



# FRDM-KL25Z and tracing

# Overview

- Development platform for Kinetis® L Series KL2x microcontrollers (MCUs) built on Arm® Cortex®-M0+ processor
  - MKL25Z128VLK4 MCU – 48 MHz, 128 KB flash, 16 KB SRAM, USB OTG, 80LQFP
- standard form factor with expansion board options
- built-in debug interface for flash programming and run-time control



# Top view

KL25Z 80 LQFP

© 2013 FREESCALE

Capacitive Touch Slider

J1 I/O Header

J10 I/O Header

J2 I/O Header

RGB LED

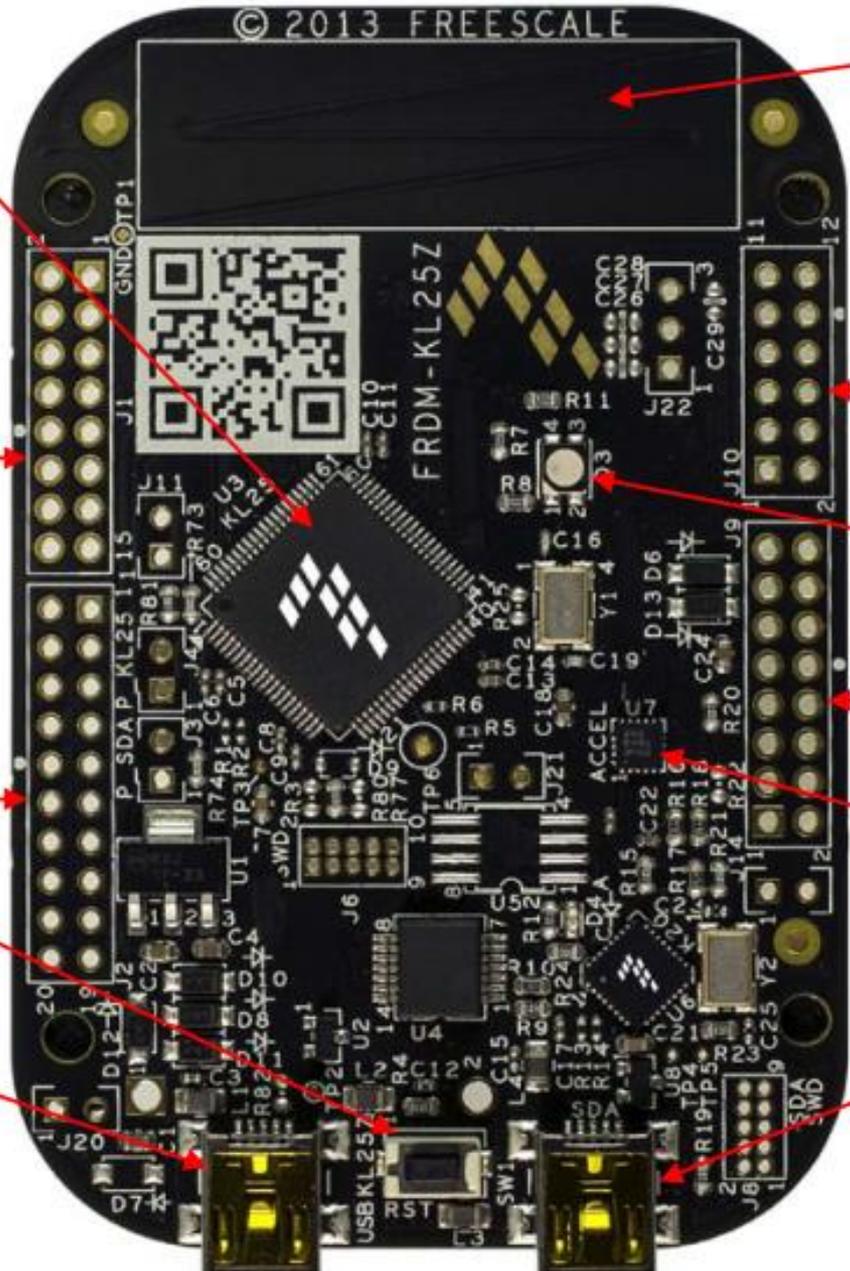
Reset

J9 I/O Header

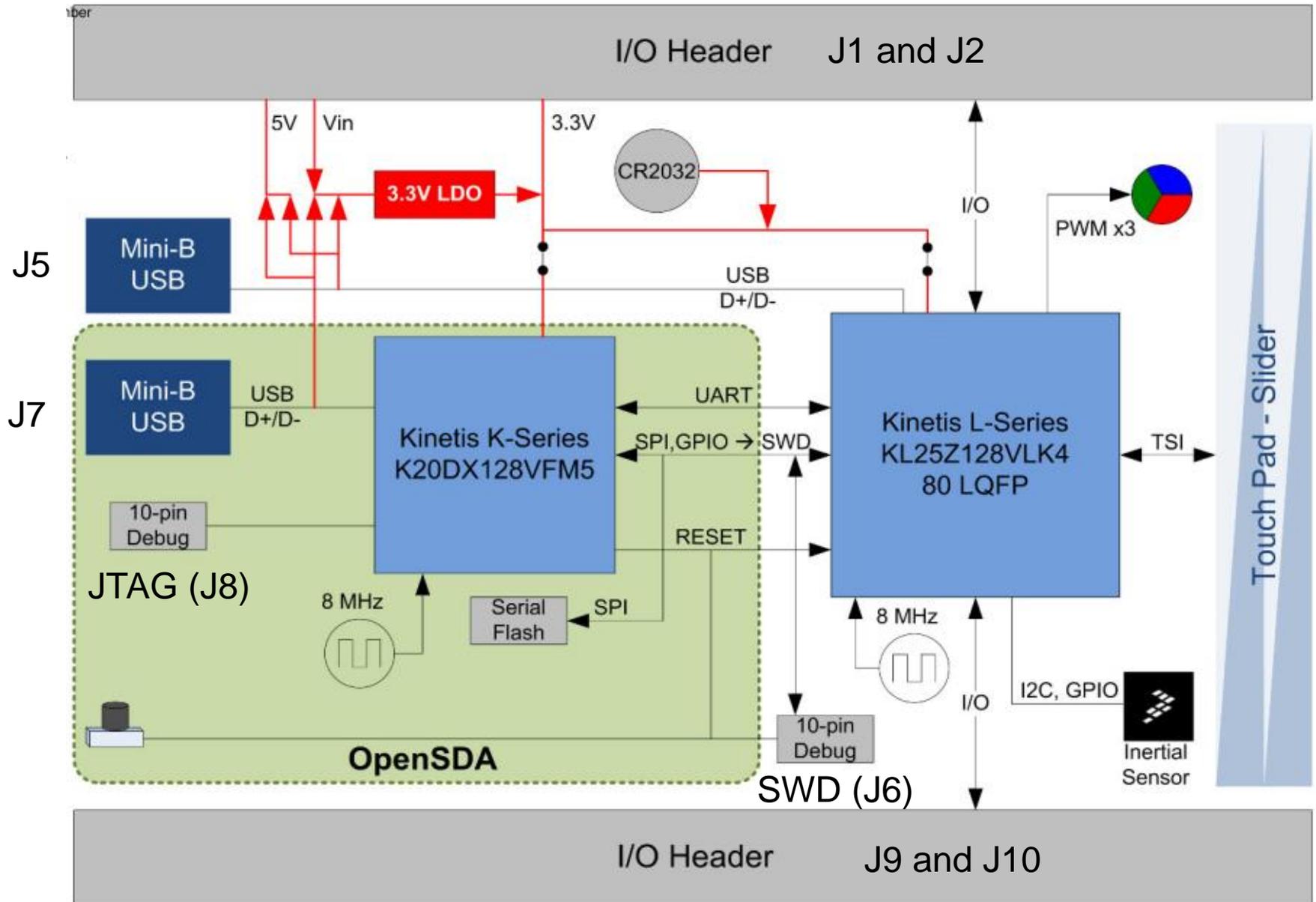
KL25Z USB

MMA8451Q

OpenSDA

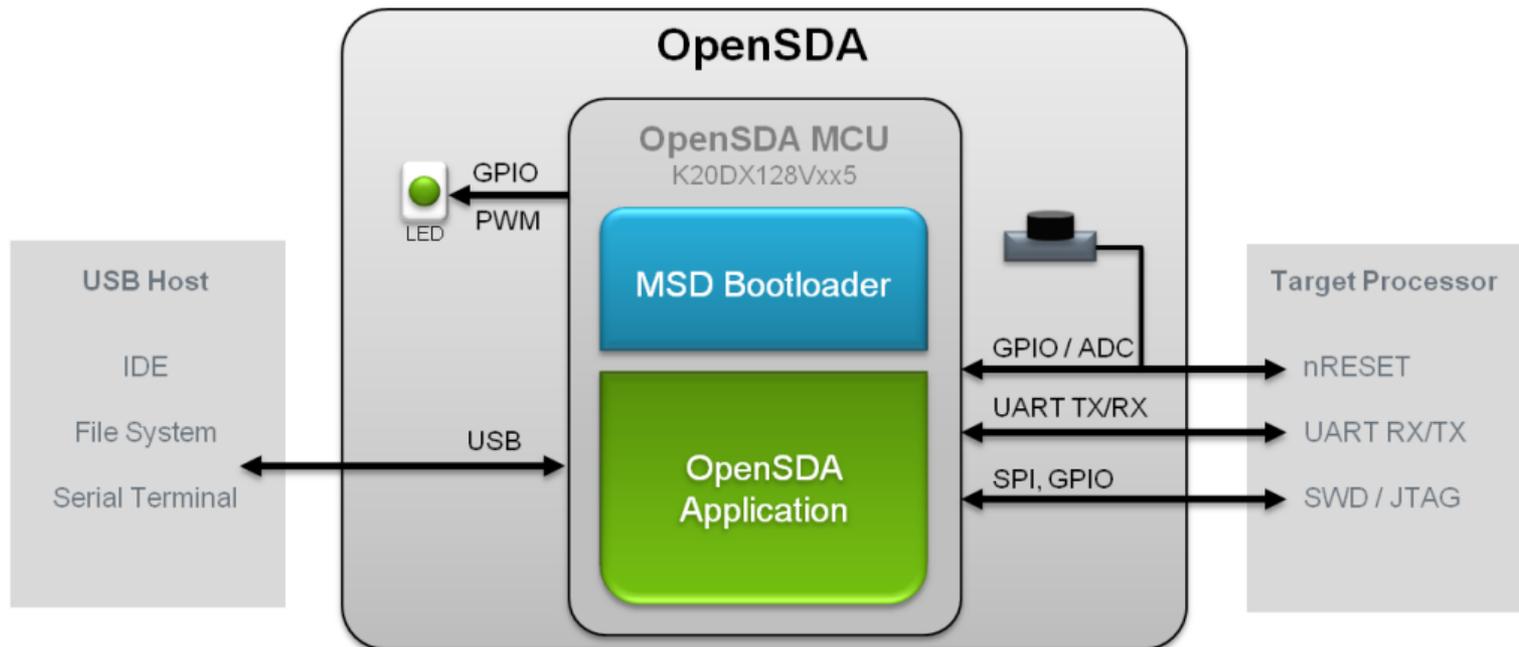


# Block diagram



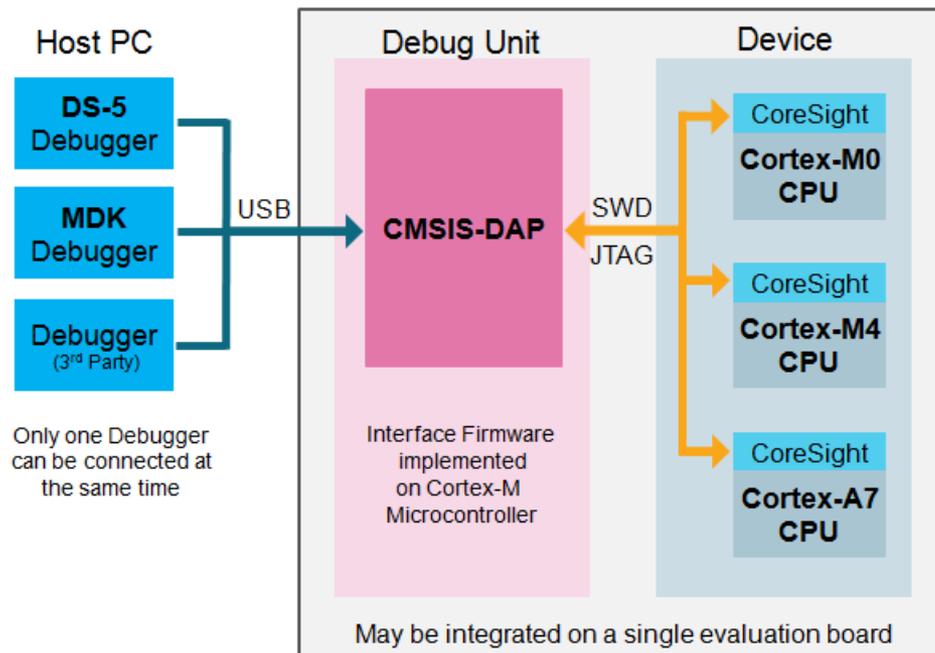
## Serial and Debug Adapter (OpenSDA)

- Based on open-sourced CMSIS-DAP bootloader and debug interface
- serial and debug communications between a USB host and an embedded target processor
- based on a Freescale Kinetis K20 family microcontroller with 128 KB of embedded flash and integrated USB controller
- features a mass storage device (MSD) bootloader



# CMSIS-DAP debugger

- CMSIS-DAP is the interface firmware for a Debug Unit that connects the Debug Port to USB.
- <http://www.keil.com/support/man/docs/dapdebug/>



# I/O

- USB Interface
- serial Port
  - SPI (2)
  - I<sup>2</sup>C (2)
  - UART (3)
- ADC (16 bit)
- DAC (1x12 bit)
- touch sensor
- GPIO (66)
- 3-axis Accelerometer (Freescale MMA8451Q low-power, three-axis accelerometer interfaced through an I<sup>2</sup>C bus)
- 3 PWM-capable signals connected to an RGB LED
- pins on the KL25Z microcontroller are named for their general purpose input/output port pin function
  - e.g., 1st pin on Port A is referred to as PTA1

# I/O

(usb) uart0\_RX  
(usb) uart0\_TX

PTA1 - 2  
PTA2 - 4  
PTD4 - 6  
PTA12 - 8  
PTA4 - 10  
PTA5 - 12  
PTC8 - 14  
PTC9 - 16



1 - PTC7  
3 - PTC0  
5 - PTC3  
7 - PTC4  
9 - PTC5  
11 - PTC6  
13 - PTC10  
15 - PTC11

PTE30 - 11  
PTE29 - 9  
PTE23 - 7  
PTE22 - 5  
PTE21 - 3  
PTE20 - 1



12 - PTC1  
10 - PTC2  
8 - PTB3  
6 - PTB2  
4 - PTB1  
2 - PTB0

(spi CS)  
spi MOSI  
spi MISO  
spi SCK

PTA13 - 2  
PTD5 - 4  
PTD0 - 6  
PTD2 - 8  
PTD3 - 10  
PTD1 - 12  
GND - 14  
VREFH - 16



1 - PTC12  
3 - PTC13  
5 - PTC16  
7 - PTC17  
9 - PTA16  
11 - PTA17  
13 - PTE31  
15 - NC

PTE5 - 15  
PTE4 - 13  
PTE3 - 11  
PTE2 - 9  
PTB11 - 7  
PTB10 - 5  
PTB9 - 3  
PTB8 - 1



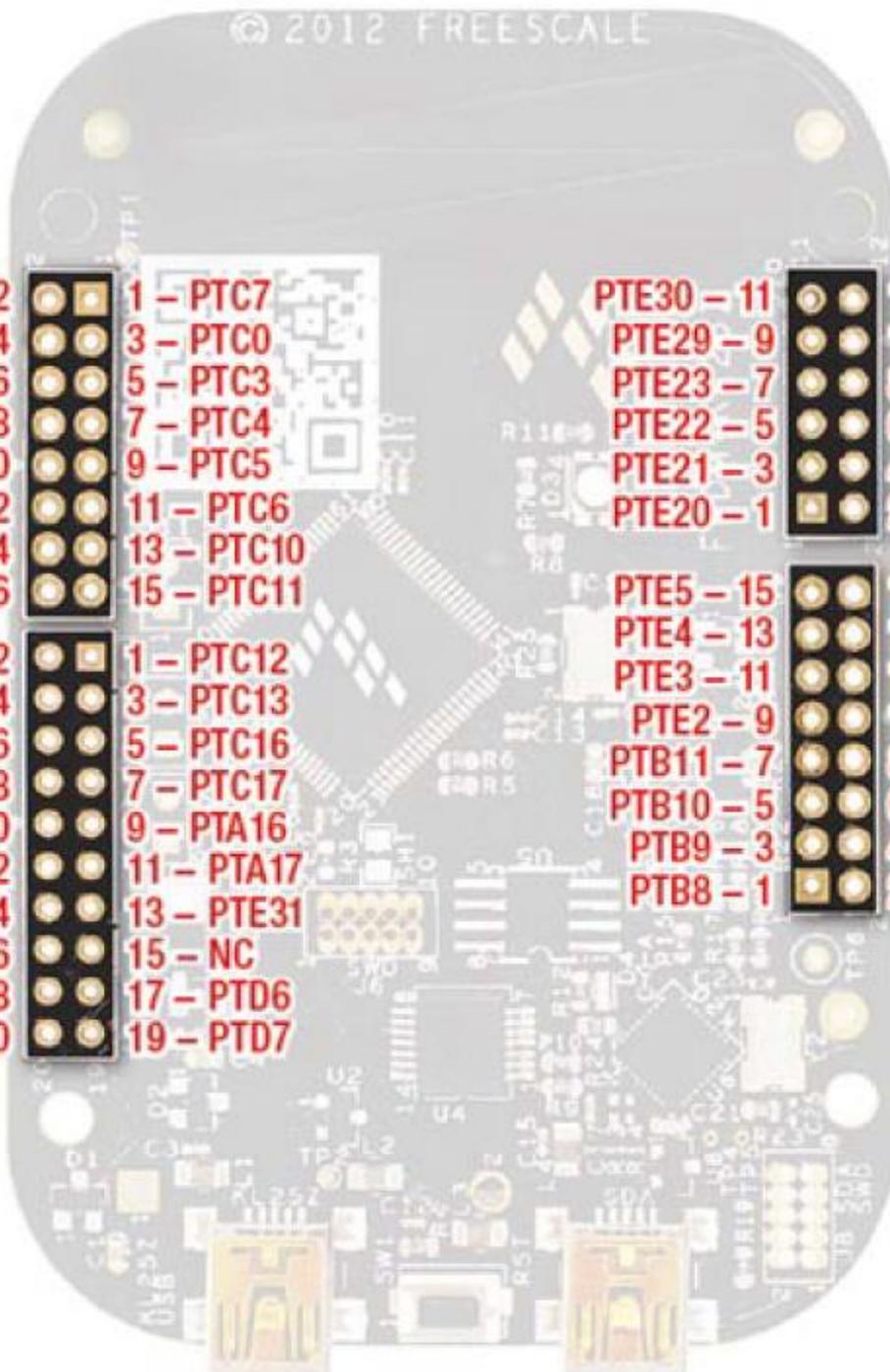
16 - P5-9V\_VIN  
14 - GND  
12 - GND  
10 - P5V\_USB  
8 - P3V3  
6 - RESET/PTA20  
4 - P3V3  
2 - SDA\_PTD5

uart1 TX; i2c SDA  
uart1 RX; i2c SCL

PTE0 - 18  
PTE1 - 20



17 - PTD6  
19 - PTD7



# Programming

Several different tools can be used for programming and debugging

- Keil Studio Cloud

- good and easy online tool, no debugging (at least with Firefox)

- Mbed Studio

- debugging available

- Mbed CLI

- command-line tool

- Keil uVision

- with debugger and tracing; full version needs a license

- MCUX

- with debugger, proprietary (NXP)

# uVision5: Blinky

- Project... Manage... Pack installer:
- File... Import: *Keil.Kinetis\_KLxx\_DFP.1.15.0.pack*
- Board: select *FRDM-KL25Z (Deprecated)*
- *Pack... ARM::CMSIS: remove version 6.1.0*
- Copy folder *FRDM-KL25* somewhere in your project home
- Project... open project... *<project\_home>... FRDM-KL25Z...*  
Blinky: *Blinky.uvprojx*
- Project... Options for target... Debug... *use CMSIS-DAP*
- Project... Options for Target... Target... ARM Compiler: *Use version 6*
- Project... Build target (1 warning generated)
- Flash... Download

## uVision5: Blinky\_257

- Project... Manage... Pack installer:
- File... Import: *Keil.Kinetis\_KLxx\_DFP.1.15.0.pack*
- Board: select *FRDM-KL25Z (Deprecated)*
- *Pack... ARM::CMSIS:* remove version 6.1.0
- Copy folder *Blinky\_257* somewhere in your project home
- Project... open project... <project\_home>... Blinky\_257...  
*Blinky.uvprojx*
- Project... Build target (1 warning generated)
- Flash... Download
  
- Note the instructions to switch on a LED...

## Example of tracing (taken from apnt\_257)

- Start the *blinky* project
- Select target: CMSIS-DAP-MTB
- Debug... Start Debug Session
- View... Register Window
- View... Disassembly Window
- Run... and then Stop
- PC will be around 0x26A-0x270:
  - while ((msTicks - curTicks) < dlyTicks);
- The following instruction is BX lr:
  - set R14=LR to 0
- BX operation is not allowed with LR=0 -> Fault!
  - last instruction in Trace Data is B to *HardFault\_Handler*

## Example of tracing (taken from apnt\_257)

- Exit and re-enter Debug
- View... Trace... Trace Data
- Repeat
  - Trace Data shows many B ☹
- Let's put a breakpoint there
  - Disassembly: click on the first column
- Exit and re-enter Debug
- Repeat
  - now Trace Data shows us how we arrived at *HardFault\_Handler*