workspace/mkprom2/bdinit
$ cp bdinit.o /workspace/samples/hello-world/b-gr740/
```

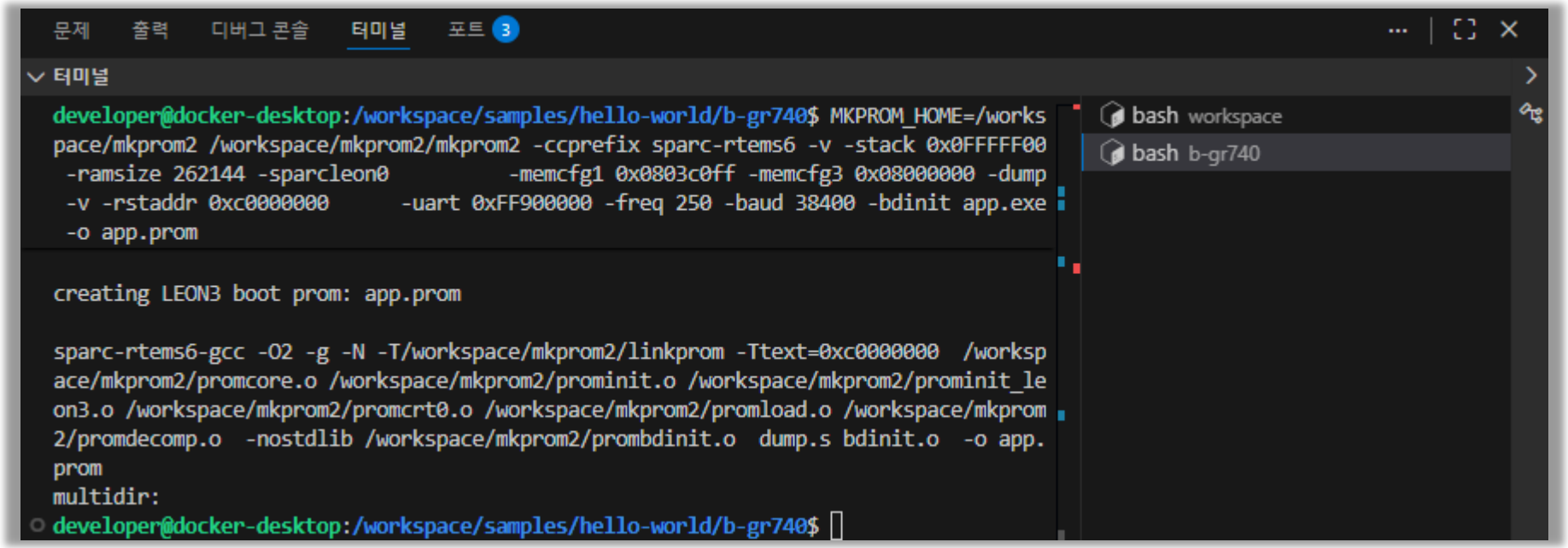
- **Build the PROM Image**

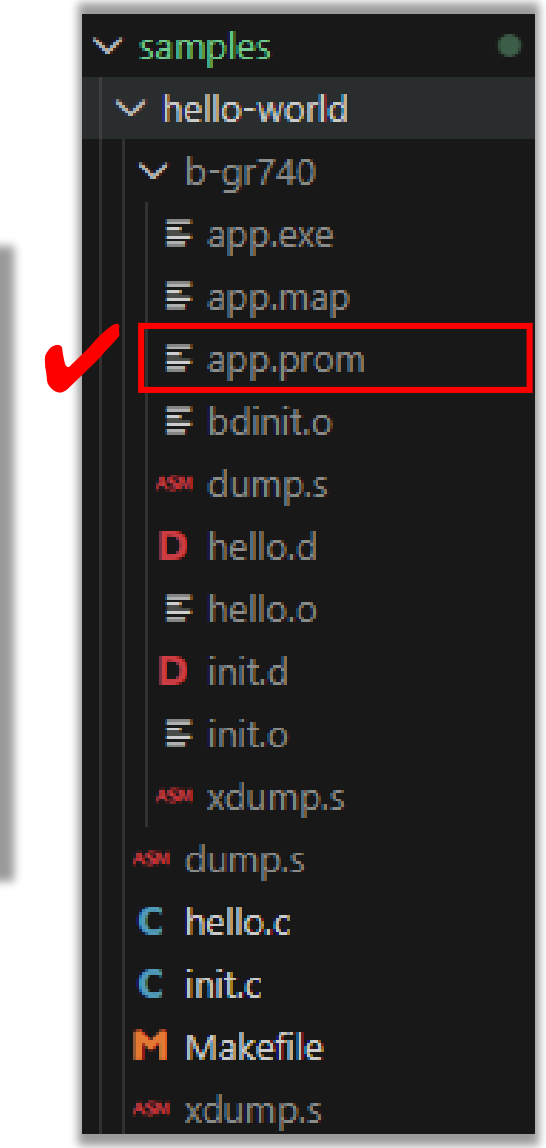
```
$ cd /workspace/samples/hello-world/b-gr740
$ export MKPROM_HOME=/workspace/mkprom2

$ $MKPROM_HOME/mkprom2 \
  -ccprefix sparc-rtems6 -v -stack   0x0FFFFF00 -ramsize 262144 -sparcleon0 \
  -memcfg1 0x0803c0ff -memcfg3 0x08000000 -rstaddr 0xc0000000 \
  -uart 0xFF900000 –freq 250 –baud 38400 -dump -bdinit  app.exe \
  -o app.prom
```

  - 띄어쓰기 모두 수정하여 명령어 입력



EMULATE WITH RENODE



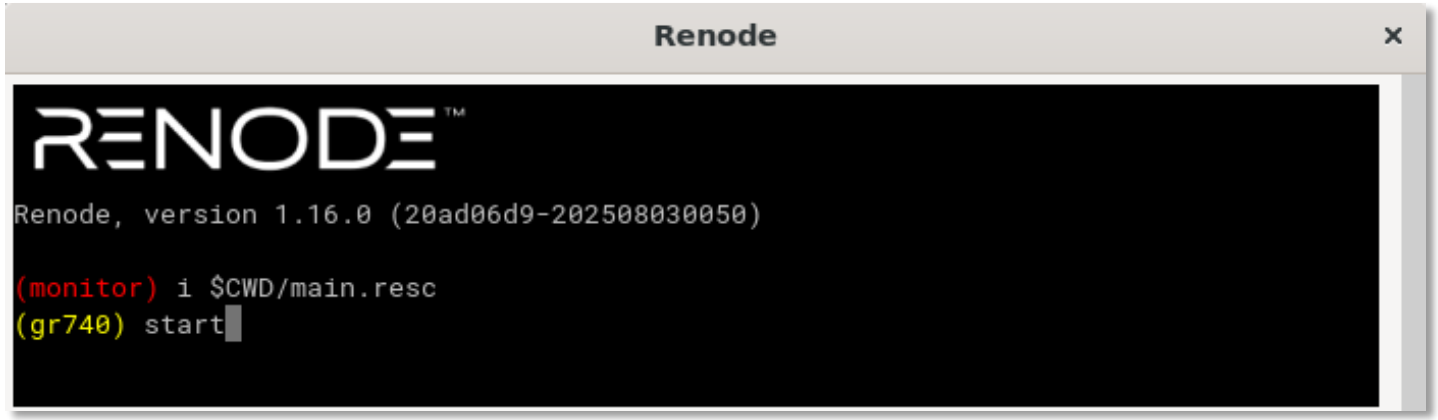


EMULATE WITH RENODE

- **Launch Renode and Start Emulation**
  - gr740.repl : GR740 보드/주변장치 정의
  - app.prom : 실행 이미지
  - **main.resc** : 이 둘을 메모리에 올린 뒤, Renode에게 실행 지시

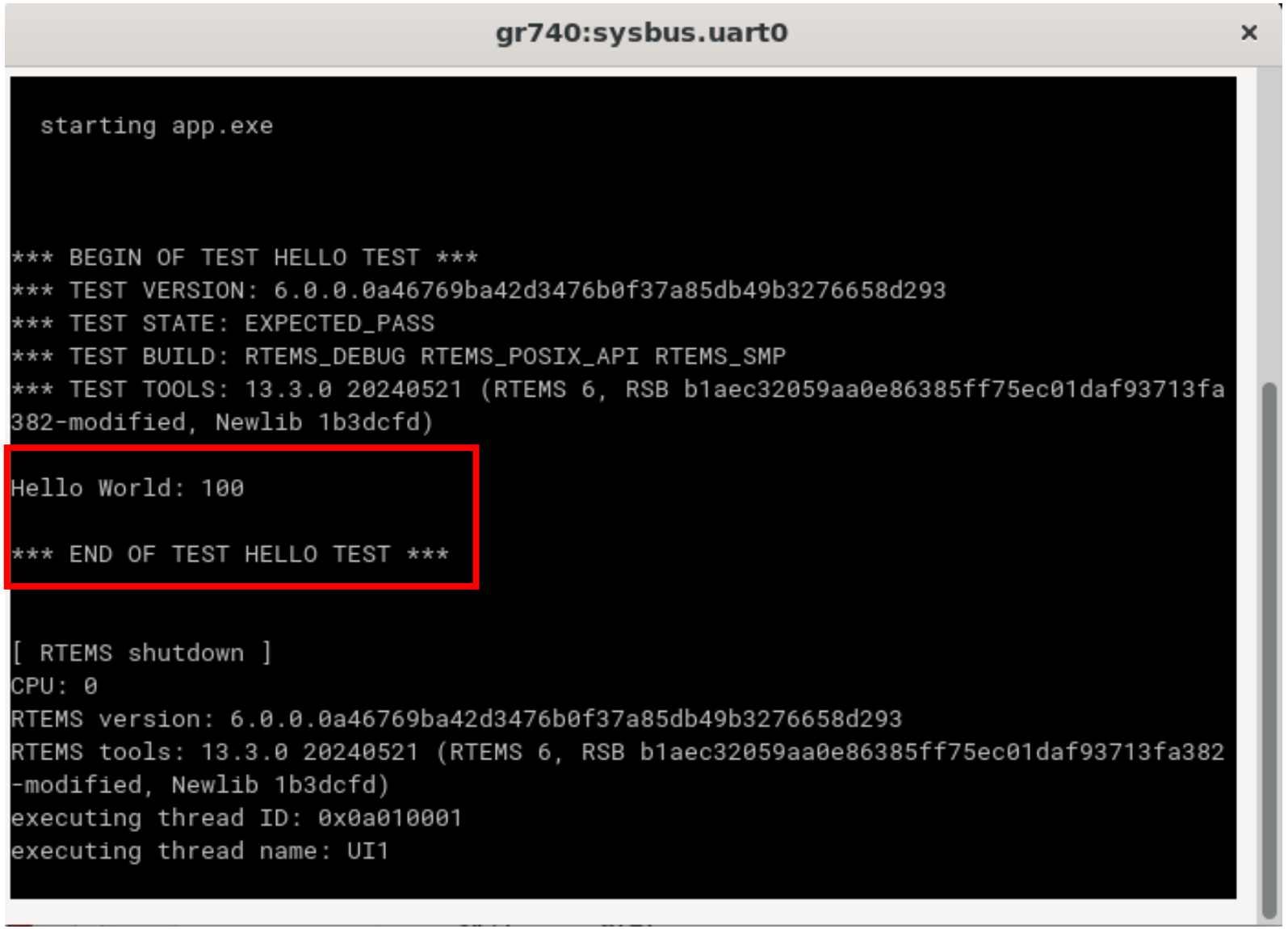
$ cd /workspace	**….. (1) Change directory to main.resc location**
$ renode main.resc	**….. (2) Execute the Renode script**

  - **Renode** GUI에서 **start** 입력



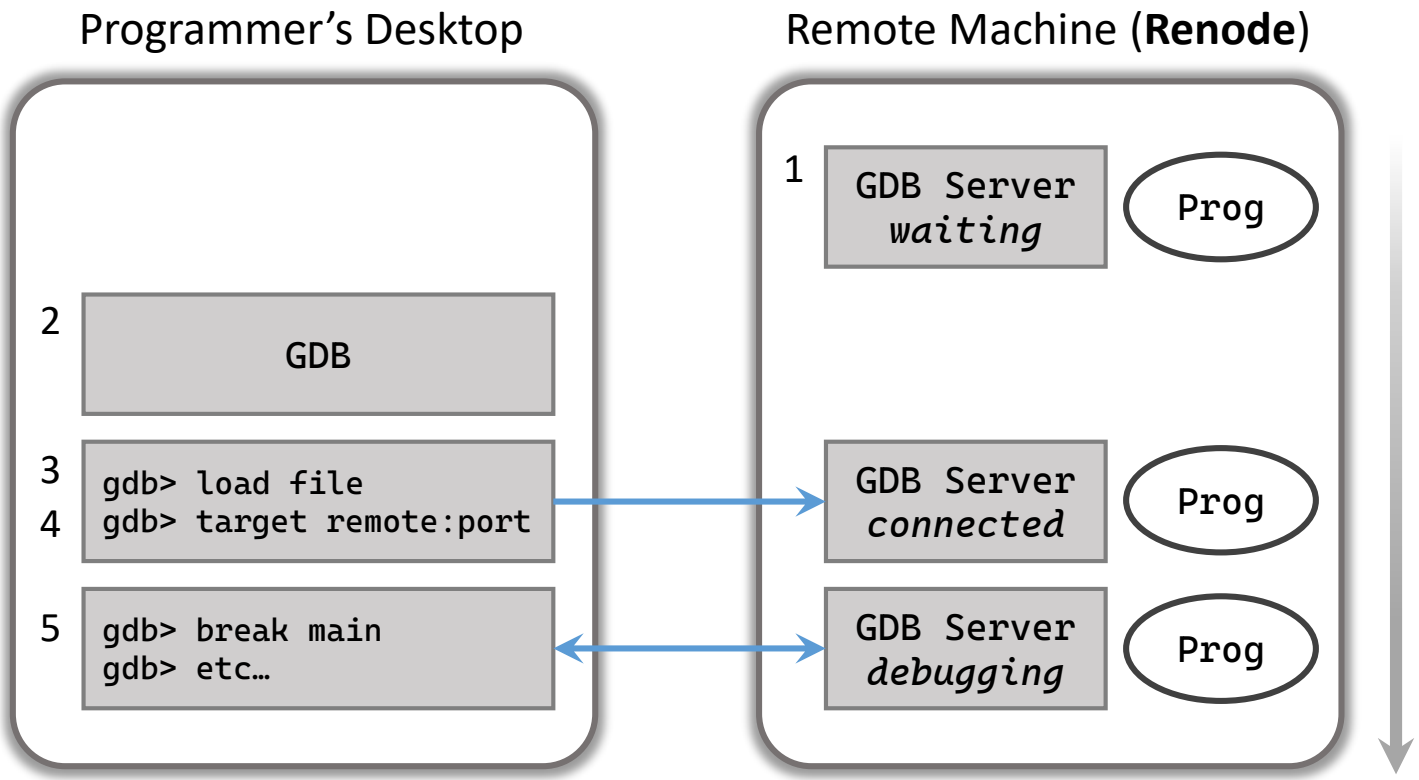
EMULATE WITH RENODE

Emulation Succeed!



DISASSEMBLE WITH RENODE

- **GDB-Based Debugging Architecture**
  - Renode/SIS는 GDB 서버를 내장한 에뮬레이터
  - 개발자는 이 서버에 연결해 원격 디버깅을 수행



DISASSEMBLE WITH RENODE

- **Launch Renode**

```
$ renode /workspace/main.resc
```

  - GDB 서버가 정상적으로 켜져있는지 확인

```
(gr740) machine StartGdbServer 3333
```

    - 이러한 메시지가 뜨면, 이미 정상적으로 작동 중

```
(gr740) machine StartGdbServer 3333
There was an error executing command 'machine StartGdbServer 3333'
CPU: gr740.cpu is already attached to an existing GDB server, running on port :3333
```

  - GDB 서버 껐다 켜기

```
(gr740) machine StopGdbServer
(gr740) machine StartGdbServer 3333
```

DISASSEMBLE WITH RENODE

- **Run GDB on Hello-World Sample**
  - 새 터미널 열기 (Ctrl + Shift + `)
  - GDB 활성화

```
$ sparc-rtems6-gdb /workspace/samples/hello-world/b-gr740/app.exe
```

    - GDB는 **심볼 정보**를 담고 있는 실행파일(.elf, .exe)을 대상으로 동작함

  - Renode GDB 서버 연결

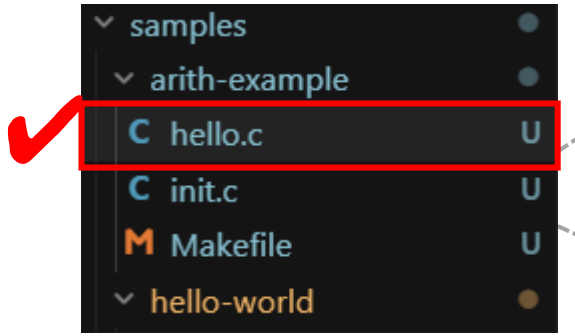
```
$ (gdb) target remote :3333
$ (gdb) disassemble Init
```

Disassemble with Renode

```
(gdb) disassemble Init
Dump of assembler code for function Init:
   0x00001258 <+0>:     save  %sp, -104, %sp
   0x0000125c <+4>:     sethi  %hi(0x20400), %i5
   0x00001260 <+8>:     clr  %o1
   0x00001264 <+12>:    call  0x2d2c <rtems_test_begin>
   0x00001268 <+16>:    or  %i5, 0x238, %o0
   0x0000126c <+20>:    mov  0x64, %g1
   0x00001270 <+24>:    st  %g1, [ %fp + -4 ]
   0x00001274 <+28>:    sethi  %hi(0), %g1
   0x00001278 <+32>:    ld  [ %fp + -4 ], %o3
   0x0000127c <+36>:    xor  %g1, -36, %g1
   0x00001280 <+40>:    ld  [ %g7 + %g1 ], %o0
   0x00001284 <+44>:    sethi  %hi(0x20400), %o2
   0x00001288 <+48>:    sethi  %hi(0x1d000), %o1
   0x0000128c <+52>:    or  %o2, 0x228, %o2
   0x00001290 <+56>:    call  0x19c40 <fiprintf>
   0x00001294 <+60>:    or  %o1, 0x190, %o1
   0x00001298 <+64>:    call  0x2d60 <rtems_test_end>
   0x0000129c <+68>:    or  %i5, 0x238, %o0
   0x000012a0 <+72>:    call  0x19ddc <exit>
   0x000012a4 <+76>:    clr  %o0
   0x000012a8 <+80>:    nop
End of assembler dump.
```

CREATE A NEW SAMPLE

- **Make new sample**
  - 새로운 코드(폴더) 생성

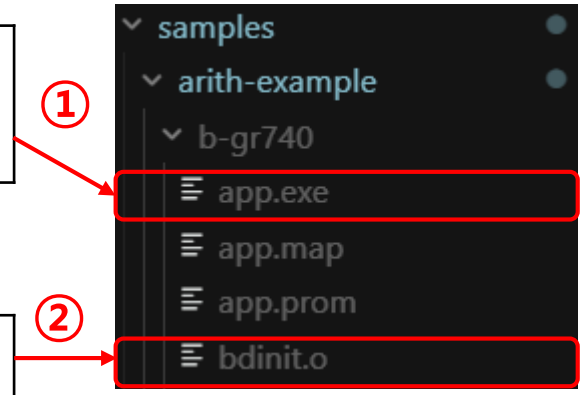



```c
TEST_BEGIN();

/* 간단한 피연산자 예제 */
int a = 10;
int b = 3;

int sum  = a + b;
int diff = a - b;

printf("\n=== Basic Arithmetic Sample ===\n");
printf("a = %d, b = %d\n", a, b);
printf("a + b = %d\n", sum);
printf("a - b = %d\n", diff);
```

samples/arith-example/hello.c

- **Build the arith-example Sample**

```
$ cd /workspace/samples/arith-example    ..... (1)  Change directory to new file
$ make                                   ..... (2)  Build the app.exe
```



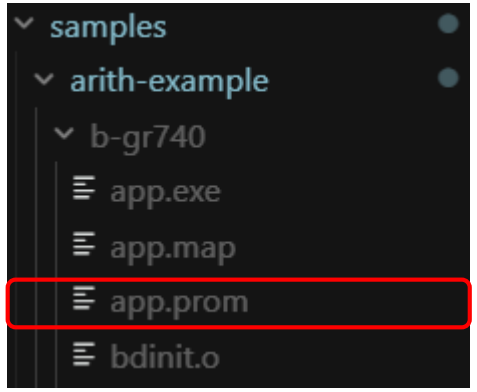
- **Copy bdinit.o to the folder**

```
$ cd /workspace/mkprom2/bdinit
$ cp bdinit.o /workspace/samples/arith-example/b-gr740/
```

CREATE A NEW SAMPLE

- **Make new code**
  - **Build the PROM Image**

```
$ cd /workspace/samples/arith-example/b-gr740
$ export MKPROM_HOME=/workspace/mkprom2

$ $MKPROM_HOME/mkprom2 -ccprefix sparc-rtems6 -v -stack   0x0FFFFF00 -ramsize
262144 -sparcleon0 -memcfg1 0x0803c0ff -memcfg3 0x08000000 -rstaddr
0xc0000000 -uart 0xFF900000 –freq 250 –baud 38400 -dump -bdinit  app.exe -o
app.prom app.exe
```



  - **Modify main.resc file**
    - main.resc는 부팅 코드와 실행 파일의 모음집
    - 새로 생성한 파일의 .prom을 가리키도록 수정

```
$name?="gr740"
$bin?=@/workspace/samples/arith-example/b-gr740/app.prom
$repl?=@/workspace/gr740.repl

using sysbus
mach create $name

machine LoadPlatformDescription $repl
showAnalyzer uart0
```

CREATE A NEW SAMPLE

- **Launch Renode and Start Emulation**

```
$ cd /workspace                    ..... (1)  Change directory to main.resc location
$ renode main.resc                 ..... (2)  Execute the Renode script
```

- **Run GDB on Hello-World Sample**
  - 새 터미널 열기 (Ctrl + Shift + `)
  - GDB 활성화

```
$ sparc-rtems6-gdb /workspace/samples/arith-example/b-gr740/app.exe
```

  - Renode GDB 서버 연결

```
$ (gdb) target remote :3333
$ (gdb) disassemble Init
```

```
(gdb) target remote :3333
Remote debugging using :3333
0xc0000000 in ?? ()
```

CREATE A NEW SAMPLE

- **Run GDB on Hello-World Sample**
  - break point 설정

$ (gdb) break Init
$ (gdb) info break

```
(gdb) break Init
Breakpoint 1 at 0x1258: file hello.c, line 21.
(gdb) info break
Num     Type           Disp Enb Address    What
1       breakpoint     keep y   0x00001258 in Init at hello.c:21
```

  - 작성한 함수 disassemble하여 제대로 동작하는지 확인

$ (gdb) disassemble Init
$ (gdb) info break

```
/* 간단한 피연산자 예제 */
int a = 10;
int b = 3;
```

samples/arith-example/hello.c

```
0x00001284 <+44>:    ld   [ %g7 + %i5 ], %o0
0x00001288 <+48>:    mov  3, %o3
0x0000128c <+52>:    mov  0xa, %o2
0x00001290 <+56>:    sethi  %hi(0x1d000), %o1
0x00001294 <+60>:    call  0x19ca0 <fiprintf>
0x00001298 <+64>:    or  %o1, 0x218, %o1     ! 0x1d218
0x0000129c <+68>:    ld   [ %g7 + %i5 ], %o0
```