Hands-On Workshop:
Advanced Debugging with MCUXpresso IDE

Clark Jarvis

SW and Tools Product Marketer

June 2018 | AMF-ENT-T3143

SECURE CONNECTIONS FOR A SMARTER WORLD

Company Public – NXP, the NXP logo, and NXP secure connections for a smarter world are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2018 NXP B.V.

Agenda

- MCUXpresso SW & Tools Overview
- Walkthrough of Tools
- Hands-on Lab
- Summary

MCUXpresso SW & Tools Overview

MCUXpresso Software and Tools

for LPC & Kinetis MCUs and i.MX RT crossover processors

MCUXpresso IDE

Edit, compile, debug and optimize in an intuitive and powerful IDE

MCUXpresso SDK

Runtime software including peripheral drivers, middleware, RTOS, demos and more

MCUXpresso Config Tools

Online and desktop tool suite for system configuration and optimization

NXP Microcontroller Enablement Consolidation

LPCXpresso

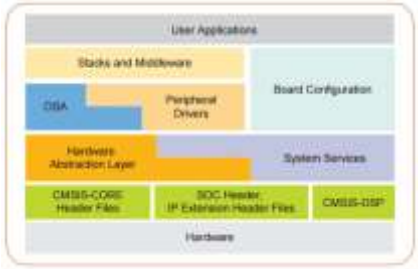
LPCware

Kinetis Design Studio

LPCOpen v2/3

Processor Expert Software

Kinetis SDK



2015 – 2016

MCUXpresso Software and Tools

- IDE
- SDK
- Config Tools

IDE SDK CFG

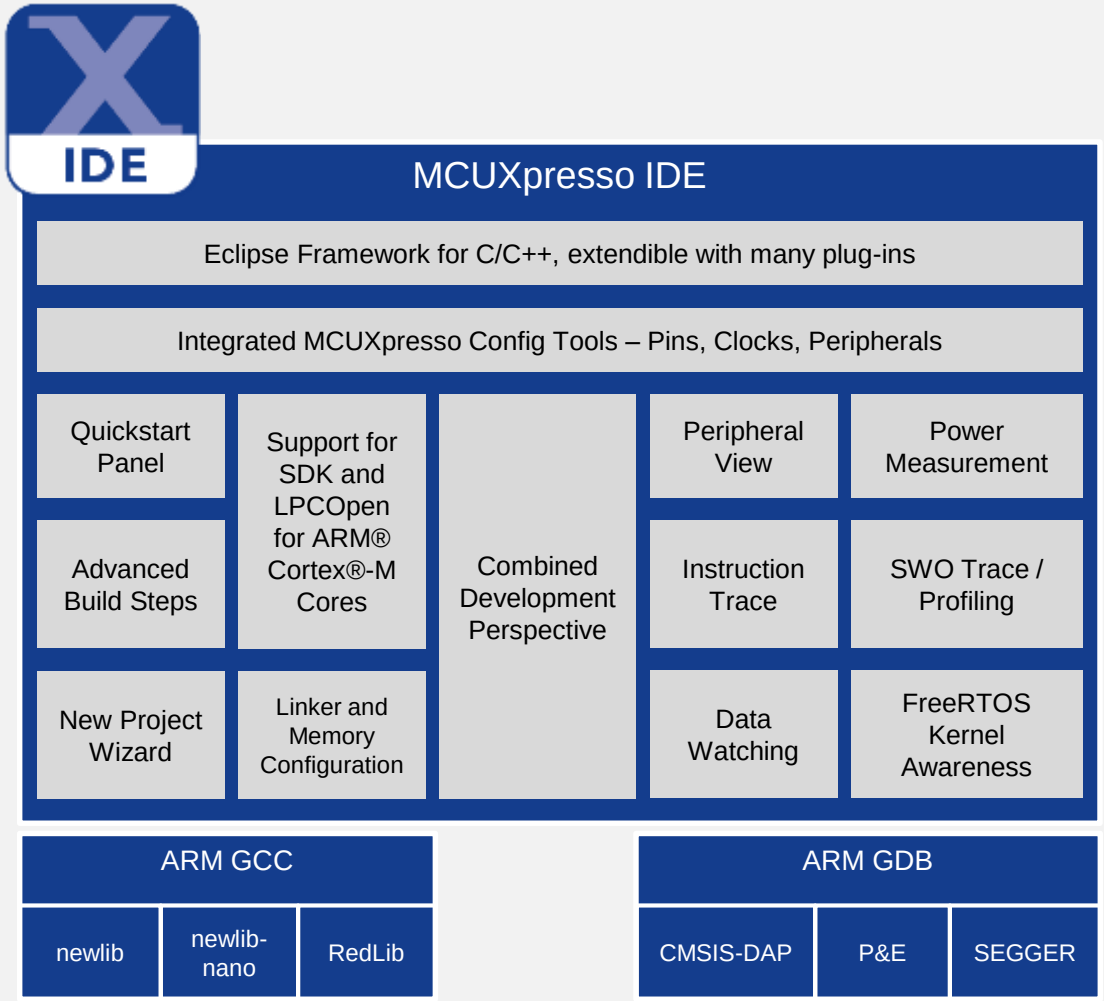
For NXP Cortex-M controllers

- Kinetis MCUs
- LPC Microcontrollers
- i.MX RT Crossover Processors

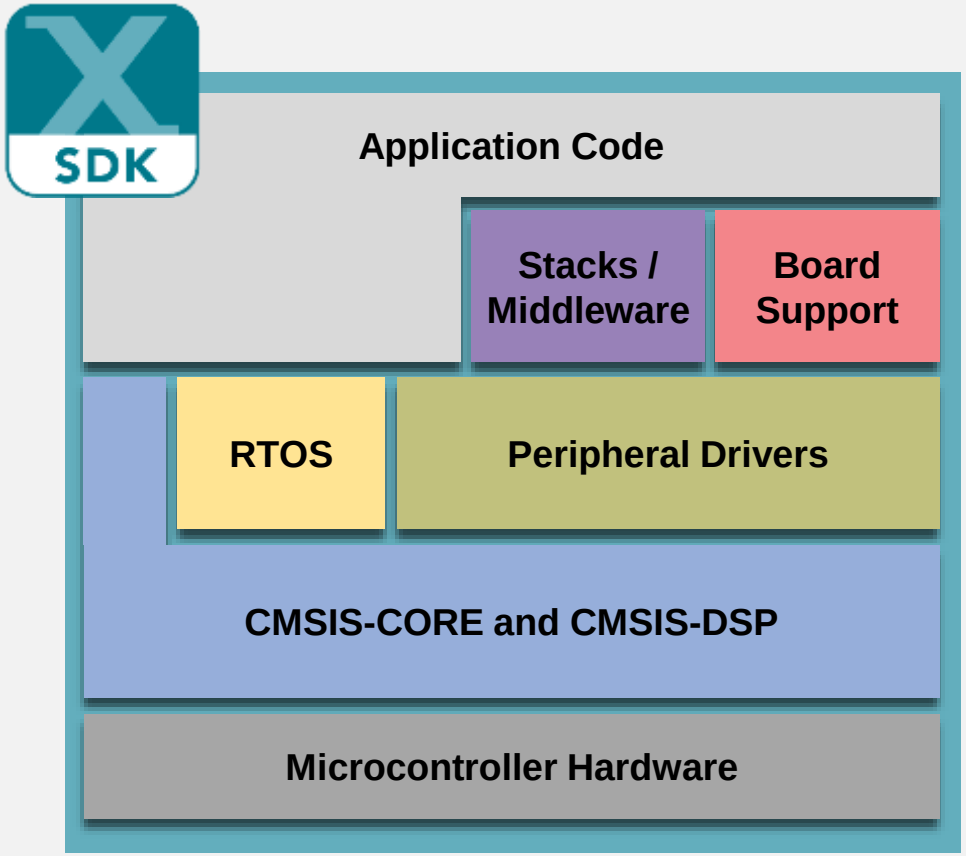
2017 – 2018

MCUXpresso IDE

Free Eclipse / GCC-based development



- **Feature-rich**, **unlimited code size**, optimized for ease-of-use, based on industry standard Eclipse framework for NXP's **Kinetis** and **LPC** MCUs and **i.MX RT** crossover processors
- Application development with Eclipse and GCC-based IDE for advanced editing, compiling and debugging
- Supports custom development boards, Freedom, Tower and LPCXpresso boards with debug probes from NXP, P&E and Segger
- **Free:** Full Featured, unlimited Code Size, no special activation needed, community based support, advanced trace capabilities, MTB and ETB instruction trace
- Works in conjunction with **MCUXpresso Config Tools** and **MCUXpresso SDK** to provide complete development environment



MCUXpresso SDK

Software framework and drivers

Architecture:

- CMSIS-CORE compatible
- Single driver for each peripheral
- Transactional APIs w/ optional DMA support for communication peripherals

Integrated RTOS:

- Amazon FreeRTOS
- RTOS-native driver wrappers

Integrated Stacks and Middleware:

- USB Host, Device and OTG
- lwIP, FatFS, LittleFS
- Crypto acceleration plus wolfSSL & mbedTLS
- SD and eMMC card support

Reference Software:

- Peripheral driver usage examples
- Application demos
- FreeRTOS usage demos
- AWS WiFi and lwIP examples

License:

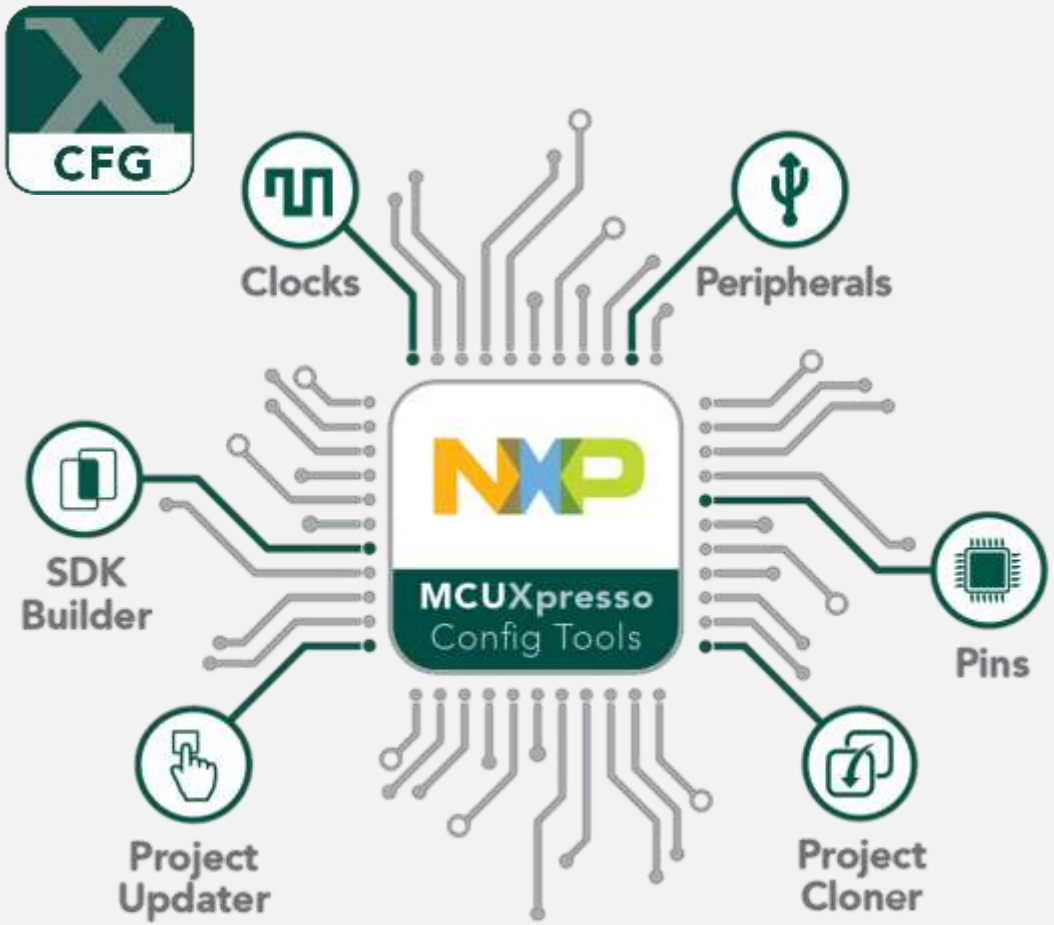
- Clear BSD 3-clause for startup, drivers, USB stack

Toolchains:

- MCUXpresso IDE
- IAR®, ARM® Keil®, GCC w/ Cmake

Quality:

- Production-grade software
- MISRA 2004 compliance
- Checked with Coverity® static analysis tools



MCUXpresso Config Tools

Configuration and Code Generation

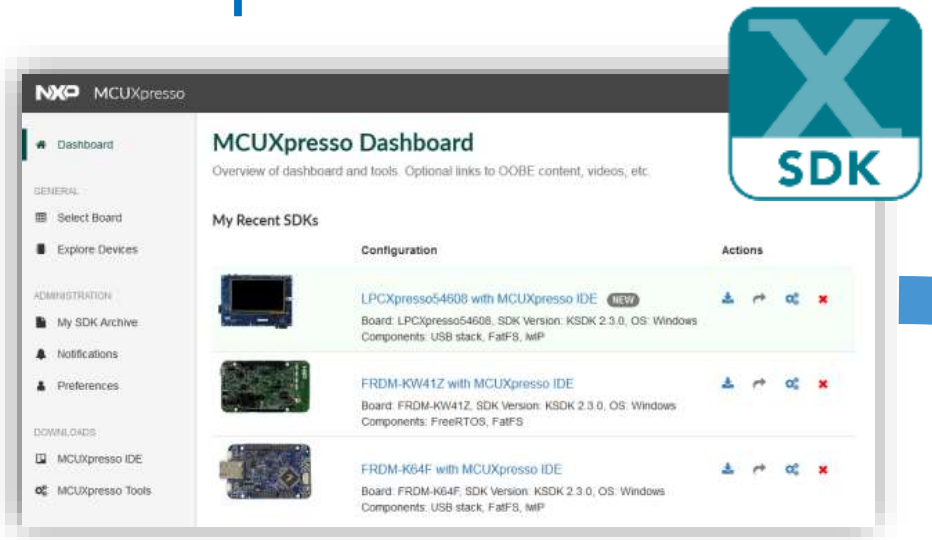
SDK Builder packages custom SDKs based on user selections of MCU, evaluation board, and optional software components.

Pins, **Clocks**, and **Peripheral** tools generate initialization C code for custom board support. Features validation of inputs and cross-tool conflict resolution.

Project Update works directly with existing SDK-based IDE projects with generated Pins, Clocks, and Peripheral source files.

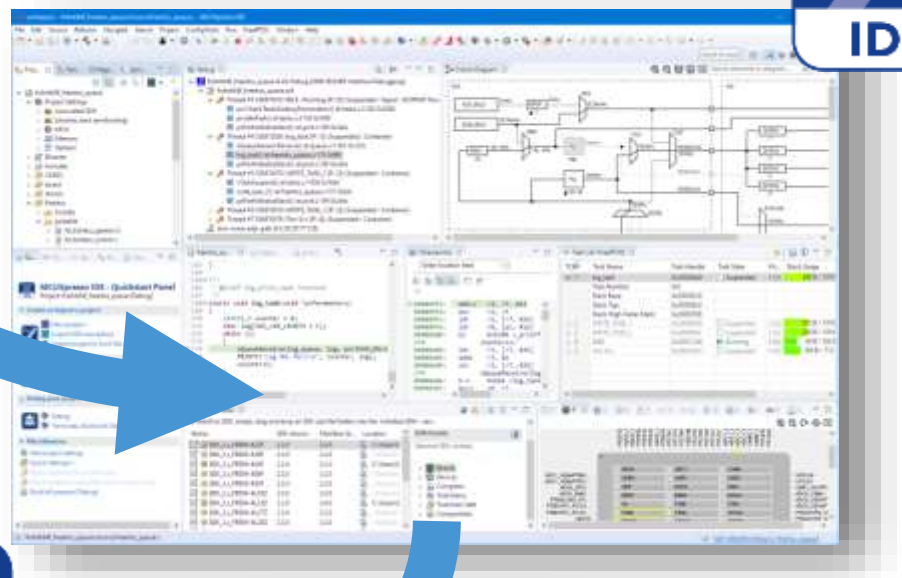
Project Cloning creates a standalone SDK project based on a example application available within SDK release.

Development Flow

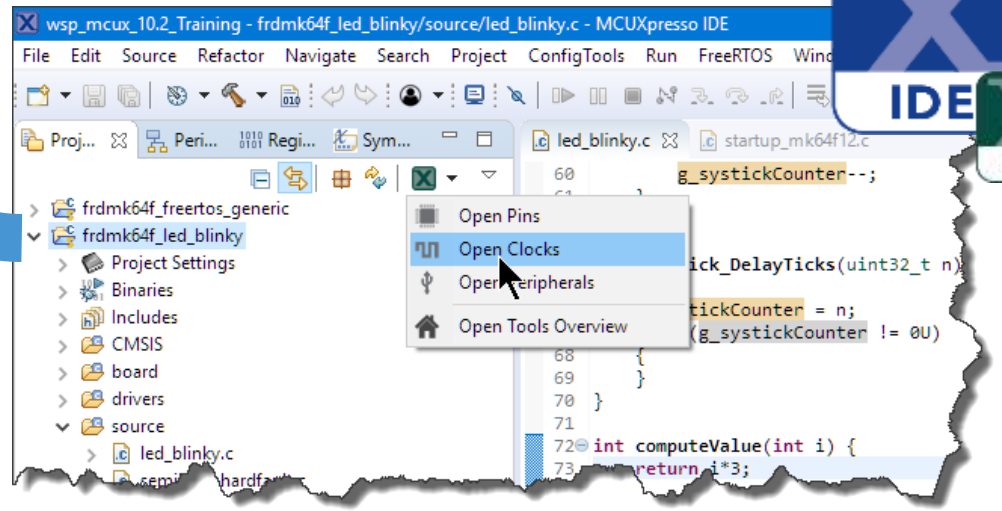


SDK

ZIP

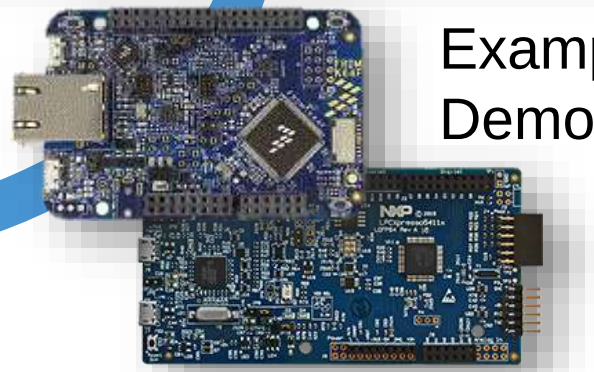


IDE



IDE

CFG



Examples,
Demo Apps

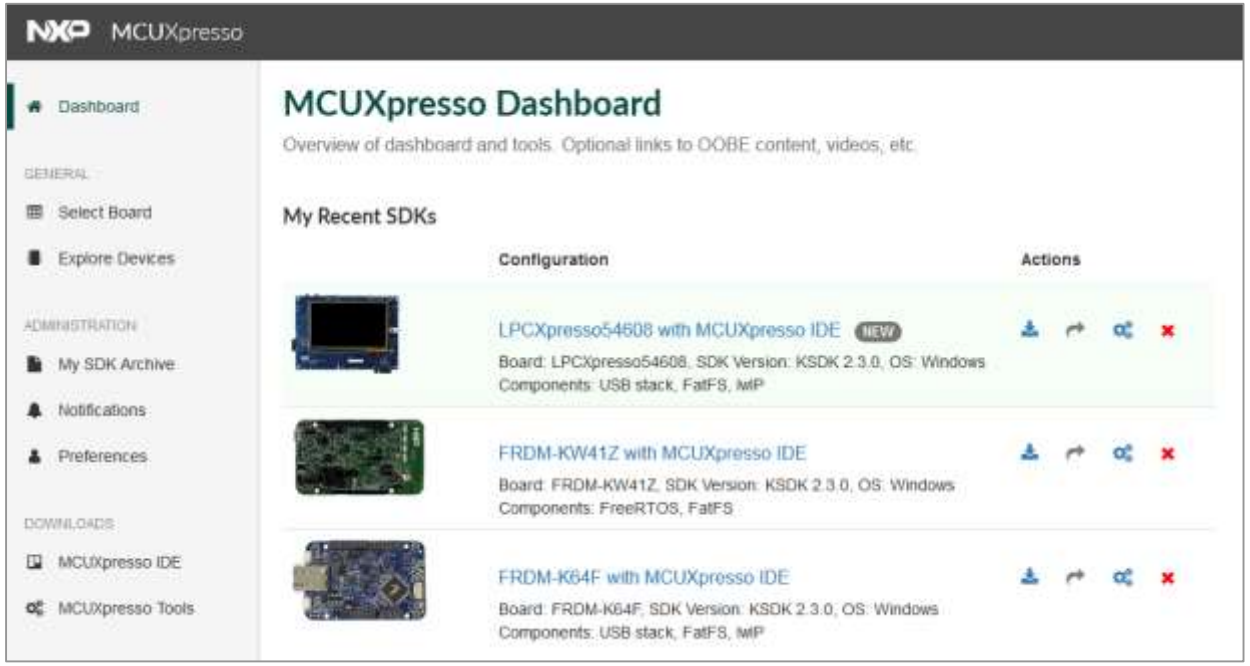
Walkthrough of Tools

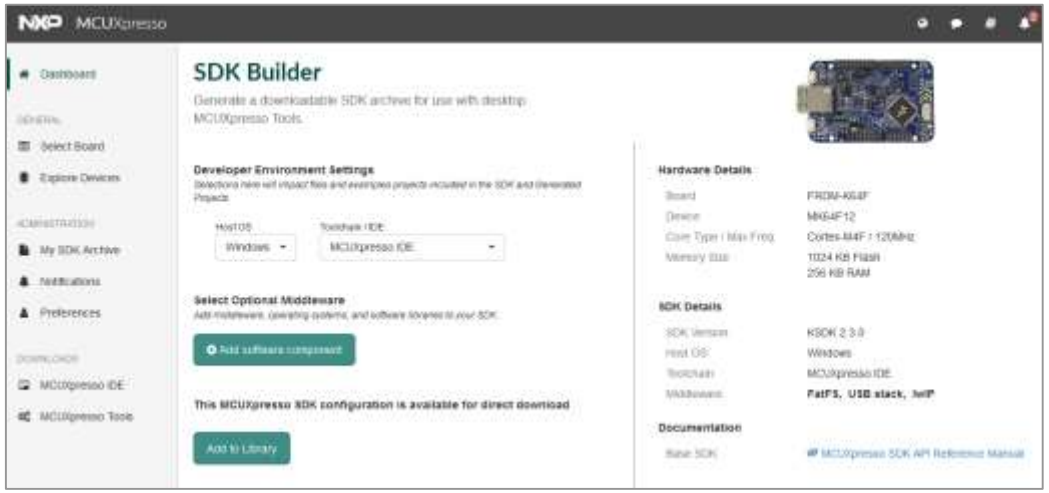
- MCUXpresso SDK Builder
- MCUXpresso Config Tool
- MCUXpresso IDE

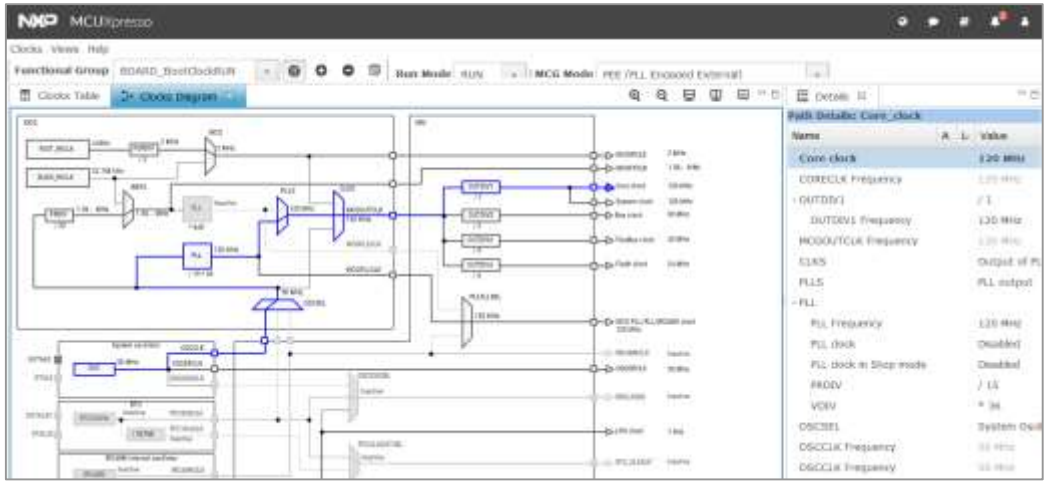
MCUXpresso SDK Builder

(http://mcuxpresso.nxp.com)

- MCUXpresso Dashboard
- MCUXpresso SDK Builder
- Online MCUXpresso Config Tools

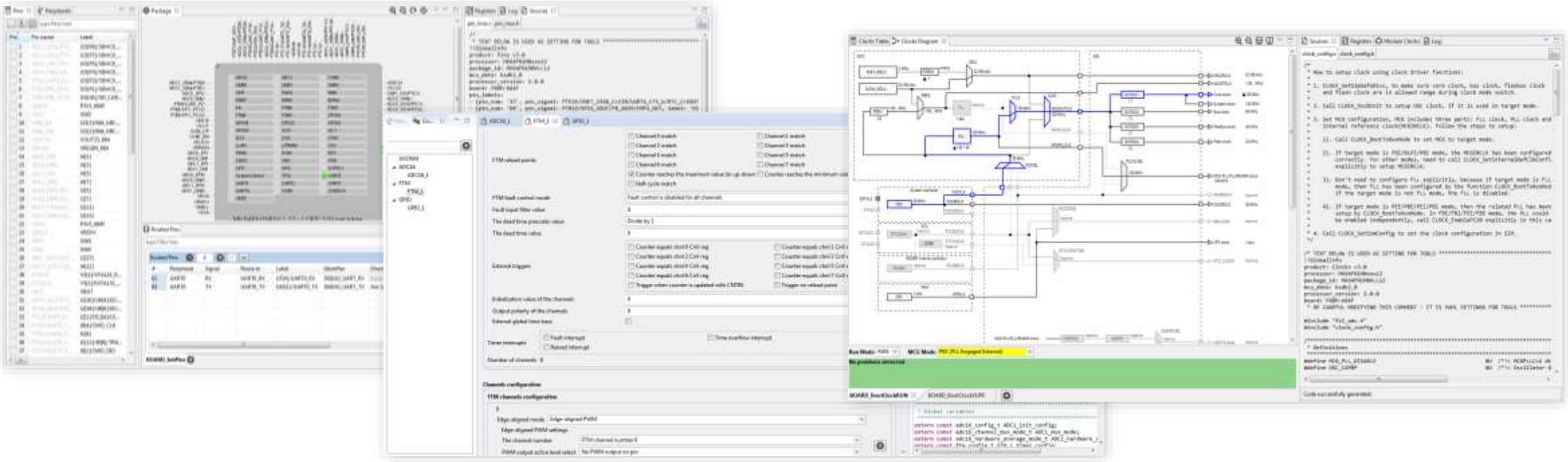






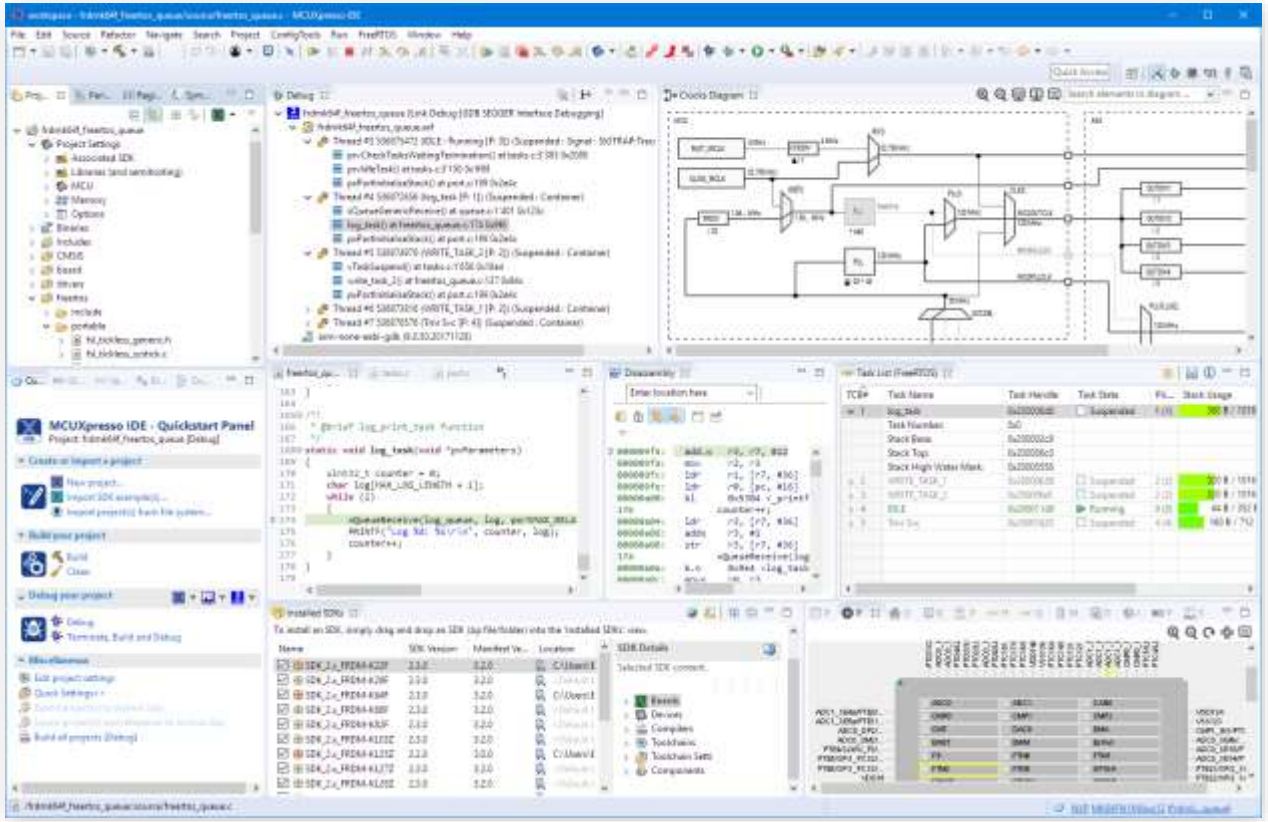
MCUXpresso Config Tools

- Integrated in IDE or standalone
- Pins, Clock, and Peripheral Initialization
- Code Generation
- Conflict Resolution



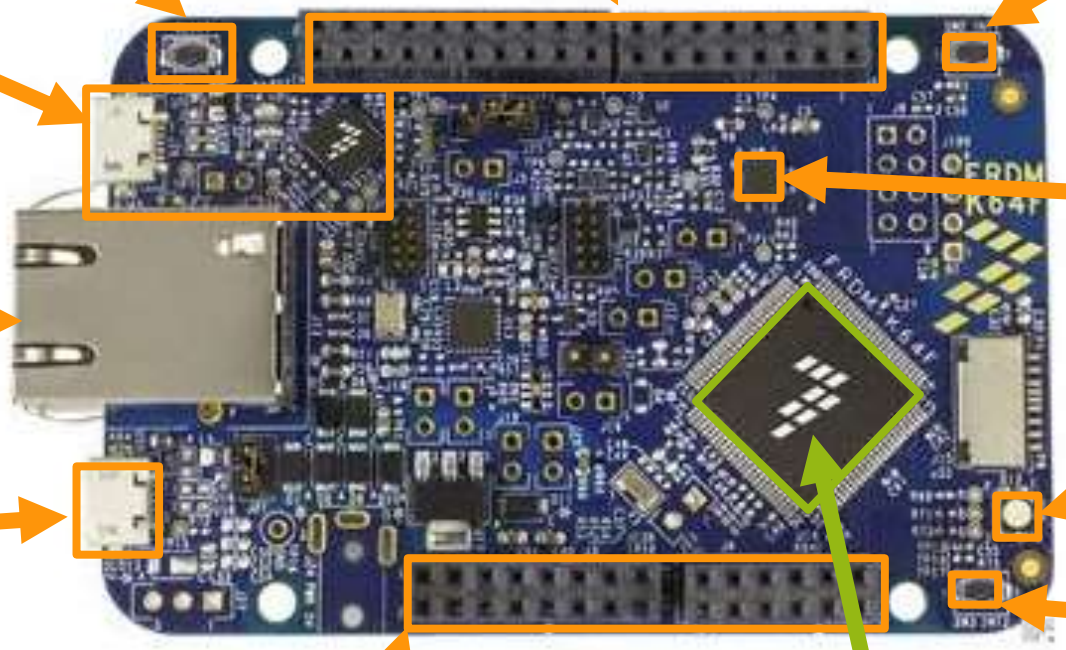
MCUXpresso IDE

- MCUXpresso SDK Installation
- Development Perspective
- Integrated MCUXpresso Config Tools



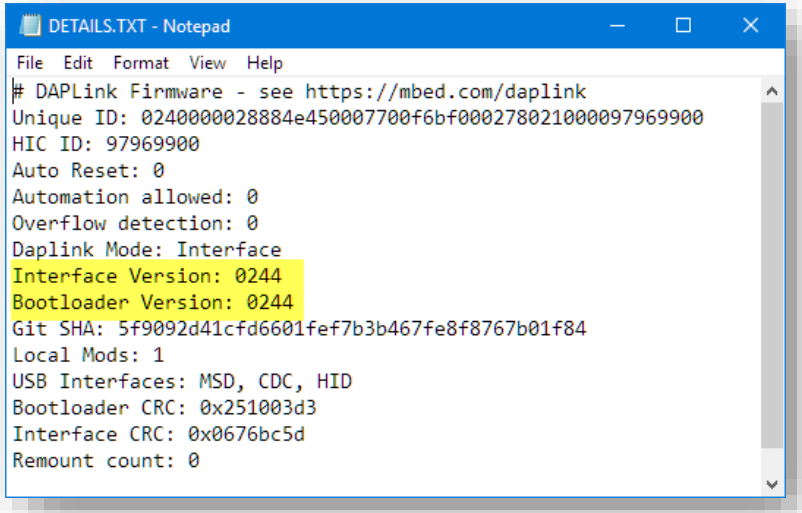
Learn more at: http://www.nxp.com/frdm-k64f

FRDM-K64F Hardware Overview

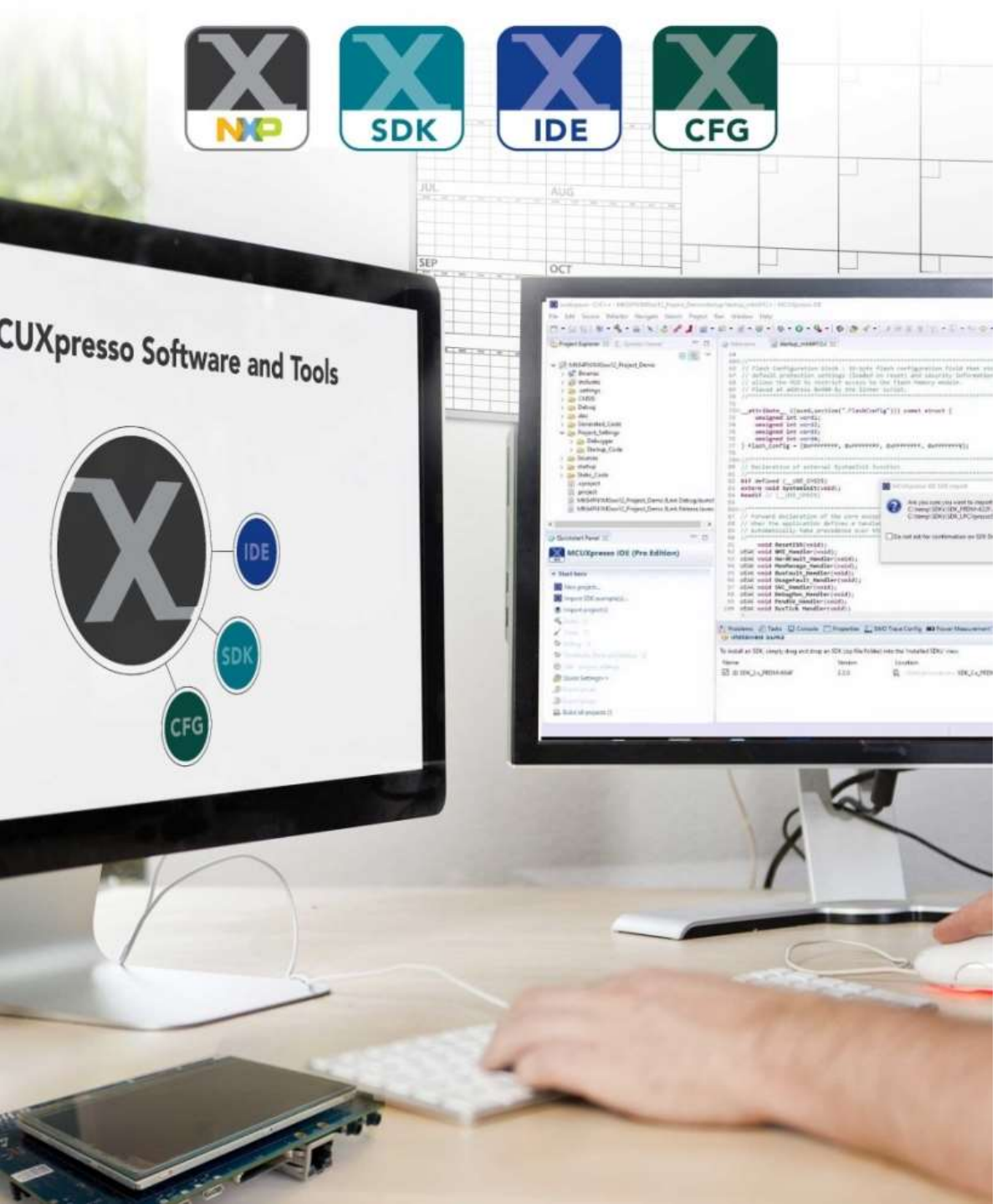


Lab Setup/Prerequisites

- MCUXpresso IDE 10.2.0 build 759
  - https://www.nxp.com/mcuxpresso/ide
- FRDM-K64 SDK V2.4.0
  - http://mcuxpresso.nxp.com/
- FRDM-K64F Board with micro-USB Cable
  - https://www.nxp.com/freedom
- FRDM-K64F Board with DAPlink/CMSIS-DAP Firmware
  - Bootloader rev0244 OpenSDA v2.2
  - DAPLink reve0244 Firmware
  - https://www.nxp.com/opensda



Hands-On Lab

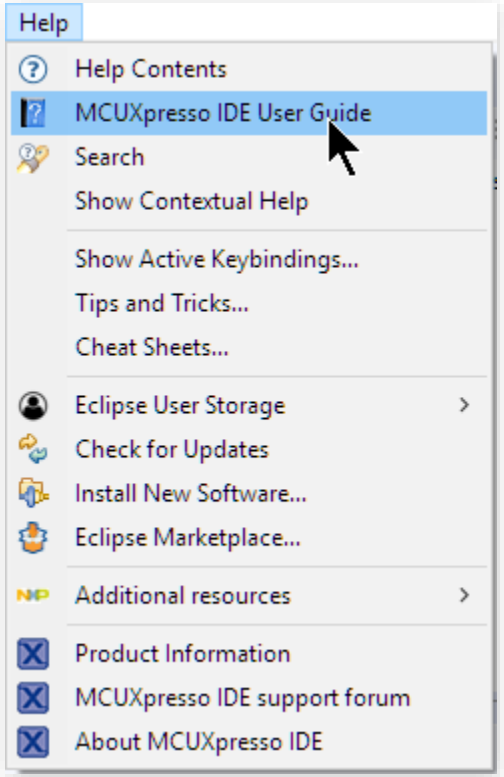


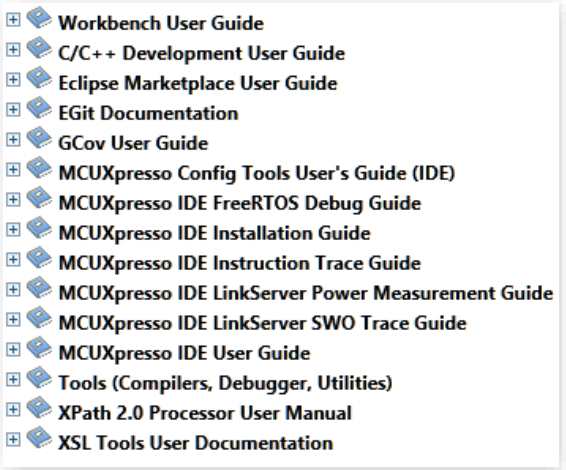
Lab Overview

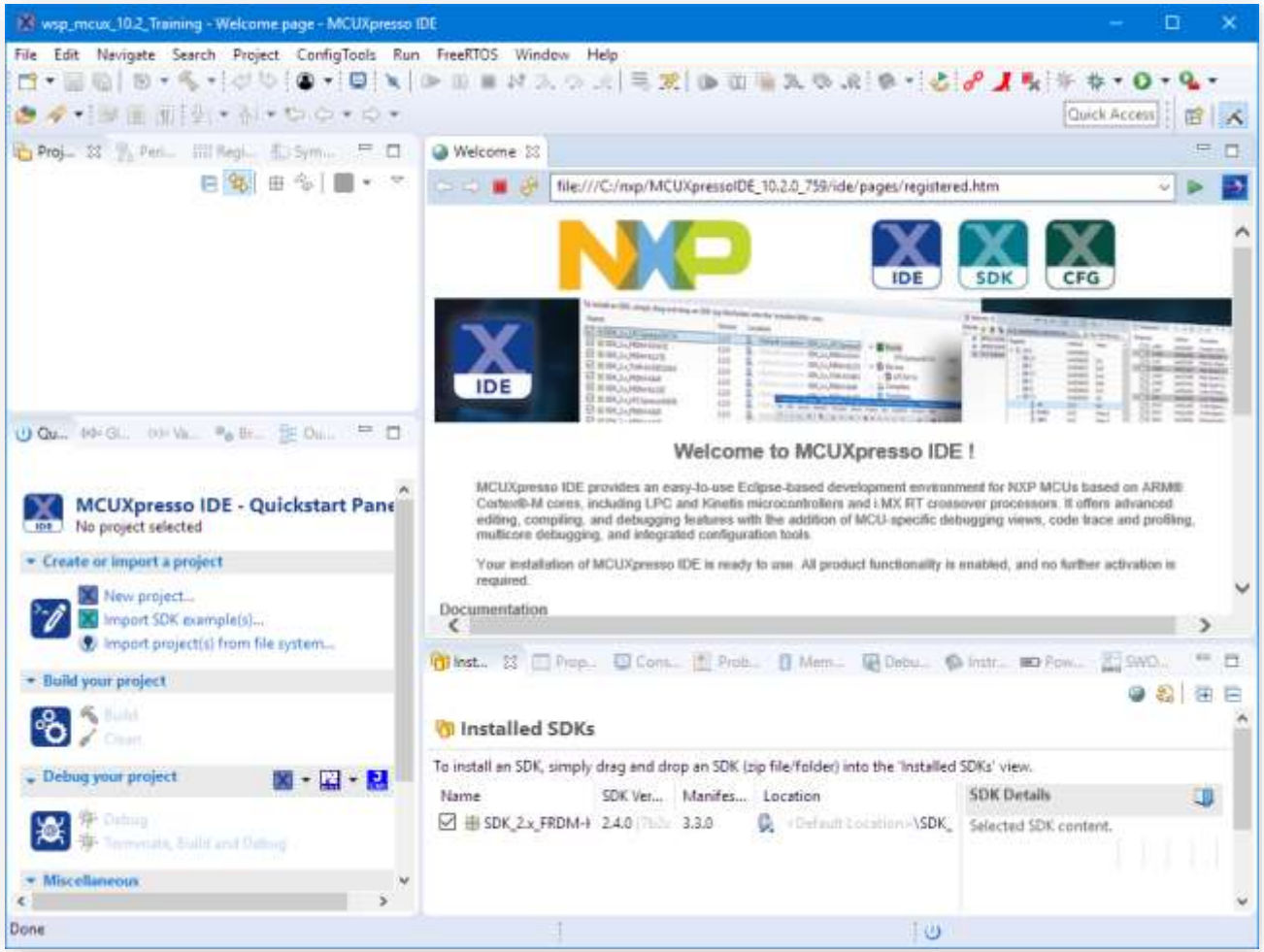
1. Creating/Cloning MCUXpresso SDK Projects in the IDE
2. Building, Basic Debugging
3. Startup, Connect, Disconnect, Attach
4. GUI Flash Tool
5. Global Variables, Variable Plots
6. Variables, Data, Peripherals
7. Breakpoints, Watchpoints
8. Trace
9. FreeRTOS TAD
10. *(Backup: SWO Features)*

MCUXpresso IDE

- Start the IDE with Shortcut
- Select workspace
- Open MCUXpresso IDE User Guide



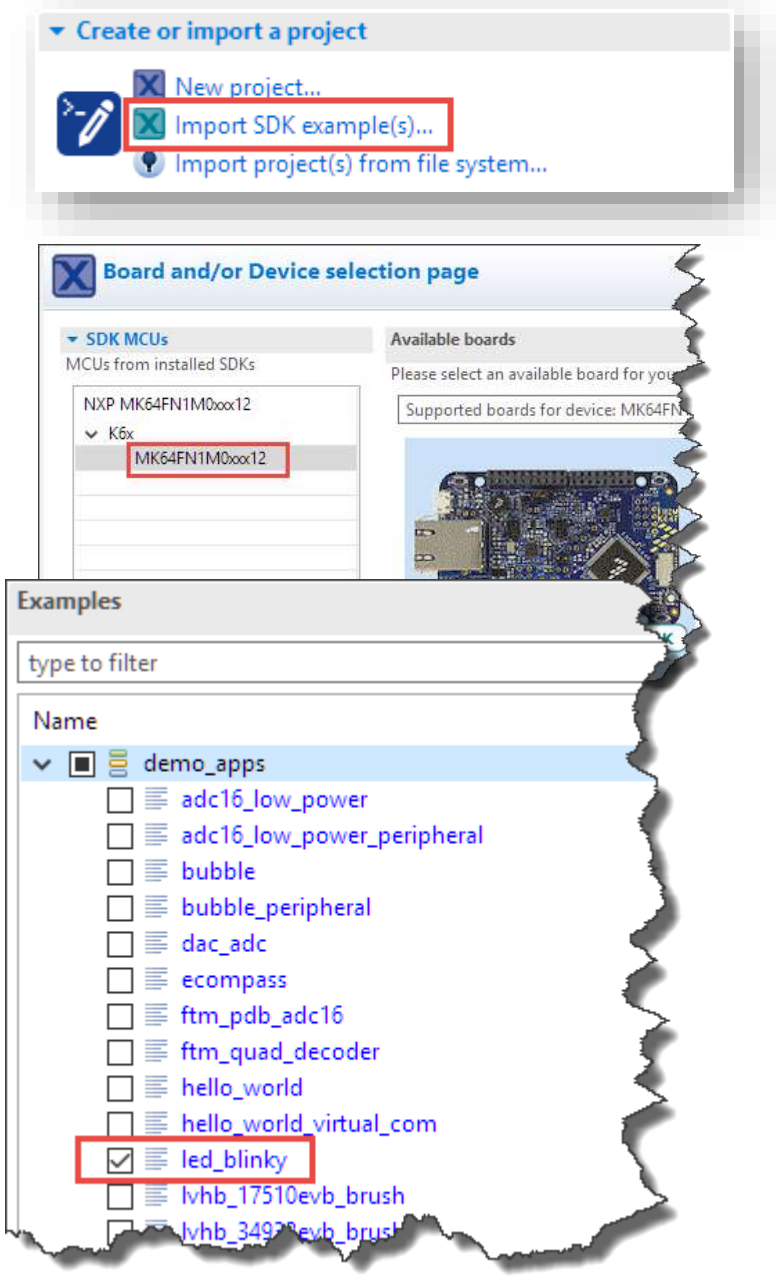




Create LED Blinky Project

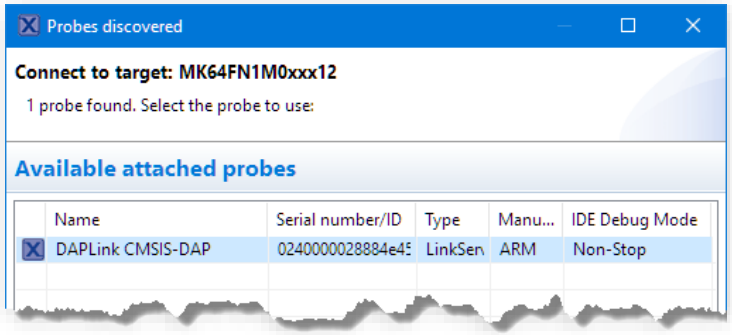
- Prerequisite:
  - **SDK_2.x_FRDM-K64F** installed in IDE

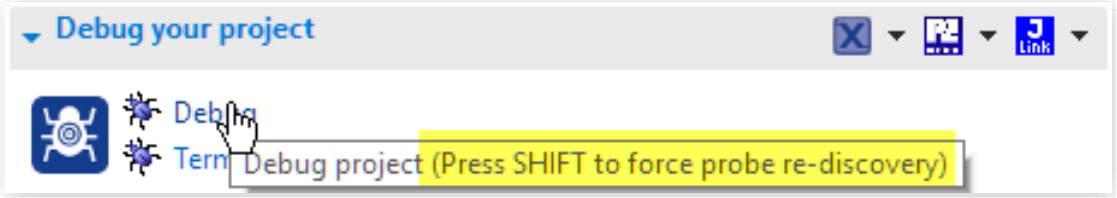
- Use Quickstart Panel with “**Import SDK example(s)**…”
  - Select **frdmk64f** board image
  - Click “**Next**”
  - Select “**demo_apps > led_blinky**”
  - Click “**Finish**”

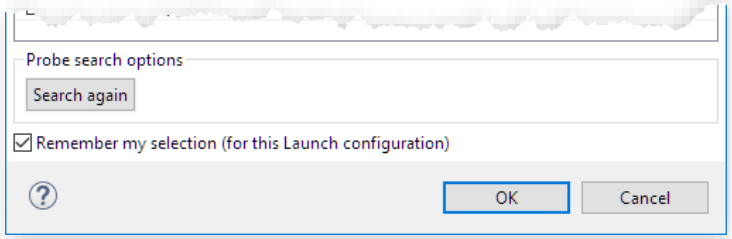


Build and Debug

- **Connect FRDM-K64F board** (micro USB to OpenSDA)
  - Windows may need to enumerate USB connection
- **Use IDE Quickstart Panel to:**
  - **Clean**
  - **Build**
  - **Debug**
- Debugger discovers probe
  - Use SHIFT to force probe re-discovery

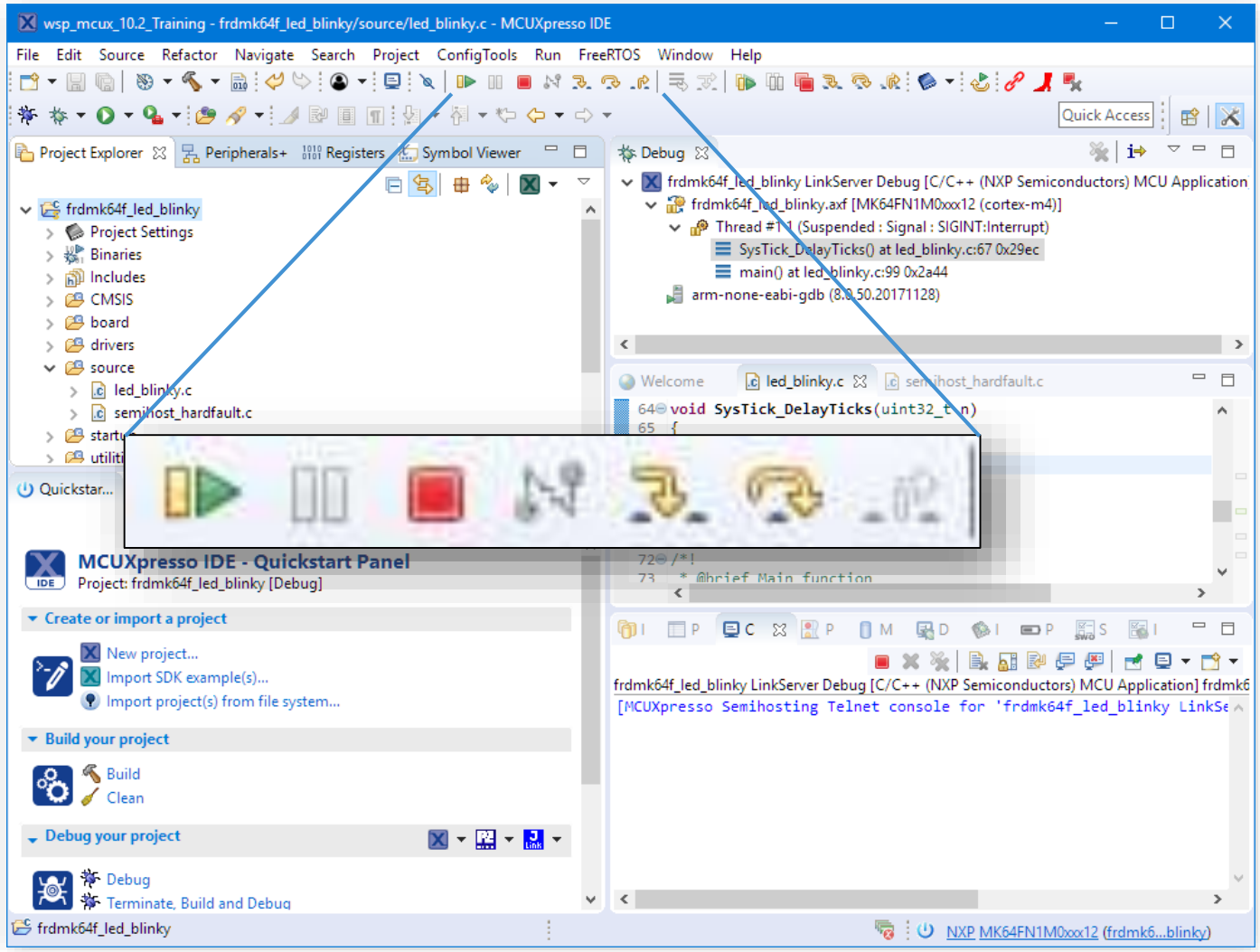






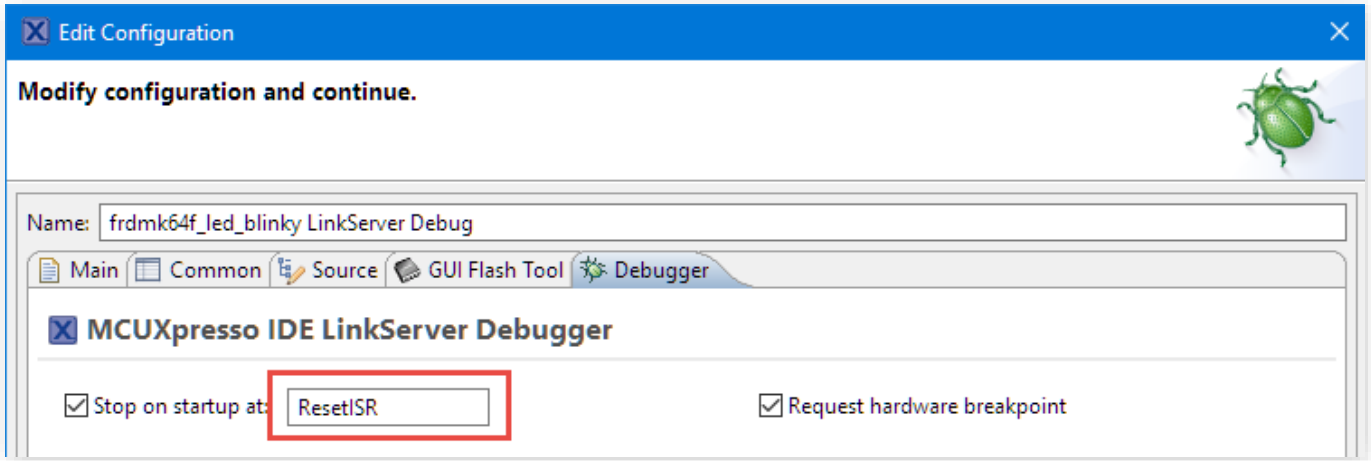
Debugger Run Control

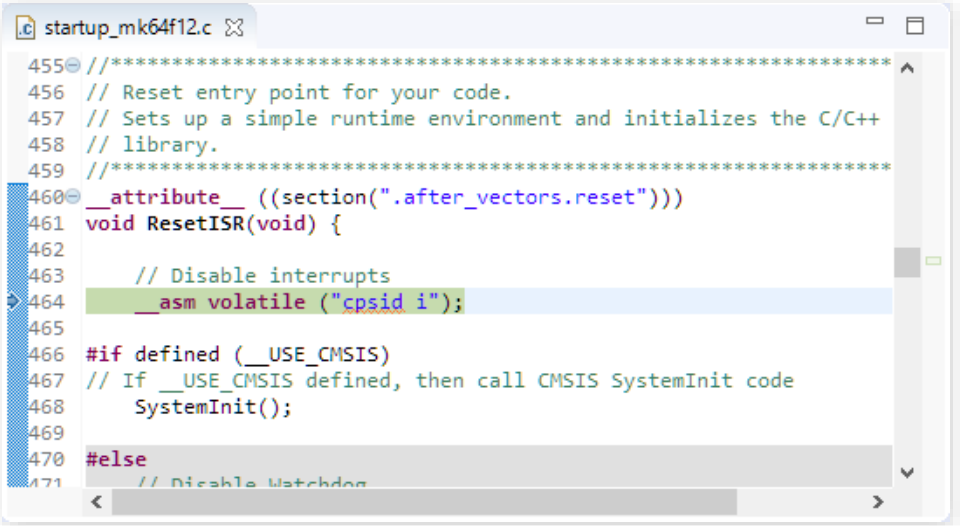
- Resume/Run
- Suspend/Pause
- Terminate/Stop
- (Disconnect)
- Step Into
- Step Over
- Step Out
- **Step trough the code**



Debug Startup Breakpoint

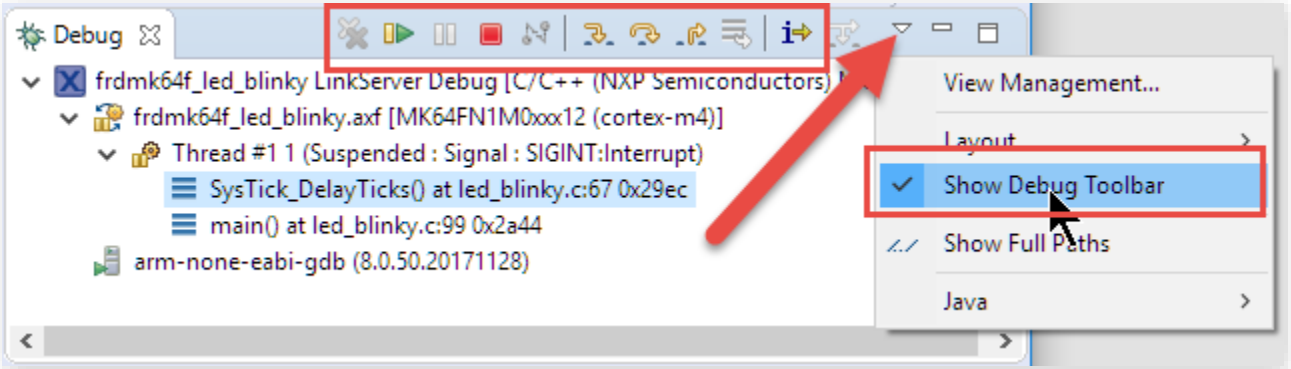
- By default, target runs until main()
- **Double-Click on Debug *.launch File in Project**
  - Opens Debug Configuration
  - Stop at **ResetISR**
- **Debug**
- Can now debug startup code

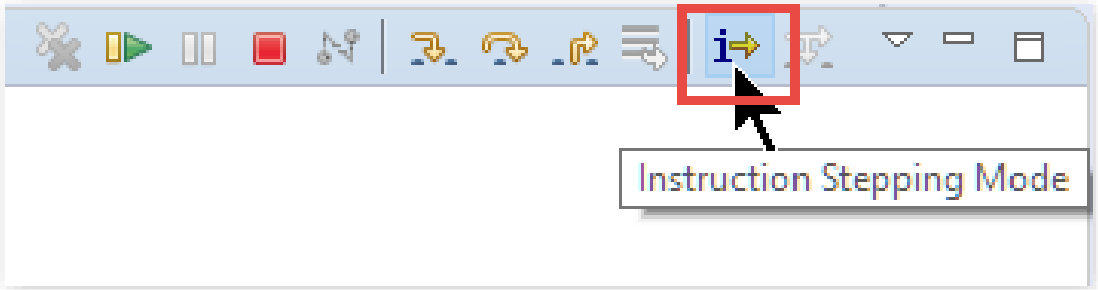


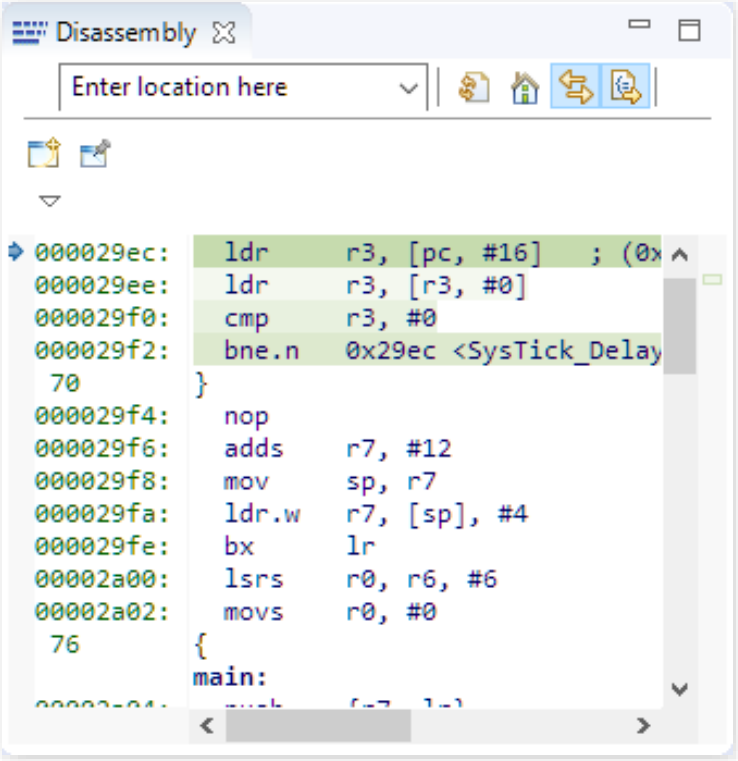


Debug Toolbar in Debug View, Assembly Stepping

- Enable debug toolbar
- Turn on instruction stepping
- Perform stepping
- Switch back to Source stepping

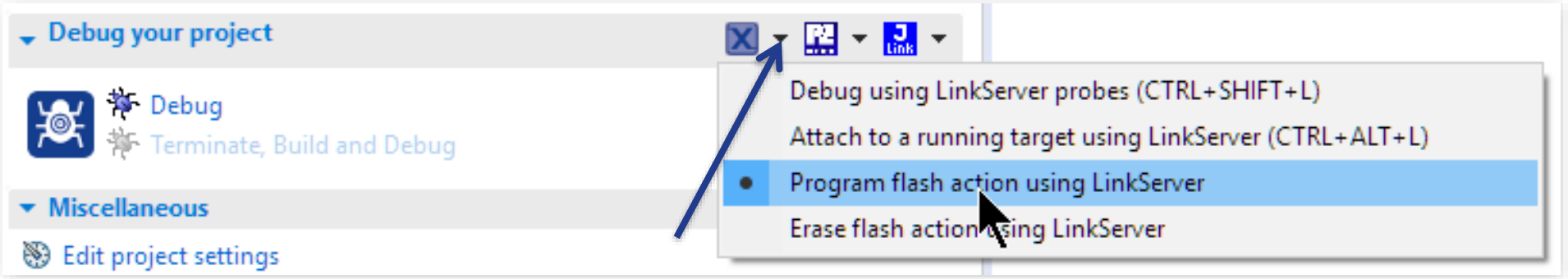






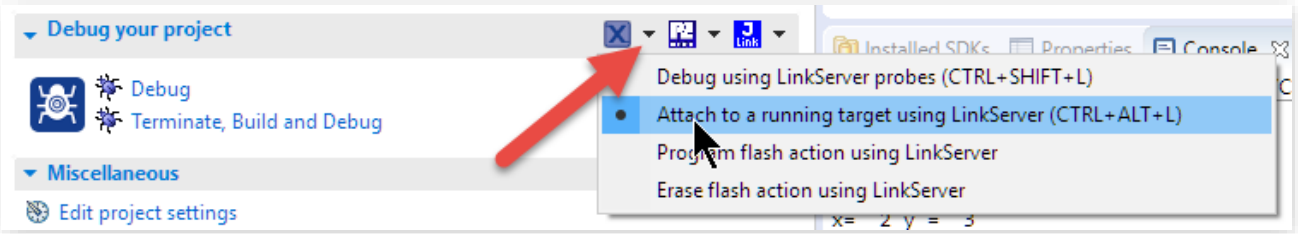
Debug Quickstart Shortcuts

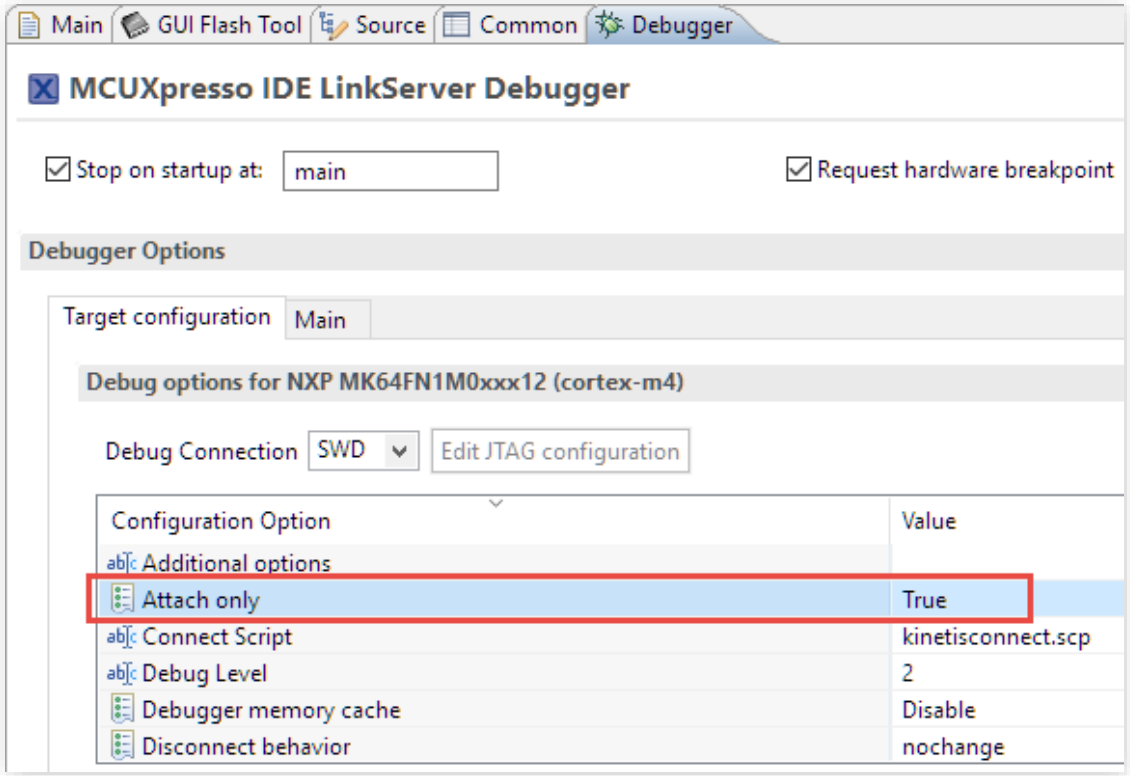
- Quick and fast way to
  - Debug (default)
  - Attach
  - Program
  - Erase
- Setting is persistent between sessions



Debug Quickstart Shortcuts: Attach

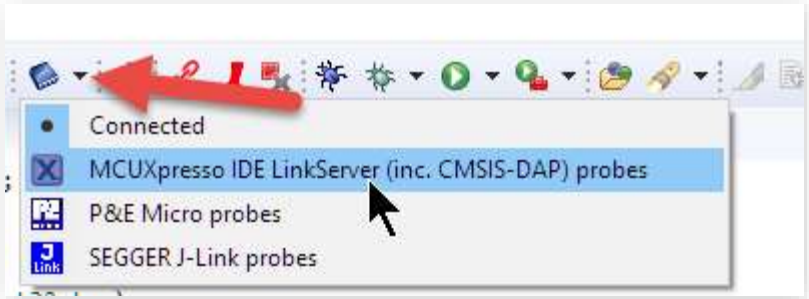
- **Use ‘Attach’ to running target**
- Launch Configuration Icon has ‘A’ Decorator added
- Persistent Setting in Setting in Debug Configuration

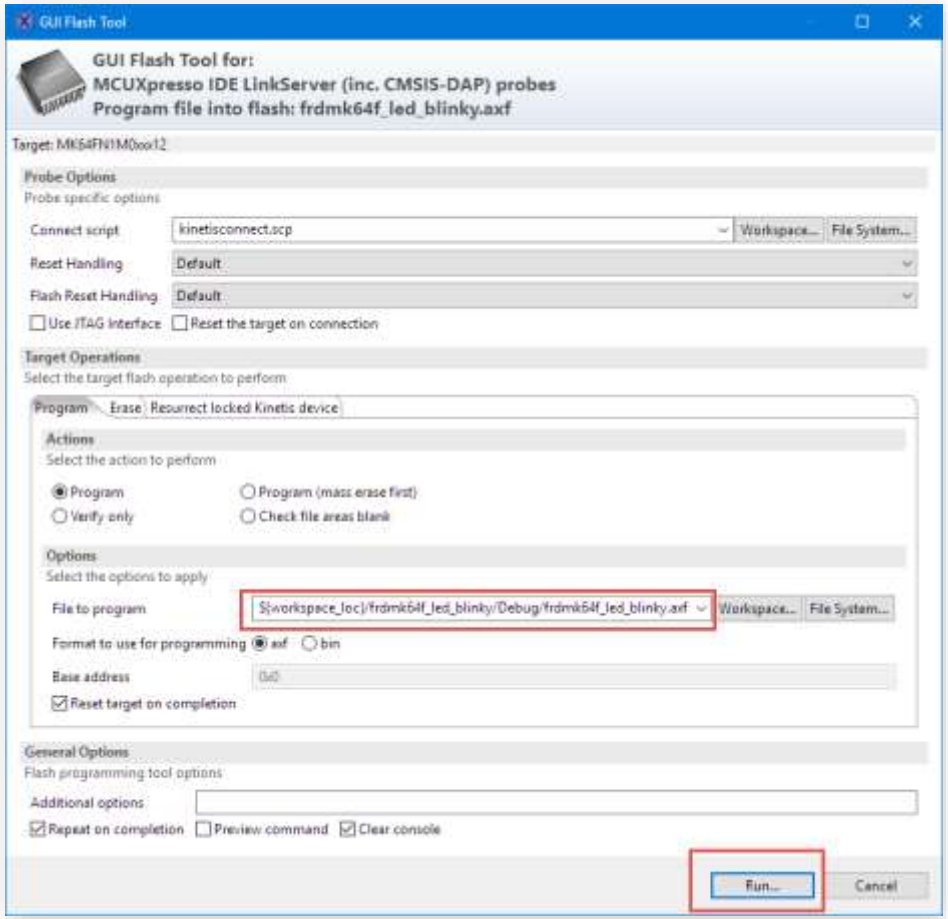


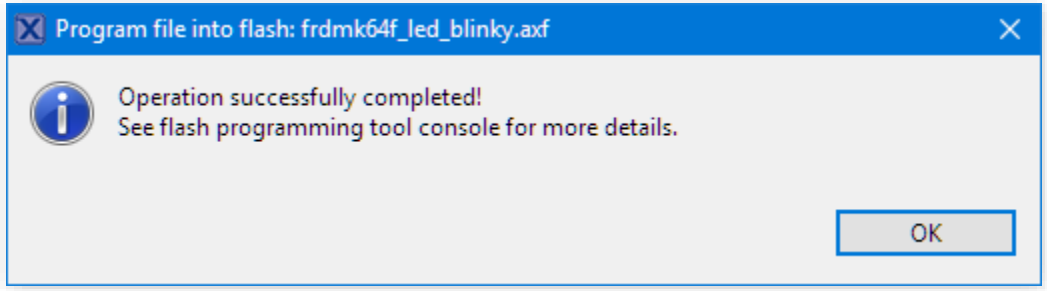


GUI Flash Tool

- Advanced Board flashing/programming
  - Program, verify, erase, check, resurrect, ...
  - Programming multiple boards
- Select **GUI Flash Tool** Icon
- **Run** Job
- Messages written to Console View
- **Check Console View**



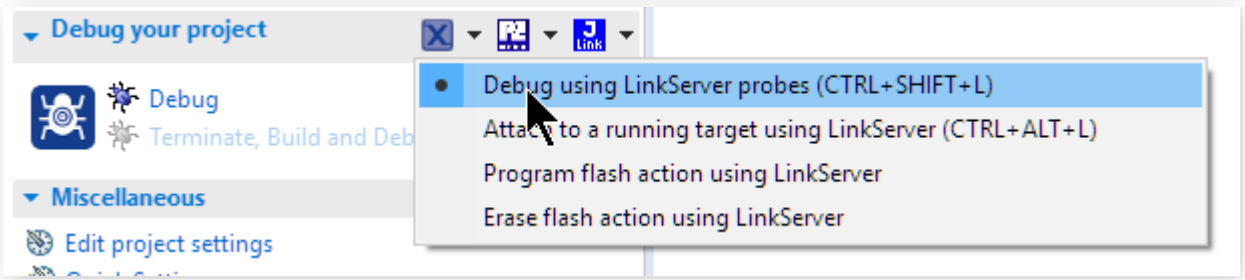


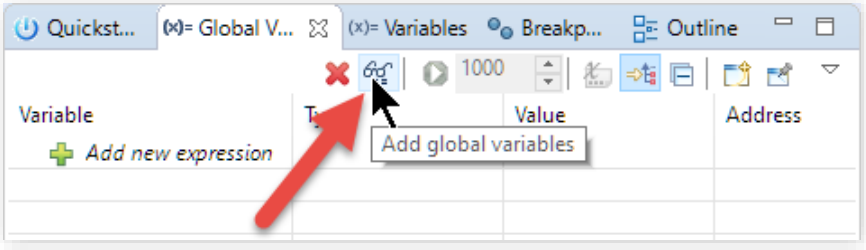


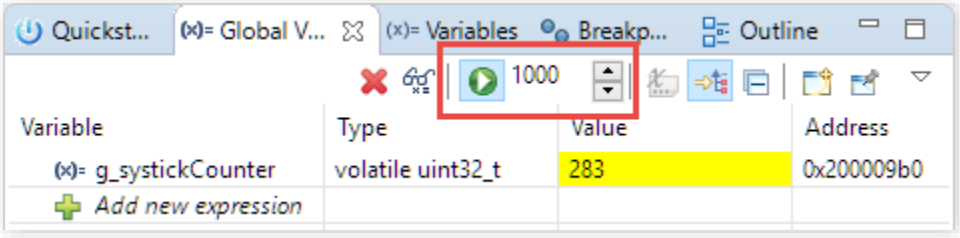
Variables, Data, Registers & Peripherals

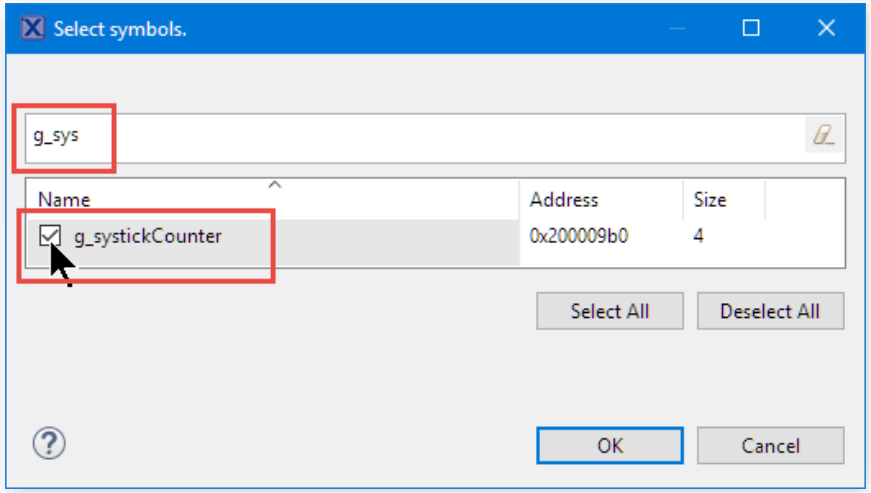
Global Variables View Update

- **Debug project**
- Global Variables View
  - **Add global variables**
  - can use Filter, press **OK**
- **Run target**
- Variable is automatically updated



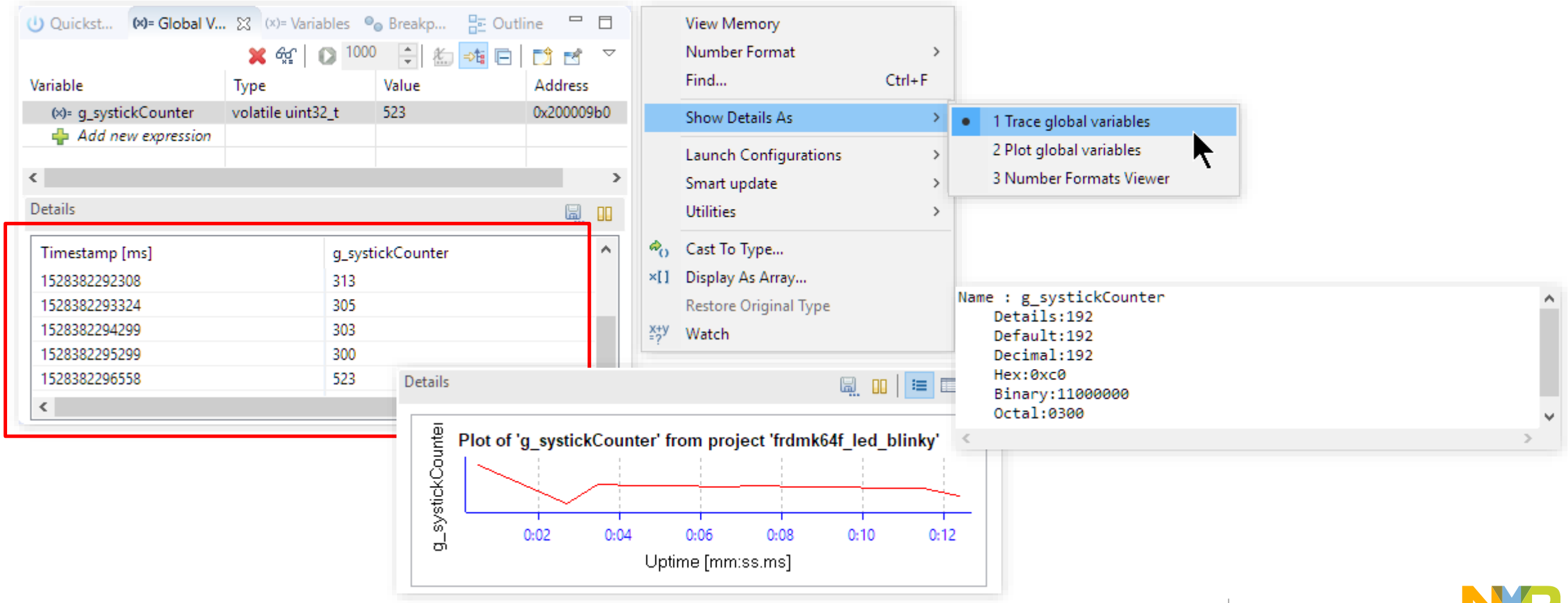






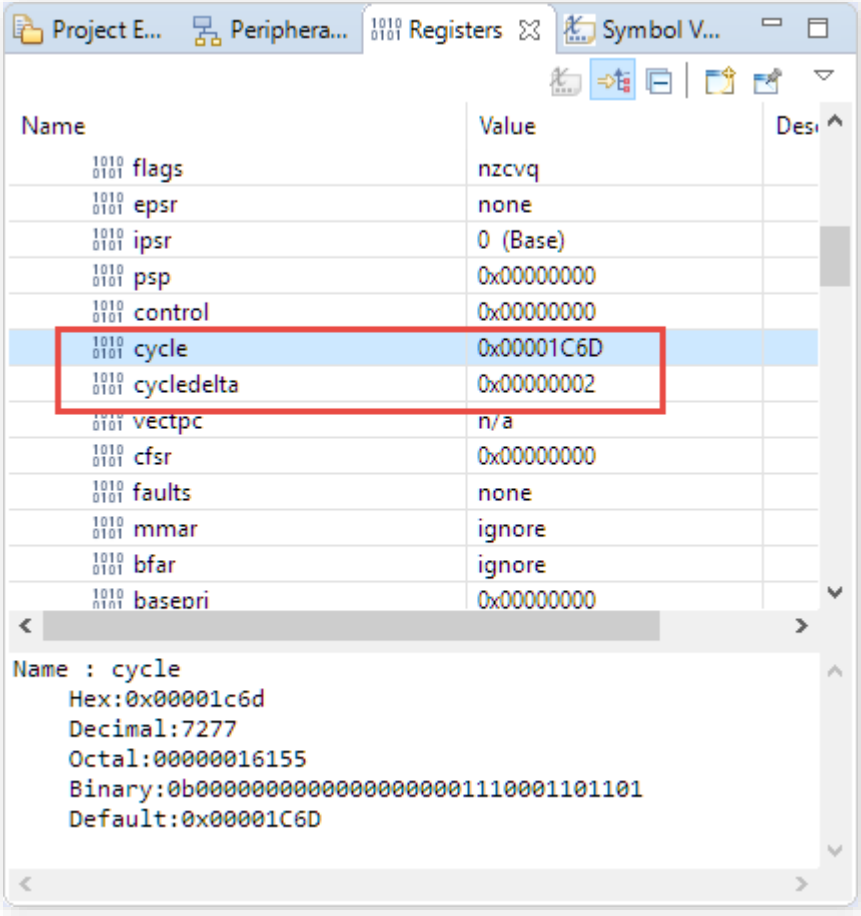
Global Variables View: Show Details

- Context menu on Variable
- **Show Details As > Trace** or **Plot** or **Number**



Registers View

- Registers View for core registers
- If view not already open: Menu *Window* > *Show View* > *Others* > *Debug* > *Registers*
- Pseudo Registers for Cycle Counters
  - **cycledelta**: cycles since the last CPU stop
- **Measure time of code execution between two breakpoints**



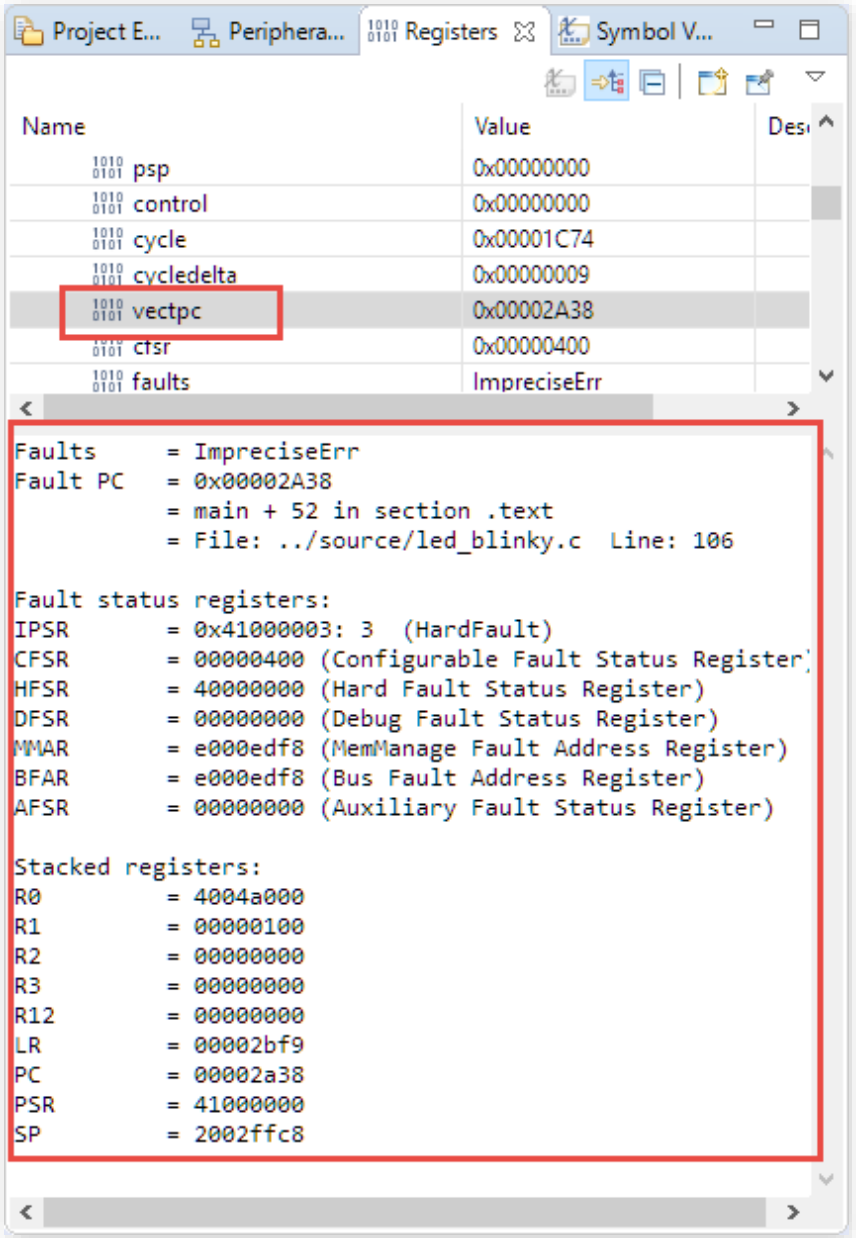
vectpc & HardFault

- **Add code writing to read-only memory**

```
*(int*)0 = 0;
```

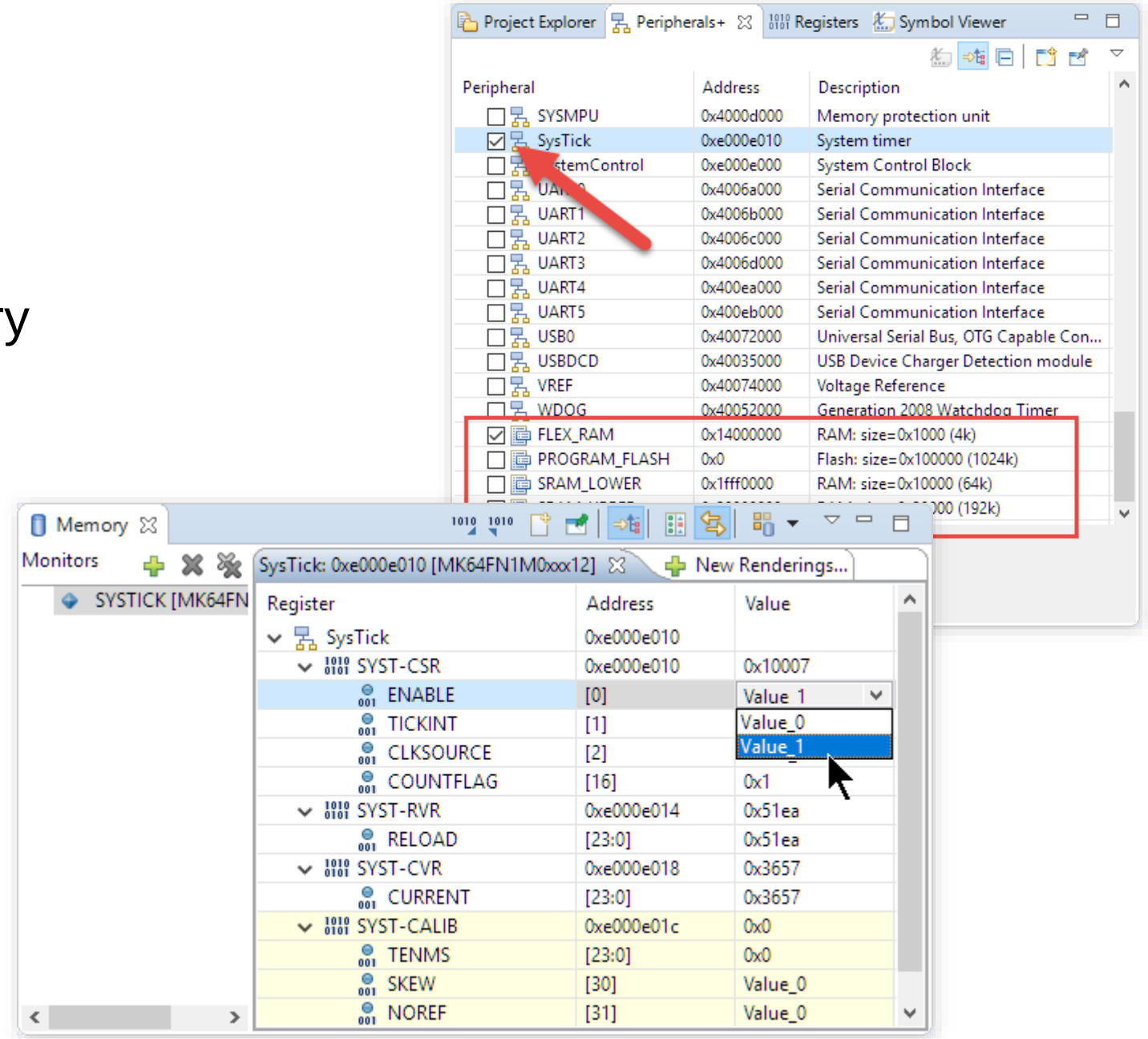
  – Causes HardFault Exception

- **Debug** and let it fault
- **vectpc** pseudo register shows fault details



Peripherals+

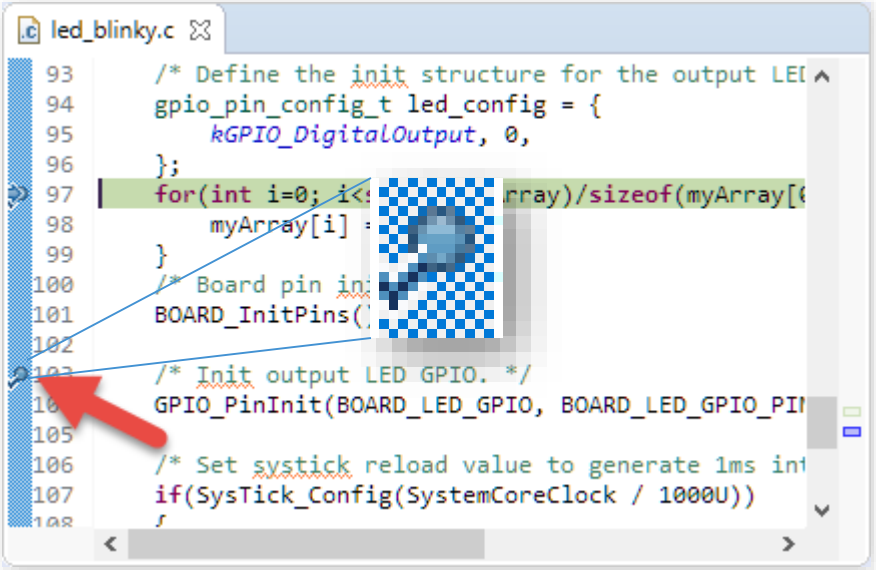
- Graphical view of memory and Peripherals
- Enabling item adds new memory rendering
- **Inspect SysTick**
- Changeable values
- Note:
  Peripheral needs
  to be clocked to read/change it

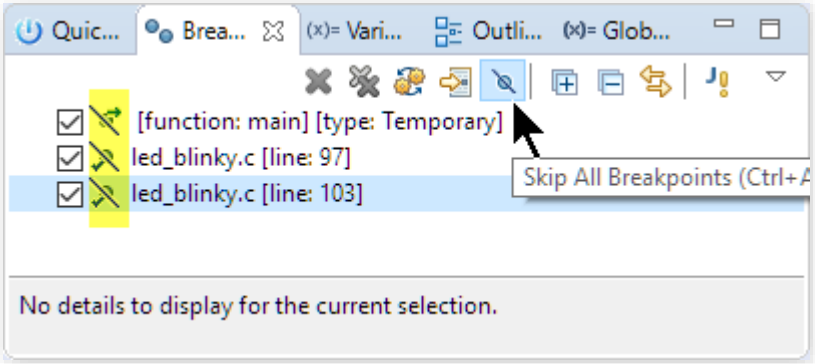


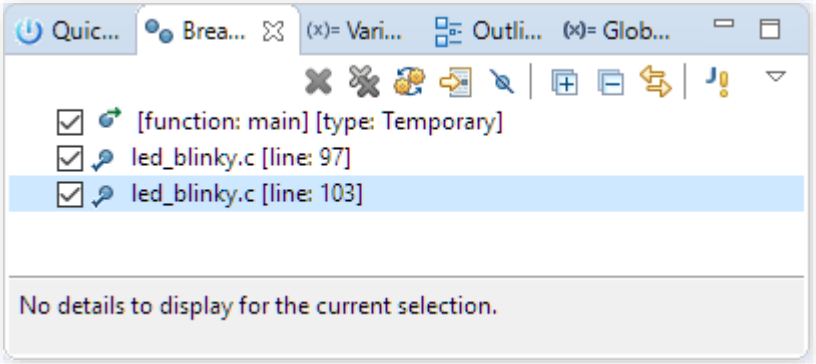
Code & Data Breakpoints

Breakpoints and Breakpoints View

- Add Breakpoint to source view
  - Double-Click into ‘blue’ ribbon
- Breakpoints are listed in Breakpoints view
- Set/Enabled breakpoint have checkmark
- Be aware of ‘Skip All Breakpoints’!

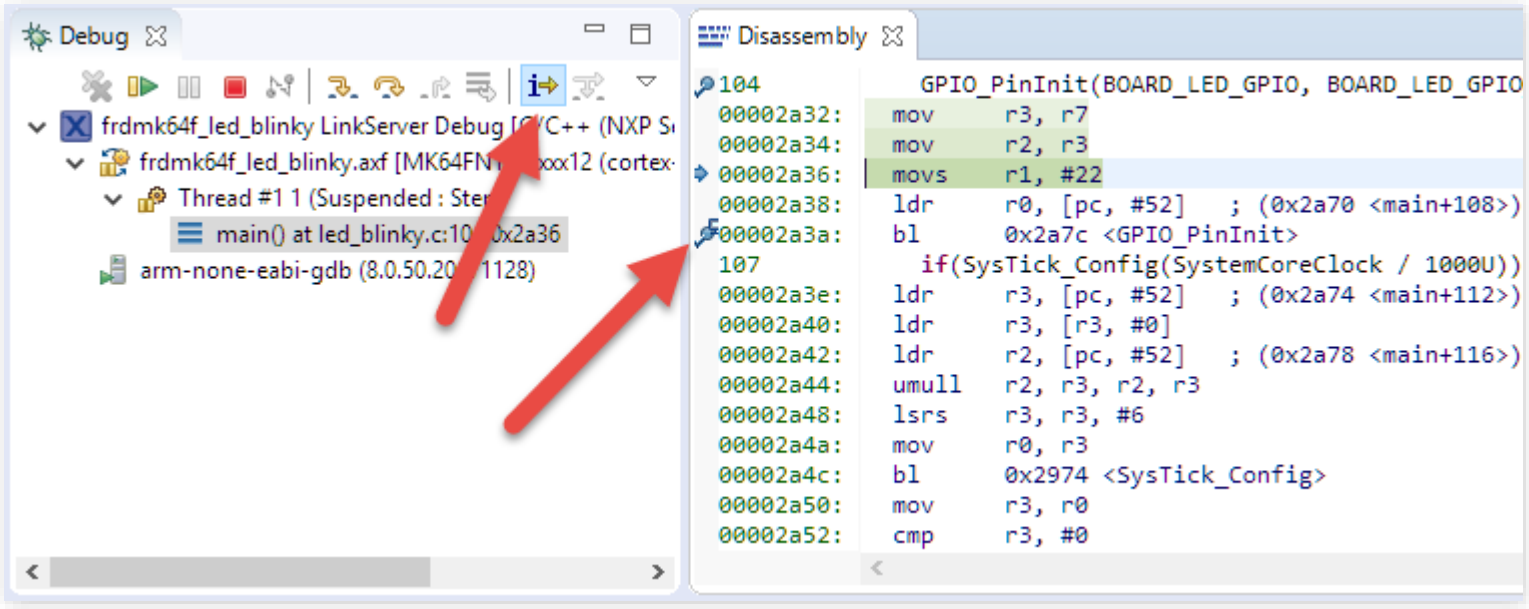






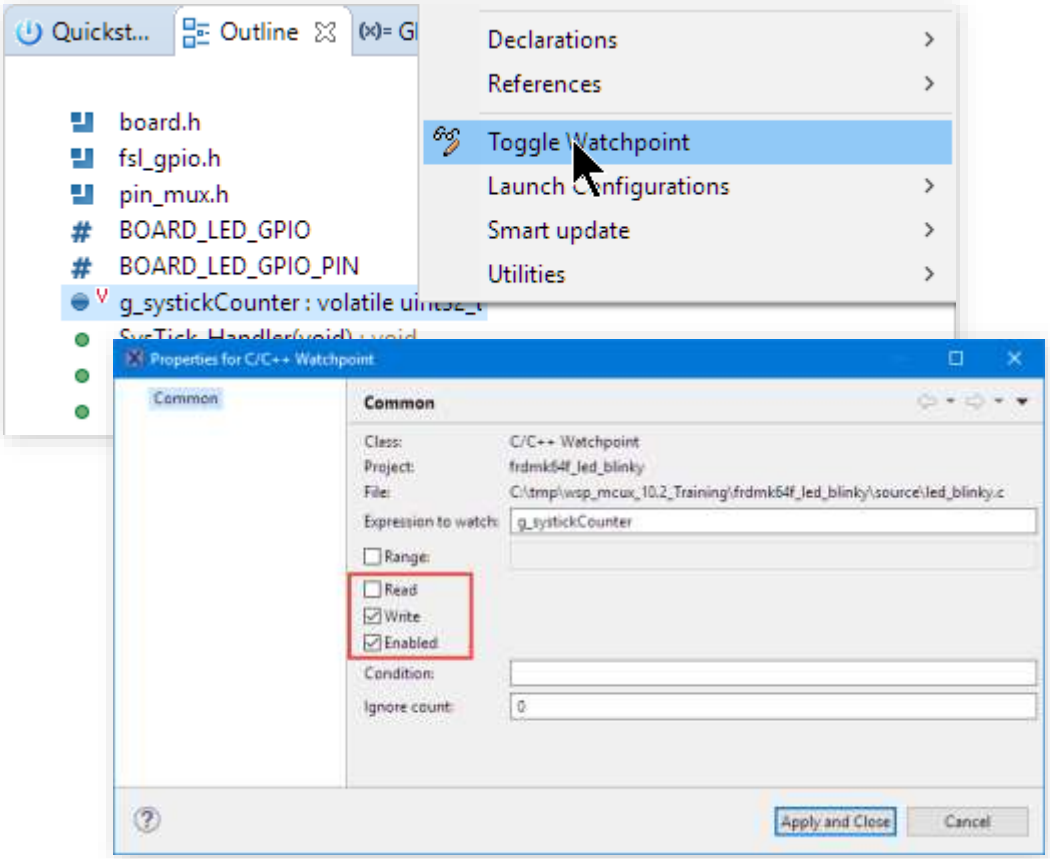
Assembly Stepping and Breakpoints

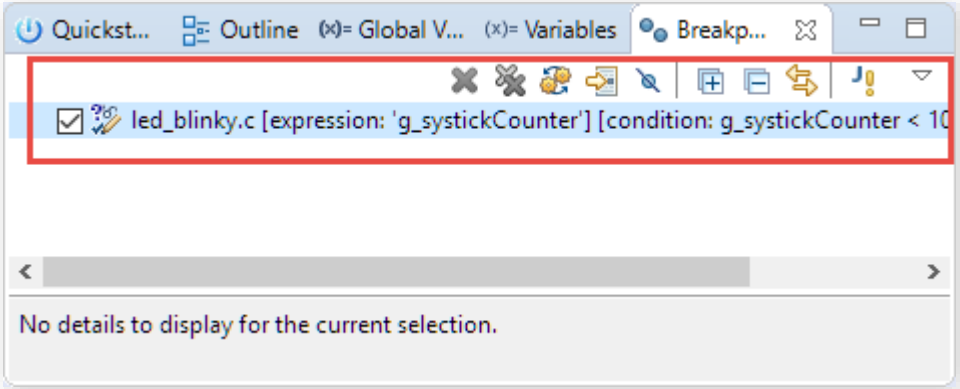
- **Turn on Instruction Stepping**
- Add/Remove **assembly instruction breakpoints**



Watchpoints

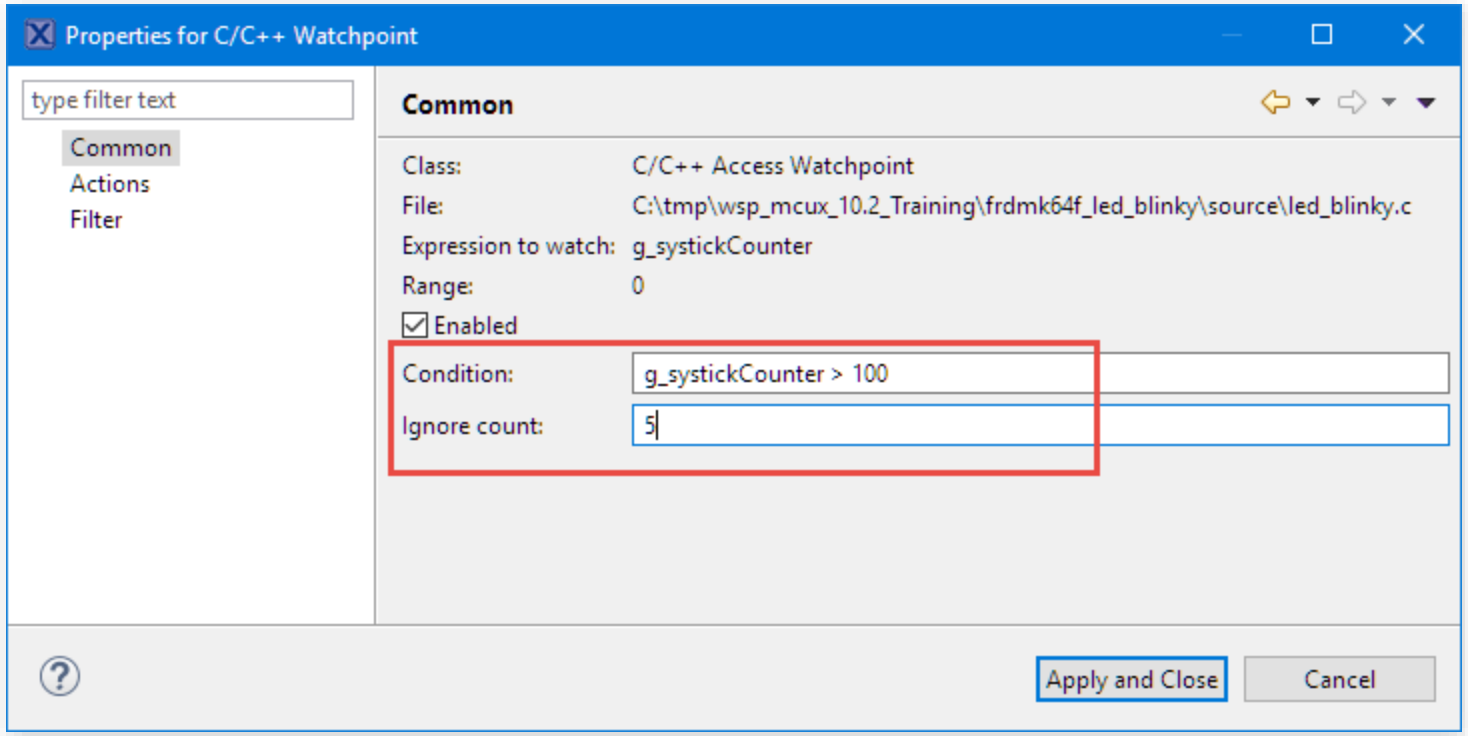
- Breakpoint on data read/write
- **Debug**
- Use **Outline** View
- **Toggle Watchpoint**
- Read/**Write** trigger condition
- Apply and Close
- **Run** Target ➔ stops on access
- Use Breakpoint View to create/update/delete watchpoints





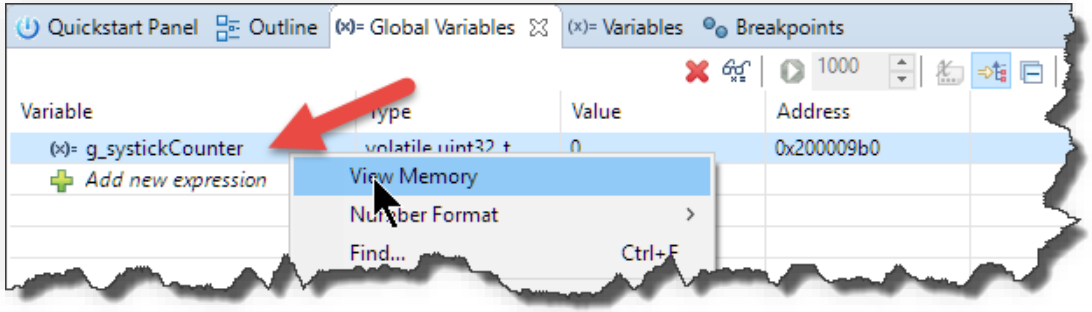
Watchpoint Properties

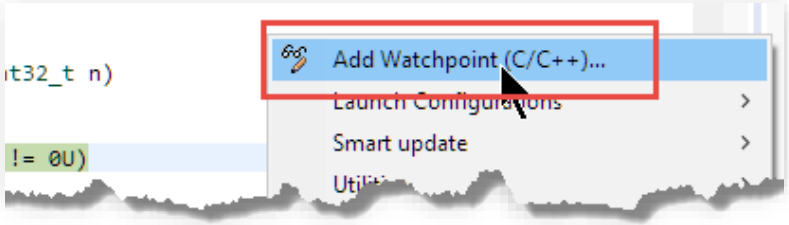
- **Read/Write** Access change:
  - Limitation of CDT: re-create watchpoint
- **Condition**: only stops on access if condition is true
- **Ignore Count**: counts down to zero until it stops
- **Actions** & **Filter**
- *Note: stops target for condition evaluation!*

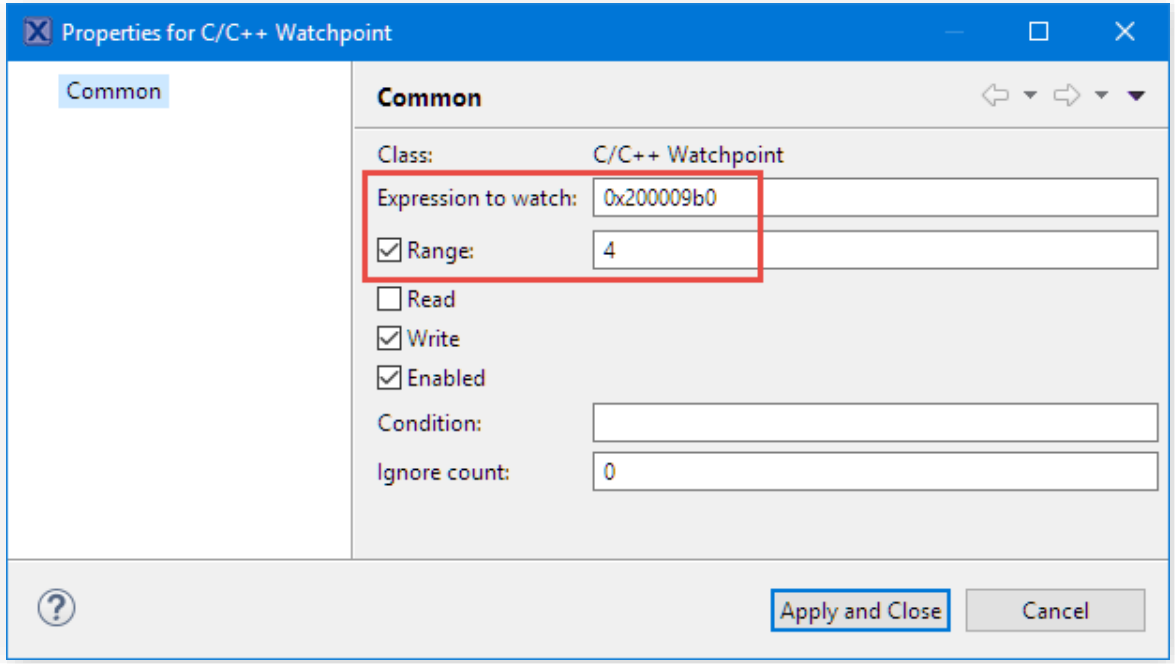


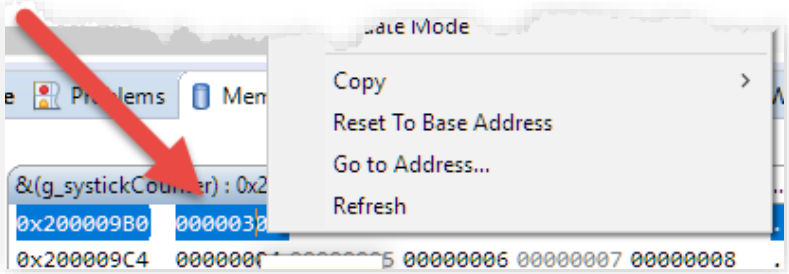
Watchpoints from Memory View

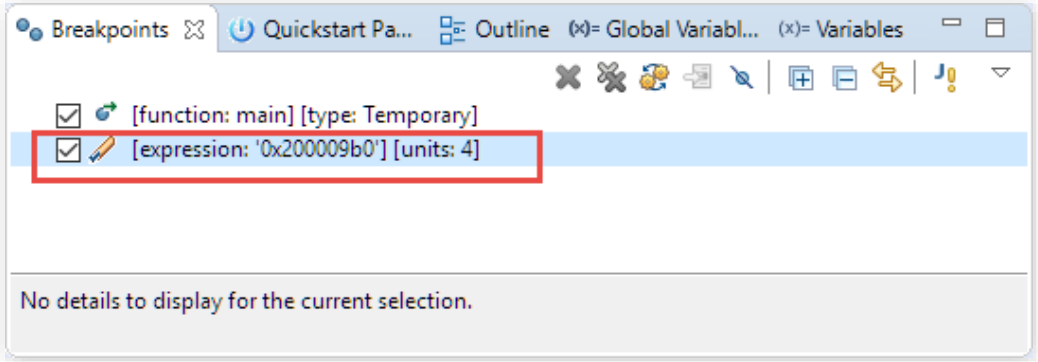
- View Memory on Global Variable
- In Memory View, select range
- Add Watchpoint (C/C++)
- Watchpoint on address range











Stepping Return Value of Functions

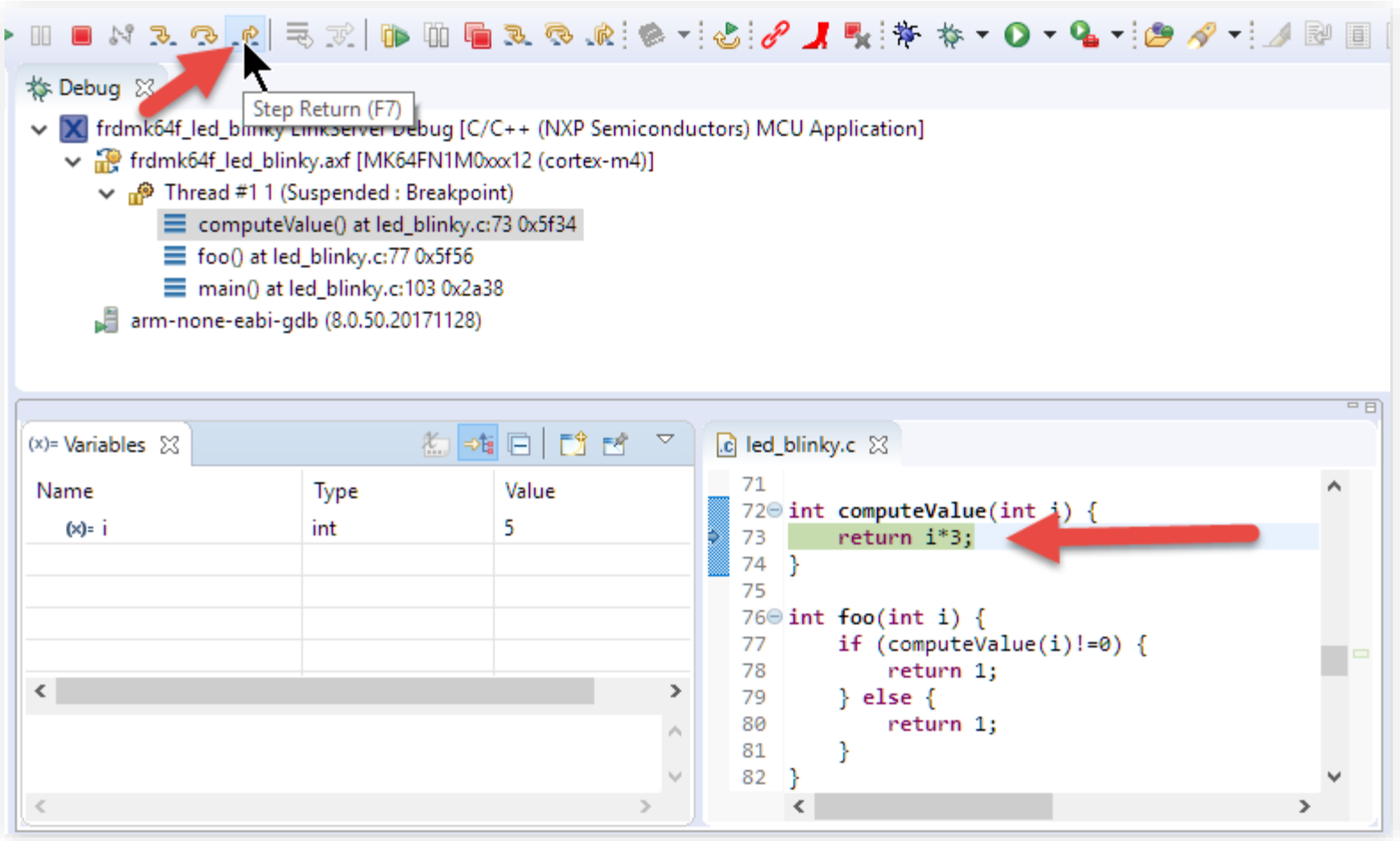
- Problem
  - How to see the calculated return value
  - Usually calculated/returned in registers
  - No local variable to inspect
- Solution
  - Step-Return Value in Variable view
- Add code to application and build (copy-paste or use snippet)

```
int computeValue(int i) {
  return i*3;
}

int foo(int i) {
  if (computeValue(i)!=0) {
    return 1;
  } else {
    return 1;
  }
}
```

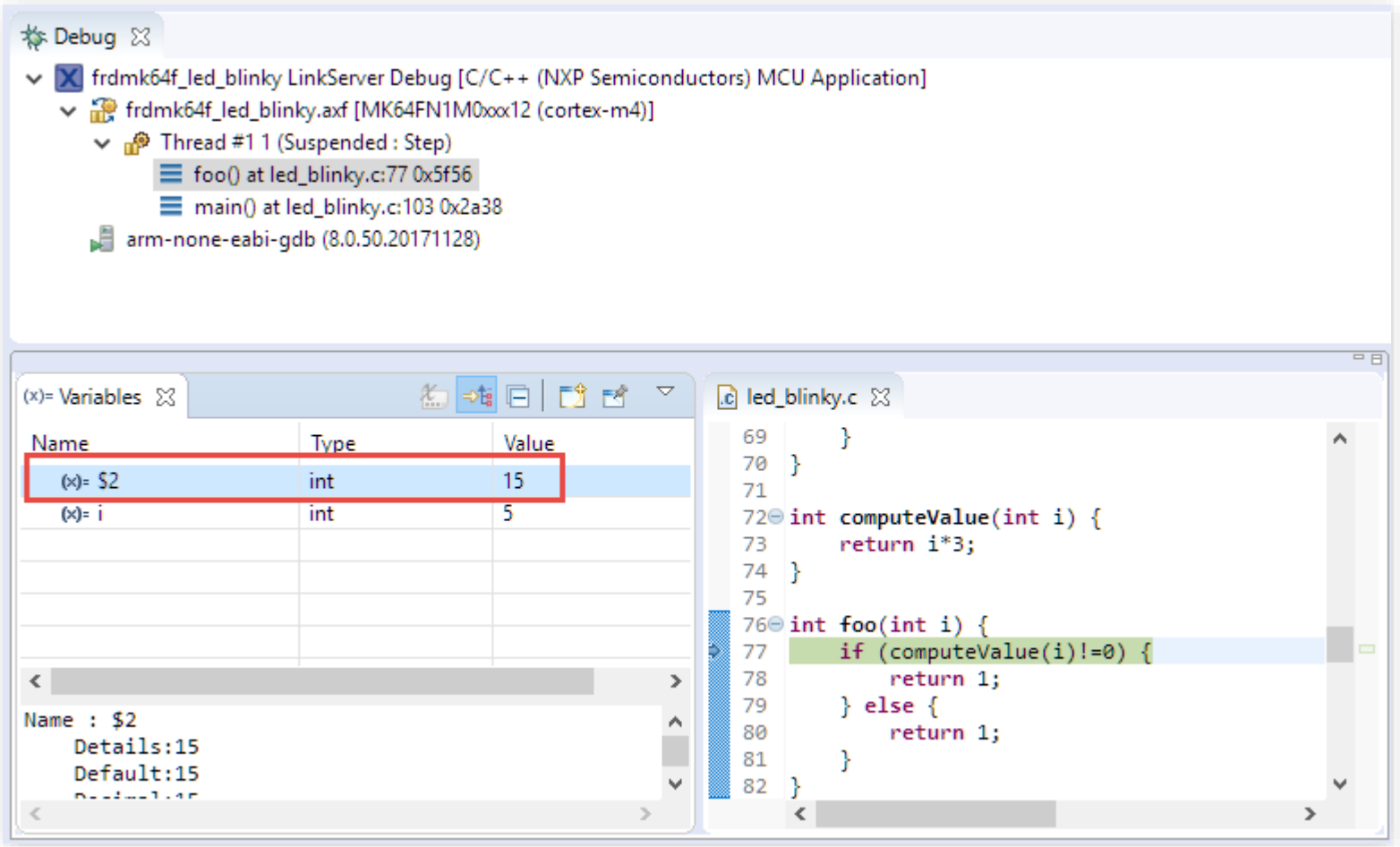
Step-Return Value

- Debug
- Perform a Step-Return from Function which returns value



Step-Return Value

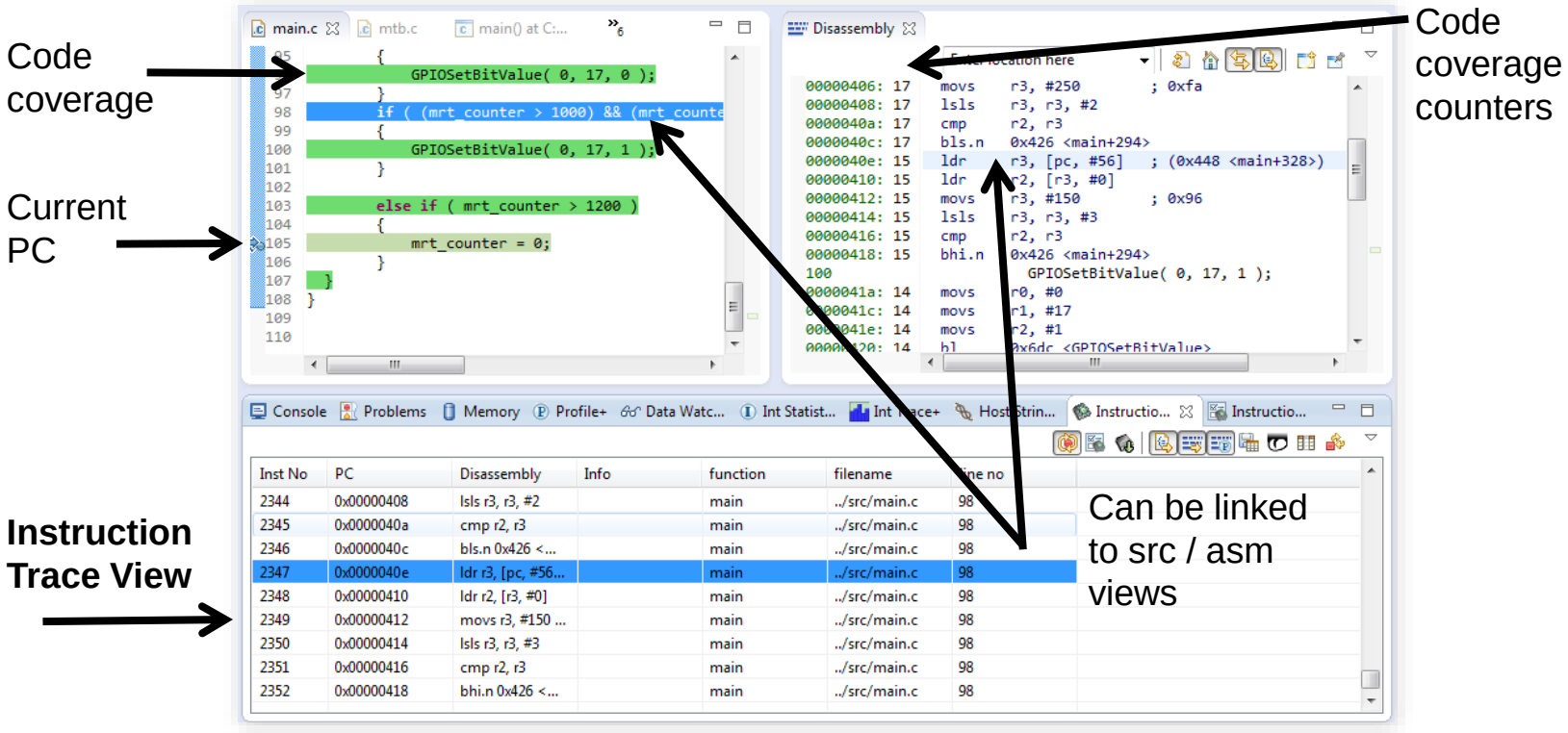
- After the Step-Return, return value is shown in Variables View



Instruction Trace

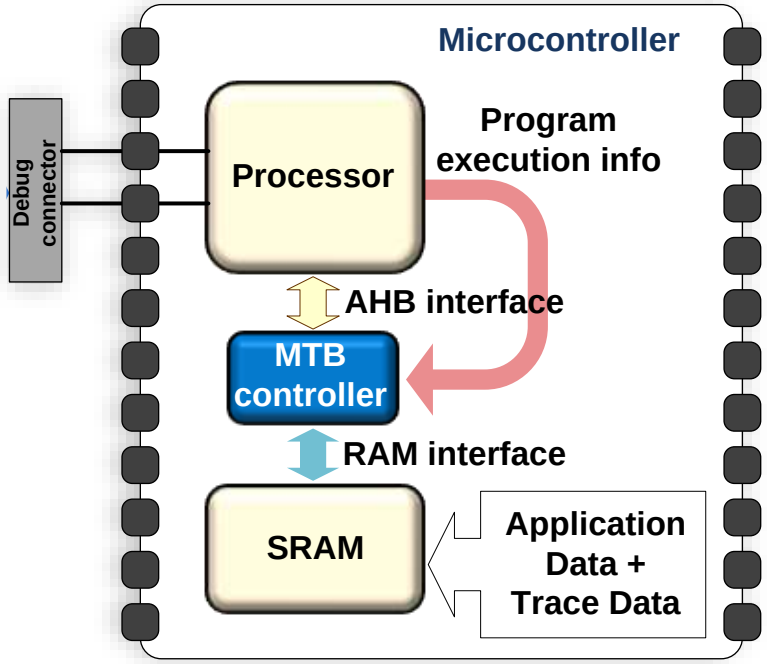
Instruction Trace

- Collects details of instructions being executed
- Allows complex program flow problems to be examined
  - Gives insight into what happening in system before a fault was encountered
- <IDE Installation Path>\MCUXpresso_IDE_Instruction_Trace.pdf



Instruction Trace – Supported Targets

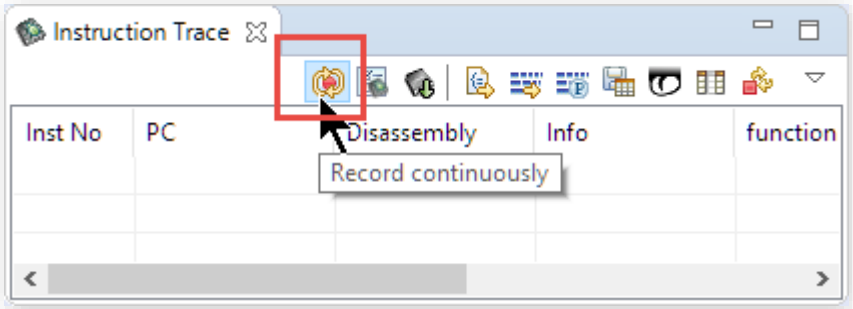
- Cortex M3/M4/M7 MCUs (Kinetis
  - Target MCU must implement both an Embedded Trace Macrocell (ETM) AND an Embedded Trace Buffer (ETB).
    - ✓Kinetis K, LPC18xx and LPC43xx parts
    - ✗LPC17xx parts
      - ✗ Does not implement an ETB
    - ✗LPC13xx parts
      - ✗ Does not implement ETM or ETB
- Cortex M0+ LPC/Kinetis MCUs
  - Target MCU must implement a Micro Trace Buffer (MTB)
    - ✓LPC8xx parts
    - ✓LPC11U6x/11E6x parts
    - ✓Kinetis L parts

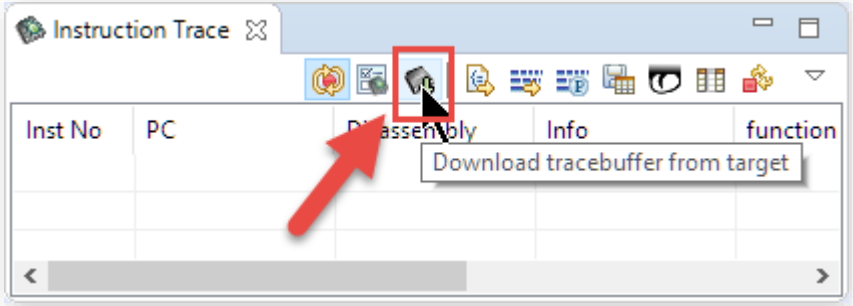


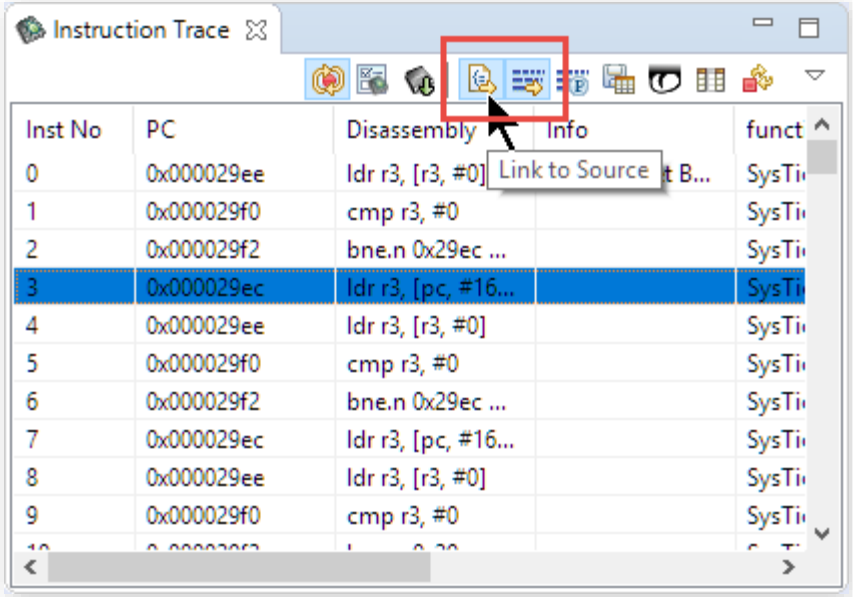
Instruction Trace in MCUXpresso can be carried out via any supported debug probe

Instruction Trace

- **Debug Application**
- Instruction Trace View
  - **Turn on 'Record continuously'** Button
- **Run** the Target and then **Suspend/Pause**
- **Download tracebuffer from target**
- Turn on **Link to Source** & **Disassembly**
- Shows corresponding program location



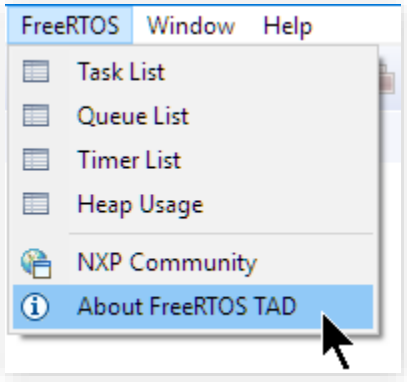


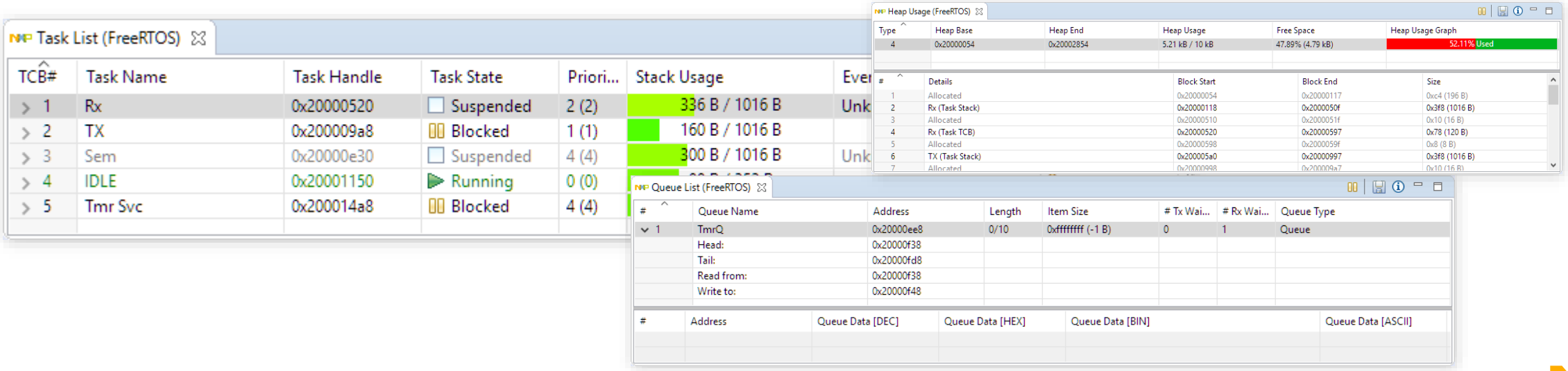


FreeRTOS
Task Aware Debug

FreeRTOS TAD

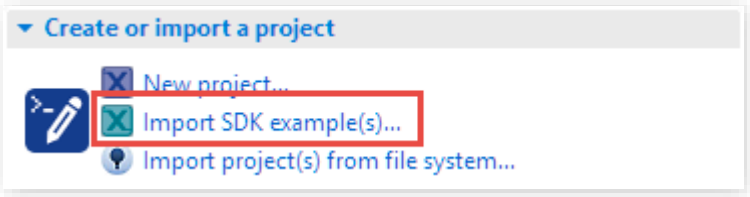
- Thread Aware Debugging
  - Show and switch between threads in Debug View
  - Views to inspect status of the RTOS
- Views read RTOS data structures while target is haltet
- Debugger needs extra information:
  <IDE Installation Path>\MCUXpresso_IDE_FreeRTOS_Debug_Guide.pdf
  - SDK projects should be updated for this

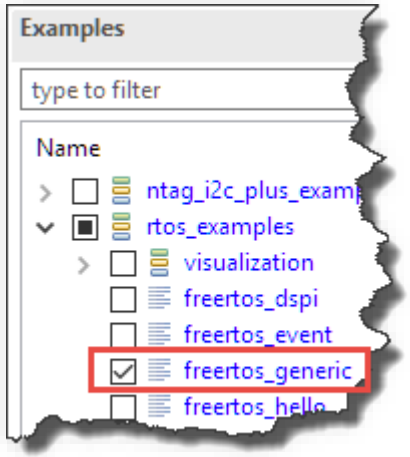




Creating FreeRTOS Project

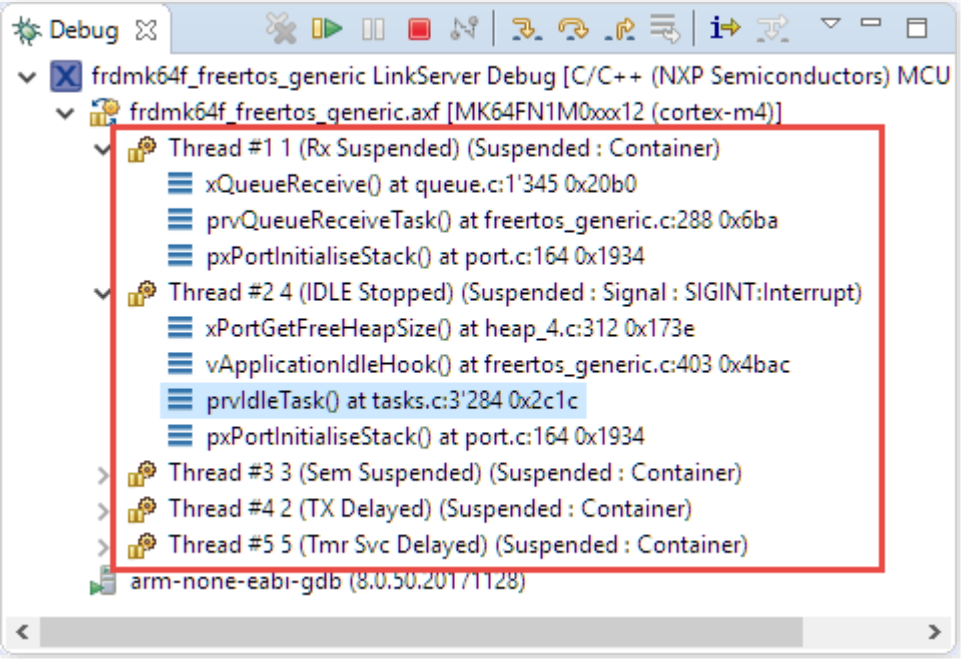
- **Quickstart Panel**
- **Import SDK example(s)**
- Select Board (**FRDM-K64F**)
- **rtos_example > freertos_generic**
- **Finish** to create project
- **Build** project

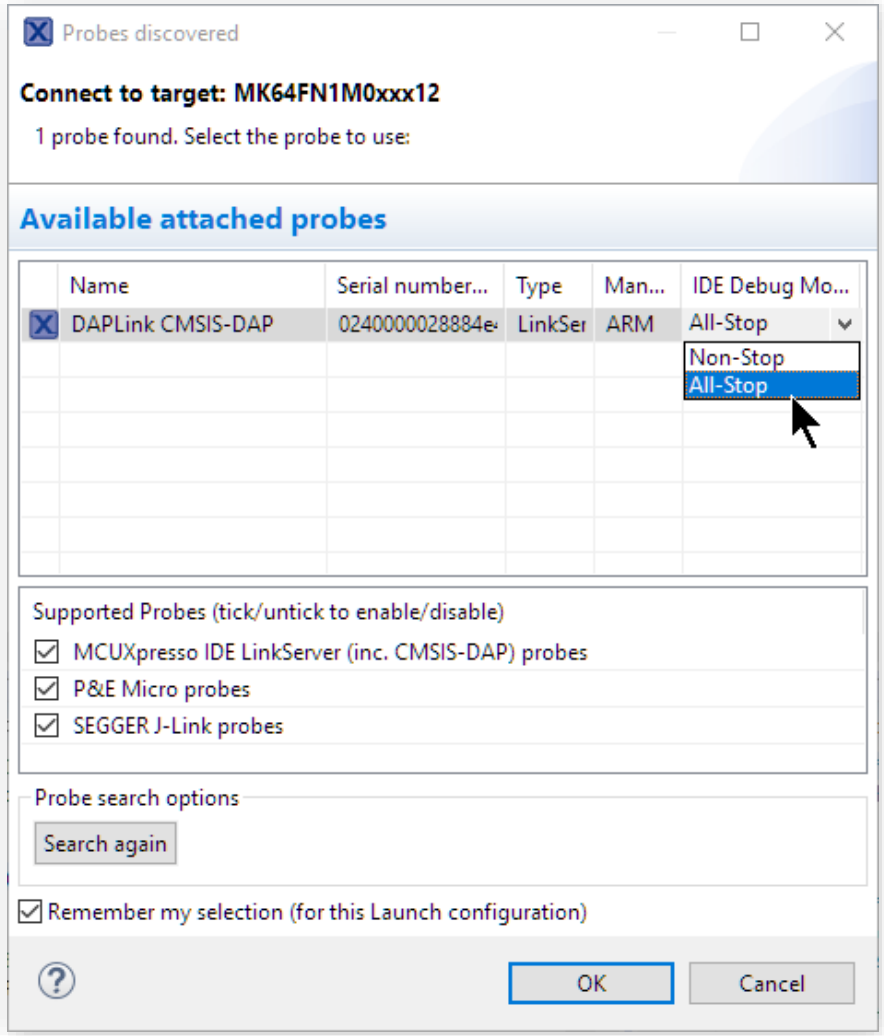




Debug Probe Connection

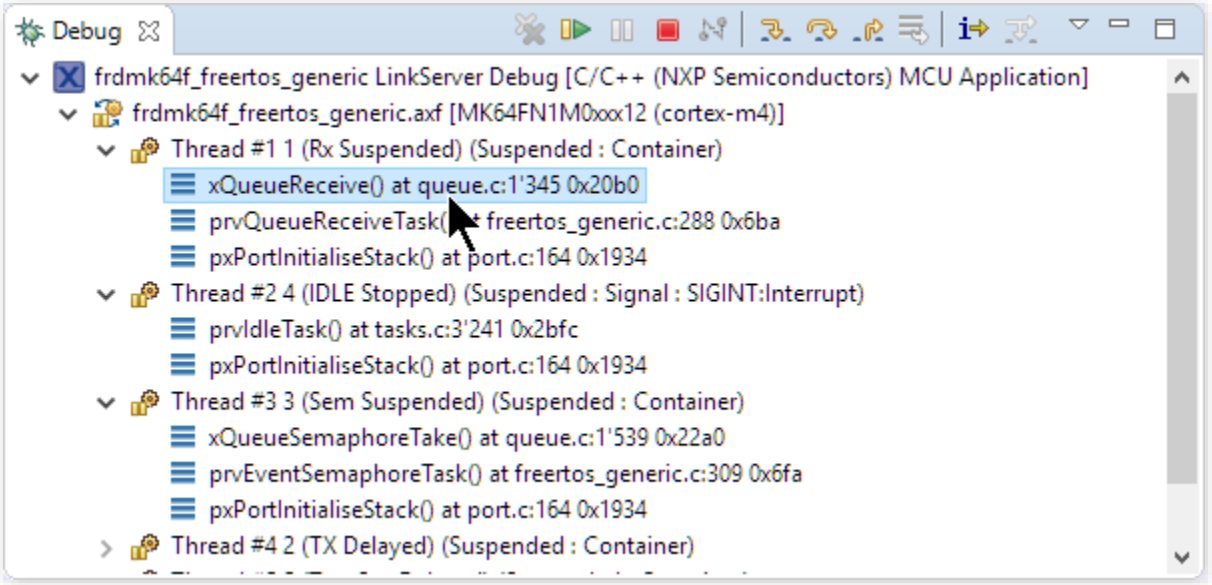
- Configure to use **All-Stop**
  - Allows thread aware debug view
  - Otherwise only current thread is shown
- **Resume**, then **Pause**





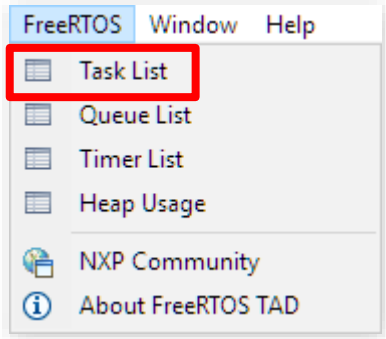
Thread Aware Debug View

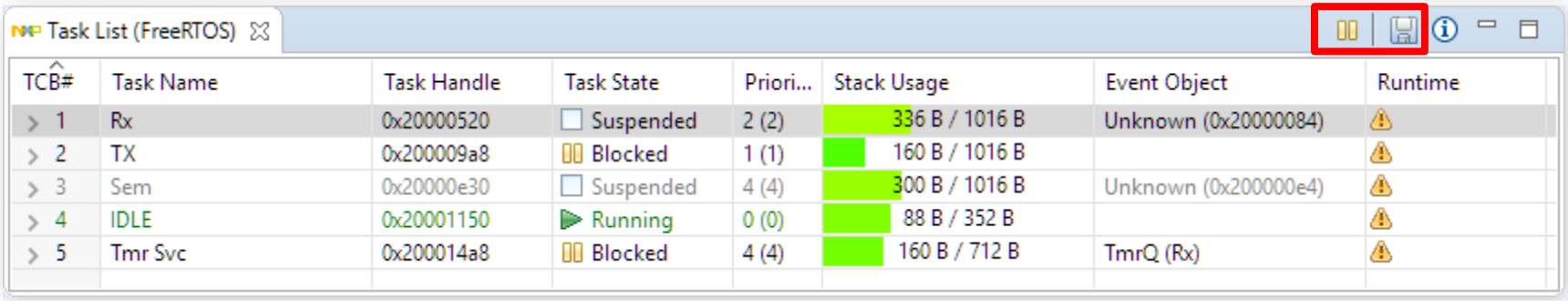
- Lists all FreeRTOS **tasks** with stacks
  - Switch debug context to thread
  - Context, registers, stack
- **Click on function** (top) of a thread
- Debug it (**step out, step over**)
- **Switch** to another thread



Task List

- Lists FreeRTOS Tasks in the System
- Menu **FreeRTOS > Task List**
- **Pause** Button: do not update data when target is stopped
- **Save** Button: Store information as .csv file

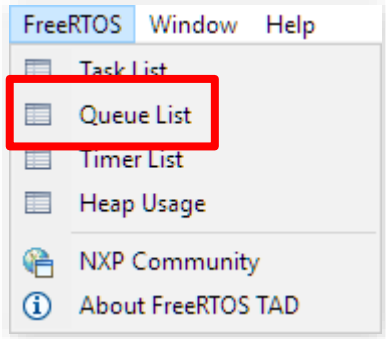


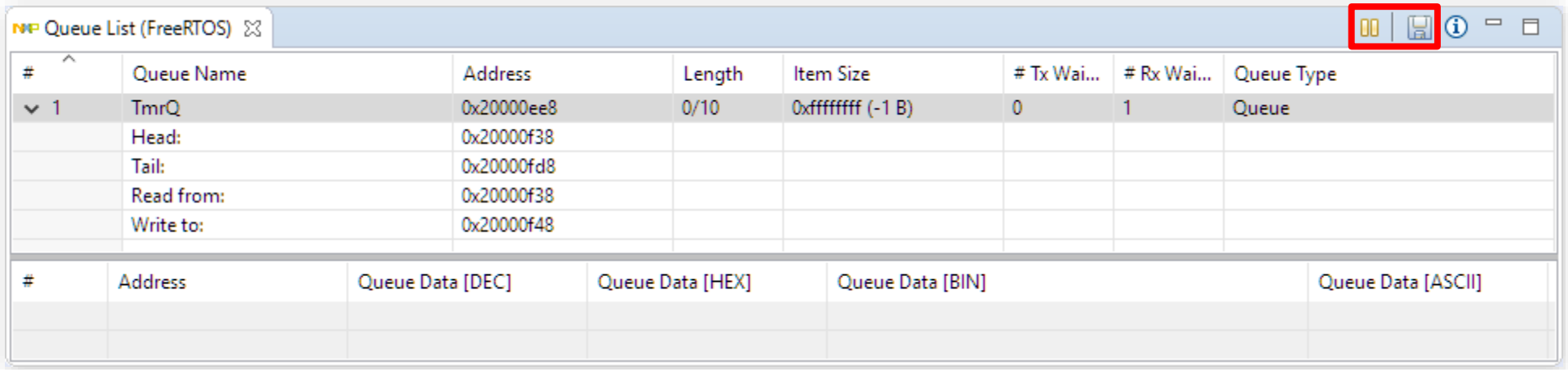


TCB#	Task Name	Task Handle	Task State	Priori...	Stack Usage	Event Object	Runtime
1	Rx	0x20000520	Suspended	2 (2)	336 B / 1016 B	Unknown (0x20000084)	
2	TX	0x200009a8	Blocked	1 (1)	160 B / 1016 B		
3	Sem	0x20000e30	Suspended	4 (4)	300 B / 1016 B	Unknown (0x200000e4)	
4	IDLE	0x20001150	Running	0 (0)	88 B / 352 B		
5	Tmr Svc	0x200014a8	Blocked	4 (4)	160 B / 712 B	TmrQ (Rx)	

Queue List

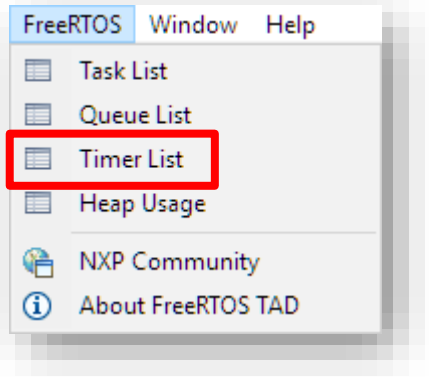
- Lists all FreeRTOS **Queues**
- Menu **FreeRTOS > Queue List**
- **Pause** Button: do not update data when target is stopped
- **Save** Button: Store information as .csv file

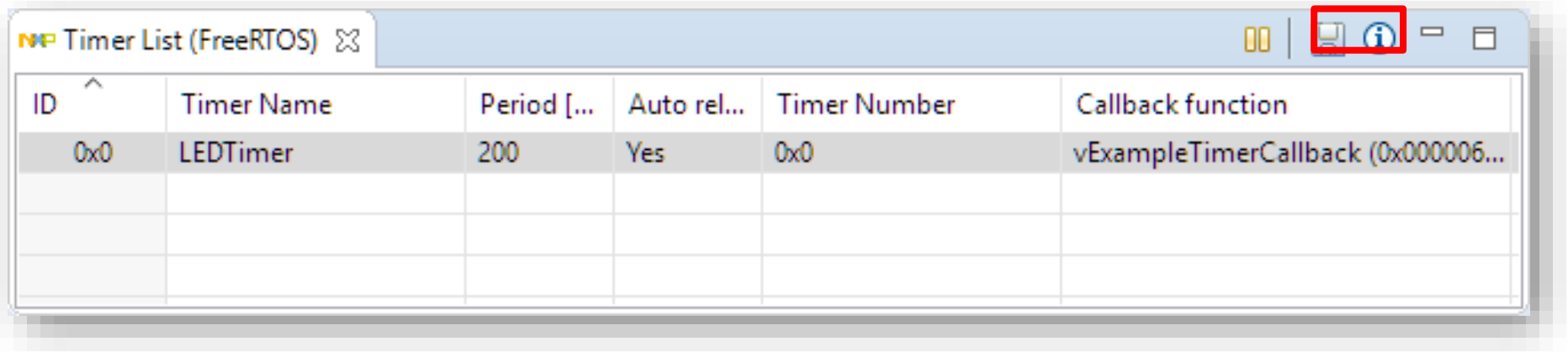




Queue Timer

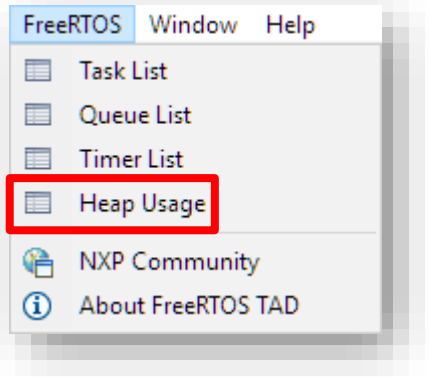
- Lists all FreeRTOS **Software Timer**
- Menu **FreeRTOS > Queue List**
- **Pause** Button: do not update data when target is stopped
- **Save** Button: Store information as .csv file

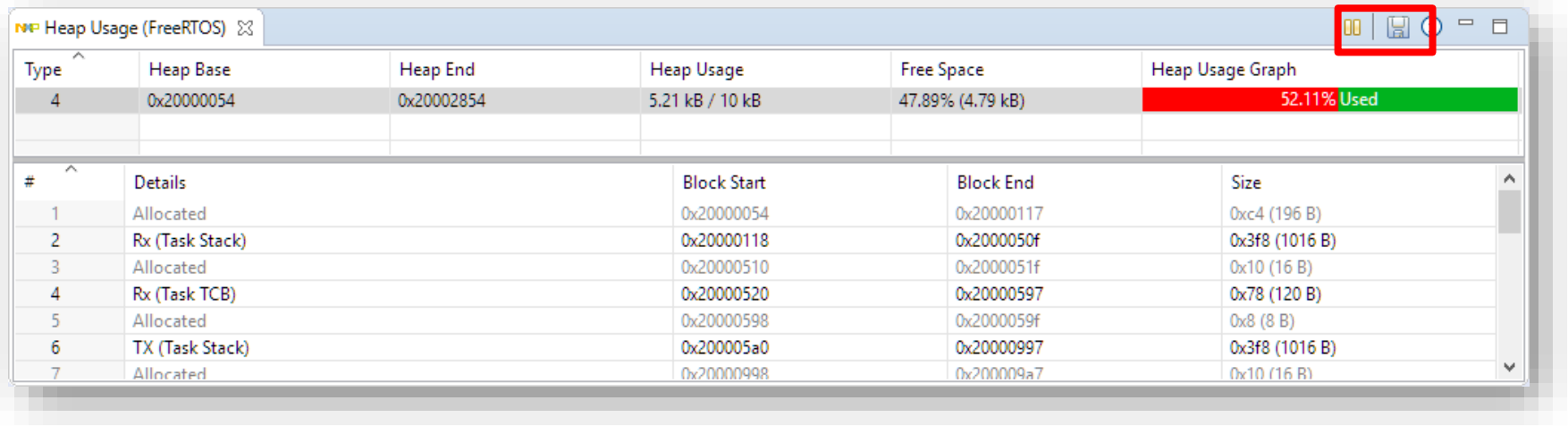




Heap Usage

- Status of FreeRTOS **Heap** and **Memory Allocation**
- Menu **FreeRTOS > Queue List**
- **Pause** Button: do not update data when target is stopped
- **Save** Button: Store information as .csv file





Summary

Summary

- Overview: MCUXpresso
  - IDE, Configuration Tools, SDK
- Building
  - Wizard, create & import projects
- Variables and Data
  - Global Variables, Expressions, Registers, Peripheral+
- Run Control
  - Stepping, Breakpoints, Watchpoints, Step-Return
- Instruction Trace
- FreeRTOS Thread Aware Debugging (TAD)
  - Tasks, Queues, Timers, Heap
- *Backup Material: SWO Debug Features and Trace*

MCUXpresso Software and Tools Additional Resources

Web pages

- MCUXpresso Software and Tools – www.nxp.com/mcuxpresso

- MCUXpresso IDE – www.nxp.com/mcuxpresso/ide
- MCUXpresso SDK – www.nxp.com/mcuxpresso/sdk
- MCUXpresso Config Tools – www.nxp.com/mcuxpresso/config

Supported Devices: Supported Devices Table (Community Doc)

Communities

- MCUXpresso Software and Tools - https://community.nxp.com/community/mcuxpresso

- MCUXpresso SDK- https://community.nxp.com/community/mcuxpresso/mcuxpresso-sdk
- MCUXpresso IDE- https://community.nxp.com/community/mcuxpresso/mcuxpresso-ide
- MCUXpresso Config Tools -https://community.nxp.com/community/mcuxpresso/mcuxpresso-config

SECURE CONNECTIONS
FOR A SMARTER WORLD

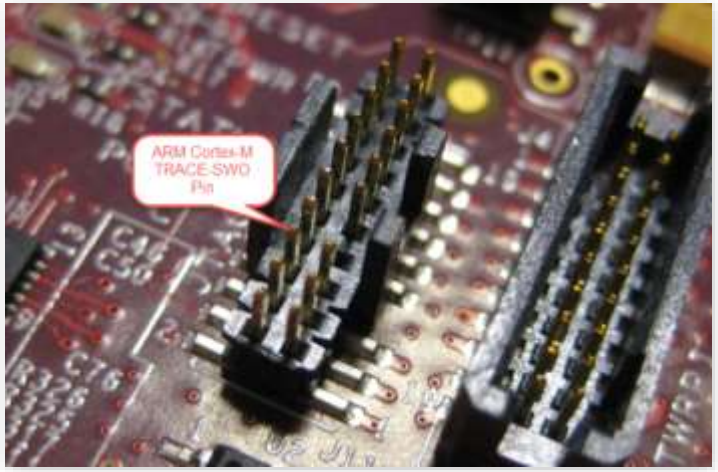
www.nxp.com

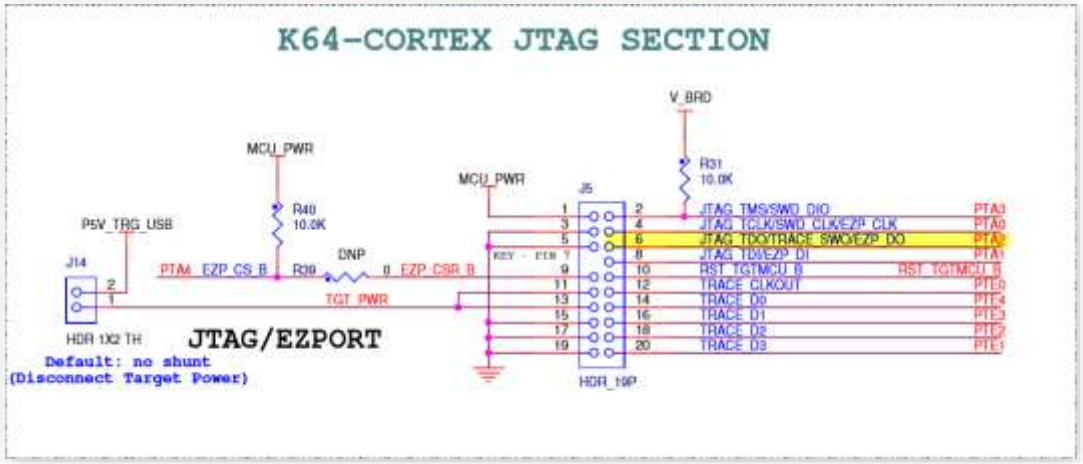
NXP, the NXP logo, and NXP secure connections for a smarter world are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2018 NXP B.V.

Backup:
SWO TRACE

Single Wire Output

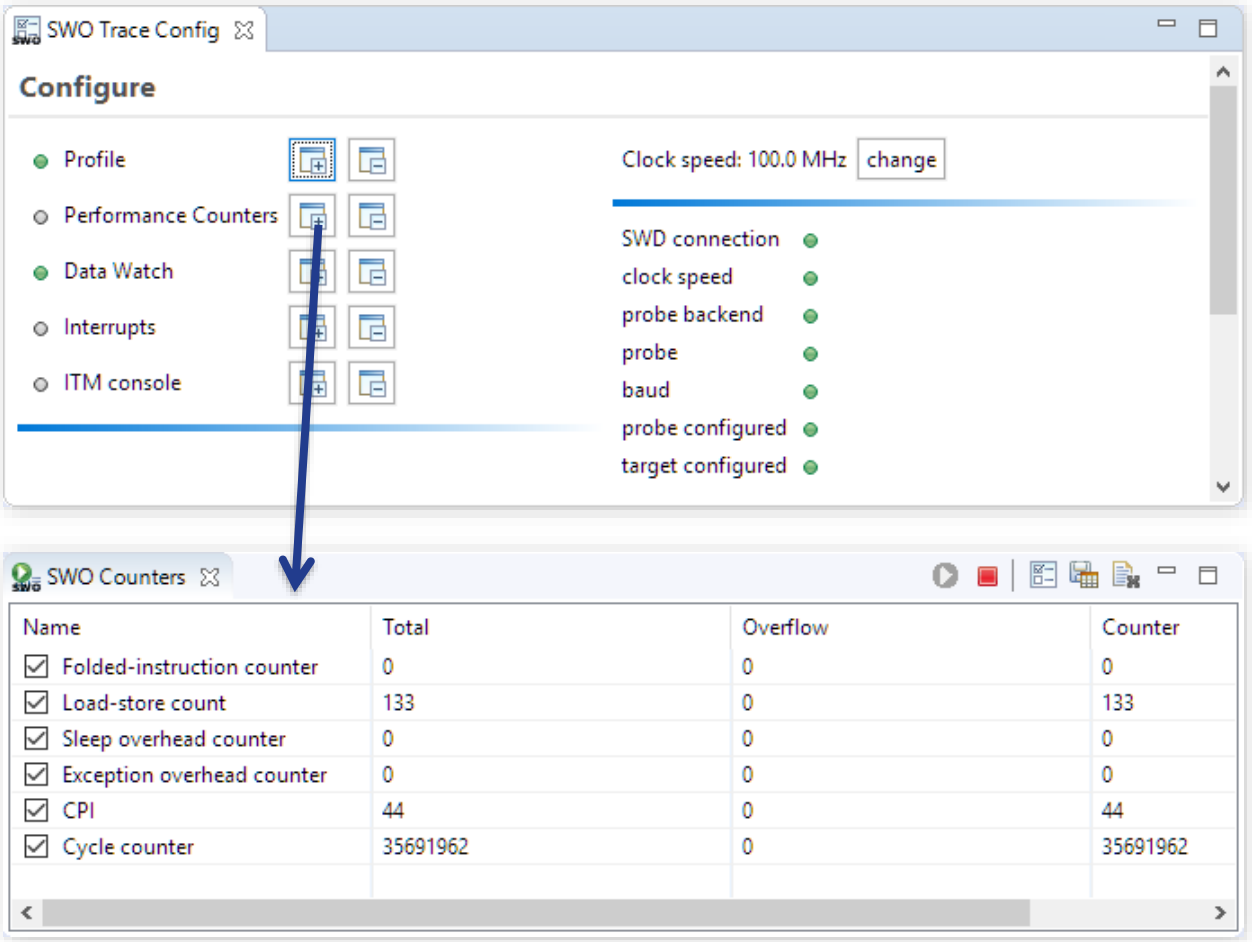
- SWO: Single Wire Output
- ARM Cortex-M Hardware Feature (1 pin)
- Requires that the SWO pin is routed to debug headers
  - Present on Tower (TWR) and LPCXpresso boards
  - Freedom (FRDM) board do not have SWO pin available
- Debug Features with SWO
  - SWO Console
  - SWO Interrupt tracing/logging
  - SWO Profiling
  - via LPC-Link2 with NXP CMSIS-DAP firmware
- <IDE Installation Path>\MCUXpresso_IDE_SWO_Trace.pdf





SWO Trace Config

- Dedicated view to configure SWO
- SWO Clock speed setting
- (+) opens other views



SWO Trace – Profile View

- Displays statistical profile of application activity
  - Based on PC sampling, typically at ~50kHz
  - Non-intrusive – does not affect application
- Benefits – Identify hotspots

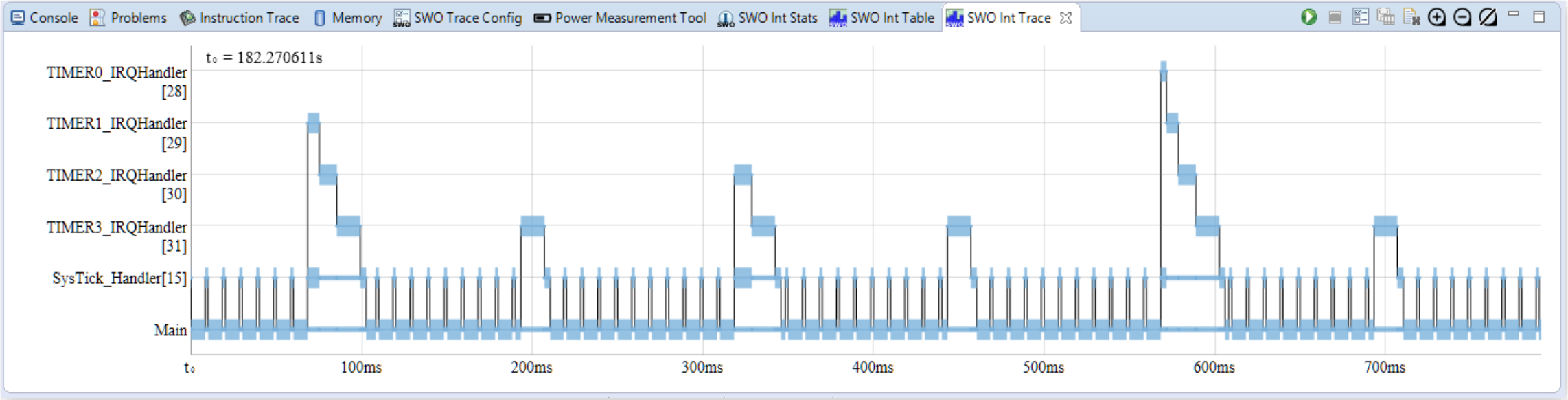
Installed S... | Properties | Problems | Progress | Memory | Debugger ... | Power Me... | SWO Trac... | Search | SWO Profile

Function	Cumulative samples	Cumulative samples %	Current samples %	Coverage %	First	Last	Since	Avg Between	Start	End	Co
prvCheckTasksWaitingTerr	680829	71.9%	71.9%	26.3%	4.8...	4.5...	0.00...	6.679us	0x3...	0x3...	11
prvTimerTask	2	0.0%	0.0%	16.7%	1.7...	3.4...	1.052s	883.538ms	0x3...	0x3...	00
__vfprintf	8	0.0%	0.0%	0.8%	553...	4.0...	472....	440.216ms	0x4...	0x5...	00
SysTick_Handler	163	0.0%	0.0%	20.0%	1.1...	3.4...	1.052s	14.378ms	0x1...	0x1...	01
__aeabi_lowlevel_memset	1	0.0%	0.0%	20.0%	2.9...	2.9...	1.644s	0.000s	0x5...	0x5...	00
vListInsert	2	0.0%	0.0%	3.5%	1.1...	2.9...	1.644s	878.621ms	0x2...	0x2...	00
xTaskResumeAll	4	0.0%	0.0%	2.9%	1.1...	4.0...	472....	732.347ms	0x2...	0x2...	00
prvProcessExpiredTimer	2	0.0%	0.0%	3.2%	1.1...	2.9...	1.644s	878.621ms	0x3...	0x3...	00
vTaskSwitchContext	1	0.0%	0.0%	1.3%	1.7...	1.7...	2.819s	0.000s	0x3...	0x3...	00
xTaskIncrementTick	71	0.0%	0.0%	3.3%	1.1...	4.5...	7.07...	47.808ms	0x2f...	0x3...	10
vPortExitCritical	5	0.0%	0.0%	10.5%	553...	4.0...	472....	704.348ms	0x1...	0x1...	00
vQueueWaitForMessageRe	1	0.0%	0.0%	1.9%	553...	553...	3.994s	0.000s	0x2...	0x2...	00
prvUnlockQueue	1	0.0%	0.0%	1.2%	4.0...	4.0...	472....	0.000s	0x2...	0x2...	00
prvIdleTask	266191	28.1%	28.1%	30.0%	8.3...	4.0...	466....	15.300us	0x3...	0x3...	00
PendSV_Handler	4	0.0%	0.0%	12.5%	1.7...	4.0...	472....	586.733ms	0x1...	0x1...	00
_printf	1	0.0%	0.0%	5.0%	1.1...	1.1...	3.401s	0.000s	0x5...	0x5...	00
vPortEnterCritical	5	0.0%	0.0%	11.9%	553...	4.0...	472....	704.342ms	0x1...	0x1...	00
prvInsertTimerInActiveList	1	0.0%	0.0%	1.6%	1.7...	1.7...	2.819s	0.000s	0x3...	0x3...	00
prvProcessTimerOrBlockTa	2	0.0%	0.0%	5.6%	553...	3.4...	1.052s	1.471s	0x3...	0x3...	00
xQueueReceive	3	0.0%	0.0%	1.8%	553...	1.1...	3.401s	197.437ms	0x2...	0x2...	10

SWO Trace – Interrupt Views

SWO Int Table

Index	ID	Event	Handler	Time	Ticks
3098	35	EXIT	I2C1_IRQHand...	16.048s	3273746073
3097	35	ENTRY	I2C1_IRQHand...	16.048s	3273745862
3096	0	RETURN		16.048s±3.294us	3273740833
3095	-3	OVERFLOW	SWO Overflow	16.048s±3.294us	3273740833
3094	0	RETURN		16.048s±3.294us	3273740833
3093	15	EXIT	SysTick_Handler	16.048s	3273740833
3092	15	ENTRY	SysTick_Handler	16.048s	3273740817
3091	31	EXIT	TIMER3_IRQH...	16.048s	3273740810
3090	31	ENTRY	TIMER3_IRQH...	16.034s	3270940679



SWO Trace – Interrupt Views

- Interrupts Stats
  - Continuous count (and other stats) of all interrupts
- Benefits
  - Determine time spent in interrupt handlers
  - Optimization of interrupt handlers

SWO Int Stats

Int handler	Count	Time in	Min(time in)	Max(time in)	Min(time between)	Max(time bet...	Total time between	Time since last	First time run	Current Time
Main	793	23.506s...	945.790us	15.747s	1.270us	7.140±2.790us	11.088ms±22.680us	1.270us	10.000ms	23.517s
CT32B0_IRQHandler[27]	8	22.850us	2.840us	2.920us	999.994ms	999.994ms	23.517s	999.994ms	16.297s	23.517s
CT32B1_IRQHandler[28]	16	46.830us	2.920us	2.930us	502.438ms	502.438ms	23.517s	502.438ms	15.809s	23.517s
SysTick_Handler[15]	778	933.52...	1.190us	1.200us	9.998ms	15.747s	23.516s	9.999ms	9.999ms	23.517s

Note: this functionality is a Technology Preview

SWO Interrupt Stats

- Capturing Interrupt Trace also populates an SWO Int Stats View
- **Switch to the SWO Int Stats View and compare the Stats with our measured values**
  - Note that for linear functions the Min (time in) and Max (time in) will be the same
    - this is not true for time spent in Main (which is the interrupted code)
  - Note the Min (time between) and Max (time between) is not guaranteed to be same
    - This relates to interrupt default priorities and pre-emption

Console | Problems | Memory | Instruction Trace | SWO Trace Config | Power Measurement Tool | SWO Int Stats

Int handler	Count	Time in	Min(time in)	Max(time in)	Min(time between)	Max(time between)	Total time be
SysTick_Handler[15]	619	1.062s	1.716ms	1.716ms	63.725ns	42.574ms	5.800s
TIMER2_IRQHandler[30]	27	277.959ms	10.295ms	10.295ms	229.427ms	249.983ms	6.584s
TIMER3_IRQHandler[31]	54	741.216ms	13.726ms	13.726ms	90.709ms	131.838ms	6.121s
TIMER0_IRQHandler[28]	6	20.593ms	3.432ms	3.432ms	996.566ms	996.566ms	6.841s
TIMER1_IRQHandler[29]	13	89.224ms	6.863ms	6.863ms	489.712ms	496.559ms	6.773s
Main	591	4.661s	3.283ms	8.284ms	1.716ms	36.033ms	2.201s

SWO Trace – DataWatch

Dynamic memory accesses

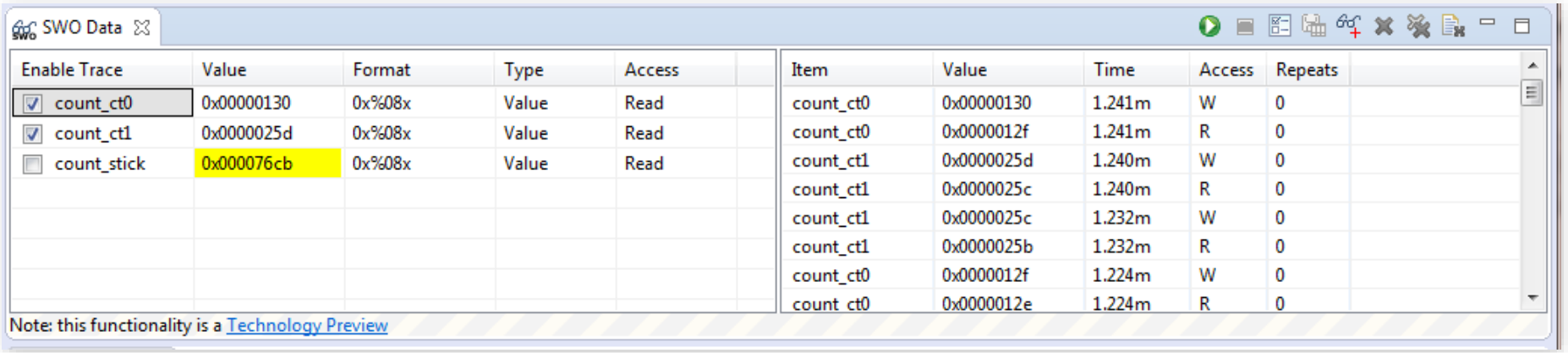
- Read and write to target memory without stopping CPU
- Non-intrusive
- Reads done on periodic basis (by default)
- Unlimited number of addresses
- Allows modifications to parameters in real time

Datawatch Trace

- Capture all accesses to memory location, without stopping CPU

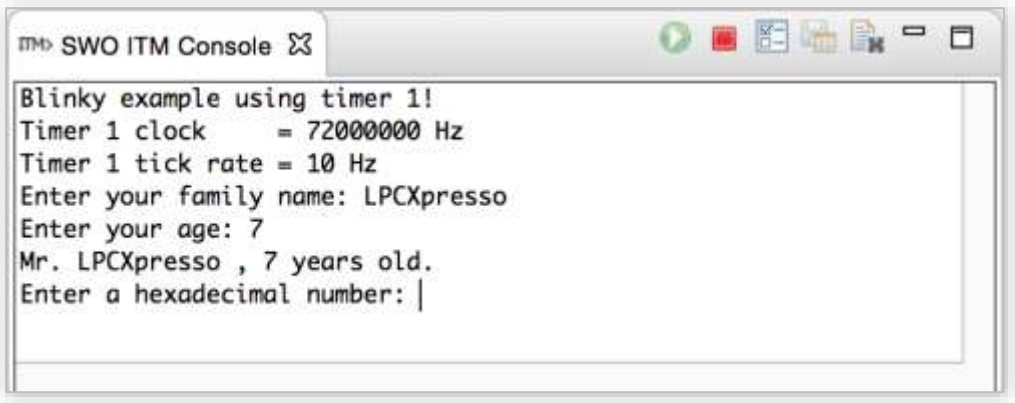
Benefits

- Monitor and analyse memory accesses
- Identify ‘rogue’ memory accesses



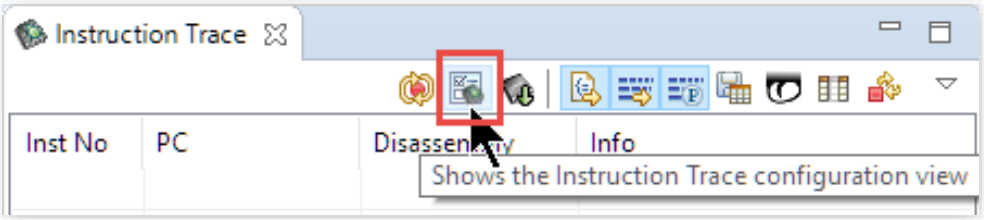
SWO Trace – ITM Printf

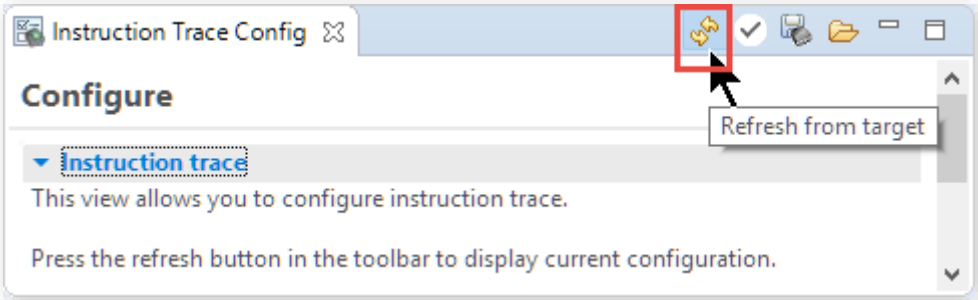
- Instrumentation Trace Macrocell (ITM) block provides a mechanism for sending data from your target to the debugger via the SWO trace stream
- MCUXpresso IDE allows user to redirect printf/scanf data by reimplementing low level Redlib function __sys_write / __sys_readc
  - Newlib reimplementation also possible
- Unlike normal semihosting, this scheme is both low bandwidth and does not halt the MCU to transfer data

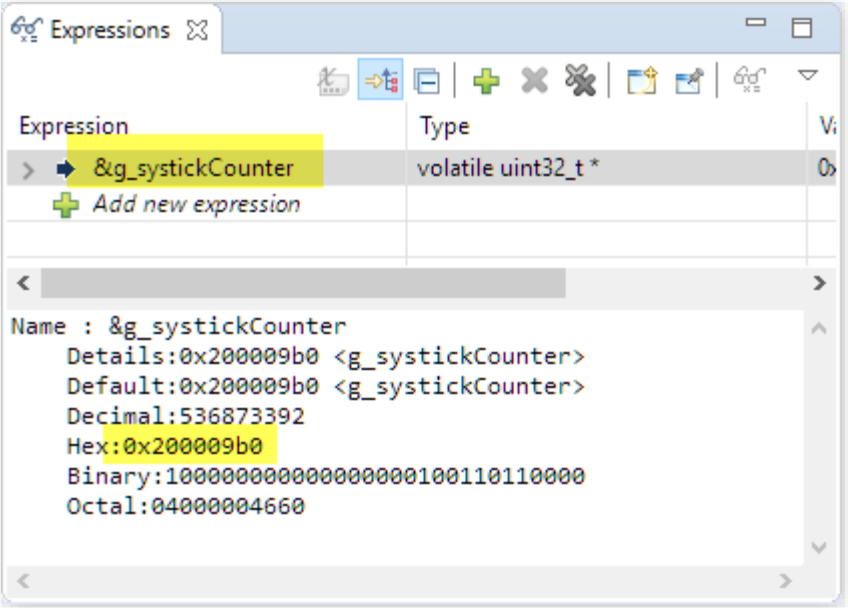


Instruction Trace Triggers

- **Debug and Suspend**
- **Show Instruction Trace Configuration**
- **Refresh from target**
- Address of global variable
  - **Expressions View**
  - **Add Expression**
  - **Address: &<variableName>**
  - **Copy-Paste address**

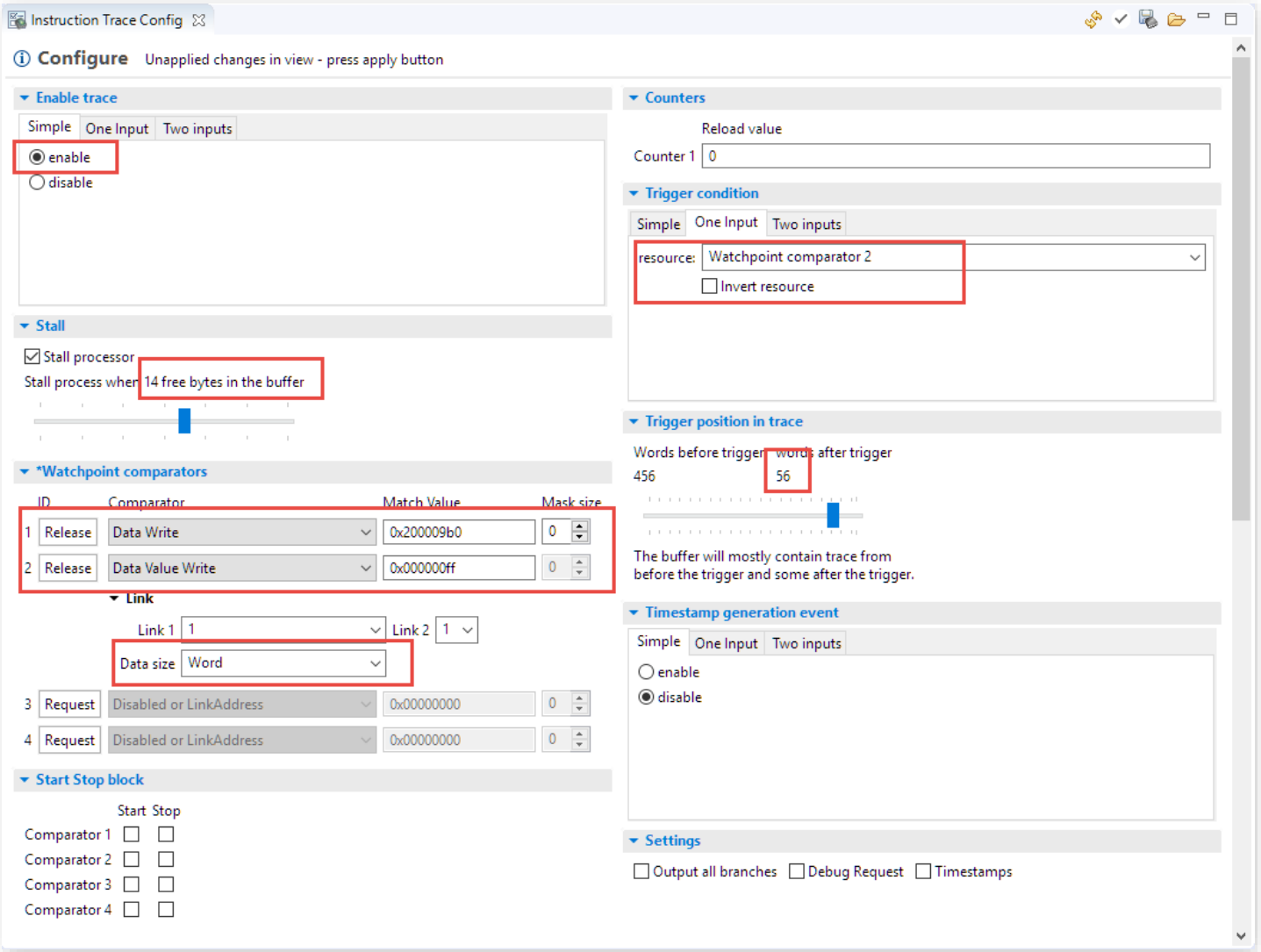






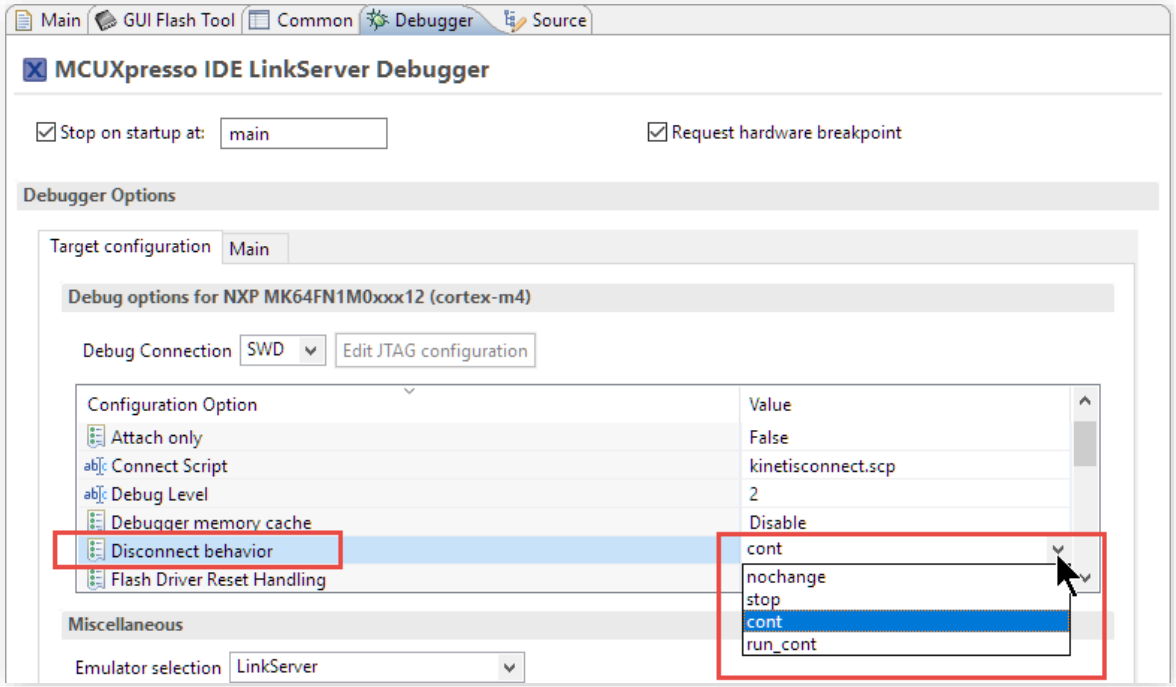
Instruction Trace Trigger Configuration

- Trace enabled
- Stall if only some free bytes in buffer
- Data write on address and value written
- Trigger on comparator 2
- Records some words before and after trigger
- Save Configuration



Disconnect

- **‘Terminate’/Stop** button disconnects from Target
- **Open Debug Configuration**
  - **Double Click on *.launch file in Project**
  - **nochange** - will leave the target in its current state
  - **stop** - will leave the target in debug state i.e. halted
  - **cont** - the default, will either start the image from its current PC value or leave it running
  - **run cont** - will reset the target and let it run
- **Change it to ‘stop’ and try it**



SECURE CONNECTIONS
FOR A SMARTER WORLD

www.nxp.com

NXP, the NXP logo, and NXP secure connections for a smarter world are trademarks of NXP B.V. All other product or service names are the property of their respective owners. © 2018 NXP B.V.