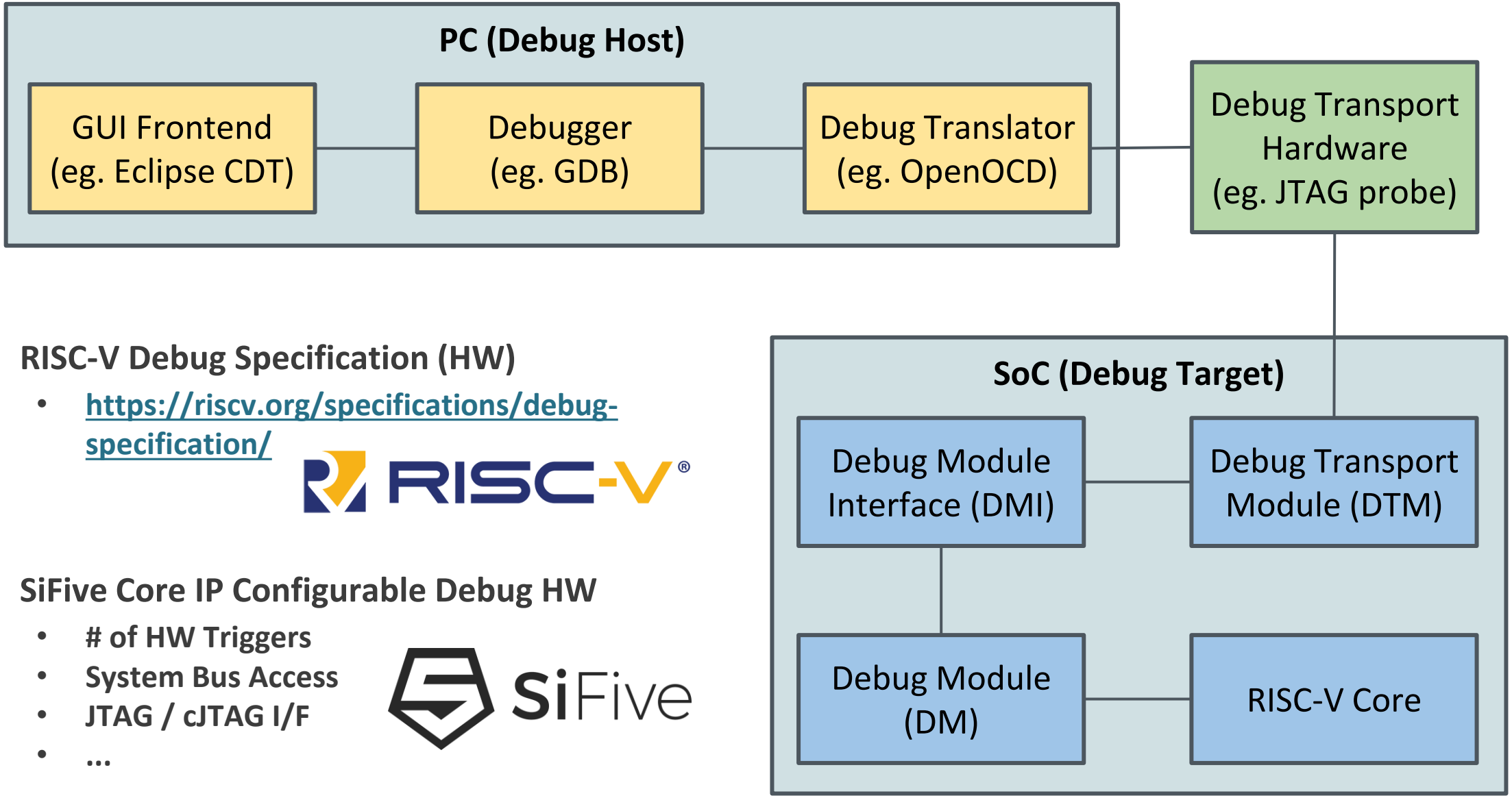
SiFive Tech Symposium 2019

RISC-V Core IP Debug SW Stack

RISC-V Core IP Debug SW+HW Stack



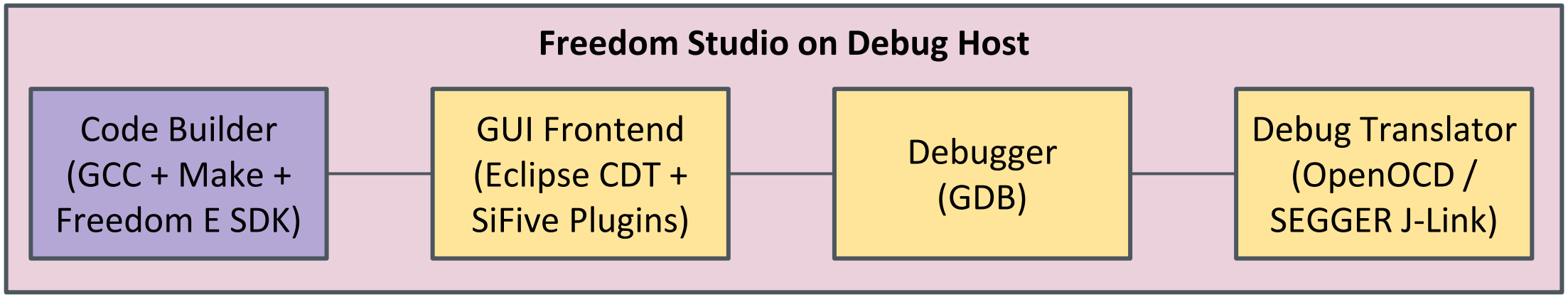
RISC-V Debug Specification (HW)

- https://riscv.org/specifications/debug-specification/

SiFive Core IP Configurable Debug HW

- # of HW Triggers
- System Bus Access
- JTAG / cJTAG I/F
- …

Freedom Studio - A Fully Integrated Developer Environment



What is Freedom Studio

- **IDE for Bare Metal embedded development & debugging on SiFive RISC-V Platforms and Core-IP**
- **Distributions available for Windows, MacOS and Linux**
- **Based on Open Source technology like Eclipse CDT**
- **Includes SiFive extensions**
  - Toolchain integration
  - Debugger interfaces
  - Project management
- **Bundled with all needed prebuilt packages**
  - Java Runtime Environment
  - RISC-V OpenOCD
  - SEGGER J-Link SW
  - GNU RISC-V Newlib Toolchain (Binutils, GCC, Newlib, GDB)
  - Freedom E SDK (BSP's, Freedom Metal HAL, example programs)

Freedom Tools - Open Source Repository for SiFive's RISC-V Tools

What is Freedom Tools

- **Public repository with open source tools and scripts for building these tools**
  - https://github.com/sifive/freedom-tools
    - GNU RISC-V Newlib Toolchain (Binutils, GCC, Newlib (+Nano), GDB)
    - RISC-V OpenOCD
- **The toolchain setup is a riscv64-unknown-elf-gcc Newlib toolchain for Bare Metal embedded development**
  - Supported multilibs: rv32e, rv32eac, rv32em, rv32emac, rv32i, rv32iac, rv32im, rv32imac, rv32imafc, rv32imafdc, rv64imac, rv64imafc, rv64imafdc
- **We use these scripts to provide prebuilt binary distributions for Windows, MacOS and Linux**
- **We provide these prebuilt binary distributions as part of Freedom Studio and as standalone on the website**

Other SW packages bundled as part of the Freedom Studio distributions

- **SEGGER J-Link SW**
  - https://www.segger.com/downloads/jlink/

- **Java Runtime Environment**
  - https://www.oracle.com/technetwork/java/javase/downloads/index.html

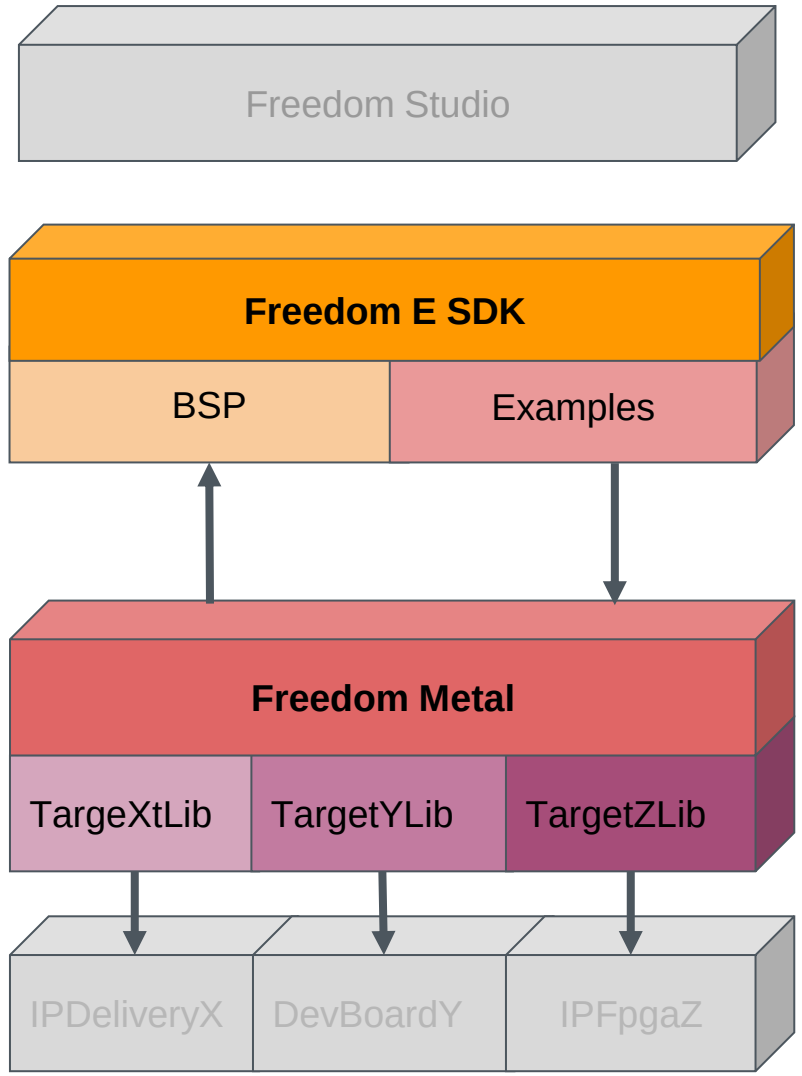
Freedom E SDK & Metal - A Bare Metal SW Stack for SiFive Devices

What is Freedom E SDK

- Embedded development kit providing a command line driven workflow with Examples and Utilities including BSP's for
  - Standard Core IP Deliverables
  - Standard Core FPGA Deliverables
  - SiFive Development Boards
- Examples uses Freedom Metal to provide portability
- Open source repository
  - https://github.com/sifive/freedom-e-sdk

What is Freedom Metal

- Library for writing Portable, Bare Metal SW for all SiFive devices
  - A Bare Metal C application environment
  - An API for controlling CPU features and peripherals
  - The ability to retarget to any SiFive RISC-V product
  - A RISC-V hardware abstraction layer (HAL)
- Uses BSP's to provide target adaptation
- Open source repository
  - https://github.com/sifive/freedom-metal

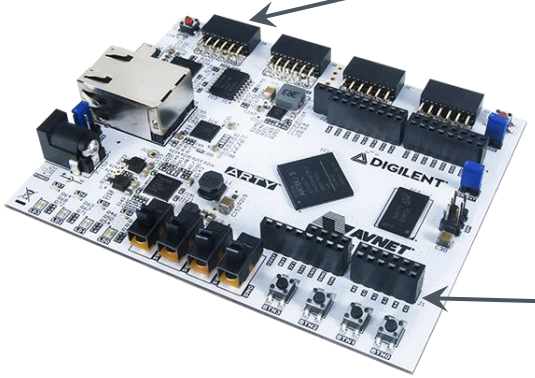


Supported Debug Transport Hardware - JTAG Probes

Olimex ARM-USB-TINY Probe

SiFive HiFive Unleashed
& HiFive1 (FTDI)

Digilent Arty FPGA



PC (Debug Host)

Debug Translator
OpenOCD

or

Debug Translator
SEGGER J-Link

SEGGER J-Link Probe

SiFive HiFive1 revB (J-Link
OB)

How do you get it?

https://www.sifive.com/boards

Prebuilt RISC-V GCC Toolchain

Save time by using one of our prebuilt toolchains which contain all the tools necessary to compile and debug programs on SiFive products. Our toolchain distributions have been carefully packaged to support both 32-bit & 64-bit ISAs.

GNU Embedded Toolchain — v2019.02.0

- Windows
- macOS
- CentOS
- Ubuntu

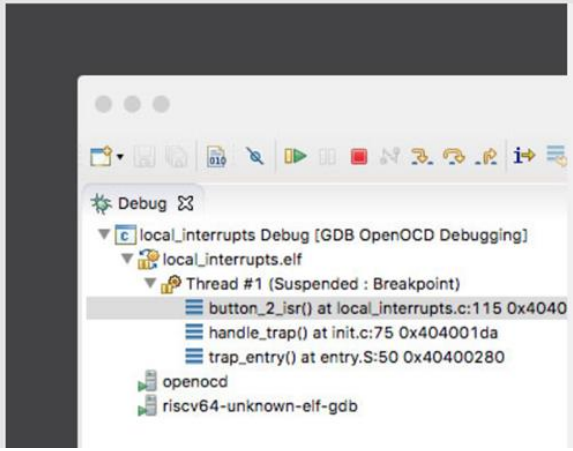
OpenOCD — v2019.02.0

- Windows
- macOS
- CentOS
- Ubuntu

https://www.sifive.com/boards

Freedom Studio

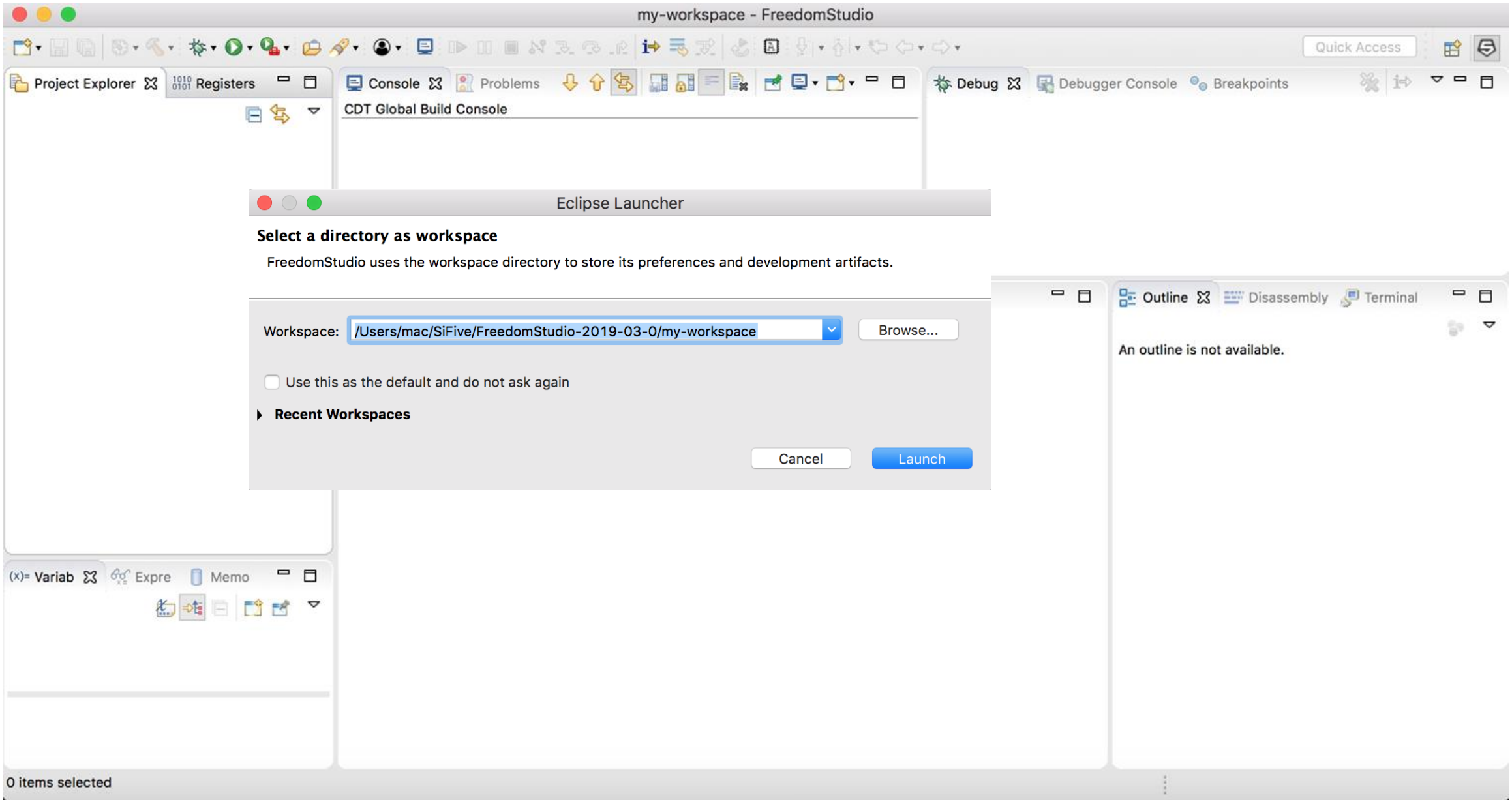
Freedom Studio is the fastest way to get started programming with your SiFive hardware. Freedom Studio is built on top of the popular Eclipse IDE and packaged with a prebuilt toolchain and example projects from the Freedom E SDK. Freedom Studio is compatible with all SiFive RISC-V development boards.



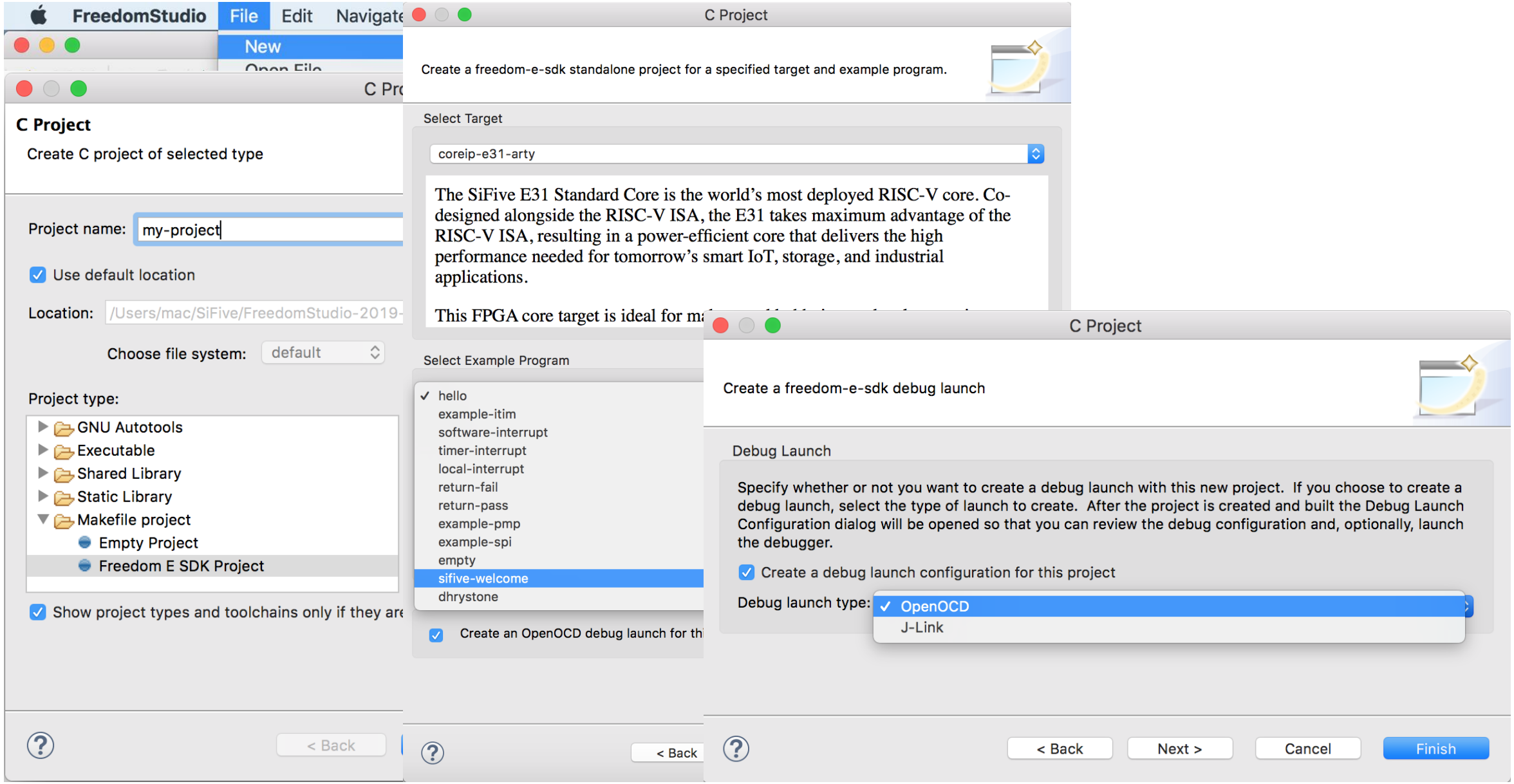
Freedom Studio

- Windows
- macOS
- Linux

How does it look?



Freedom E SDK New Project Wizard



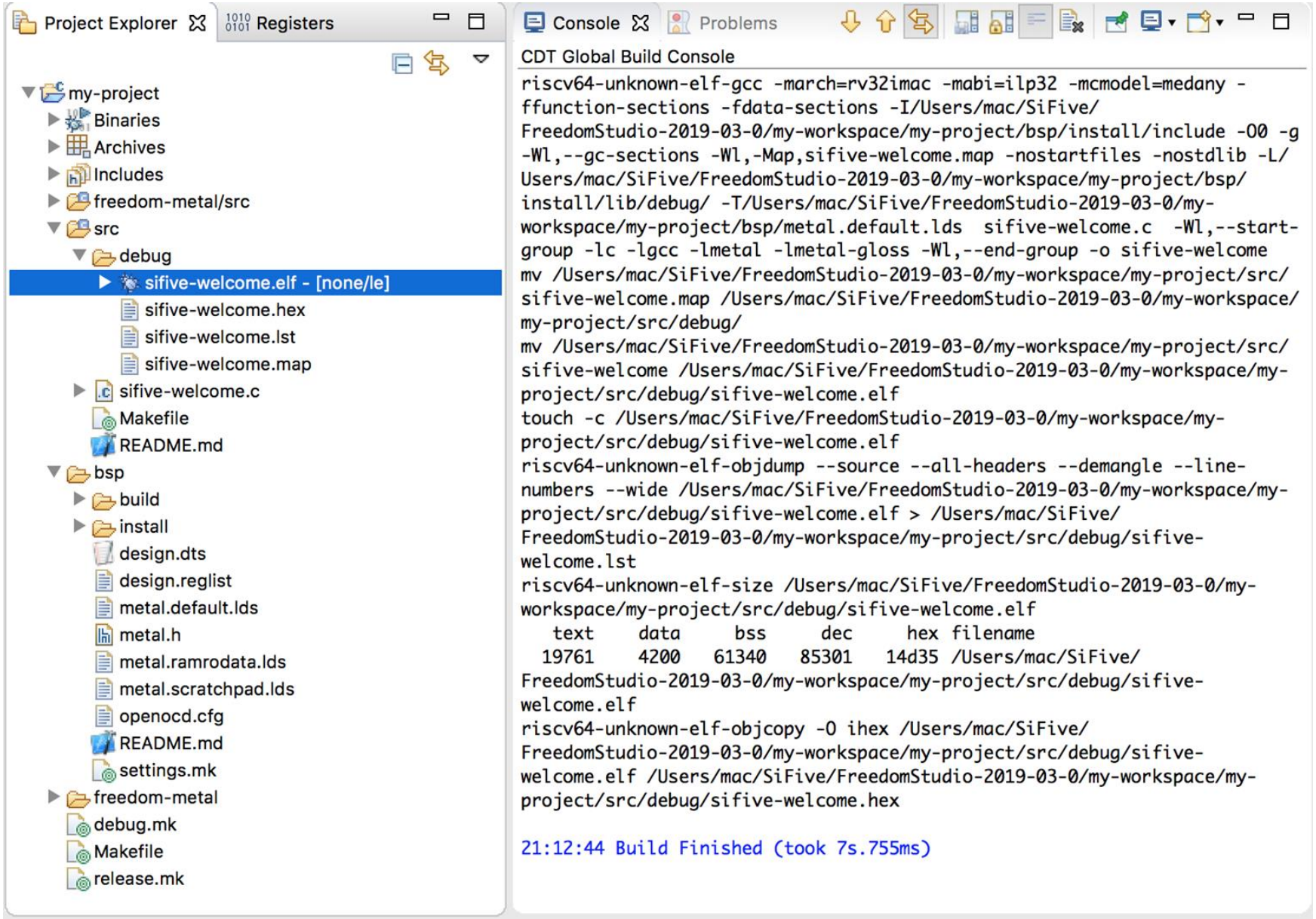
Project Explorer and Console after project is created

Freedom-E-SDK:

```
make standalone
STANDALONE_DEST=/Users
/mac/SiFive/FreedomStu
dio-2019-03-0/my-
workspace/my-project
TARGET=coreip-e31-arty
PROGRAM=sifive-welcome
RISCV_PATH=/Users/mac/
SiFive/FreedomStudio-
2019-03-
0/FreedomStudio.app/Co
ntents/Eclipse/SiFive/
toolchain/riscv64-
unknown-elf-gcc-8.2.0-
2019.02.0
```

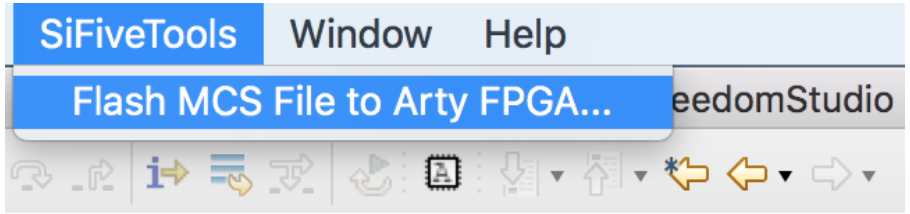
Standalone-Project:

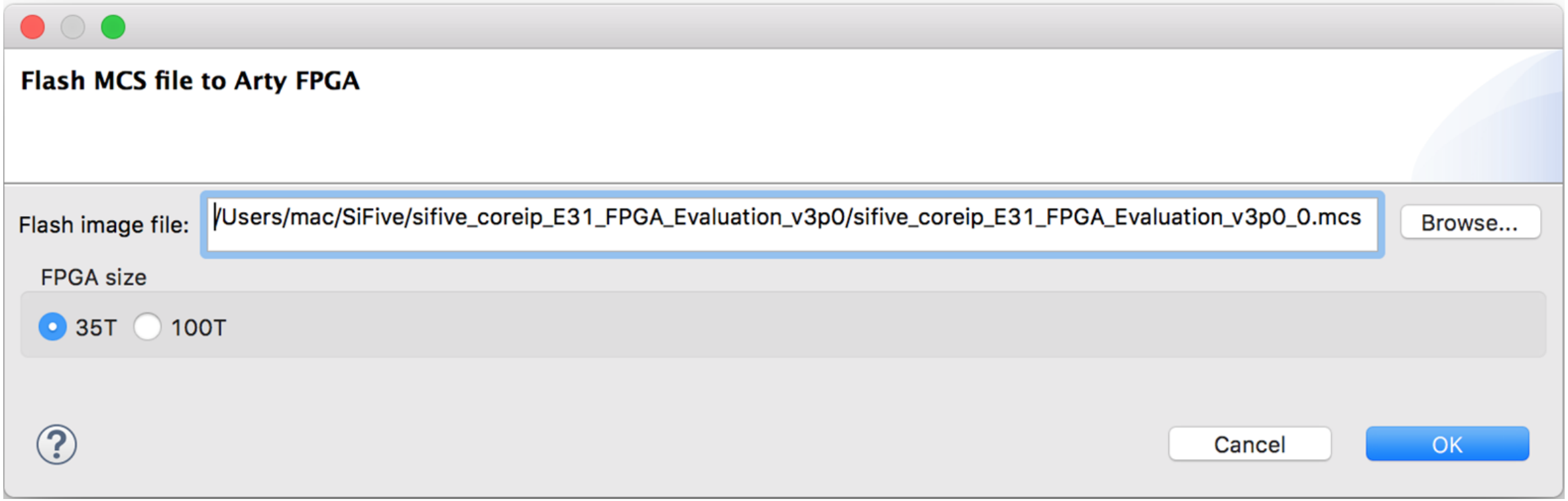
```
make all
CONFIGURATION=debug
```



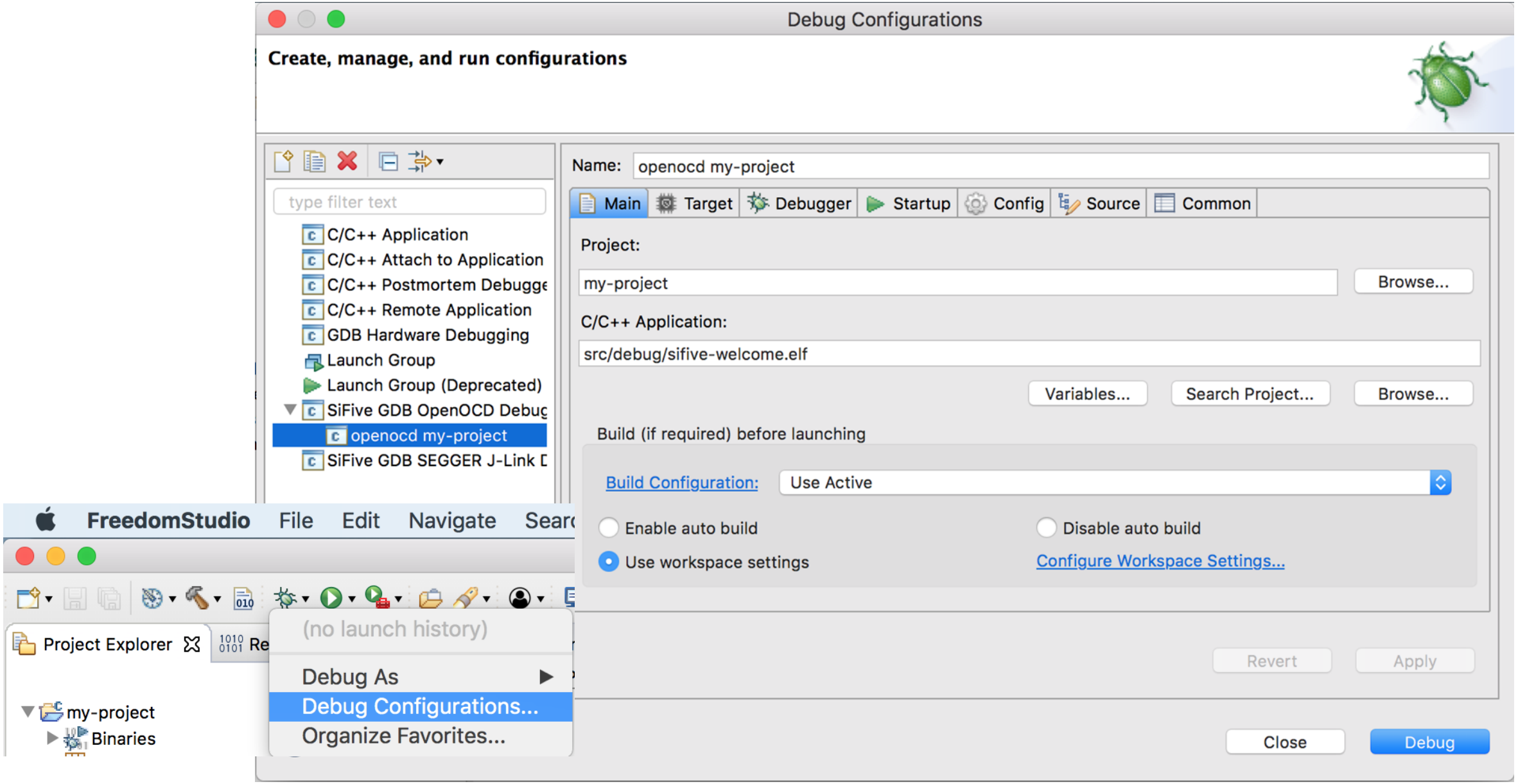
Flashing Arty FPGA Image

- **Writes driver .bit file to FPGA via Xc3sprog (Digilent)**
  - E31 image in FPGA RAM (not Flash)
  - E31 image allows OpenOCD to access Flash
- **Writes MCS (ihex) file to Flash via OpenOCD (Olimex)**

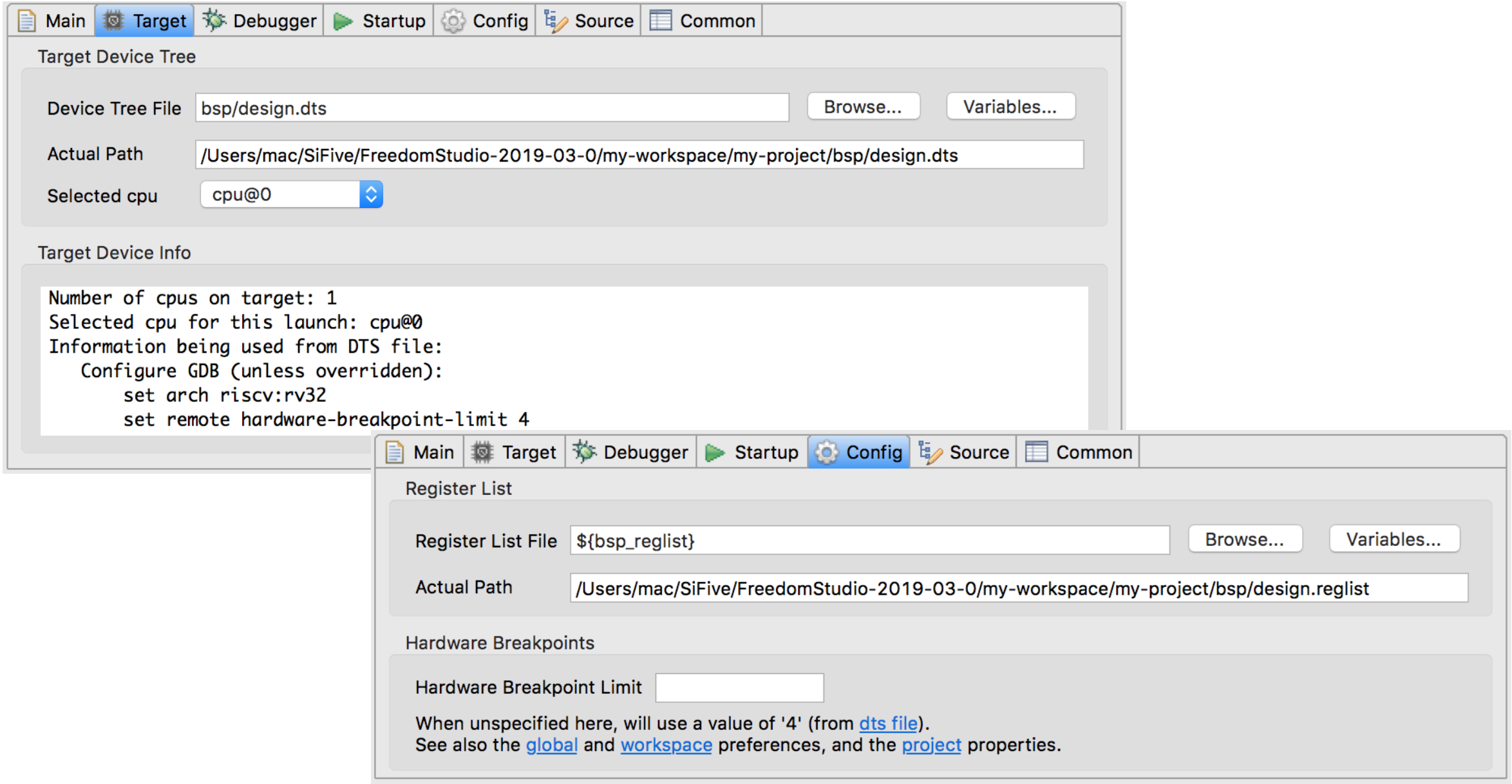




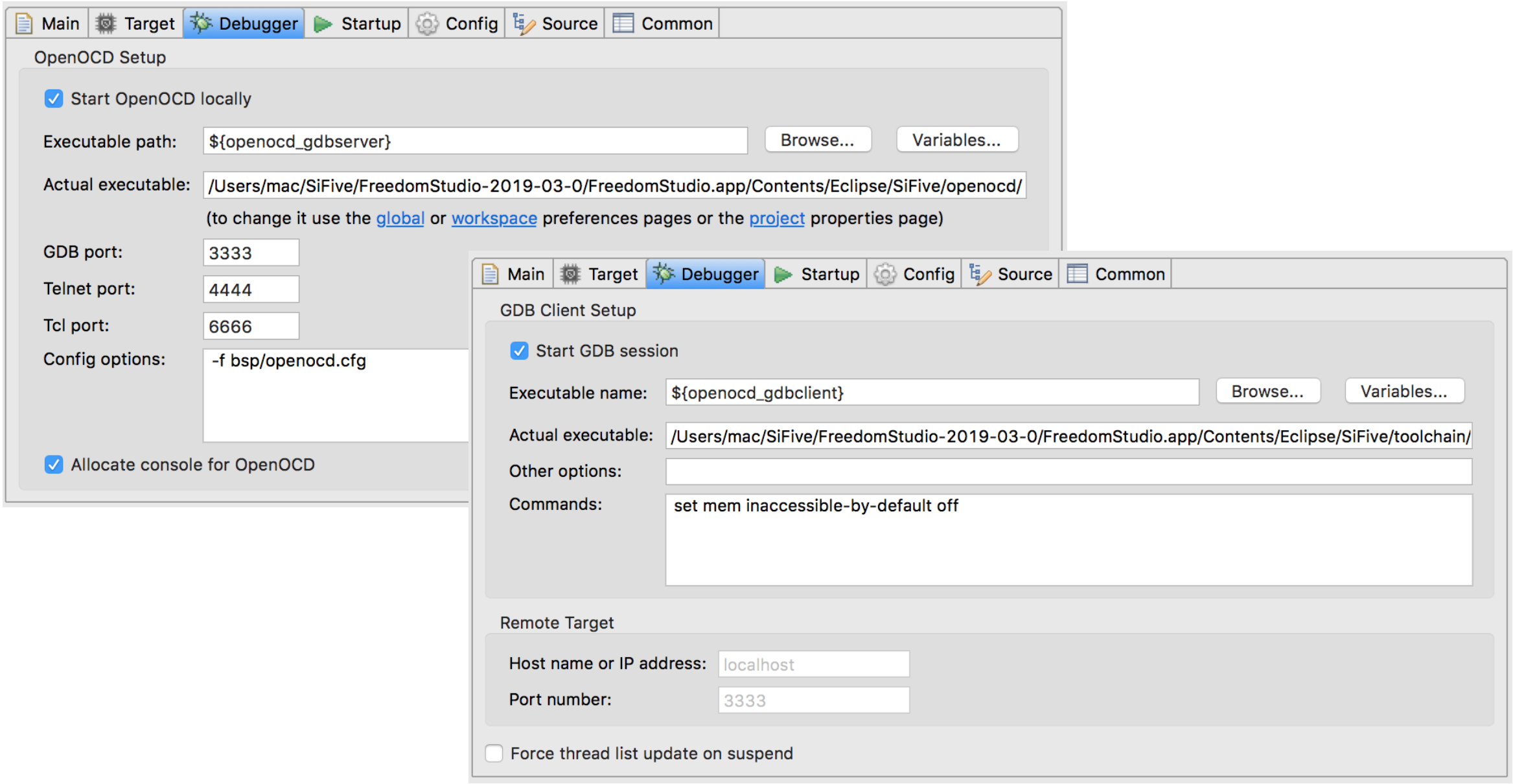
Debug configurations



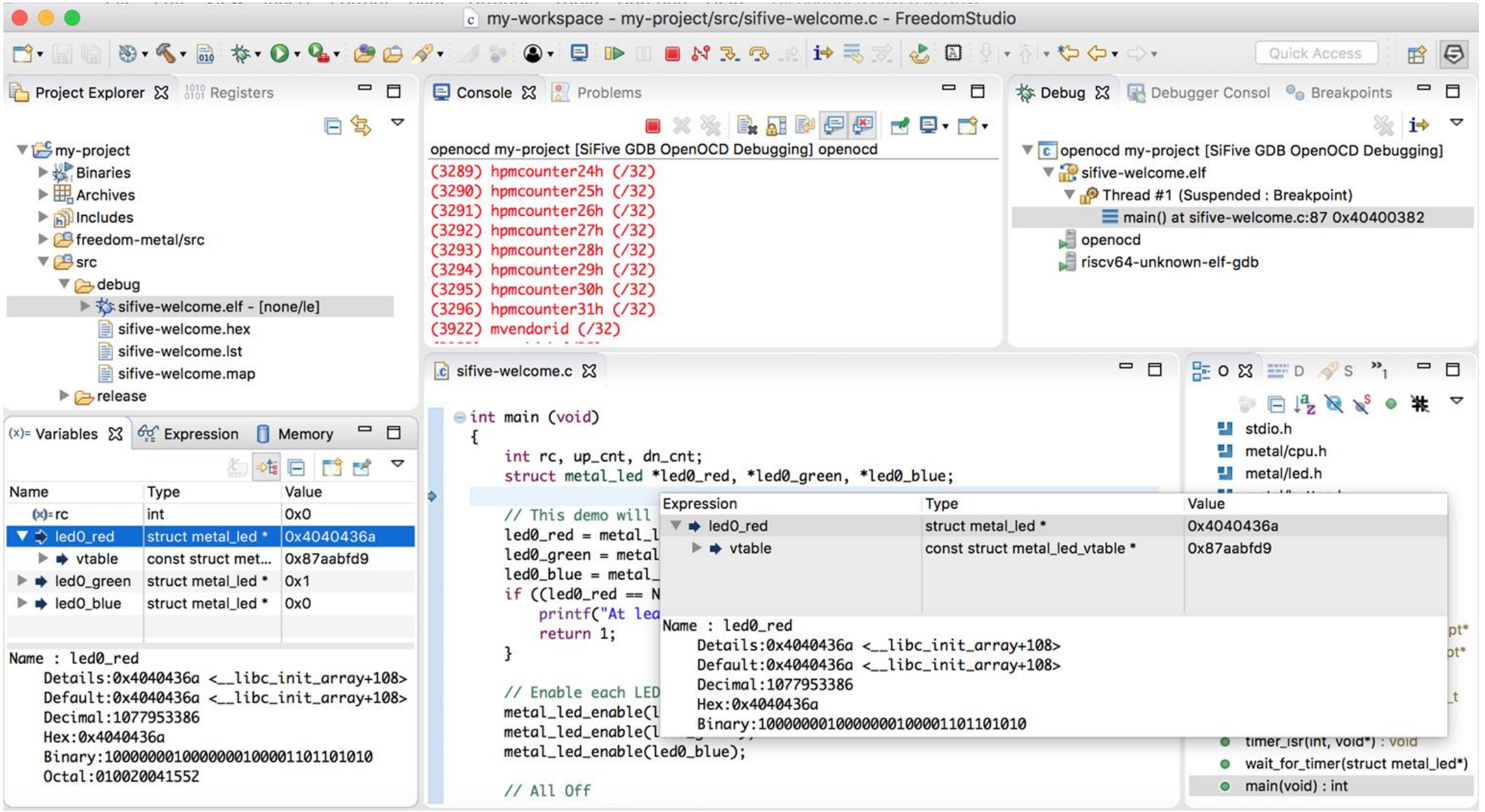
Target Device information and configuration



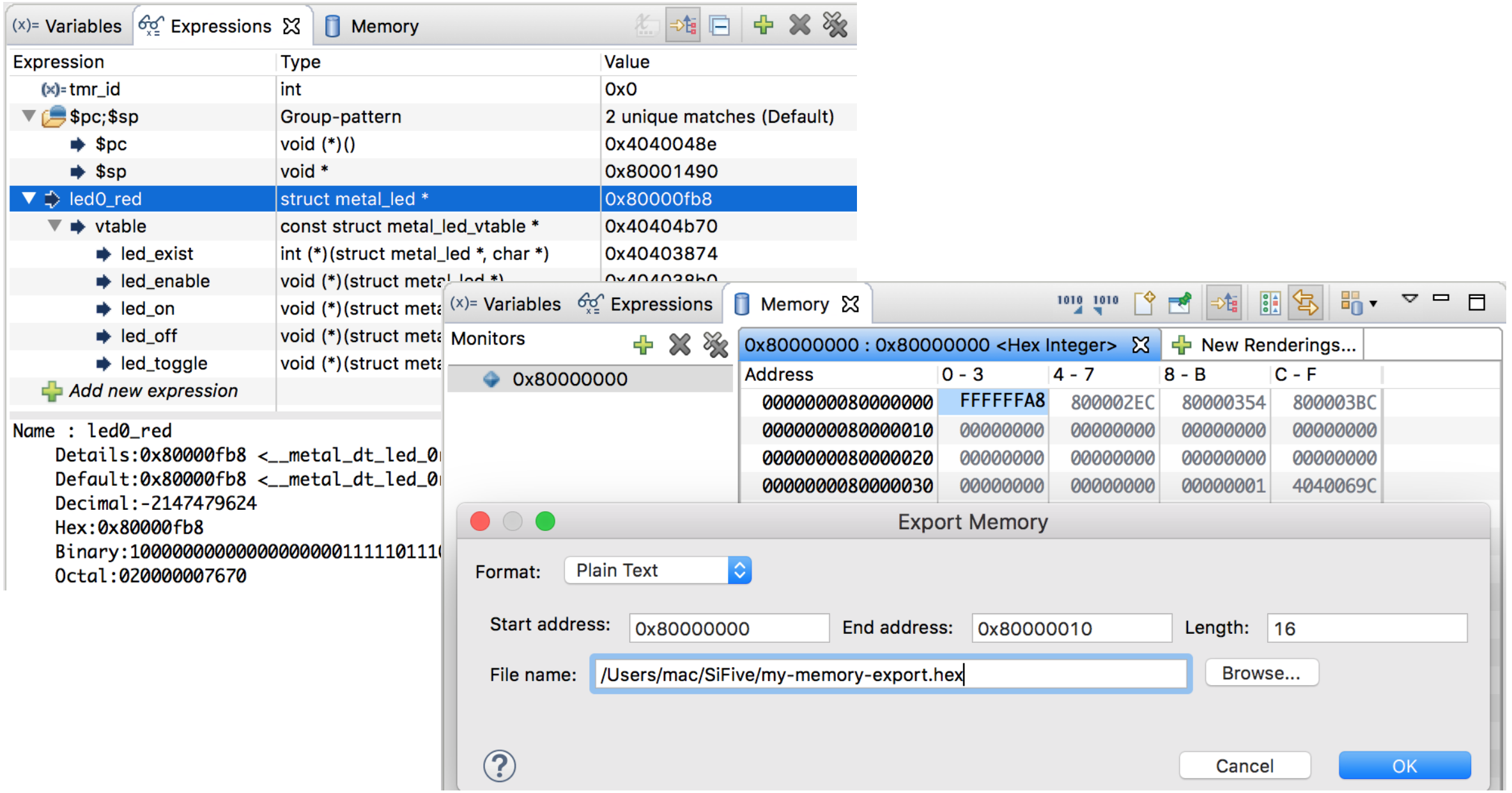
Debugger setup - GDB client and OpenOCD server



Start debugging



Viewing elements and memory



Viewing registers and terminal output

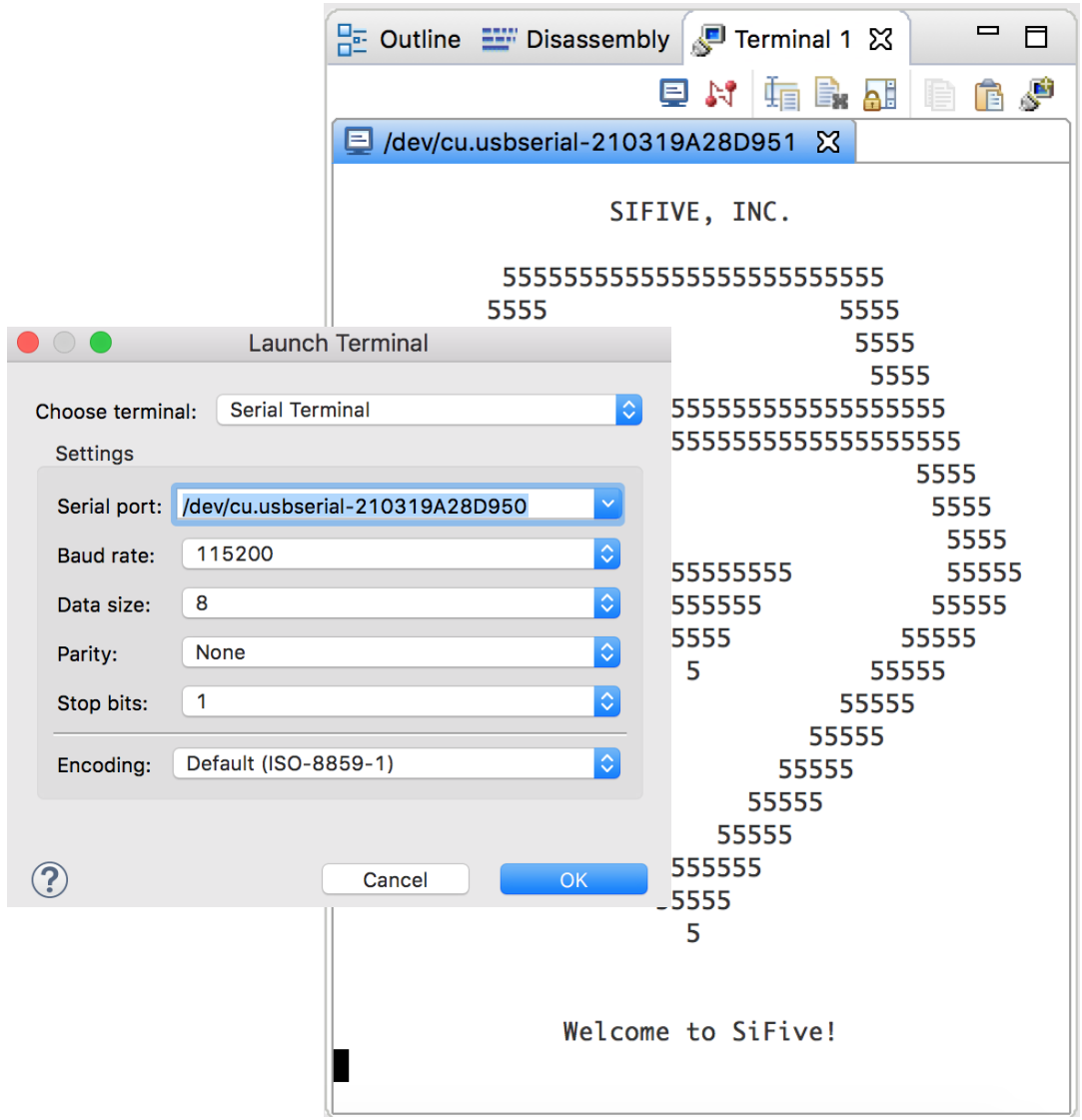
<project>/bsp/design.reglist file:

```
zero
ra
sp
gp
tp
t0
t1
t2
fp
s1
a0
a1
a2
a3
a4
a5
a6
a7
s2
...
```

Project Explorer | Registers

Name	Value	Description
General Registers		General Purpose
zero	0x0	
ra	0x4040048e	
sp	0x80001490	
gp	0x80001840	
tp	0x0	
t0	0x40401212	
t1	0x40000	
t2	0xffffffff	
fp	0x800014b0	
s1	0x0	
a0	0xffffffff	
a1	0xa	
a2	0x22	
a3	0x22	
a4	0x22	
a5	0xffffffff	
a6	0x1f	
a7	0x0	
s2	0x40403b10	

```
Name : ra
    Hex:0x4040048e
    Decimal:1077937294
    Octal:010020002216
    Binary:10000000100000000000010010001110
    Default:0x4040048e <main+276>
```



Viewing disassembly

sifive-welcome.c

```
        return 2;
    }
    cpu_intr = metal_cpu_interrupt_controller(cpu0);
    if (cpu_intr == NULL) {
        printf("CPU interrupt controller is null.\n");
        return 3;
    }
    metal_interrupt_init(cpu_intr);

    // display welcome banner
    display_banner();

    // Setup Timer and its interrupt so we can toggle LEDs on 1s cadence
    tmr_intr = metal_cpu_timer_interrupt_controller(cpu0);
    if (tmr_intr == NULL) {
        printf("TIMER interrupt controller is  null.\n");
        return 4;
    }
    metal_interrupt_init(tmr_intr);
    tmr_id = metal_cpu_timer_get_interrupt_id(cpu0);
    rc = metal_interrupt_register_handler(tmr_intr, tmr_id, timer_isr, cpu0);
    if (rc < 0) {
        printf("TIMER interrupt handler registration failed\n");
        return (rc * -1);
    }

    // Lastly CPU interrupt
    if (metal_interrupt_enable(cpu_intr, 0) == -1) {
        printf("CPU interrupt enable failed\n");
        return 6;
    }
```

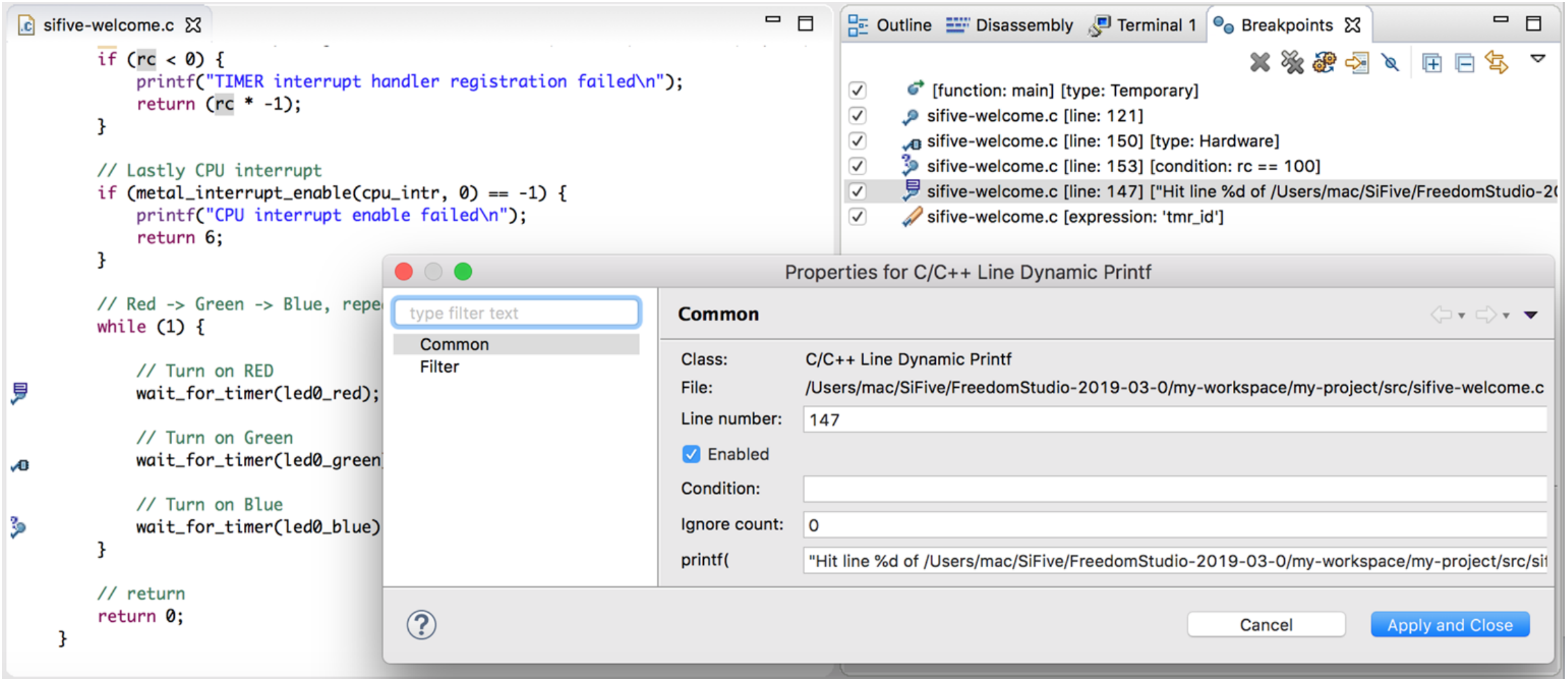
Outline | Disassembly | Terminal 1

Enter location here

```
114         if (cpu_intr == NULL) {
40400478:   li      a5,3
4040047a:   j       0x4040057a <main+512>
116             return 3;
4040047c:   auipc   a5,0x3fc01
40400480:   addi    a5,a5,-992 # 0x8000109c <cpu_intr>
40400484:   lw      a5,0(a5)
40400486:   mv      a0,a5
40400488:   jal     ra,0x40402132 <metal_interrupt_init>
119
4040048c:   jal     0x40400180 <display_banner>
122
4040048e:   auipc   a5,0x3fc01
40400492:   addi    a5,a5,-1006 # 0x800010a0 <cpu0>
40400496:   lw      a5,0(a5)
40400498:   mv      a0,a5
4040049a:   jal     ra,0x40402092 <metal_cpu_timer_interrupt_controll
4040049e:   mv      a4,a0
404004a0:   auipc   a5,0x3fc01
404004a4:   addi    a5,a5,-1032 # 0x80001098 <tmr_intr>
404004a8:   sw      a4,0(a5)
123         // Setup Timer and its interrupt so we can toggle LEDs
404004aa:   auipc   a5,0x3fc01
404004ae:   addi    a5,a5,-1042 # 0x80001098 <tmr_intr>
404004b2:   lw      a5,0(a5)
404004b4:   bnez    a5,0x404004c4 <main+330>
124           tmr_intr = metal_cpu_timer_interrupt_controller(cpu0);
404004b6:   auipc   a0,0x4
404004ba:   addi    a0,a0,1430 # 0x40404a4c
404004be:   jal     0x4040060e <puts>
```

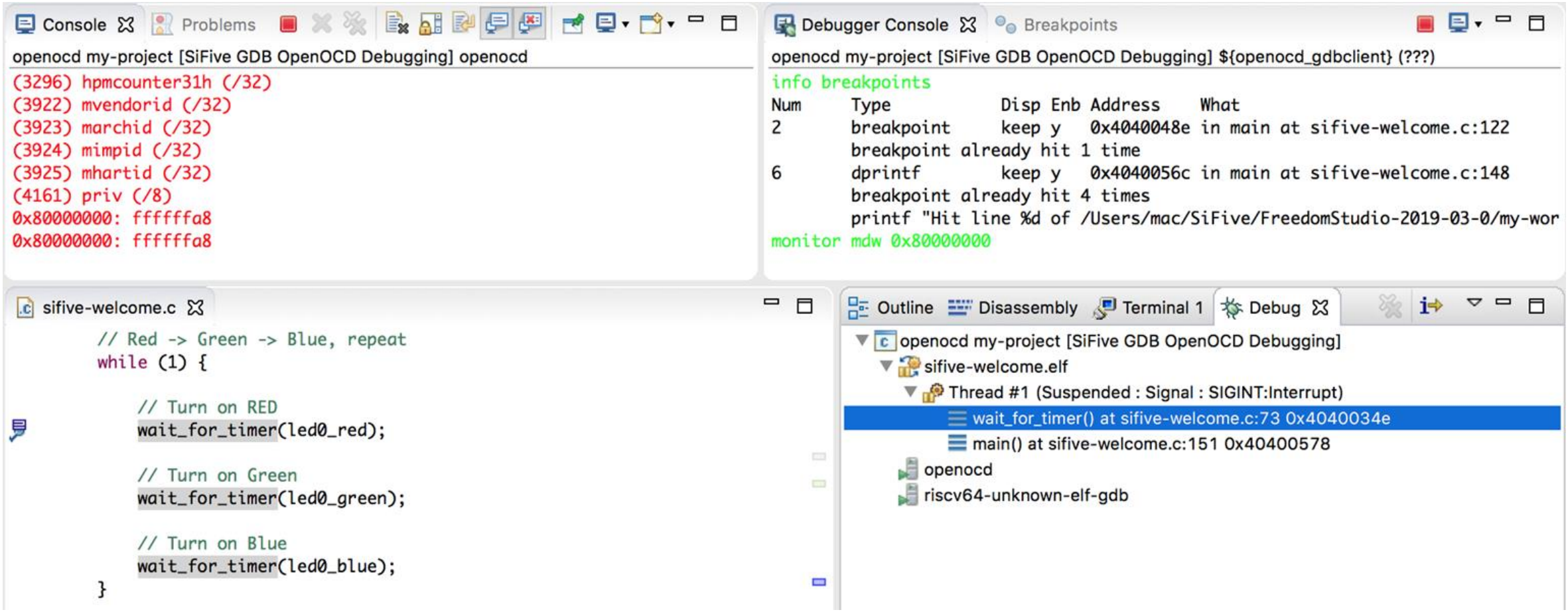
Setting breakpoints

- Types: temporary, regular, hardware, conditional, dynamic printf, data watchpoint

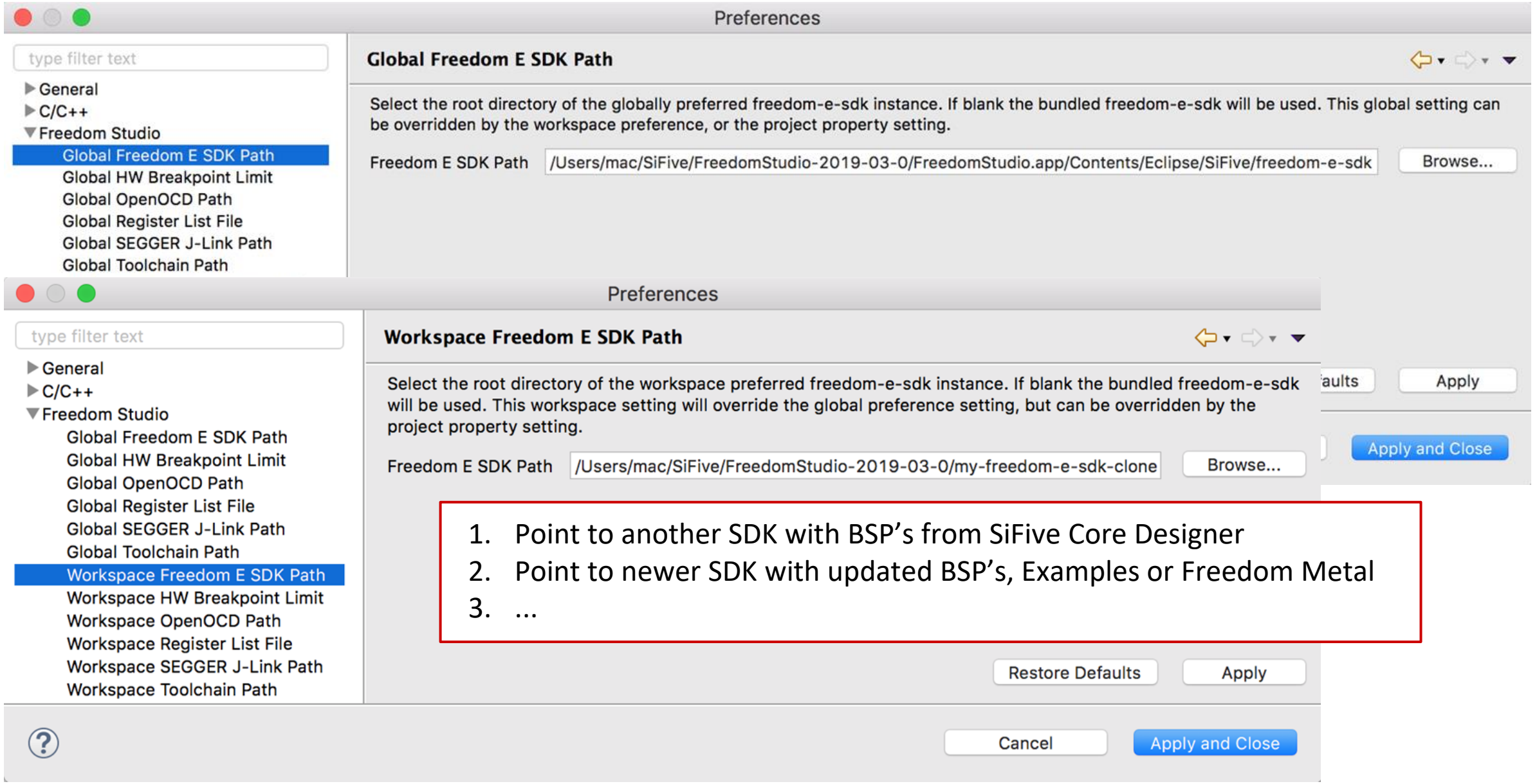


Debugger interfaces

- Debugger Console is a direct interface to the GDB command shell
- GDB “monitor” command feeds directly to OpenOCD with output in Console
- Debug View shows the state of the debugging session

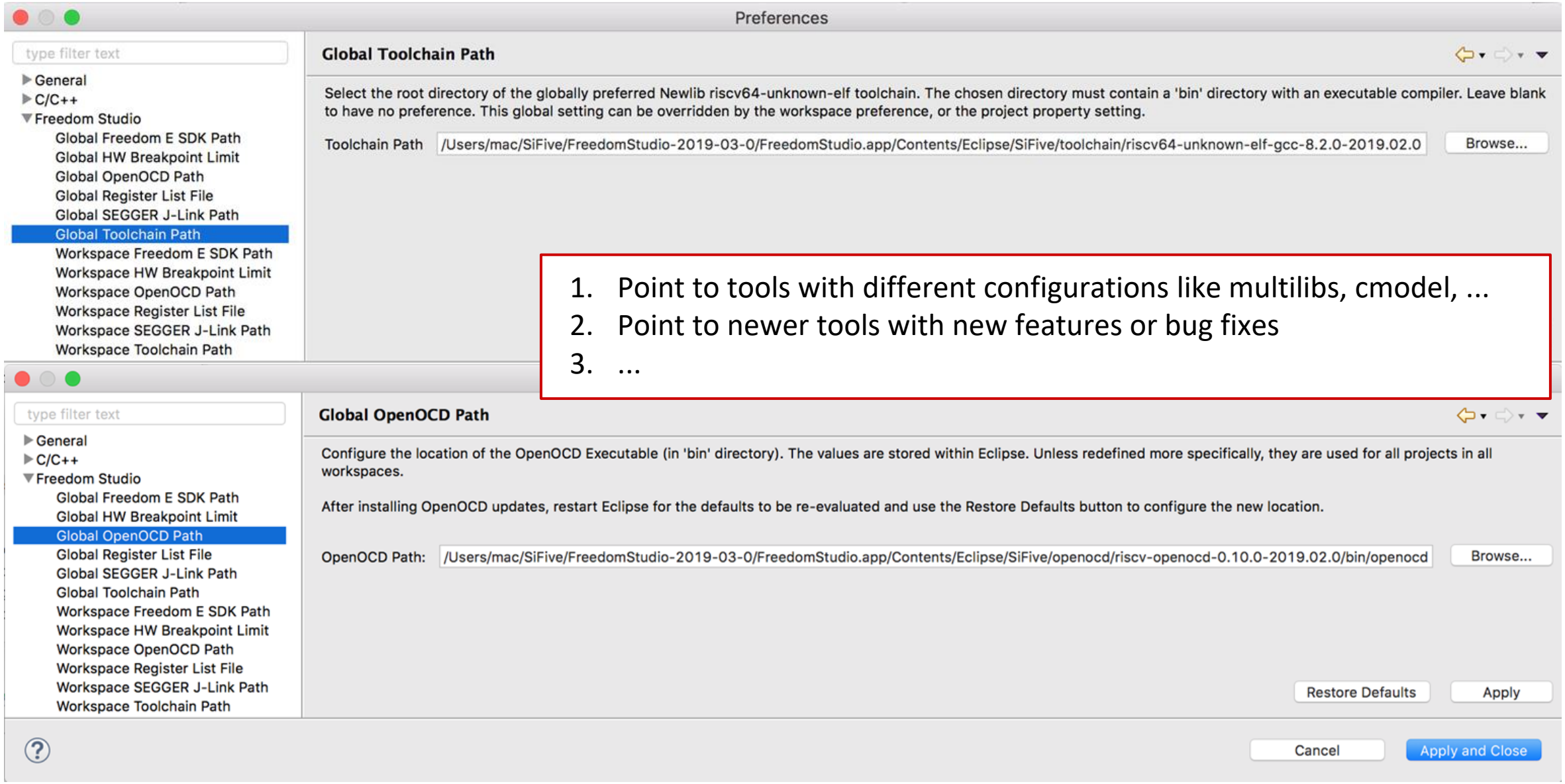


Using other versions of Freedom-E-SDK than the bundled one

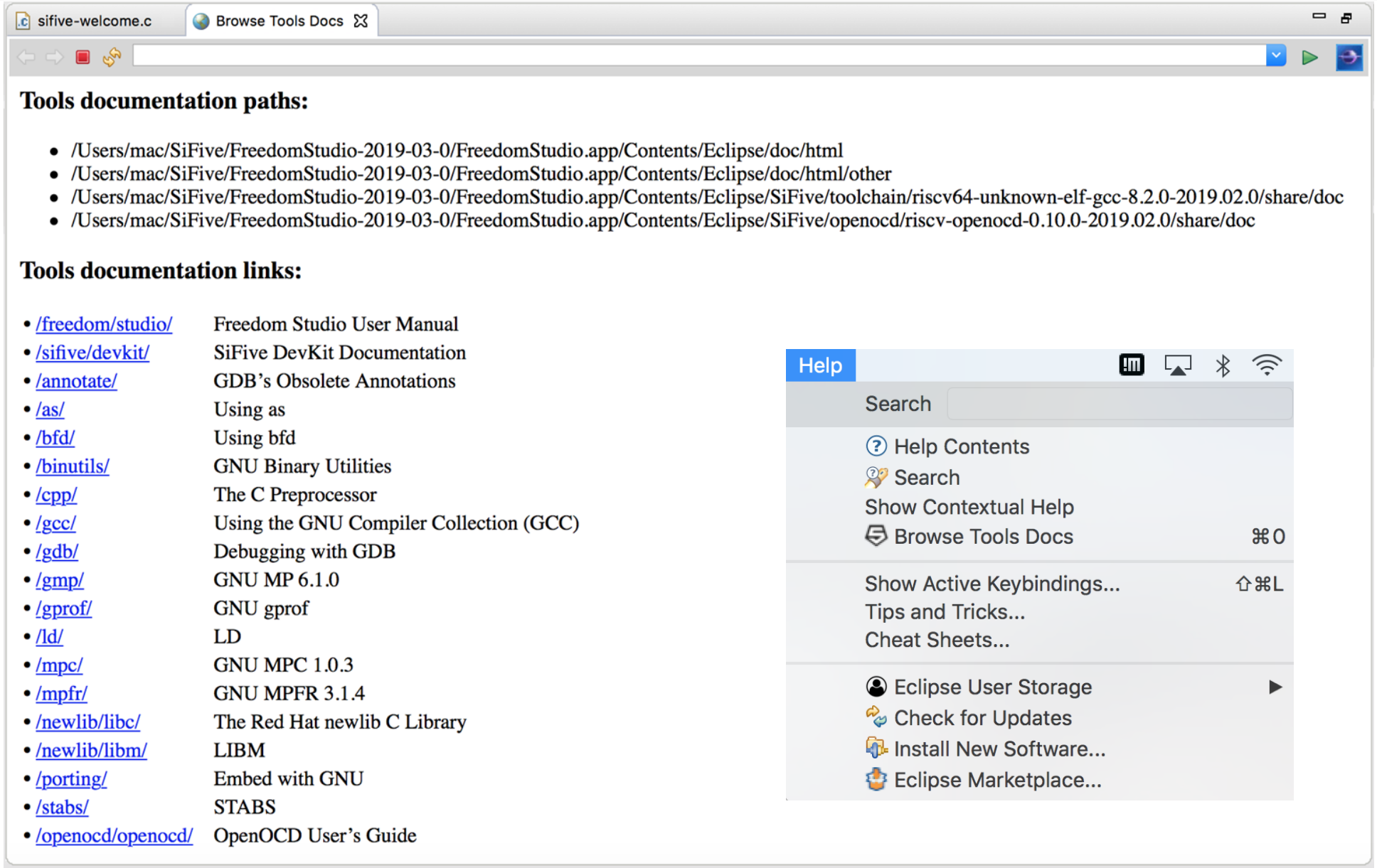


1. Point to another SDK with BSP's from SiFive Core Designer
2. Point to newer SDK with updated BSP's, Examples or Freedom Metal
3. ...

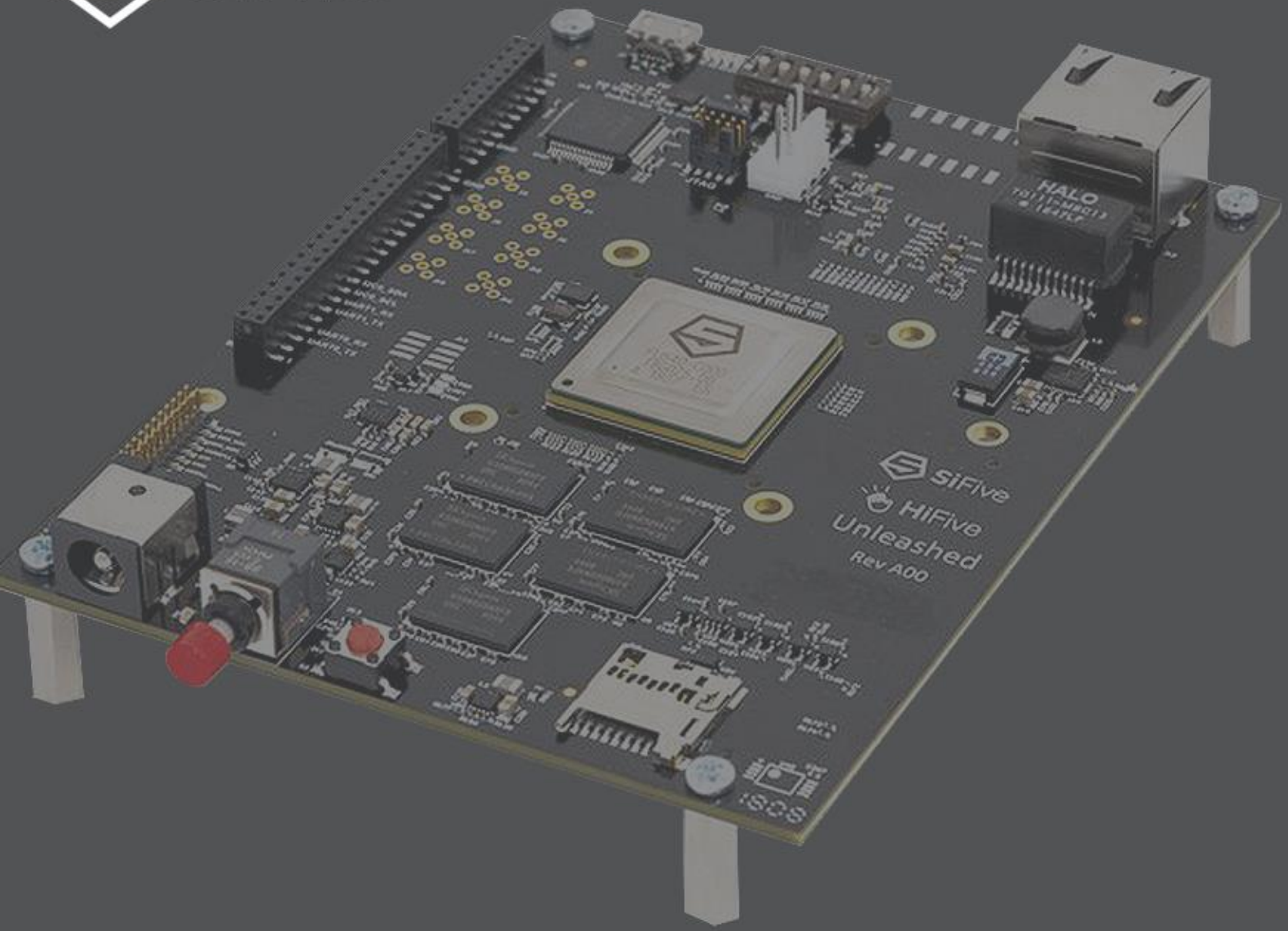
Using other tools than the bundled ones



Accessing bundled documentation



SiFive



Silicon verified. Market proven.

The most advanced configurable core IP and silicon solutions from the inventors of RISC-V.

Microcontrollers ■ Embedded ■ Linux ■ Multicore

■ Networking ■ Storage ■ Computing ■ AI
■ Industrial ■ IoT ■ Consumer ■ Automotive

www.sifive.com