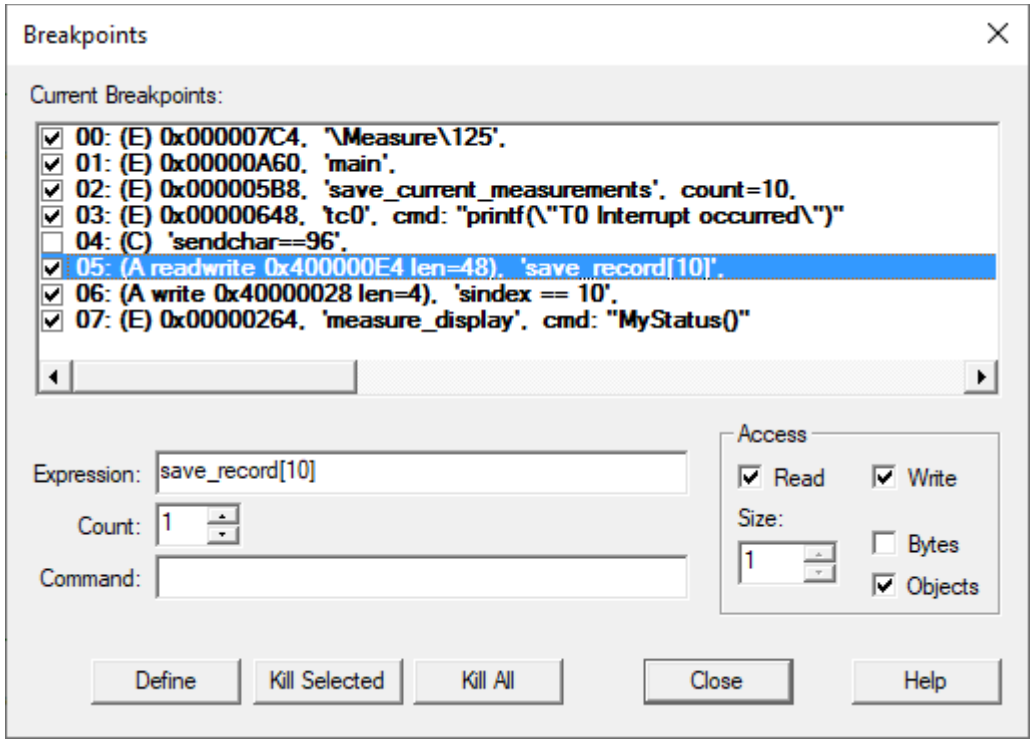
임베디드 시스템

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 IDE**의 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:

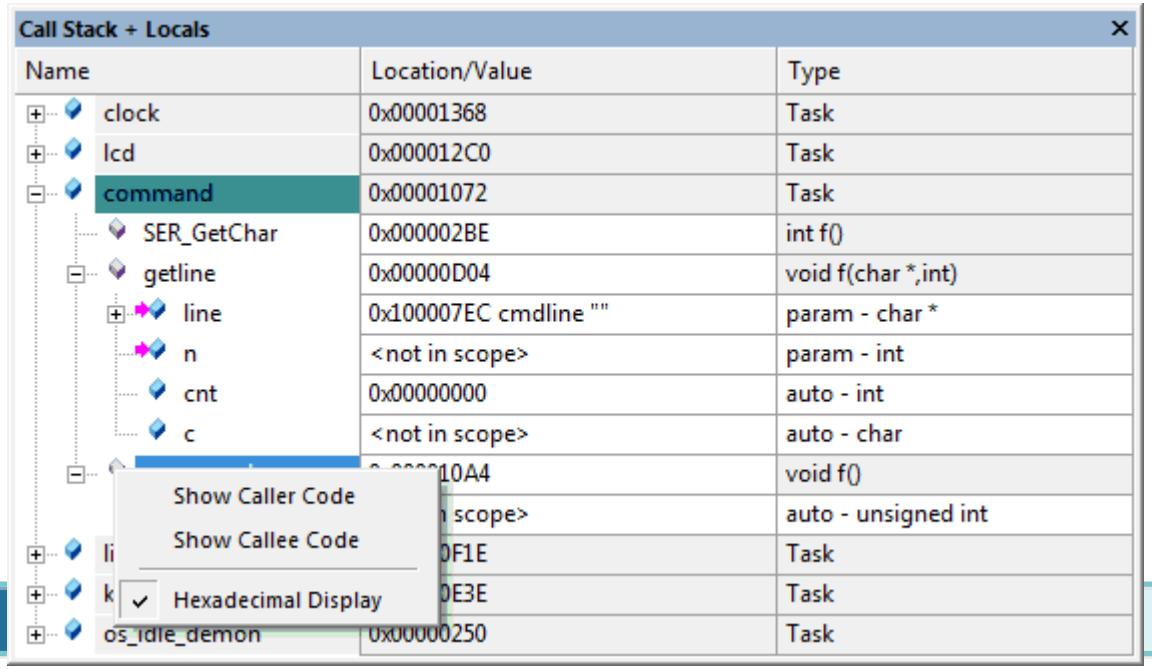


Keil uVision MDK의 디버깅

- **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

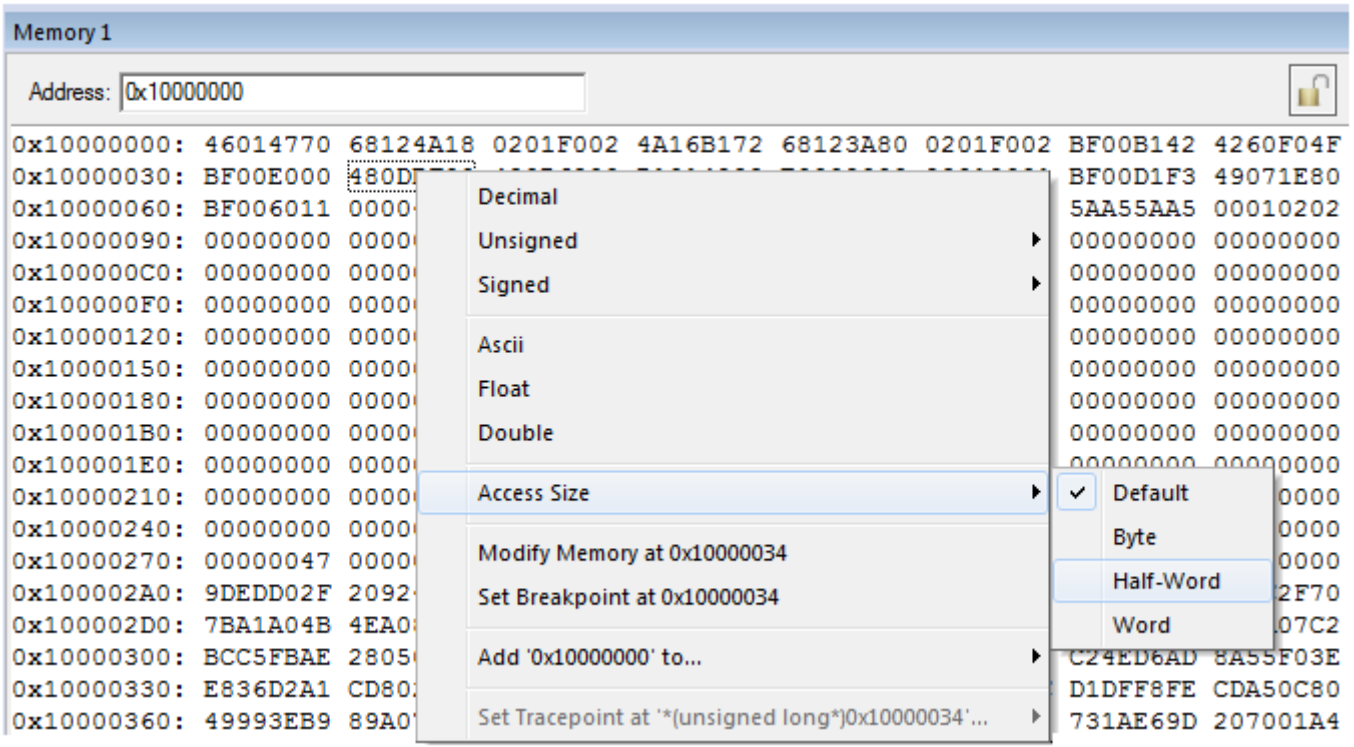
Keil uVision MDK의 디버깅

- **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.



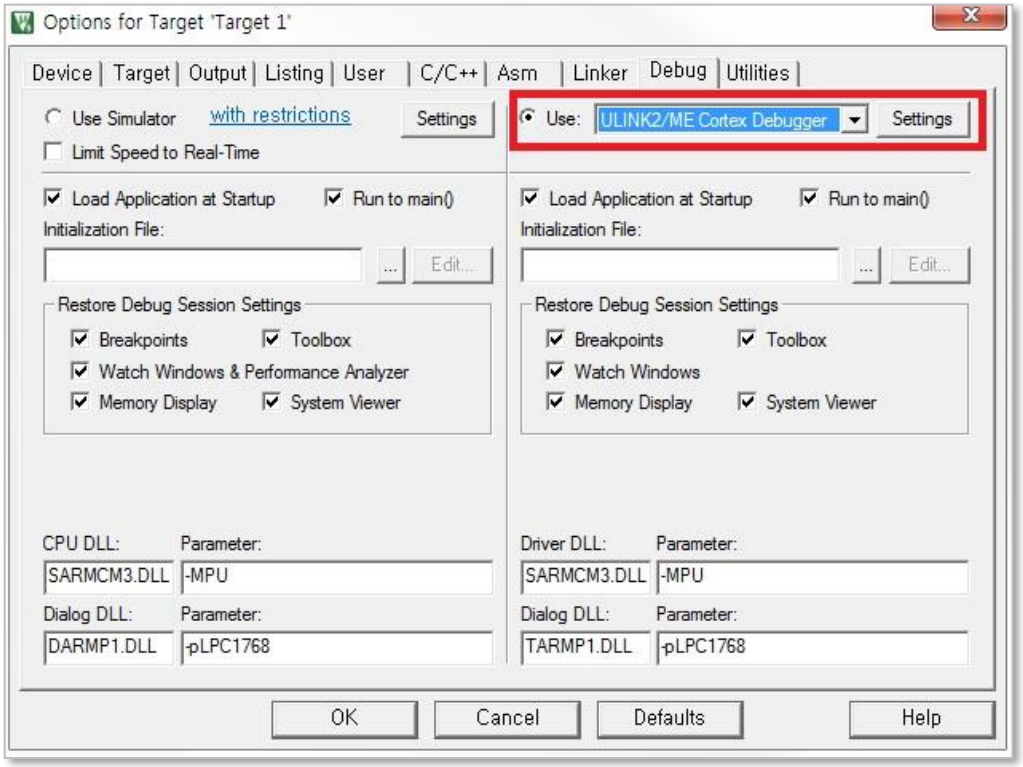
Keil uVision MDK의 디버깅

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



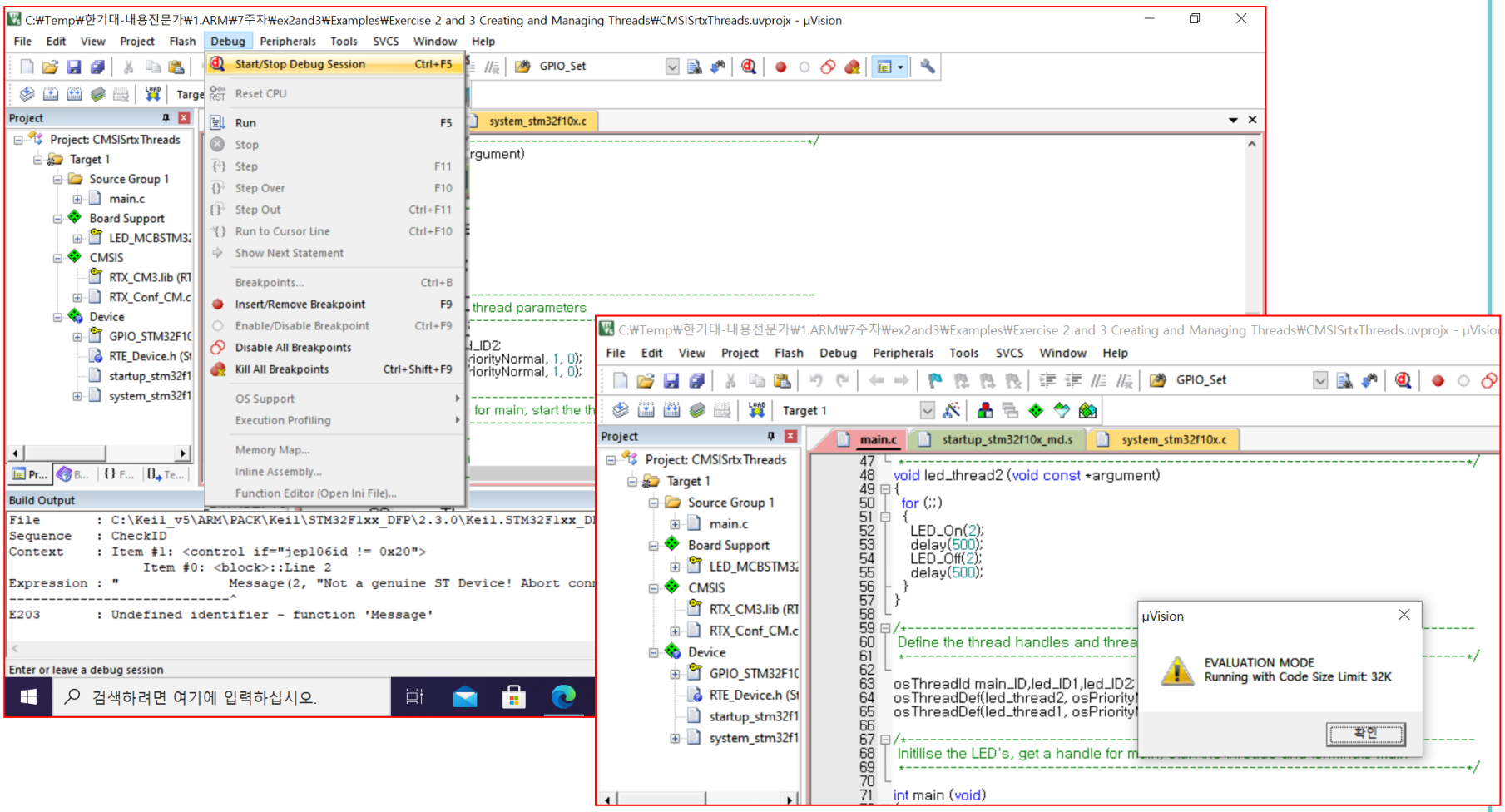
Keil uVision MDK의 디버깅

- **GPIO** 예제에 대한 **Keil uVision MDK** 디버깅



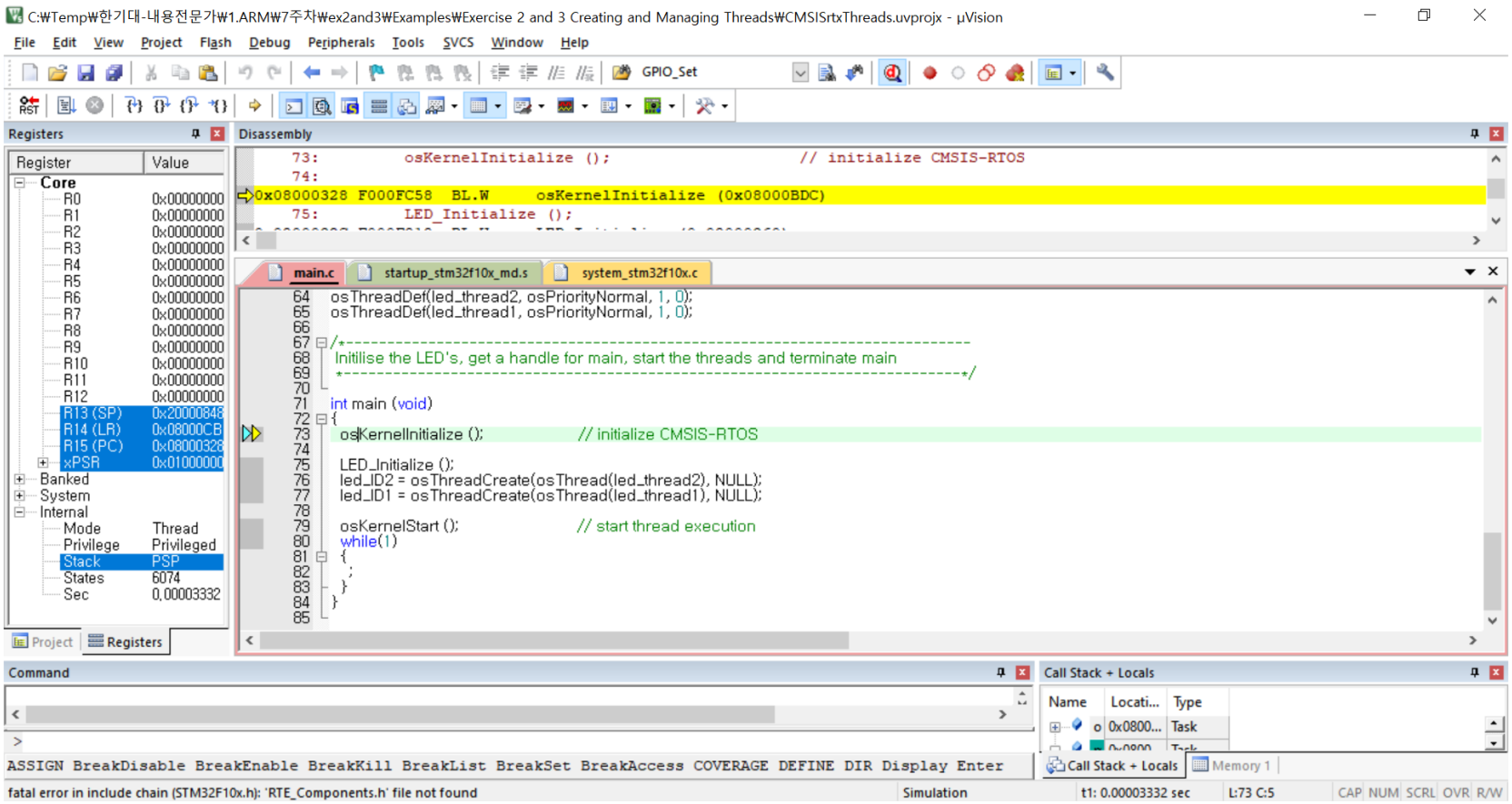
Keil uVision MDK의 디버깅

- **GPIO** 예제에 대한 **Keil uVision MDK** 디버깅



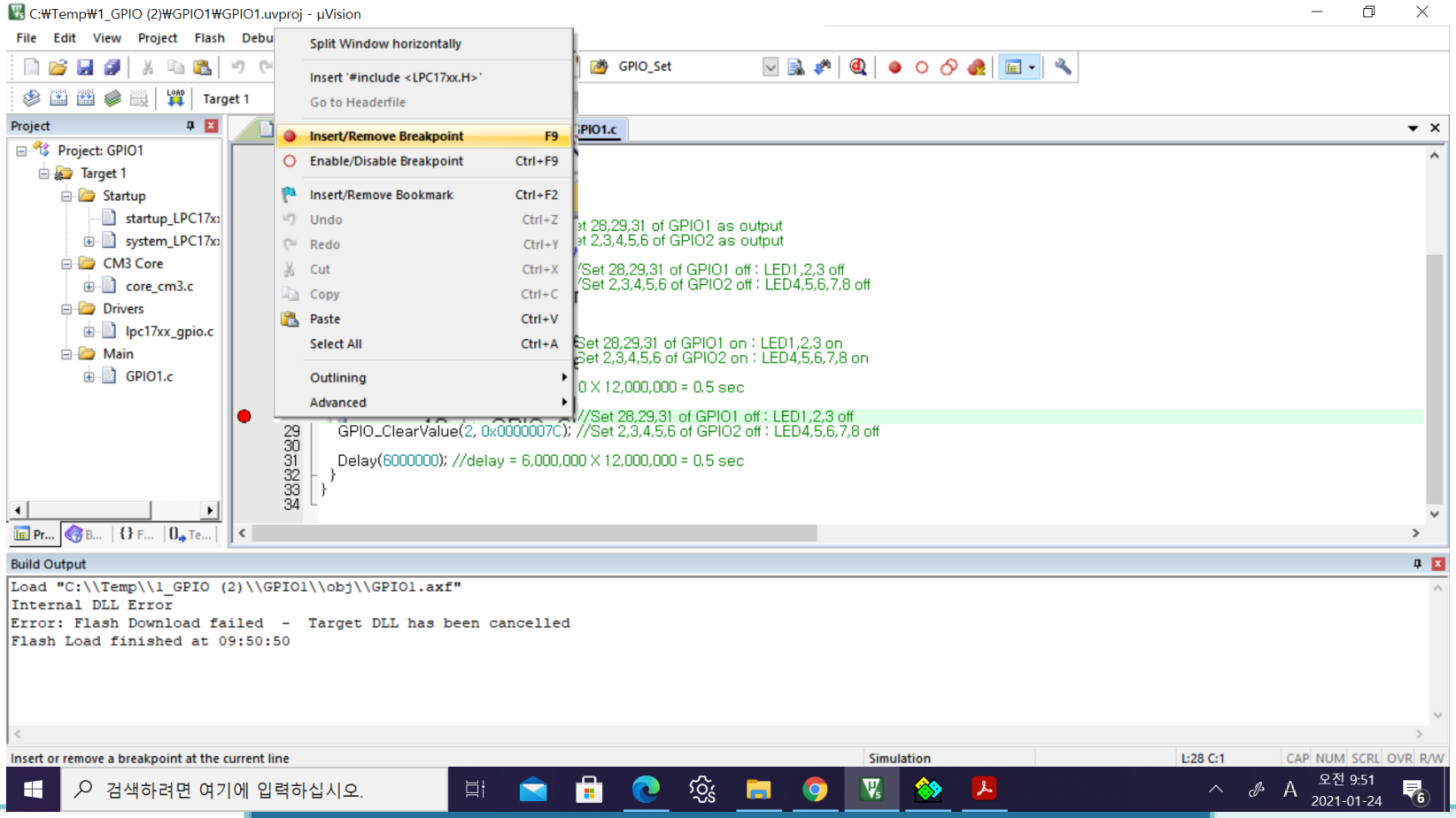
Keil uVision MDK의 디버깅

- **GPIO** 예제에 대한 **Keil uVision MDK** 디버깅



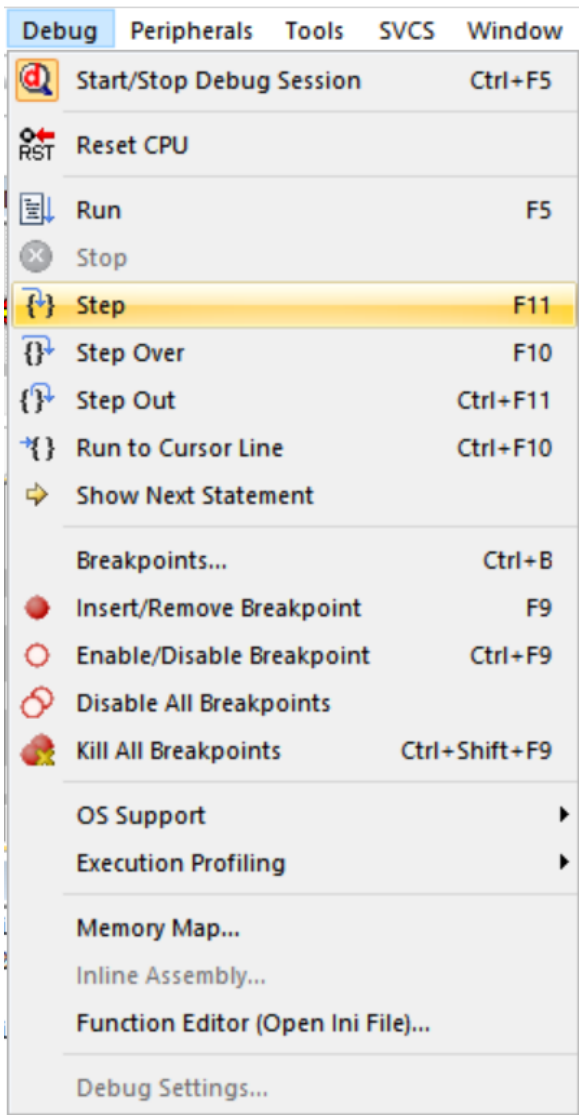
Keil uVision MDK의 디버깅

- **GPIO** 예제에 대한 **Keil uVision MDK** 디버깅
  - Insert/Remove Breakpoint



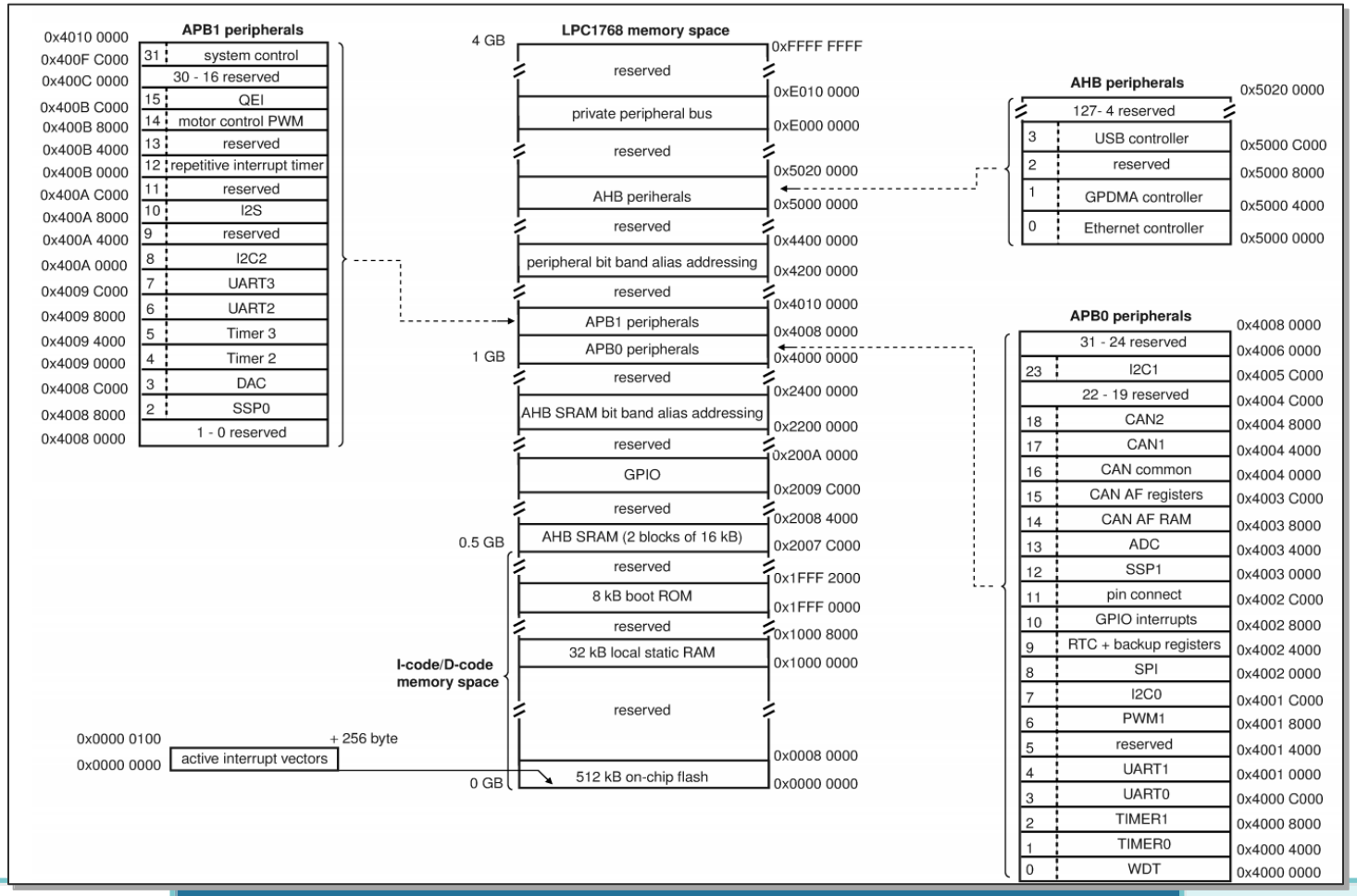
Keil uVision MDK의 디버깅

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



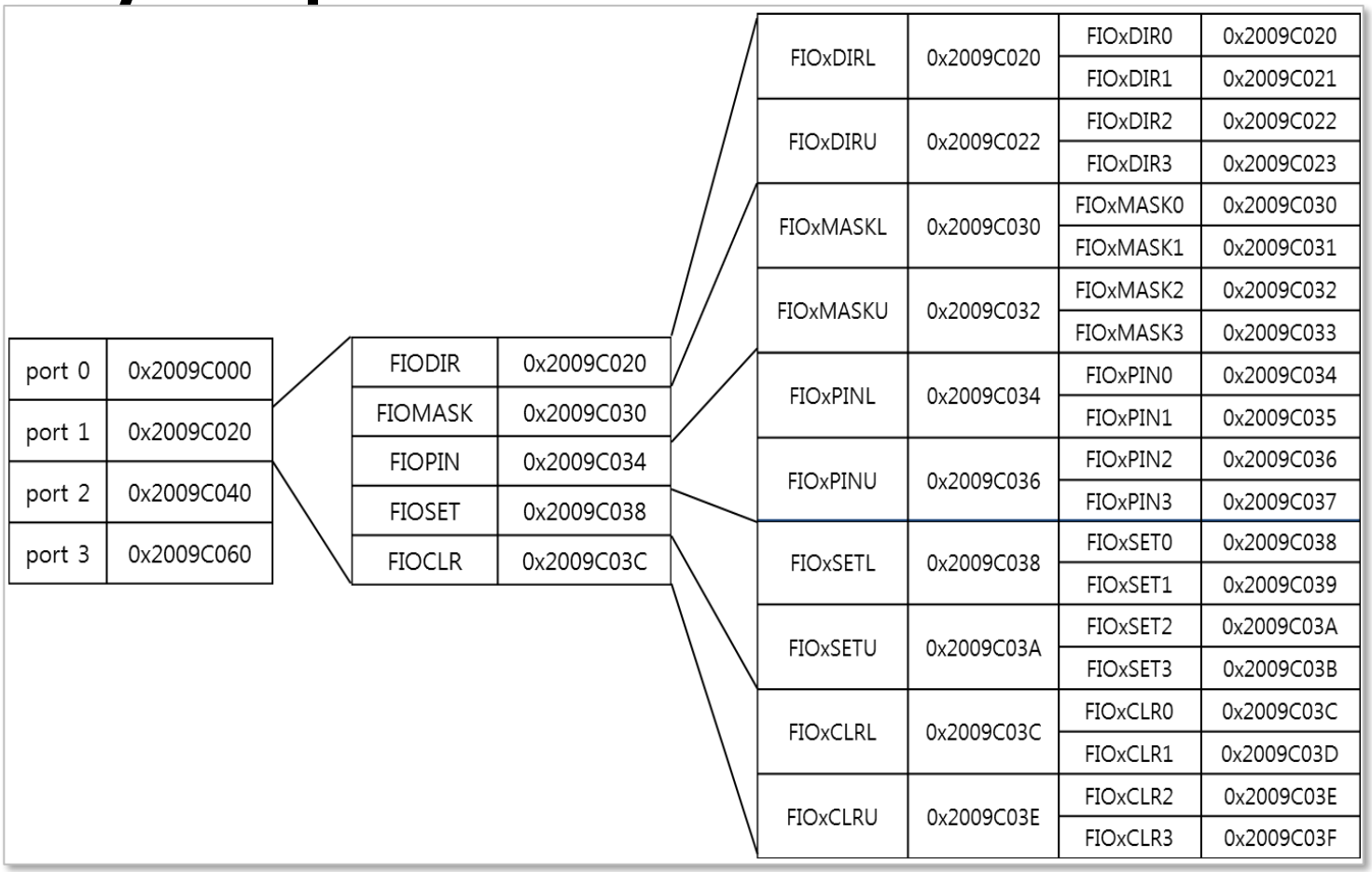
Keil uVision MDK의 디버깅

- **Memory Dump**



Keil uVision MDK의 디버깅

- **Memory Dump**



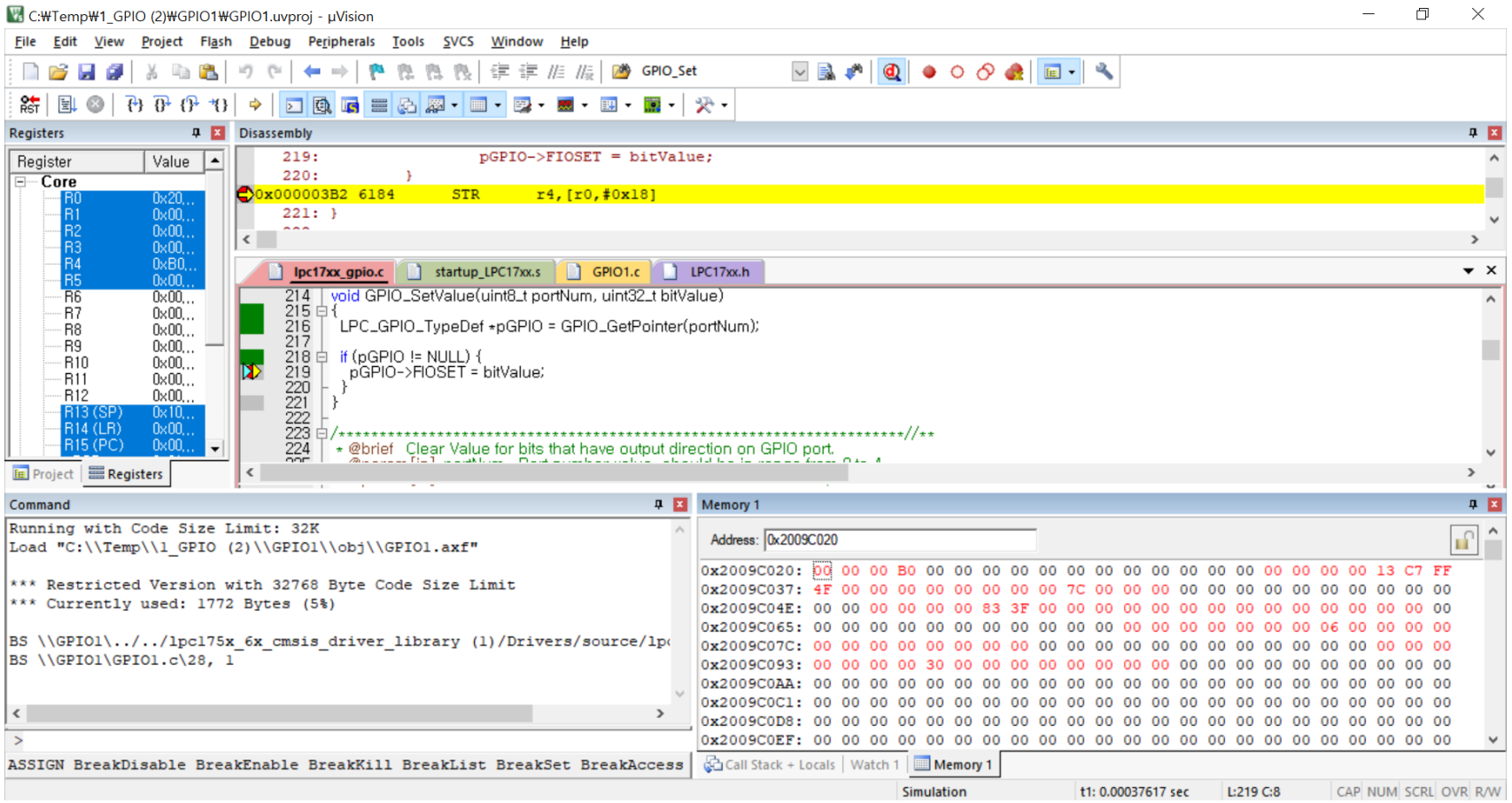
port 0	0x2009C000
port 1	0x2009C020
port 2	0x2009C040
port 3	0x2009C060

FIODIR	0x2009C020
FIOMASK	0x2009C030
FIOPIN	0x2009C034
FIOSET	0x2009C038
FIOCLR	0x2009C03C

FIOxDIRL	0x2009C020	FIOxDIR0	0x2009C020
		FIOxDIR1	0x2009C021
FIOxDIRU	0x2009C022	FIOxDIR2	0x2009C022
		FIOxDIR3	0x2009C023
FIOxMASKL	0x2009C030	FIOxMASK0	0x2009C030
		FIOxMASK1	0x2009C031
FIOxMASKU	0x2009C032	FIOxMASK2	0x2009C032
		FIOxMASK3	0x2009C033
FIOxPINL	0x2009C034	FIOxPIN0	0x2009C034
		FIOxPIN1	0x2009C035
FIOxPINU	0x2009C036	FIOxPIN2	0x2009C036
		FIOxPIN3	0x2009C037
FIOxSETL	0x2009C038	FIOxSET0	0x2009C038
		FIOxSET1	0x2009C039
FIOxSETU	0x2009C03A	FIOxSET2	0x2009C03A
		FIOxSET3	0x2009C03B
FIOxCLRL	0x2009C03C	FIOxCLR0	0x2009C03C
		FIOxCLR1	0x2009C03D
FIOxCLRU	0x2009C03E	FIOxCLR2	0x2009C03E
		FIOxCLR3	0x2009C03F

Keil uVision MDK의 디버깅

- **Memory Dump**

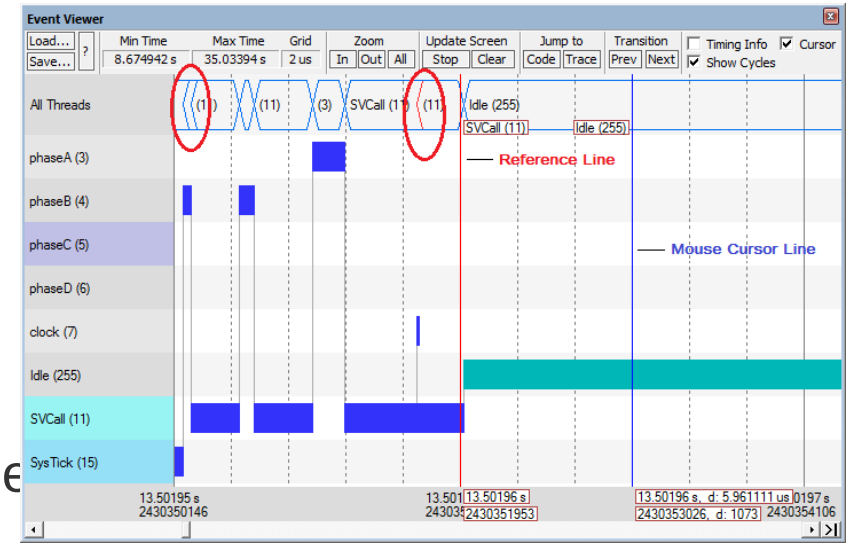


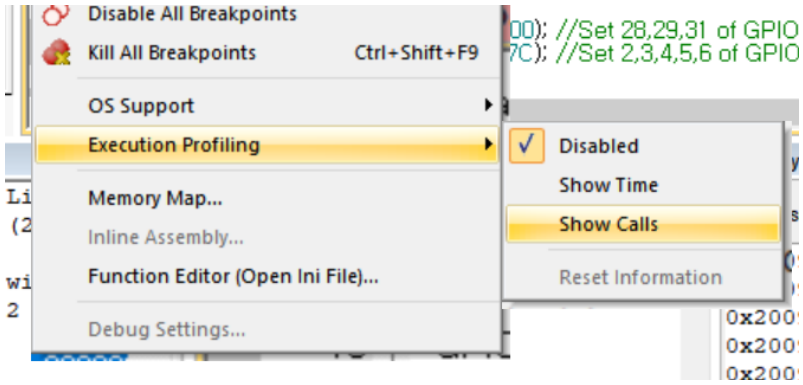
Keil uVision MDK의 디버깅

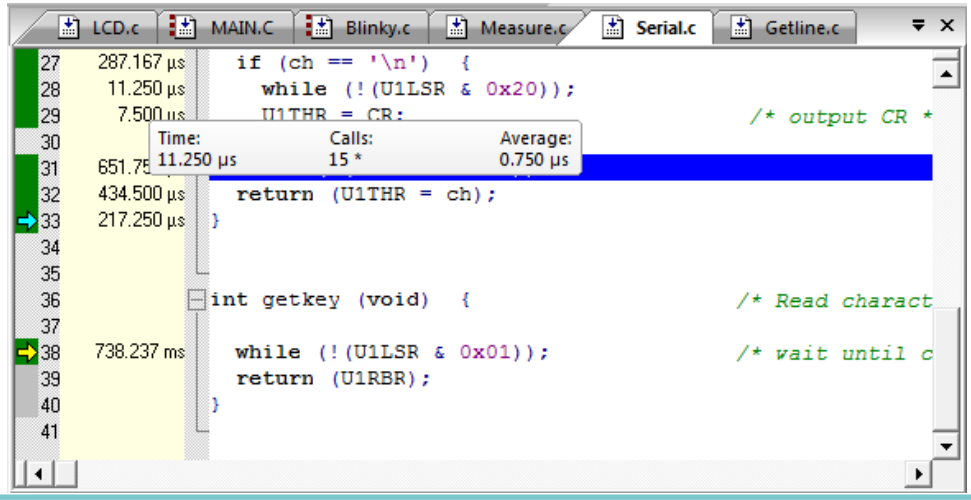
- **Memory Dump**
  - Black - CODE memory area or uninitialized RAM.
  - Red - CONST data in Flash or ROM that has been accessed at least once.
  - Gold - memory location that has been initialized, but not accessed yet.
  - Green - memory location has been accessed at least once.

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 구조

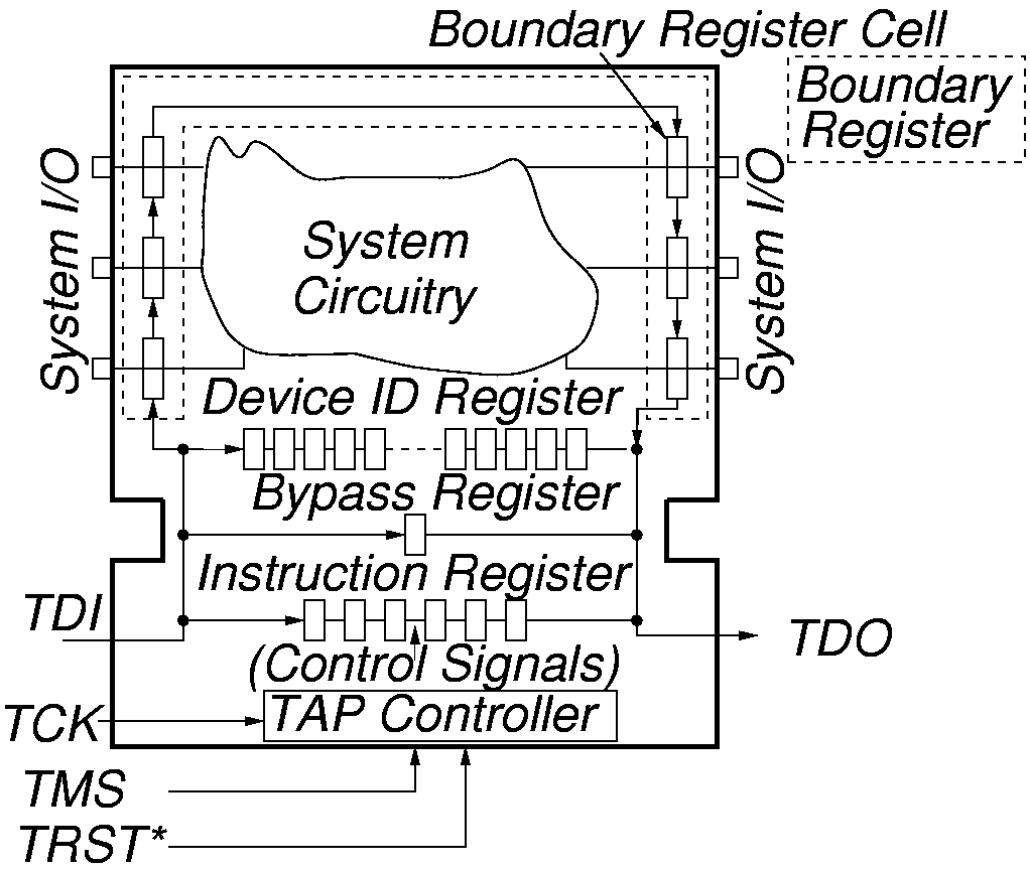
- **JTAG(Joint Test Action Group)**
  - IEEE 1149.1에 표준
  - 디지털 회로의 칩 외부와 연결되는 핀의 입출력회로에 적용하여 활용
  - 회로 설계에 따라 디지털 회로의 내부로 전송하거나 핀의 외부로 데이터를 출력할 수도 있고 상태를 읽을 수도 있음
  - JTAG을 통해 데이터를 동기식 직렬 통신 전송하는 방식은 boundary scan을 통해 구현
  - Access to the scan chain data via 2 pins, Test Data In (TDI) and Test Data 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

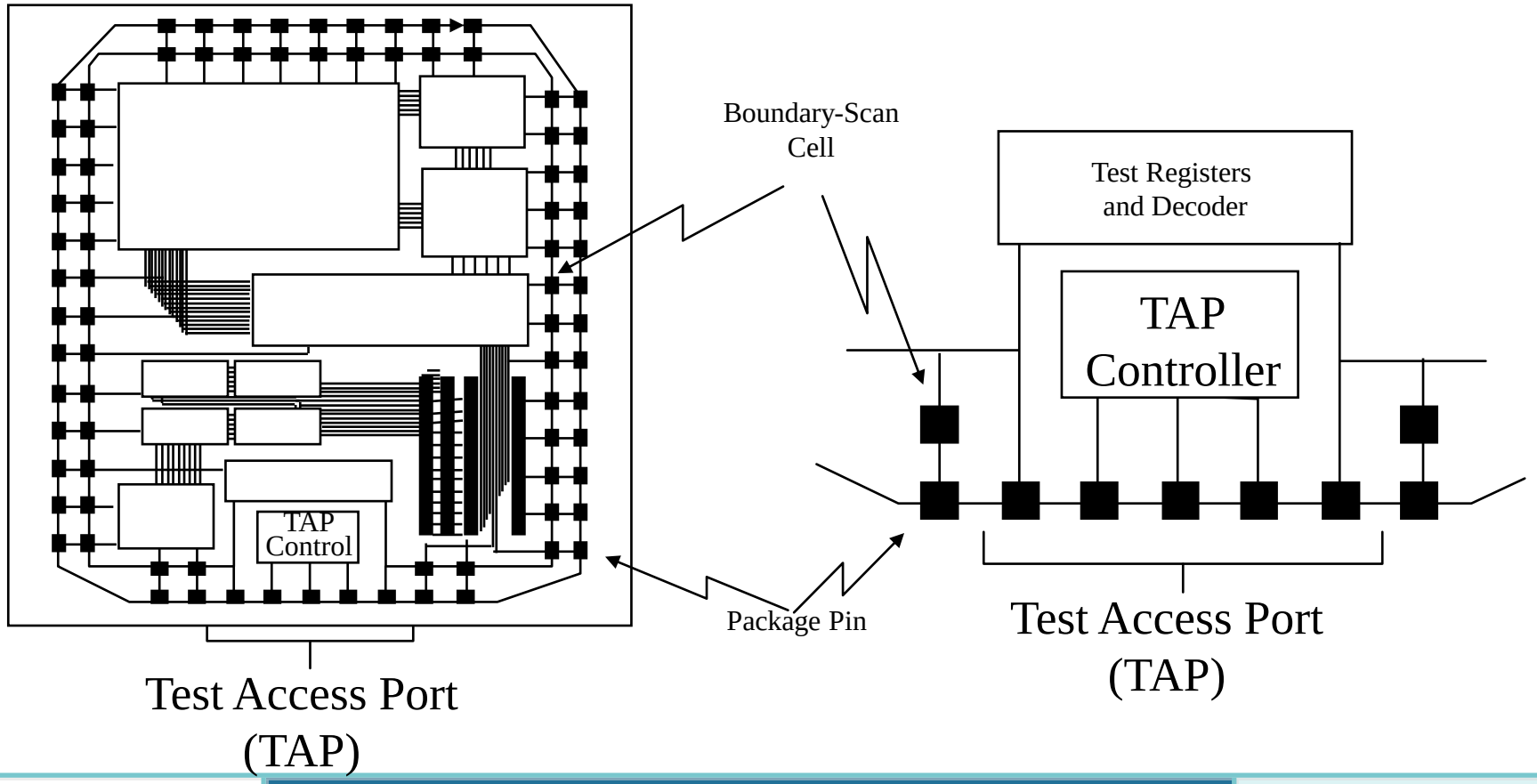
- 시스템 테스트 논리 구조



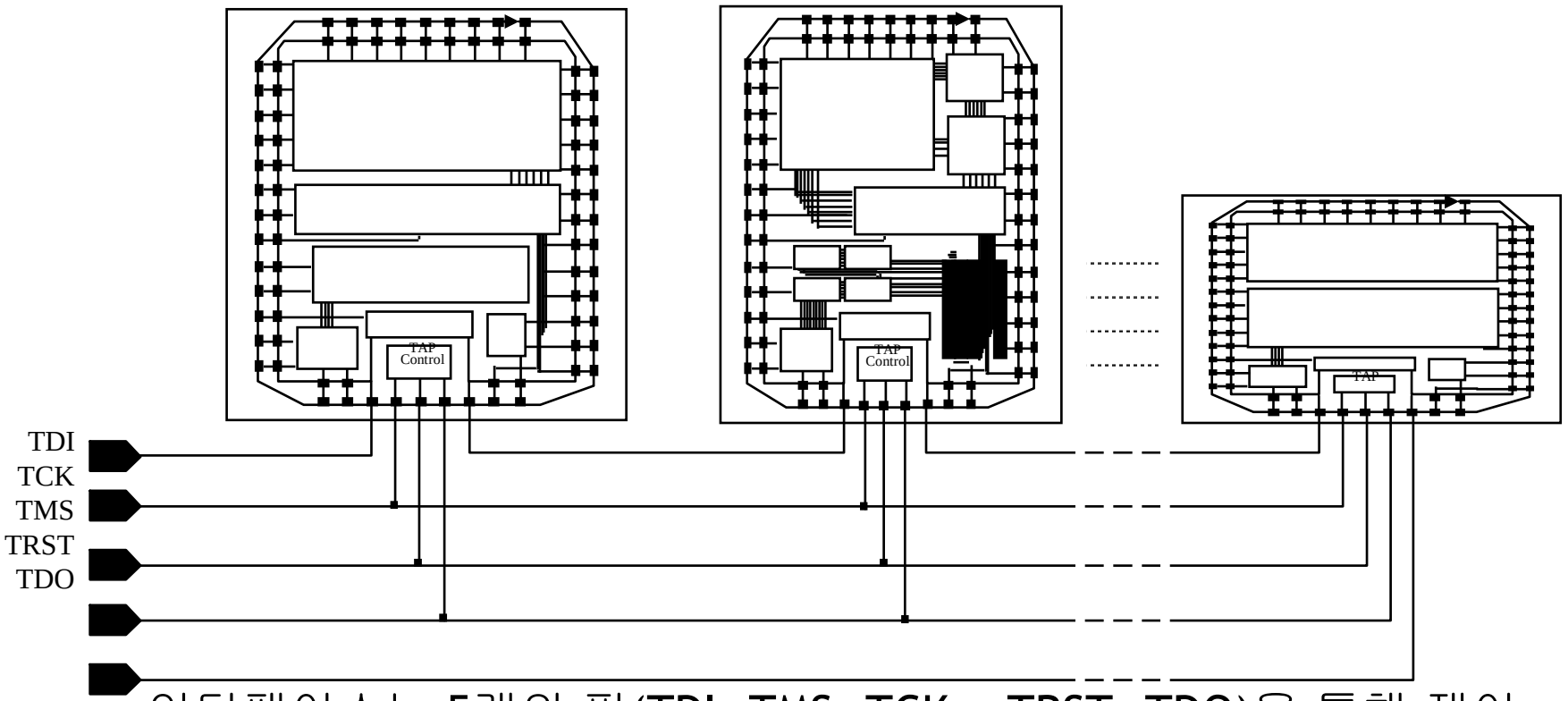
M. L. Bushnell, V. D. Agrawal, Essentials of Electronic Testing (for Digital Memory & Mixed-Signal VLSI Circuits)

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

정리하기