

임베디드 시스템

최 민

● 학습목표

- JTAG 디버거 개념과 Keil MDK 의 디버깅 기능

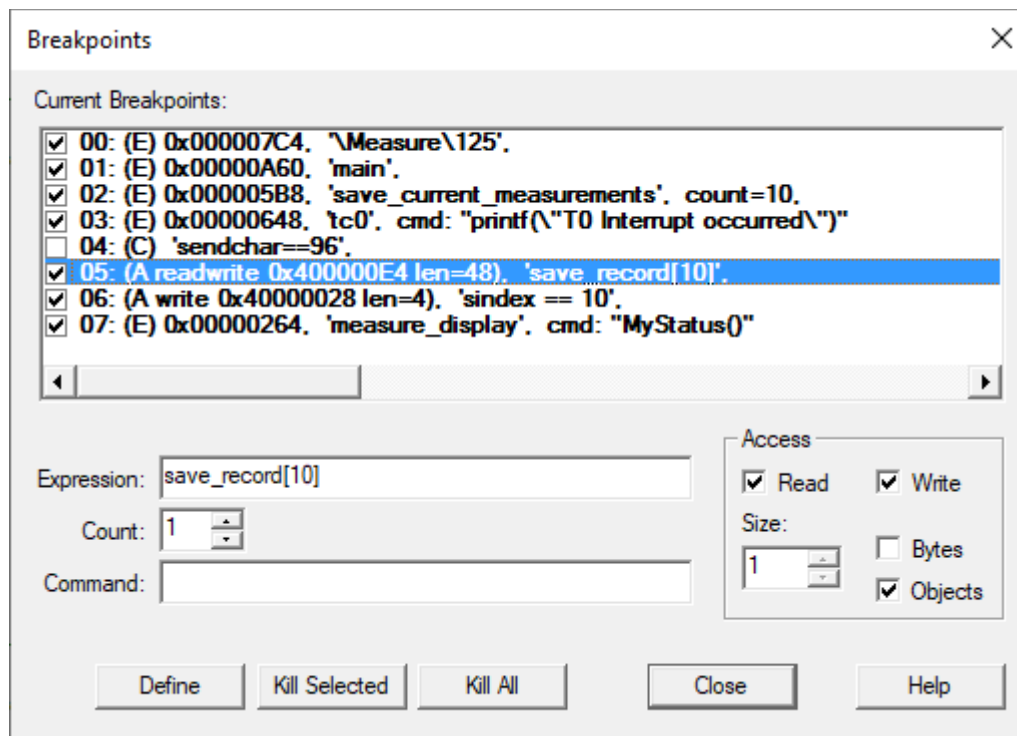
● 학습활동

- JTAG 디버거 개념에 대하여 이해합니다.
- Keil MDK 에서 제공하는 디버깅 기능을 살펴봅니다.

Keil uVision MDK의 디버깅

● Keil uVision MDK IDE의 디버깅 기능

- Breakpoints are program addresses or expressions that, when TRUE, halt program execution or execute a specified command. Breakpoints can be defined and modified in several ways:



Keil uVision MDK의 디버깅

● Access Break (A)

- defined when the flags Read, or Write, or both are set. The breakpoint is triggered when the specified memory access occurs.

● Execution Break (E)

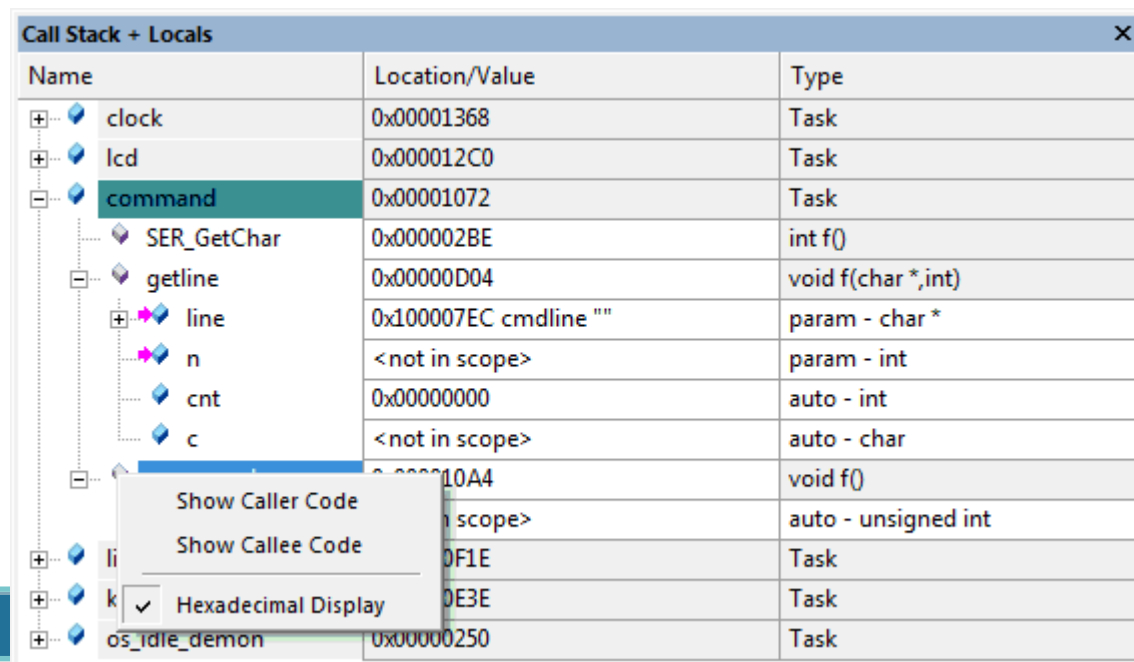
- defined when Expression resolves to a code address. The breakpoint triggers when the specified code address is reached

● Conditional Break (C)

- defined when Expression cannot be reduced to an address. The breakpoint triggers when the conditional expression becomes TRUE

Keil uVision MDK의 디버깅

- **The Call Stack + Locals window shows objects that are currently on stack.**
 - RTX-RTOS를 사용하는 application에 대한 예시
 - 각 object는 location/value, type에 대해서 제시됨
 - 리스트의 top에 현재 수행중인 function이, 이를 호출한 function이 그 밑에 표시됨.

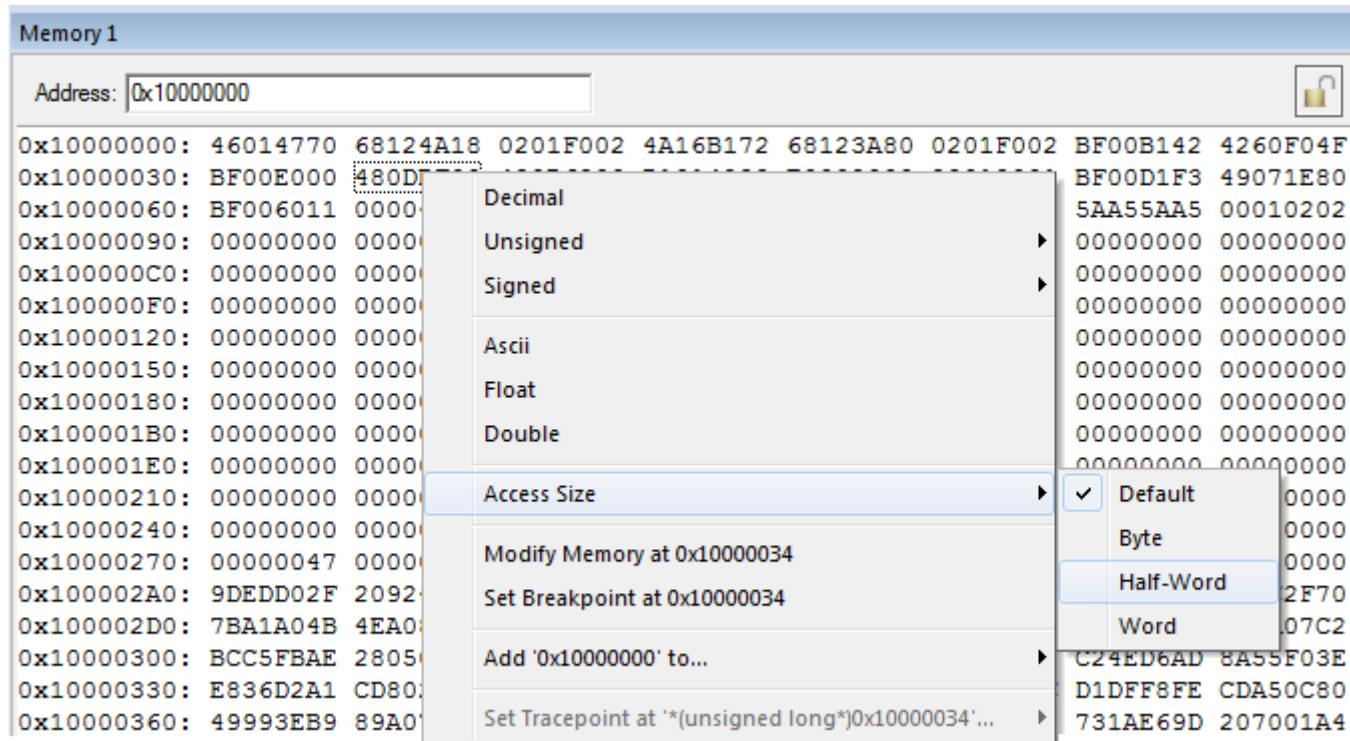


Name	Location/Value	Type
clock	0x00001368	Task
lcd	0x000012C0	Task
command	0x00001072	Task
SER_GetChar	0x000002BE	int f()
getline	0x00000D04	void f(char *,int)
line	0x100007EC cmdline ""	param - char *
n	<not in scope>	param - int
cnt	0x00000000	auto - int
c	<not in scope>	auto - char
	0x000010A4	void f()
	<not in scope>	auto - unsigned int
li	0x00000F1E	Task
k	0x00000E3E	Task
os_idle_demon	0x00000250	Task

Keil uVision MDK의 디버깅

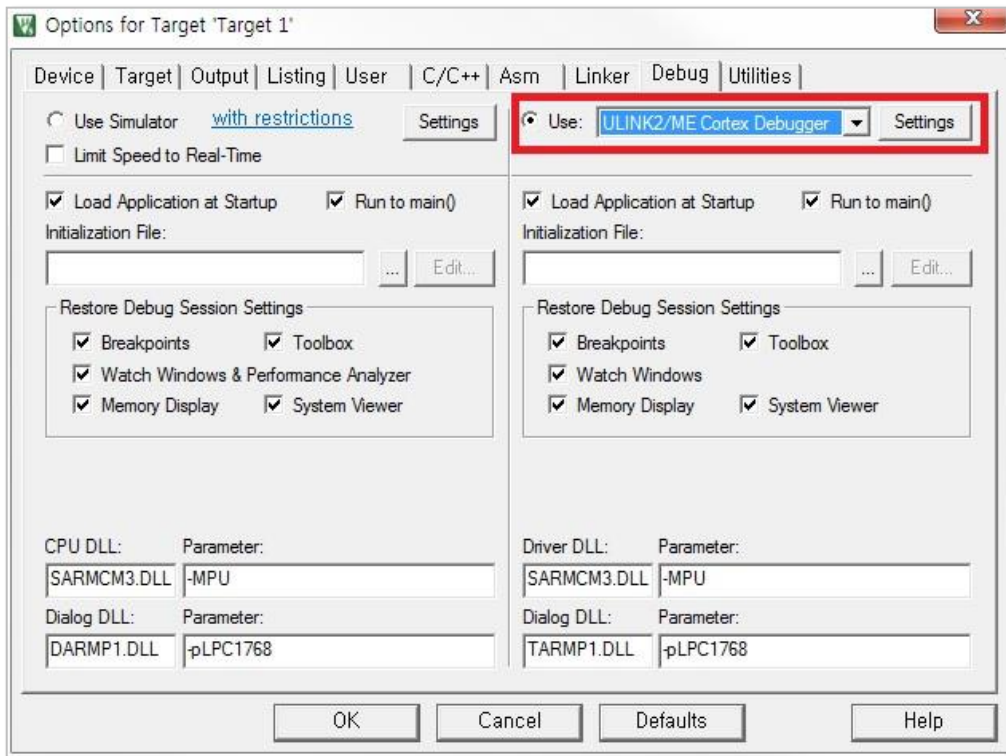
● Memory Window

- displays the memory area content. Several memory windows can be used at a time.



Keil uVision MDK의 디버깅

● GPIO 예제에 대한 Keil uVision MDK 디버깅



Keil uVision MDK의 디버깅

GPIO 예제에 대한 Keil uVision MDK 디버깅

The screenshot shows the Keil uVision MDK interface. The **Debug** menu is open, displaying options like **Start/Stop Debug Session** (Ctrl+F5), **Run** (F5), **Stop**, **Step** (F11), **Step Over** (F10), **Step Out** (Ctrl+F11), **Run to Cursor Line** (Ctrl+F10), **Show Next Statement**, **Breakpoints...** (Ctrl+B), **Insert/Remove Breakpoint** (F9), **Enable/Disable Breakpoint** (Ctrl+F9), **Disable All Breakpoints**, **Kill All Breakpoints** (Ctrl+Shift+F9), **OS Support**, **Execution Profiling**, **Memory Map...**, **Inline Assembly...**, and **Function Editor (Open Ini File)...**. The **Build Output** window at the bottom shows the following error:

```
File      : C:\Keil_v5\ARM\PACK\Keil\STM32F1xx_DFP\2.3.0\Keil.STM32F1xx_DFP
Sequence  : CheckID
Context   : Item #1: <control if="jep106id != 0x20">
            Item #0: <block>::Line 2
Expression : "          Message(2, "Not a genuine ST Device! Abort con
E203      : Undefined identifier - function 'Message'
```

The screenshot shows the Keil uVision MDK interface with the **main.c** file open. The code defines a thread `led_thread2` and a `main` function. A warning dialog box is displayed in the foreground, indicating that the software is in **EVALUATION MODE** and running with a **Code Size Limit 32K**. The dialog box has a yellow warning icon and a **확인** (OK) button.

Keil uVision MDK의 디버깅

GPIO 예제에 대한 Keil uVision MDK 디버깅

C:\Temp\한기대-내용전문가\W1.ARMW7주차\Wex2and3\WExamples\WExercise 2 and 3 Creating and Managing Threads\WCMSISrTxThreads.uvprojx - uVision

File Edit View Project Flash Debug Peripherals Tools SVCS Window Help

GPIO_Set

Registers

Register	Value
Core	
R0	0x00000000
R1	0x00000000
R2	0x00000000
R3	0x00000000
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000
R11	0x00000000
R12	0x00000000
R13 (SP)	0x20000848
R14 (LR)	0x08000CB
R15 (PC)	0x08000328
xPSR	0x01000000
Banked	
System	
Internal	
Mode	Thread
Privilege	Privileged
Stack	PSP
States	6074
Sec	0.00003332

Disassembly

```
73:      osKernelInitialize ();          // initialize CMSIS-RTOS
74:      <0x08000328 F000FC58 BL.W      osKernelInitialize (0x08000BDC)
75:      LED_Initialize ();
```

main.c startup_stm32f10x_md.s system_stm32f10x.c

```
64  osThreadDef(led_thread2, osPriorityNormal, 1, 0);
65  osThreadDef(led_thread1, osPriorityNormal, 1, 0);
66
67  /*-----
68  Initilise the LED's, get a handle for main, start the threads and terminate main
69  -----*/
70
71  int main (void)
72  {
73      osKernelInitialize ();          // initialize CMSIS-RTOS
74
75      LED_Initialize ();
76      led_ID2 = osThreadCreate(osThread(led_thread2), NULL);
77      led_ID1 = osThreadCreate(osThread(led_thread1), NULL);
78
79      osKernelStart ();              // start thread execution
80      while(1)
81      {
82      }
83  }
```

Command

ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess COVERAGE DEFINE DIR Display Enter

fatal error in include chain (STM32F10x.h): 'RTE_Components.h' file not found

Simulation t1: 0.00003332 sec L:73 C:5 CAP NUM SCRL OVR R/W

Call Stack + Locals

Name	Locati...	Type
0x0800...		Task
0x0800...		Task

Call Stack + Locals Memory 1

Keil uVision MDK의 디버깅

GPIO 예제에 대한 Keil uVision MDK 디버깅

Insert/Remove Breakpoint

The screenshot displays the Keil uVision MDK IDE interface. The 'Project' window on the left shows a project named 'GPIO1' with a 'Main' folder containing 'GPIO1.c'. The 'GPIO1.c' file is open in the main editor, showing C code for GPIO control. A red dot indicates a breakpoint is set at line 29. The 'Debug' menu is open, showing the 'Insert/Remove Breakpoint' option (F9). The 'Build Output' window at the bottom shows an error: 'Internal DLL Error' and 'Error: Flash Download failed - Target DLL has been cancelled'. The status bar at the bottom indicates 'Simulation' mode.

Project: GPIO1

Target 1

Startup

startup_LPC17xx

system_LPC17xx

CM3 Core

core_cm3.c

Drivers

lpc17xx_gpio.c

Main

GPIO1.c

GPIO_Set

GPIO1.c

29 GPIO_ClearValue(2, 0x0000007C); //Set 28,29,31 of GPIO1 off : LED1,2,3 off

30

31 Delay(6000000); //delay = 6,000,000 X 12,000,000 = 0.5 sec

32

33

34

Build Output

Load "C:\\Temp\\1_GPIO (2)\\GPIO1\\obj\\GPIO1.axf"

Internal DLL Error

Error: Flash Download failed - Target DLL has been cancelled

Flash Load finished at 09:50:50

Insert or remove a breakpoint at the current line

Simulation

L:28 C:1

CAP NUM SCRL OVR R/W

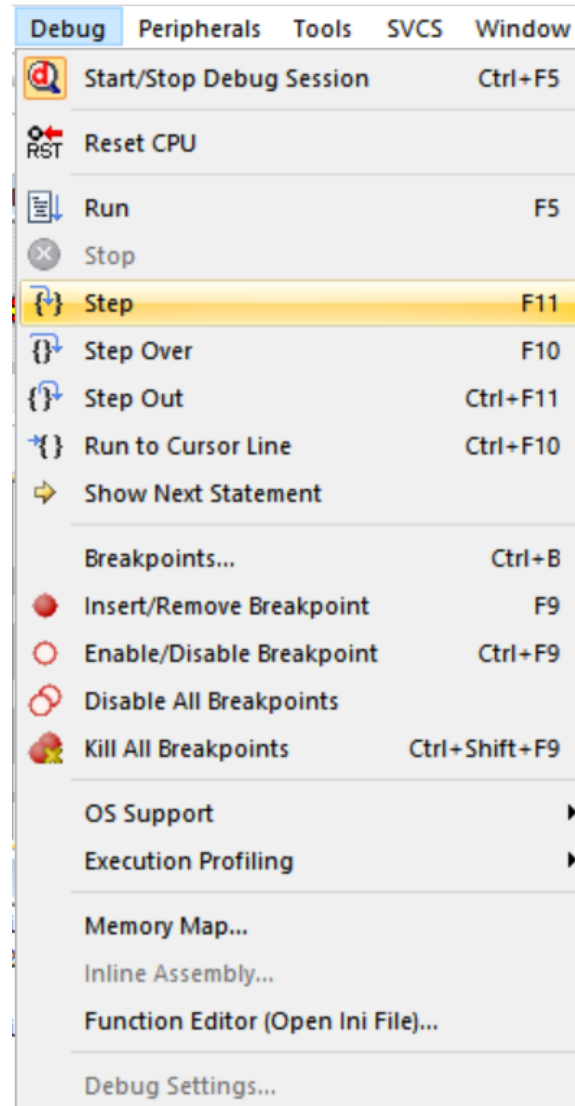
오전 9:51

2021-01-24

Keil uVision MDK의 디버깅

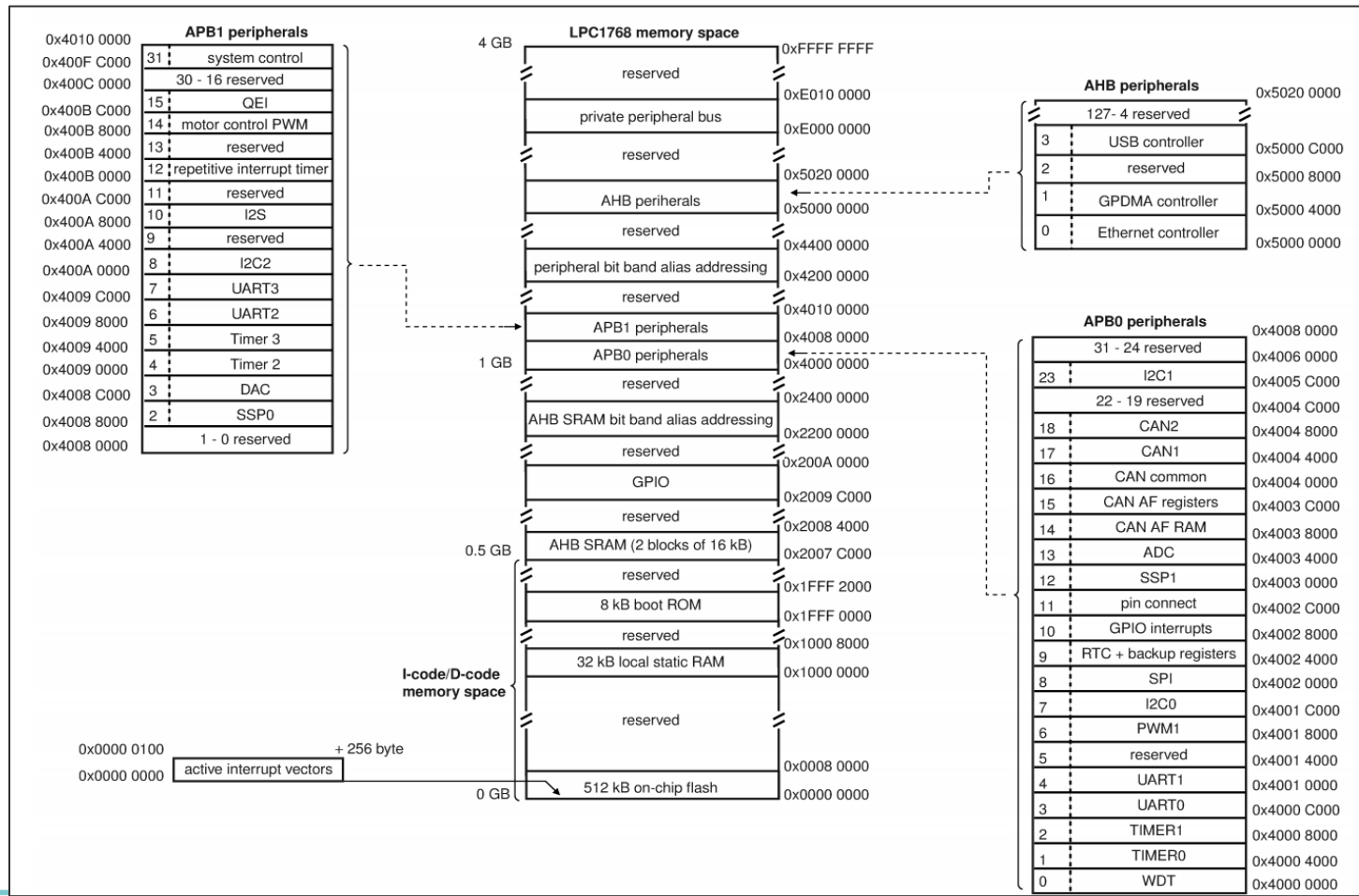
● Debug 메뉴

- Run
- Step (in) - F11
- Step Over - F10
- Step Out - Ctrl + F11
- Run to Cursor Line Ctrl + F10



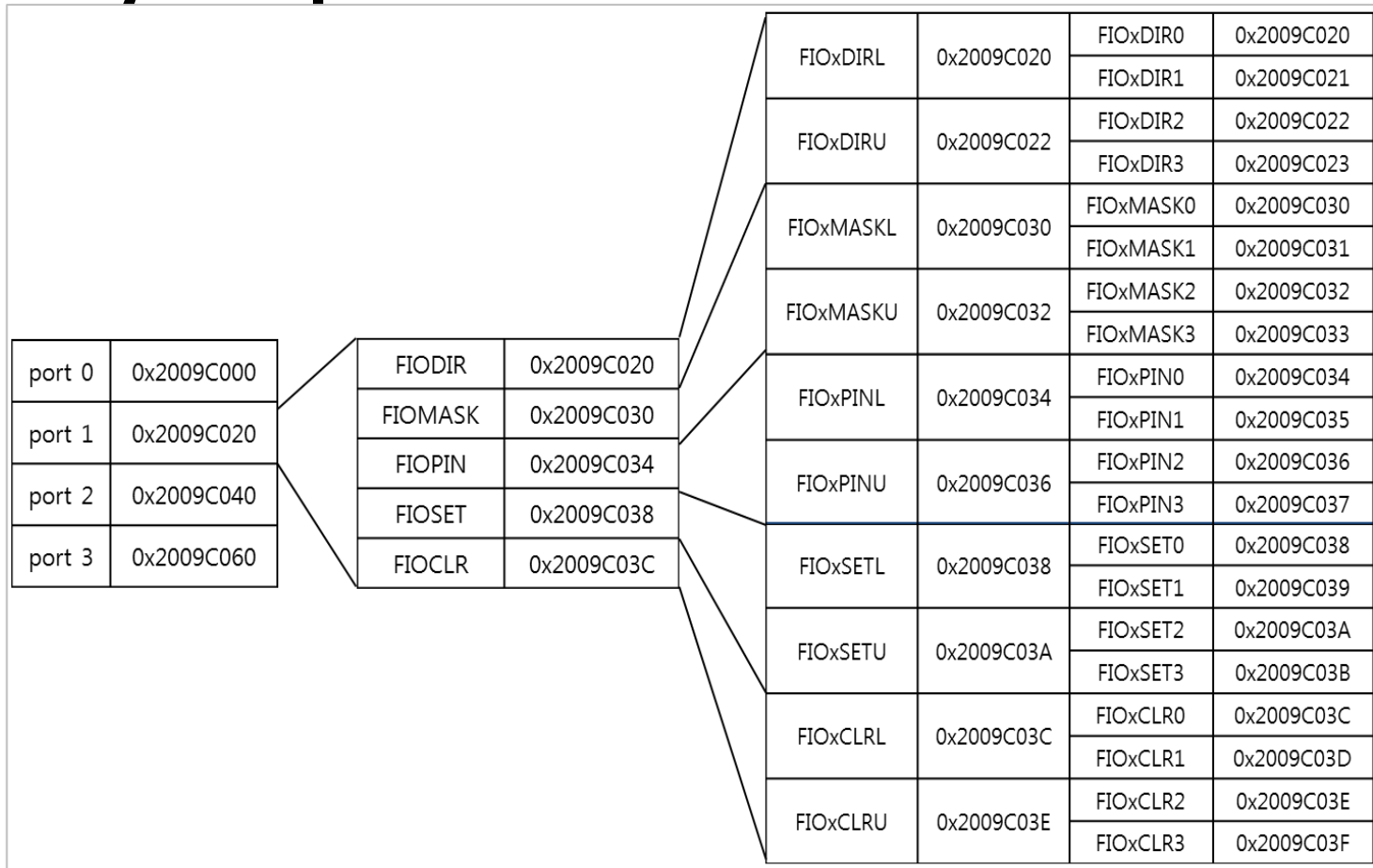
Keil uVision MDK의 디버깅

Memory Dump

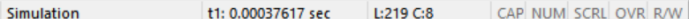


Keil uVision MDK의 디버깅

● Memory Dump



Memory Dump



● Memory Dump

- Black - CODE memory area or uninitialized RAM.
- Red - CONST data in Flash or ROM that has been accessed at least once.
- Gold - memory location that has been initialized, but not accessed yet.
- Green - memory location has been accessed at least once.

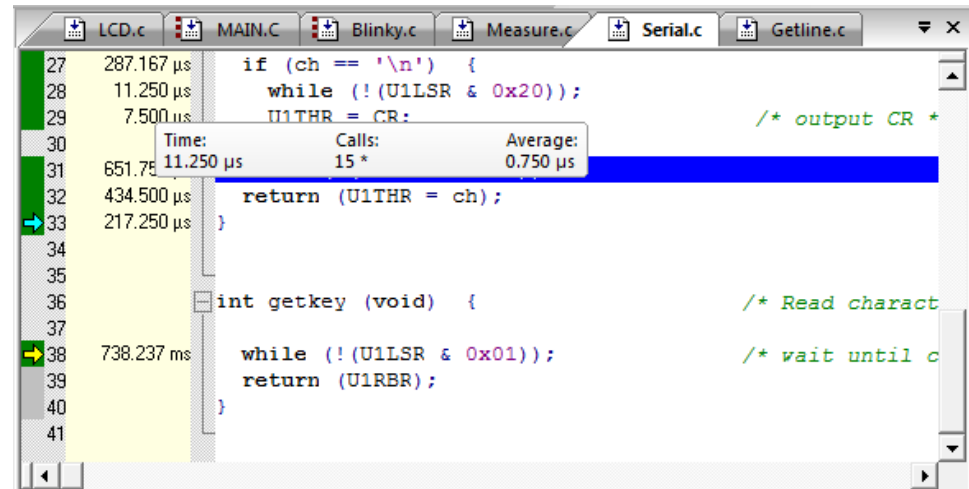
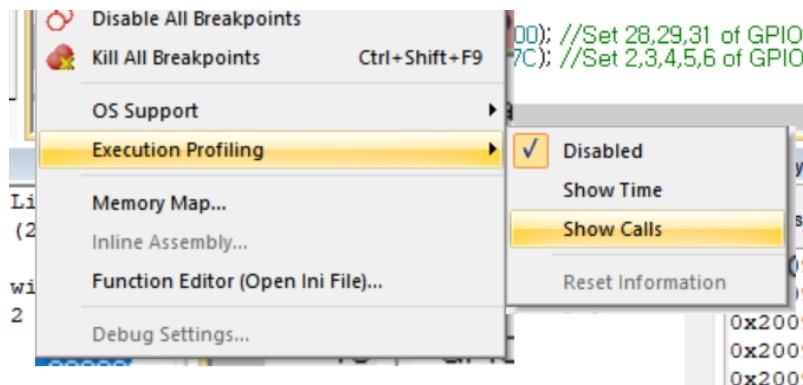
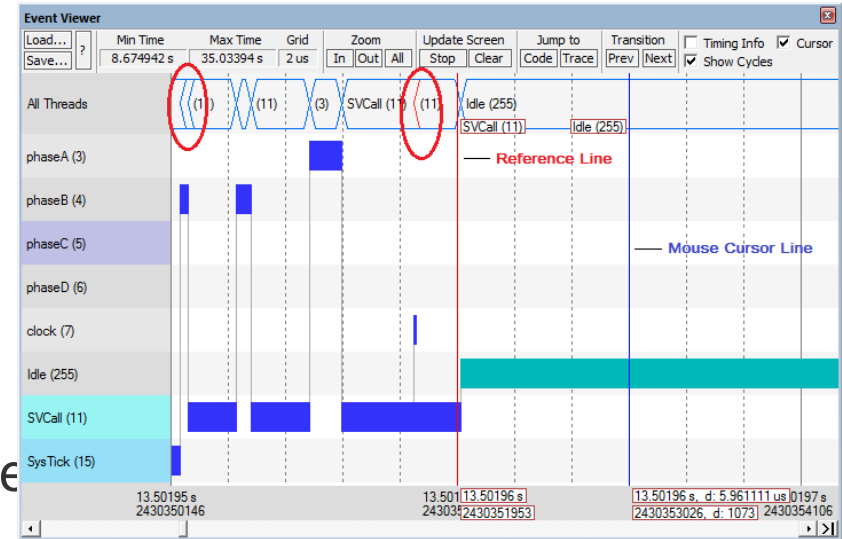
Keil uVision MDK의 디버깅

Event Viewer

- Debug - OS Support - Event Viewer

Execution Profiler

- records timing and execution statistics about instructions for the complete program code.



● JTAG(Joint Test Action Group)

- IEEE 1149.1에 표준
- 디지털 회로의 칩 외부와 연결되는 핀의 입출력회로에 적용하여 활용
- 회로 설계에 따라 디지털 회로의 내부로 전송하거나 핀의 외부로 데이터를 출력할 수도 있고 상태를 읽을 수도 있음
- JTAG을 통해 데이터를 동기식 직렬 통신 전송하는 방식은 boundary scan을 통해 구현
- Access to the scan chain data via 2 pins, Test Data In (TDI) and Test Data Out (TDO).
- Data can be applied serially on TDI to set up the system state, while state data can be read serially on TDO.

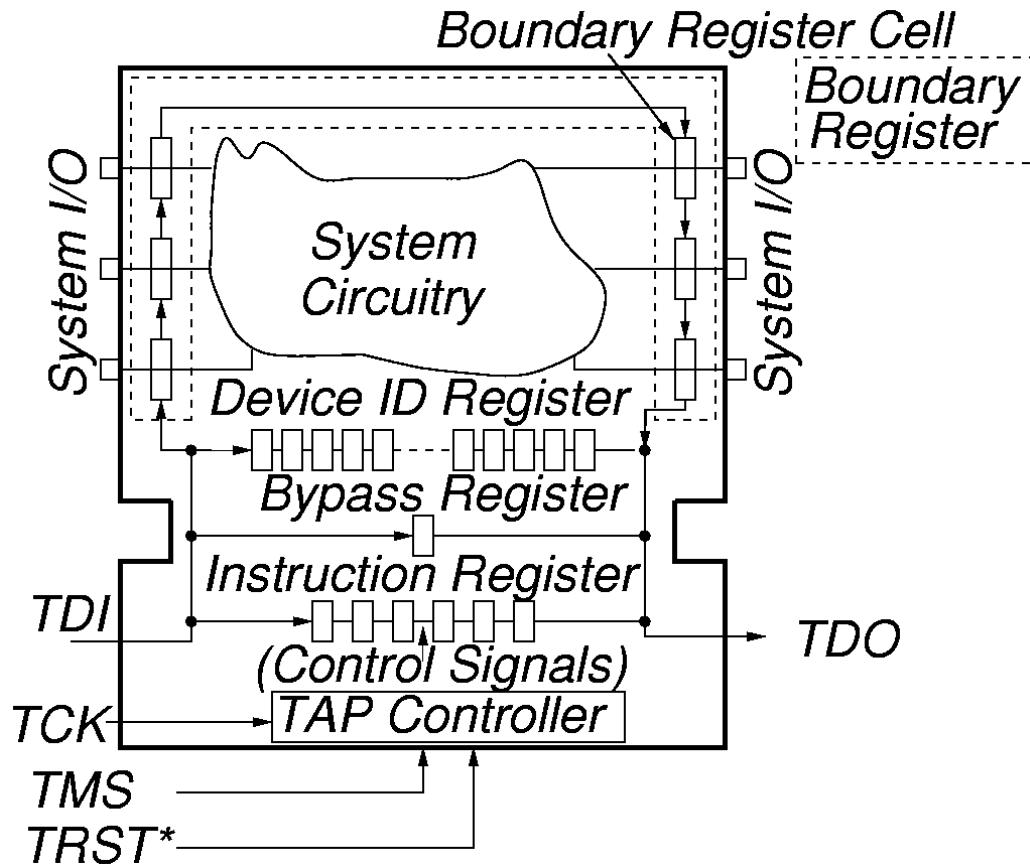
JTAG 구조

● JTAG 인터페이스

- TDI (데이터 입력) : Test하기 위한 데이터 신호. TMS에 의해 전이된 TAP state에 따라, TDI가 command/data 가 결정됨
- TDO (데이터 출력) : Test한 결과를 외부에서 모니터링 하기 위한 pin, 이 역시 TAP state에 따라 address/data가 될 수 있음.
- TCK (클럭) : Test clock
- TMS (모드) : Test Mode로 전환하기 위한 제어 신호
- TRST (리셋)

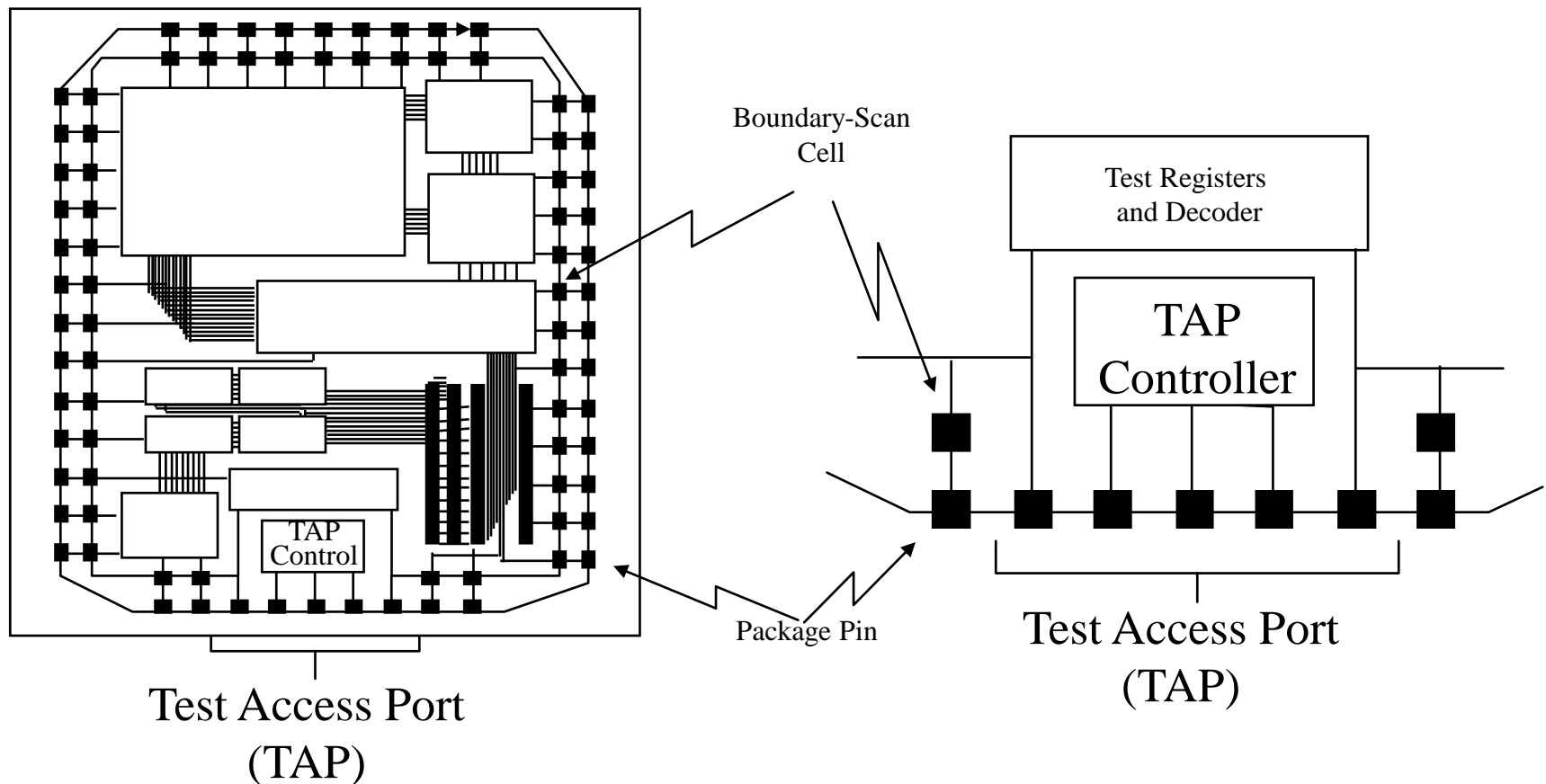
JTAG 구조

● 시스템 테스트 논리 구조

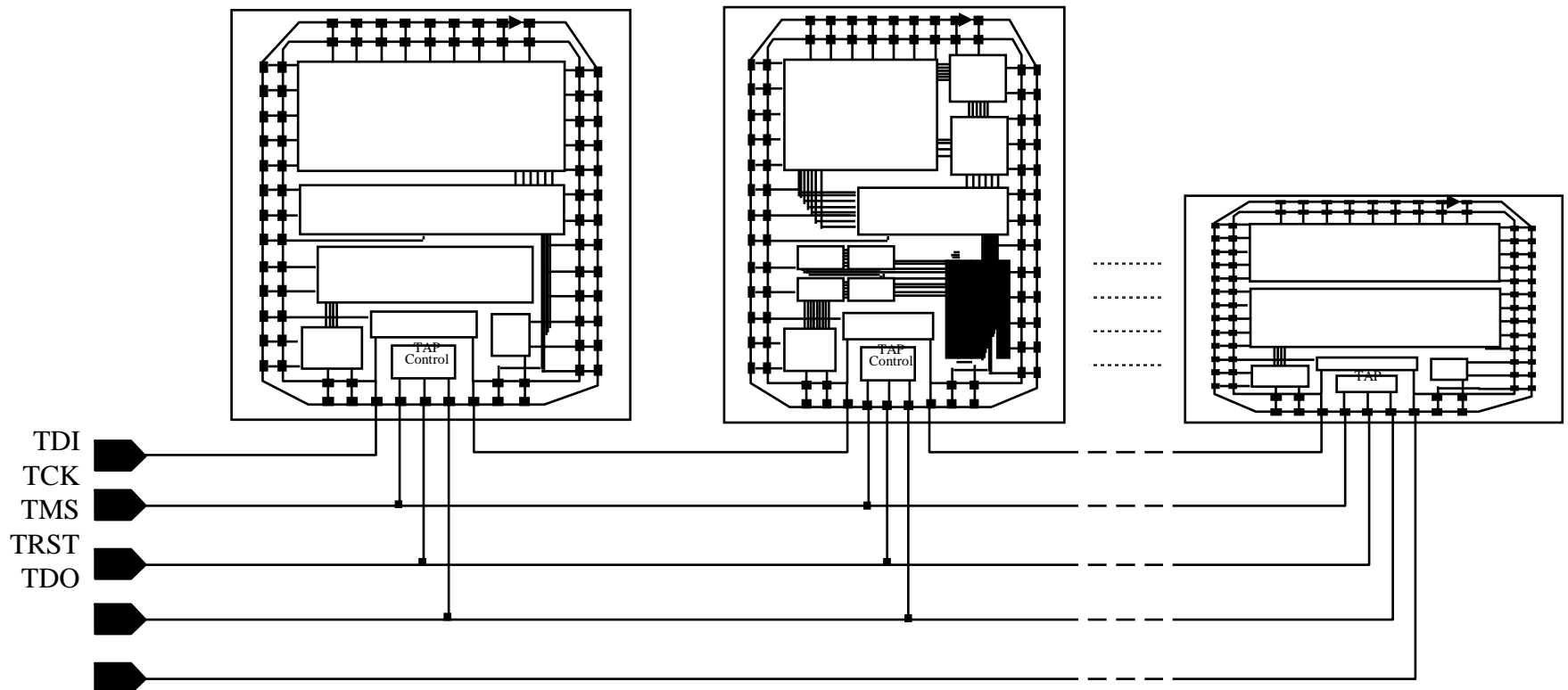


JTAG 구조

- JTAG은 디바이스 내 각각의 핀을 Boundary Cell과 일대일로 연결, 각 Cell은 boundary scan register를 형성하기 위해 서로 연결



JTAG 구조



- 인터페이스는 5개의 핀(TDI, TMS, TCK, nTRST, TDO)을 통해 제어
- 디바이스 간의 연결 상태를 테스트하거나, 플래시 메모리에 퓨징 (fusing)하는 기능

정리/하기