

A Leading Provider of Microcontroller, Security,  
Mixed-Signal, Analog & Flash-IP Solutions



**SoftConsole Renode Webinar: 3**  
***Presented by Hugh Breslin***  
***Thursday July 4<sup>th</sup>***



# First Thursday's

---

May 2 - Webinar 1: Discover Renode for PolarFire™ SoC Design and Debug

June 6 - Webinar 2: How to Get Started with Renode for PolarFire SoC

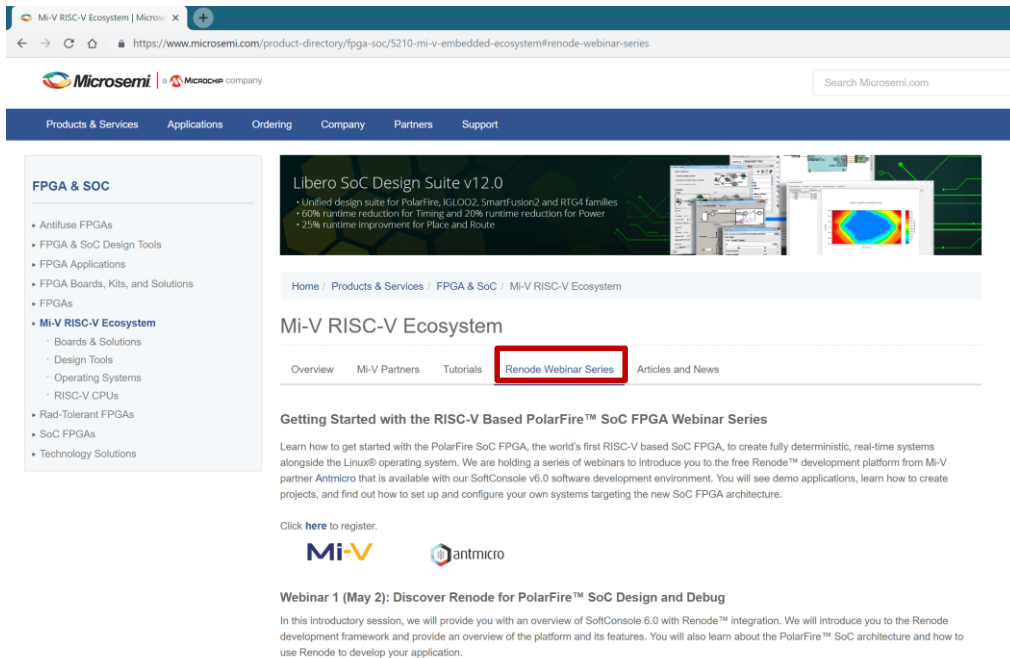
July 4 - Webinar 3: Learn to Debug a Bare-Metal PolarFire SoC Application with Renode

Aug. 1 - Webinar 4: Tips and Tricks for Even Easier PolarFire SoC Debug with Renode

Sept. 5 - Webinar 5: Add and Debug PolarFire SoC Peripherals with Renode

Oct. 3 - Webinar 6: Intro to PolarFire SoC MSS Configuration and Software Flow

# Supporting Content



Mi-V RISC-V Ecosystem | Microsemi

https://www.microsemi.com/product-directory/fpga-soc/5210-mi-v-embedded-ecosystem#renode-webinar-series

Microsemi | a MICROCHIP company

Search Microsemi.com

Products & Services Applications Ordering Company Partners Support

**FPGA & SOC**

- Antifuse FPGAs
- FPGA & SoC Design Tools
- FPGA Applications
- FPGA Boards, Kits, and Solutions
- FPGAs
- **Mi-V RISC-V Ecosystem**
  - Boards & Solutions
  - Design Tools
  - Operating Systems
  - RISC-V CPUs
- Rad-Tolerant FPGAs
- SoC FPGAs
- Technology Solutions

**Libero SoC Design Suite v12.0**

- Unified design suite for PolarFire, IGLOO2, SmartFusion2 and RTG4 families
- 60% runtime reduction for Timing and 20% runtime reduction for Power
- 25% runtime improvement for Place and Route

Home / Products & Services / FPGA & SoC / Mi-V RISC-V Ecosystem


**Mi-V RISC-V Ecosystem**

Overview Mi-V Partners Tutorials **Renode Webinar Series** Articles and News

**Getting Started with the RISC-V Based PolarFire™ SoC FPGA Webinar Series**

Learn how to get started with the PolarFire SoC FPGA, the world's first RISC-V based SoC FPGA, to create fully deterministic, real-time systems alongside the Linux® operating system. We are holding a series of webinars to introduce you to the free Renode™ development platform from Mi-V partner Antmicro that is available with our SoftConsole v6.0 software development environment. You will see demo applications, learn how to create projects, and find out how to set up and configure your own systems targeting the new SoC FPGA architecture.

Click [here](#) to register.

**Mi-V** 

**Webinar 1 (May 2): Discover Renode for PolarFire™ SoC Design and Debug**

In this introductory session, we will provide you with an overview of SoftConsole 6.0 with Renode™ integration. We will introduce you to the Renode development framework and provide an overview of the platform and its features. You will also learn about the PolarFire™ SoC architecture and how to use Renode to develop your application.

[www.microsemi.com/Mi-V](https://www.microsemi.com/Mi-V) “Renode Webinar Series”



# Learn to Debug a Bare-Metal PolarFire SoC Application with Renode

---

- **Configuring debug sessions, external tools and launch groups**
- **PolarFire SoC on Renode**
- **PSE-Blinky project**
- **Demo**

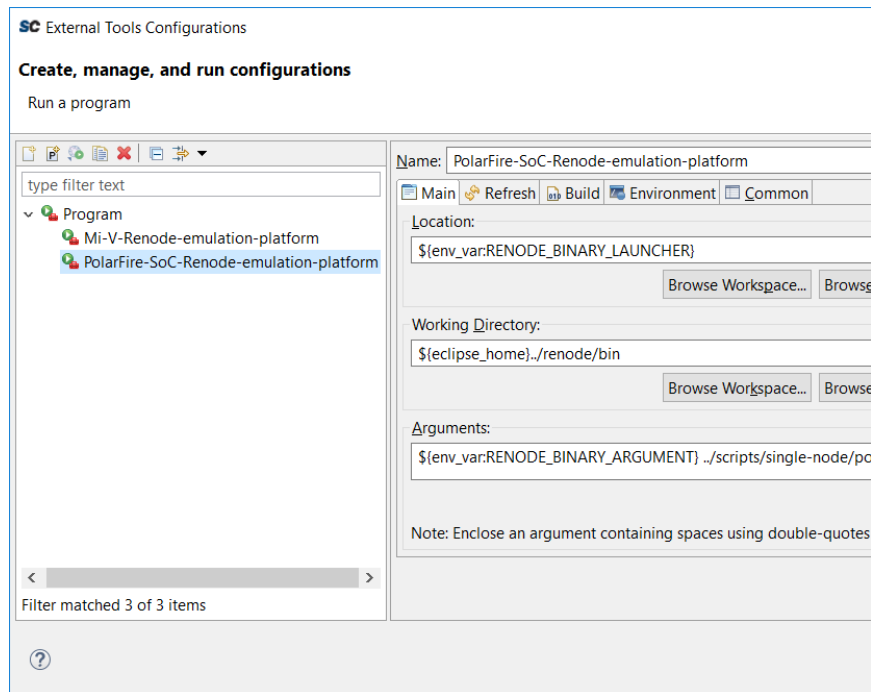


# **Learn to Debug a Bare-Metal PolarFire SoC Application with Renode**

**Configuring debug sessions, external tools and launch groups**

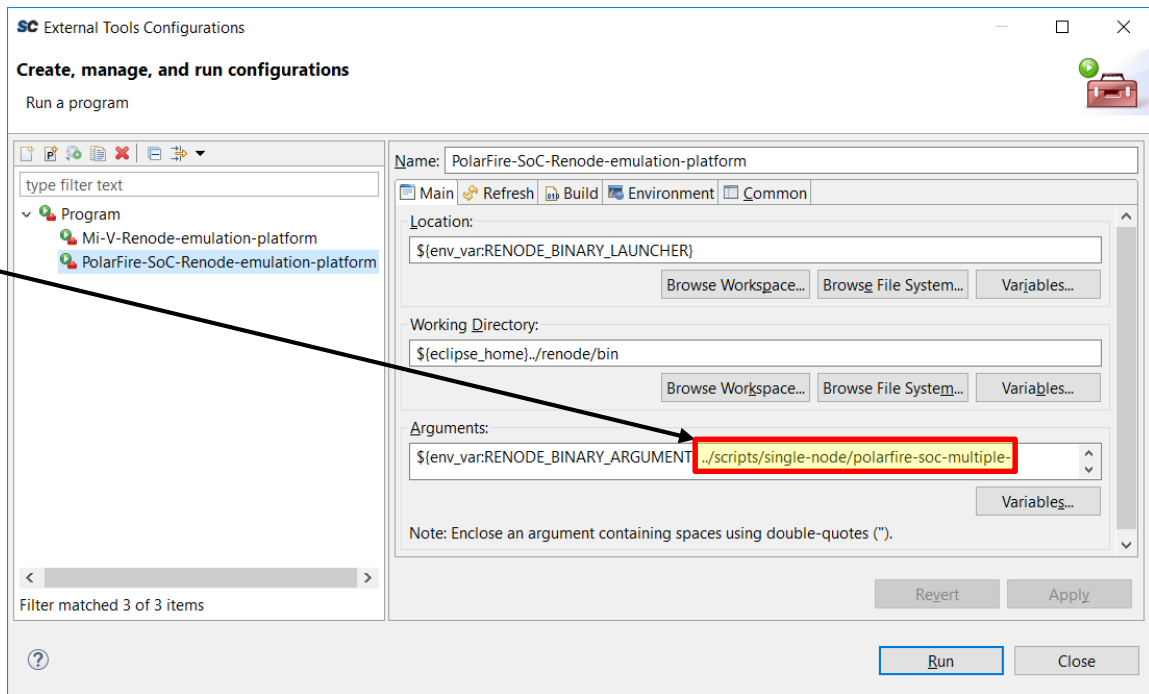
# Configuring debug sessions, external tools and launch groups

- **Renode is an external tool that needs to be running before you try to debug**
- **Launching a debug session without Renode is like trying to debug with the board off**
- **Two ways to do this:**
  - a) Manually launch Renode followed by GDB
  - b) Use a launch group to start Renode and then GDB



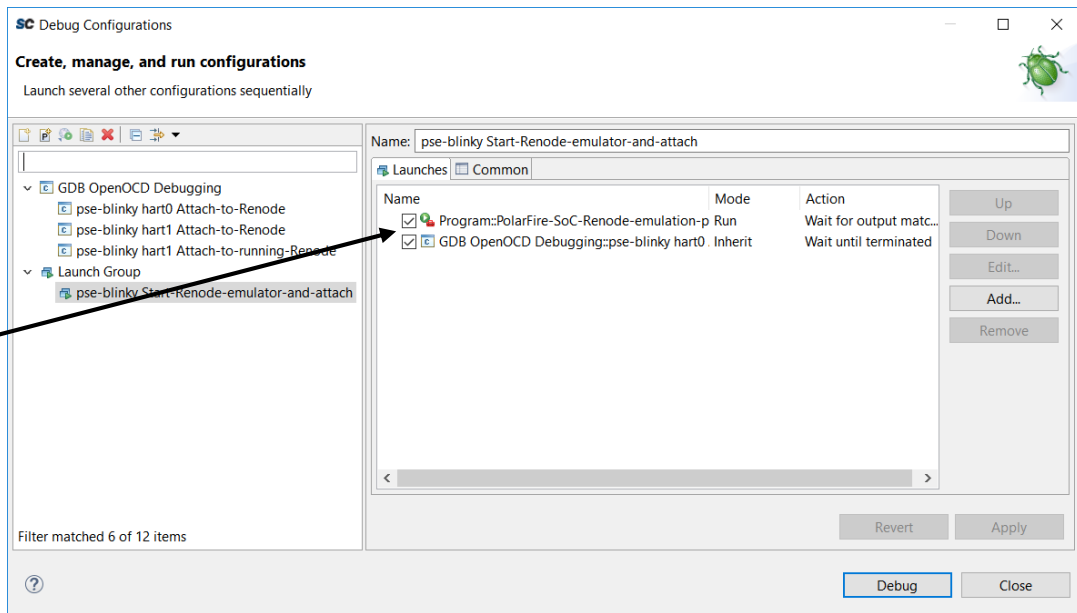
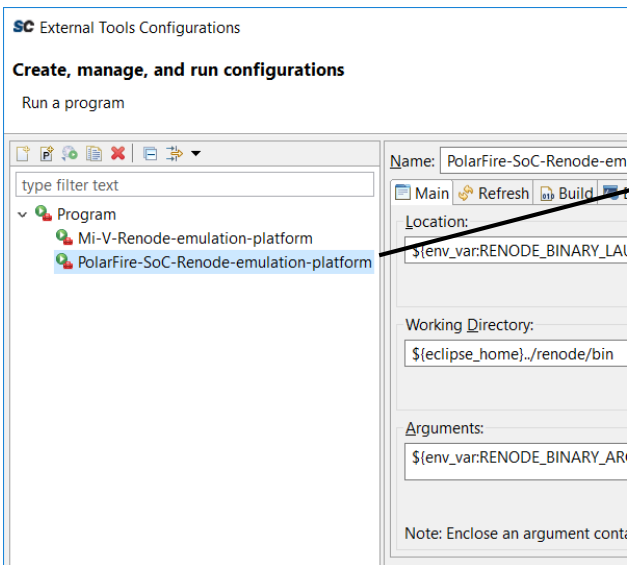
# Configuring debug sessions, external tools and launch groups

- Renode launches and a script is passed as an argument to tell it what platform to load
- It can be launched independently or as part of a launch group



# Configuring debug sessions, external tools and launch groups

- External tool in launch group





# Configuring debug sessions, external tools and launch groups

## Debug configuration for a hardware target

Starts openOCD (1)

OpenOCD acts as GDB server (2)

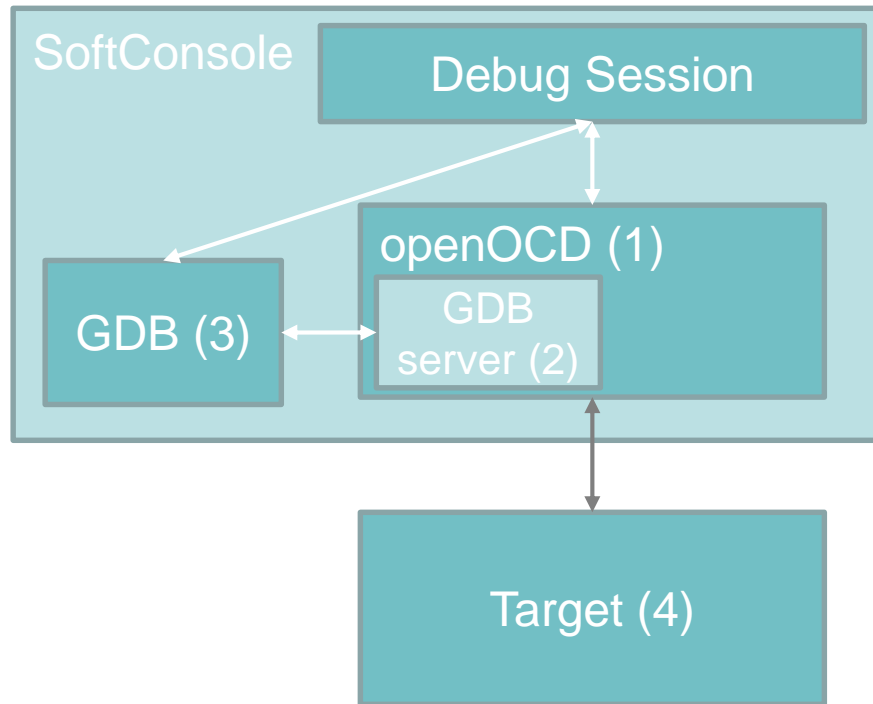
GDB starts (3) and connects (4)

## GDB startup configuration for a hardware target

Symbols and executable loaded from ELF

Breakpoint set on main

Continue selected



# Configuring debug sessions, external tools and launch groups

## Debug configuration for a Renode target

Starts Renode (1)

Renode acts as GDB server (2)

GDB starts (3) and connects (4)

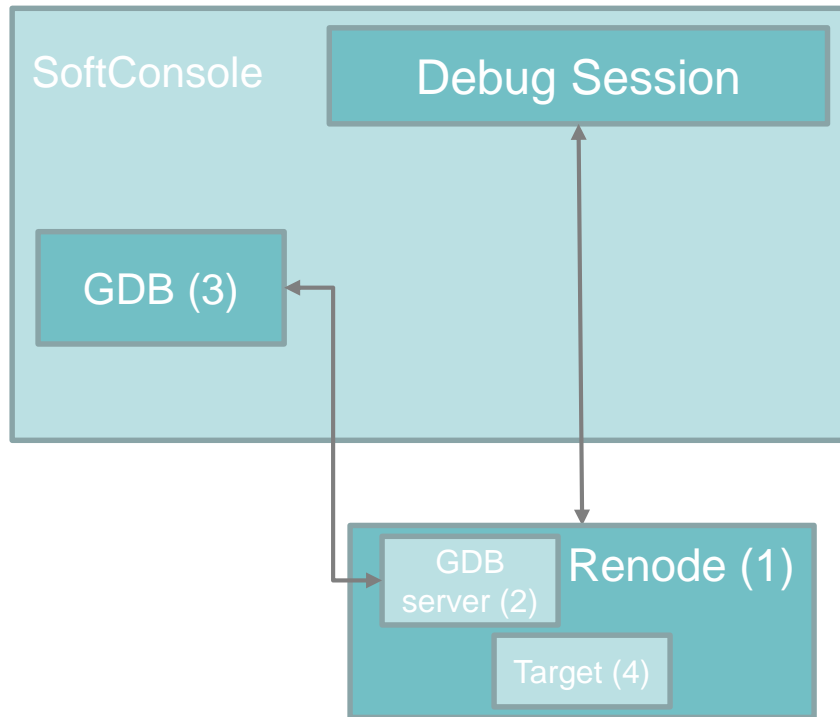
## GDB startup configuration for a Renode target

Initial reset selected

Symbols and executable loaded from ELF

Breakpoint set on main

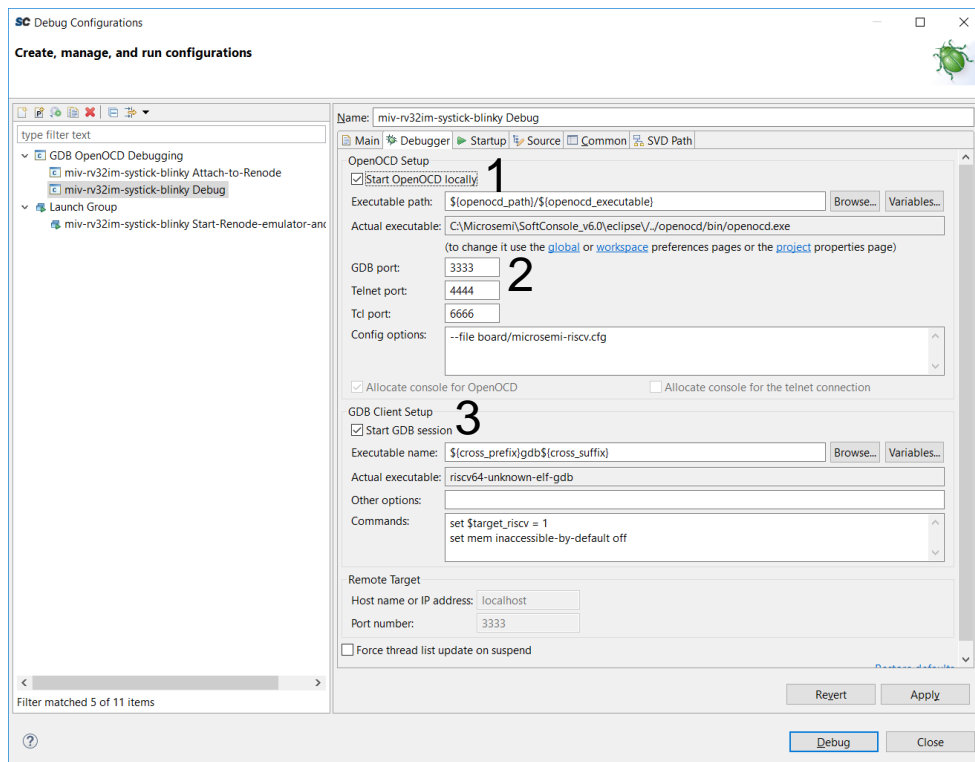
Continue selected



# Configuring debug sessions, external tools and launch groups

## Debug configuration for a hardware target

1. Starts openOCD
2. openOCD creates a port for GDB
3. GDB starts and connects



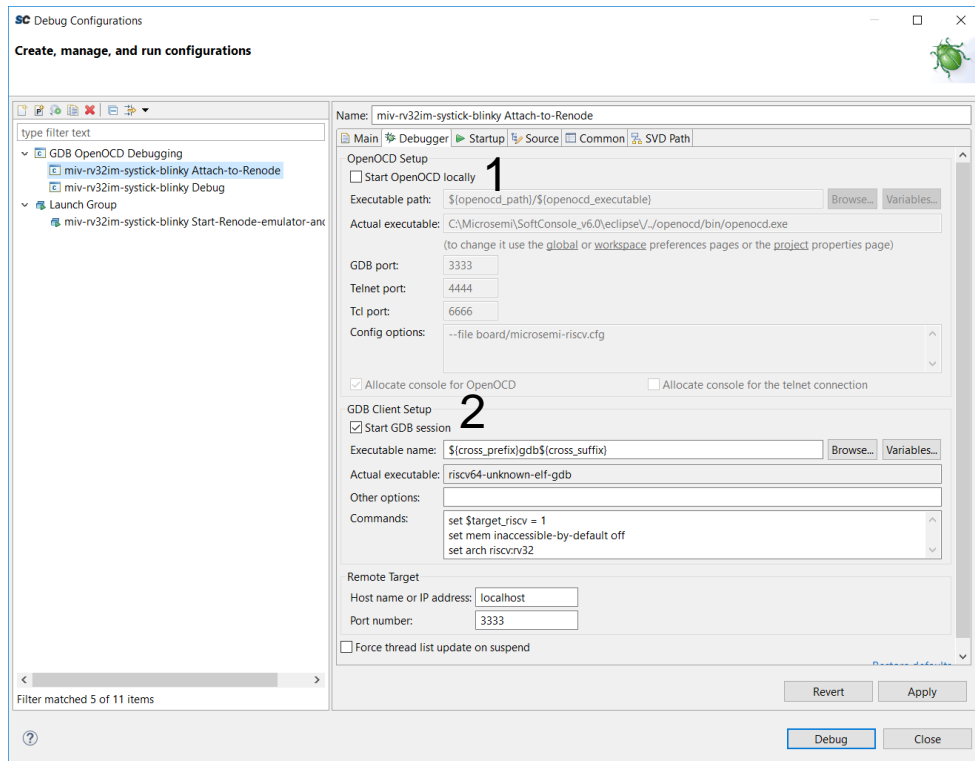
# Configuring debug sessions, external tools and launch groups

## Debug configuration for a Renode target

- Assumes Renode is running and has created a port for GDB

1. openOCD not needed

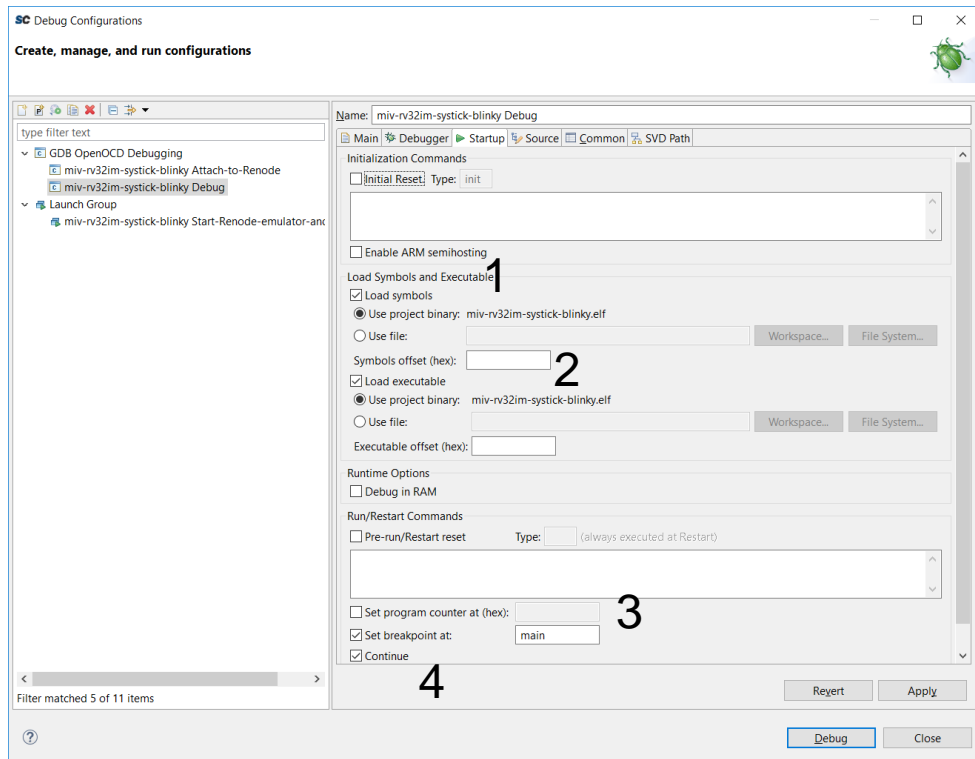
2. GDB starts and connects



# Configuring debug sessions, external tools and launch groups

## Startup configuration for a hardware target

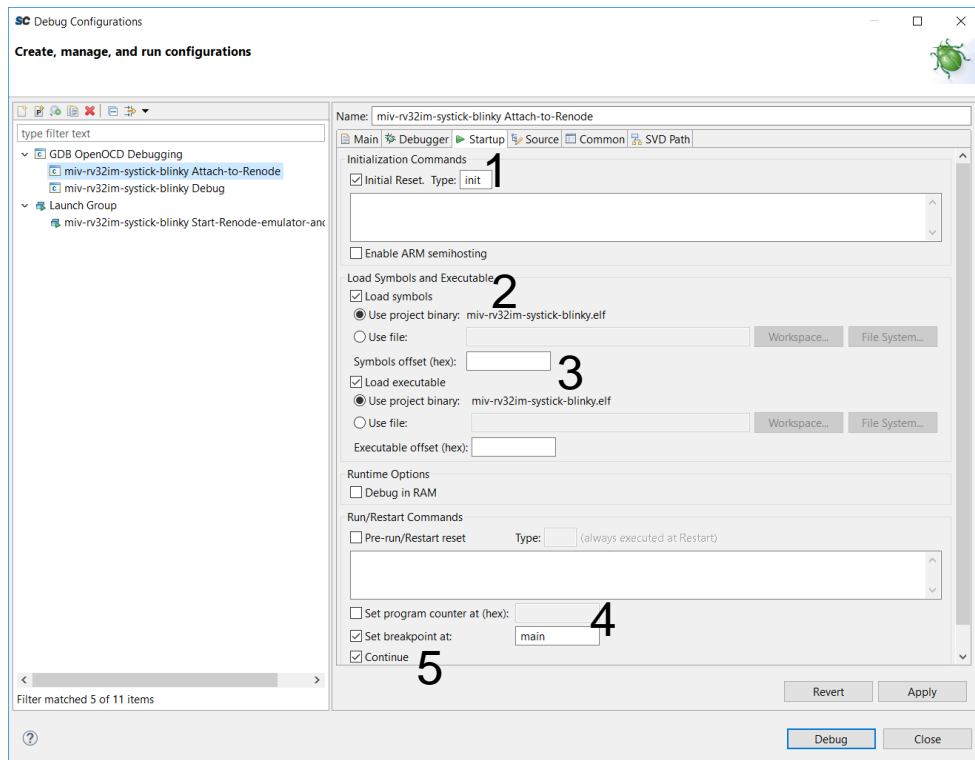
1. Symbols loaded from ELF
2. Executable loaded from ELF
3. Breakpoint set at main
4. Continue selected



# Configuring debug sessions, external tools and launch groups

## Startup configuration for a Renode target

1. Does an initial reset
2. Symbols loaded from ELF
3. Executable loaded from ELF
4. Breakpoint set at main
5. Continue selected

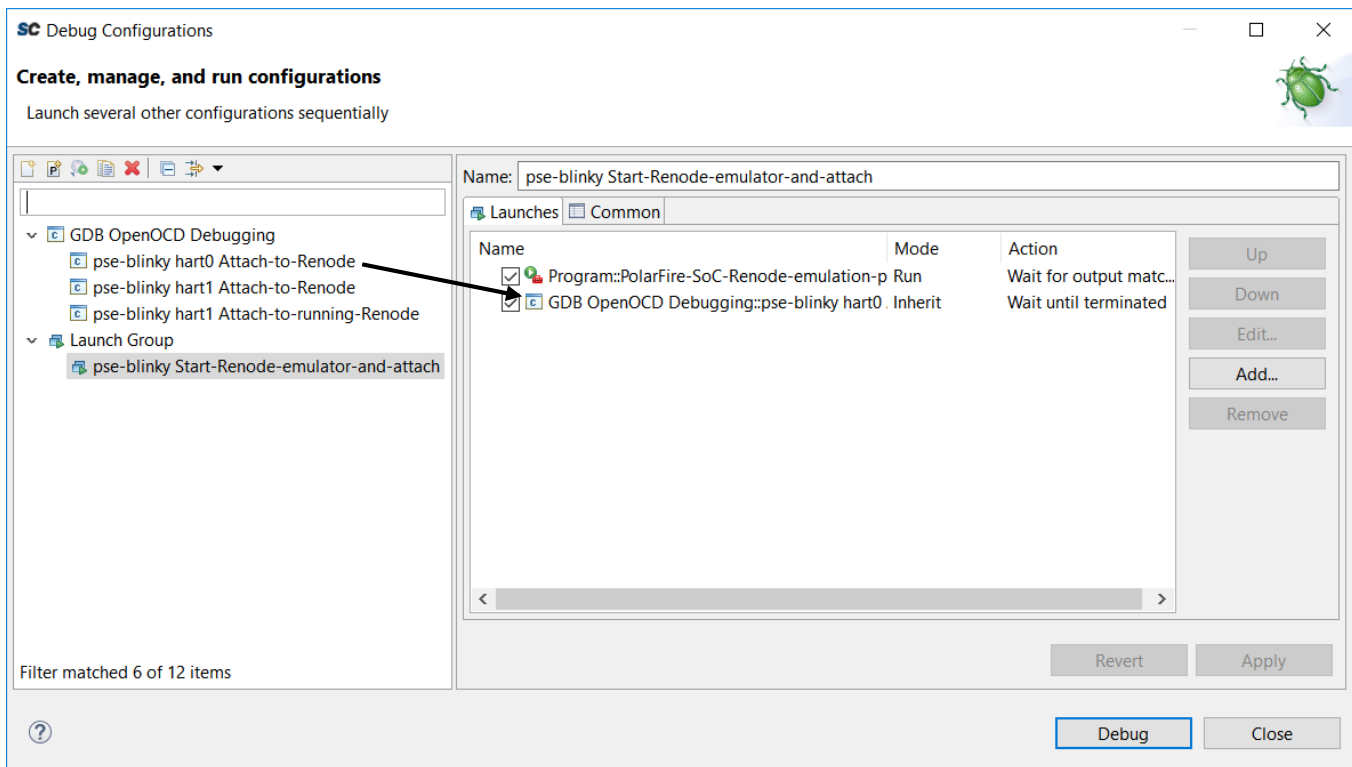


# Configuring debug sessions, external tools and launch groups

<b>Debug configuration for a hardware target</b>	<b>Debug configuration for a Renode target</b>
Start openOCD	Start Renode
OpenOCD acts as GDB server	Renode acts as GDB server
GDB starts and connects	GDB starts and connects

<b>Startup configuration for a hardware target</b>	<b>Startup configuration for a Renode target</b>
	Initial reset selected
Symbols and executable loaded from ELF	Symbols and executable loaded from ELF
Breakpoint set on main	Breakpoint set on main
Continue selected	Continue selected

# Configuring debug sessions, external tools and launch groups



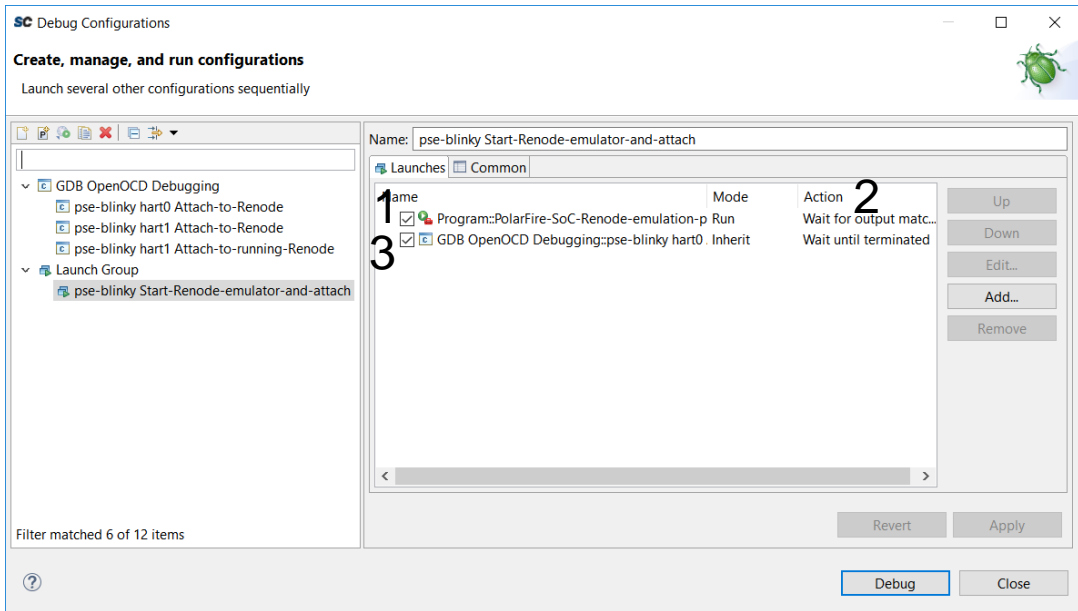


# Configuring debug sessions, external tools and launch groups

- Multiple tools can be launched as part of a launch group

- Flow to use a launch group

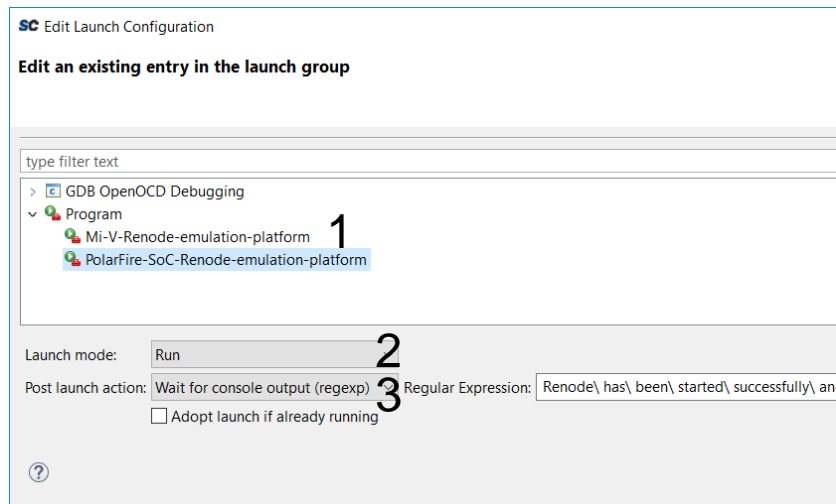
1. Start Renode external tool
2. Wait for expression in console  
"Renode\ has\ been\ started\ successfully\ and\ is\ ready\ for\ a\ gdb\ connection"
3. Start GDB debug session



# Configuring debug sessions, external tools and launch groups

## Adding a Renode platform to a launch group

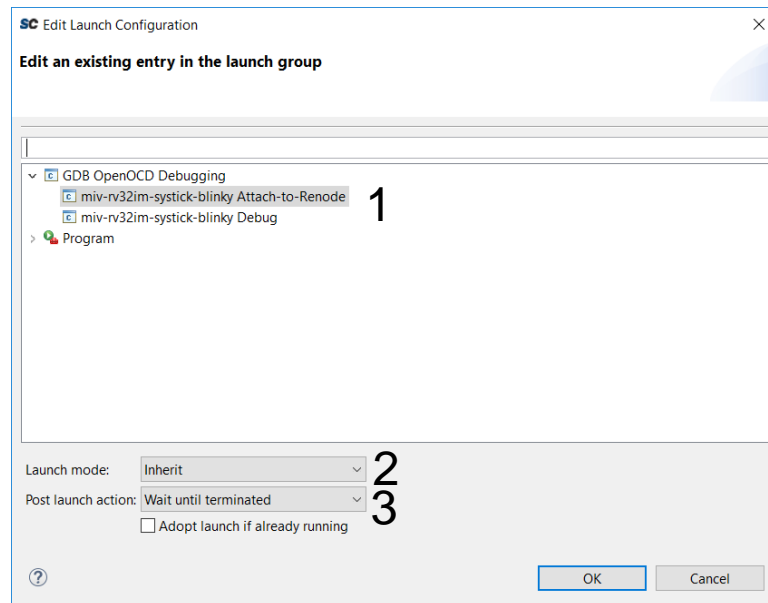
1. Add the Renode platform to launch as a program
2. Launch mode: “Run”
3. Post launch action: “Wait for console output (regexp)”
4. Regular Expression: “Renode\ has\ been\ started\ successfully\ and\ is\ ready\ for\ a\ gdb\ connection”



# Configuring debug sessions, external tools and launch groups

## Adding a debug configuration to a launch group

1. Select the debug session to be run
2. Launch mode: “Inherit”
3. Post launch action: “Wait until terminated”



# Configuring debug sessions, external tools and launch groups

**SC Debug Configurations**

Create, manage, and run configurations

Launch several other configurations sequentially

Filter matched 6 of 12 items

- GDB OpenOCD Debugging
  - pse-blinky hart0 Attach-to-Renode
  - pse-blinky hart1 Attach-to-Renode
  - pse-blinky hart1 Attach-to-running-Renode
- Launch Group
  - pse-blinky Start-Renode-emulator-and-attach

Name: pse-blinky Start-Renode-emulator-and-attach

Name	Mode	Action
<input checked="" type="checkbox"/> Program::PolarFire-SoC-Renode-emulation-p	Run	Wait for output match
<input checked="" type="checkbox"/> GDB OpenOCD Debugging::pse-blinky hart0 . Inherit	Inherit	Wait until terminated

Up Down Edit... Add... Remove

Revert Apply

Debug Close

**SC External Tools Configurations**

Create, manage, and run configurations

Run a program

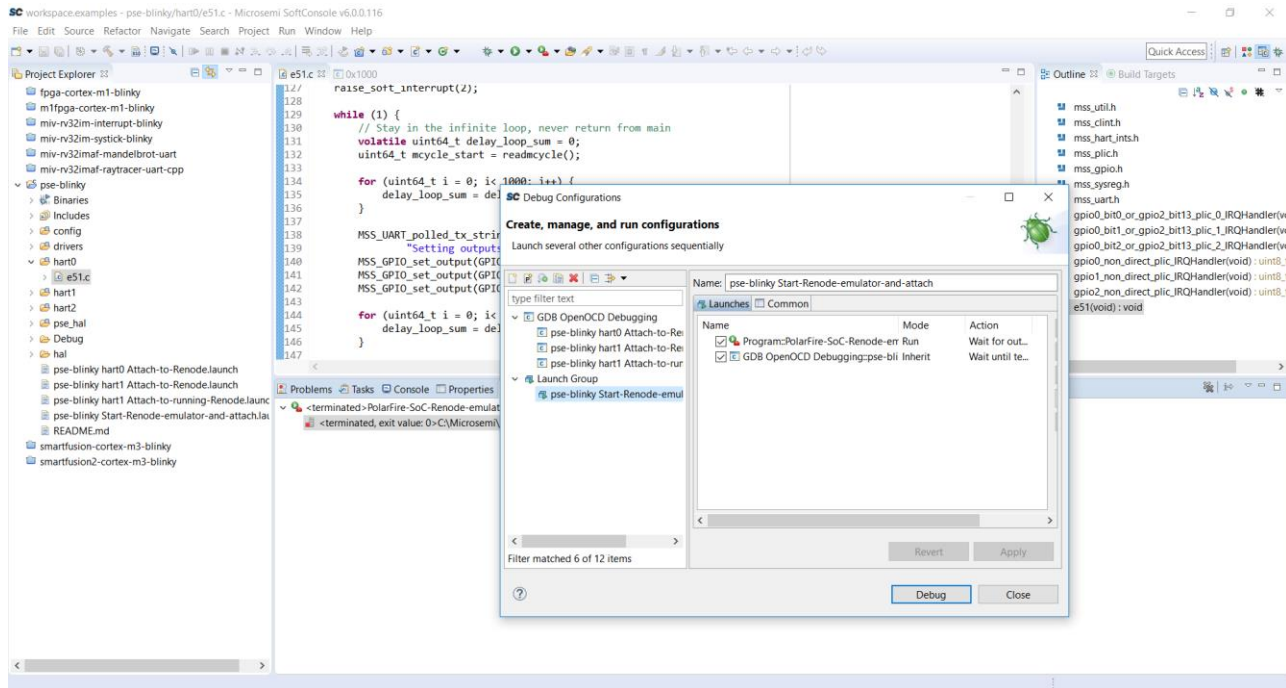
type filter text

- Program
  - Mi-V-Renode-emulation-platform
  - PolarFire-SoC-Renode-emulation-platform

Filter matched 3 of 3 items

# Configuring debug sessions, external tools and launch groups

- All Mi-V example projects have an included pre-configured launch group to use with Renode





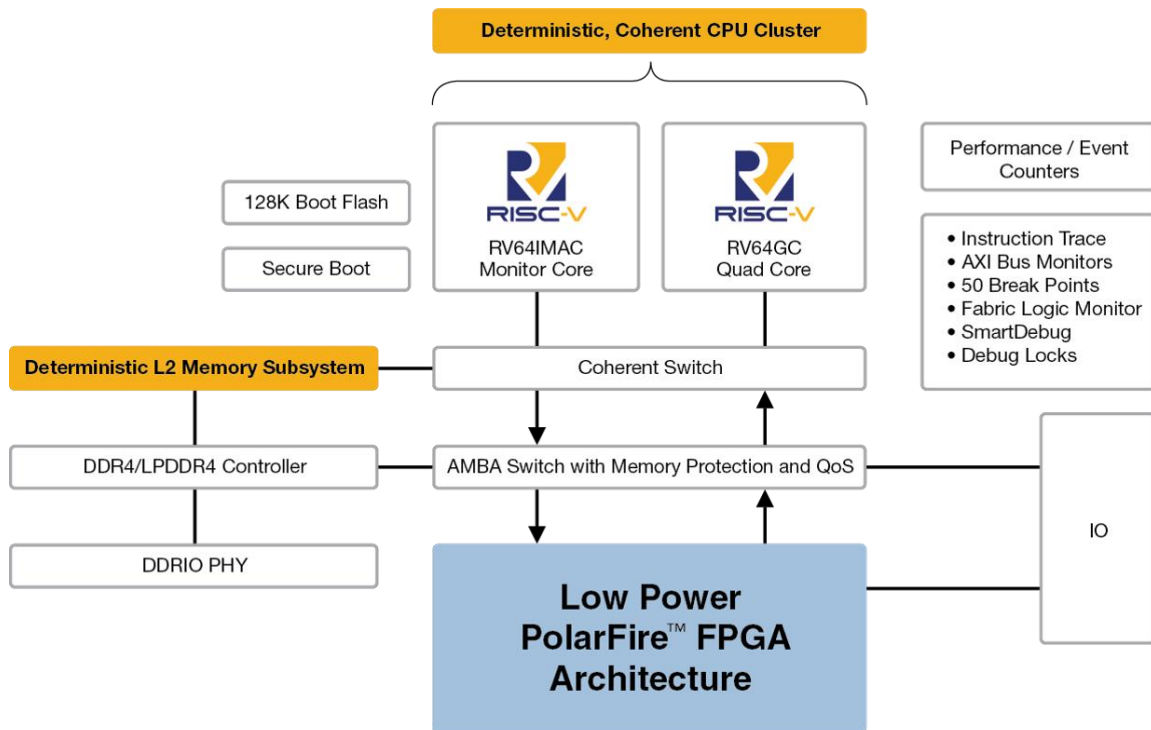
# Learn to Debug a Bare-Metal PolarFire SoC Application with Renode

---

## PolarFire SoC on Renode

# PolarFire SoC on Renode

- RV64IMAC Monitor Core: **E51**
- RV64GC Quad Core: **U54**
  - Quad core: 4 x harts
  - U54\_1, U54\_2, U54\_3, U54\_4
- Memory mapped peripherals and storage
  - GPIO
  - UART
  - CAN
  - MAC
  - DDR
  - ...



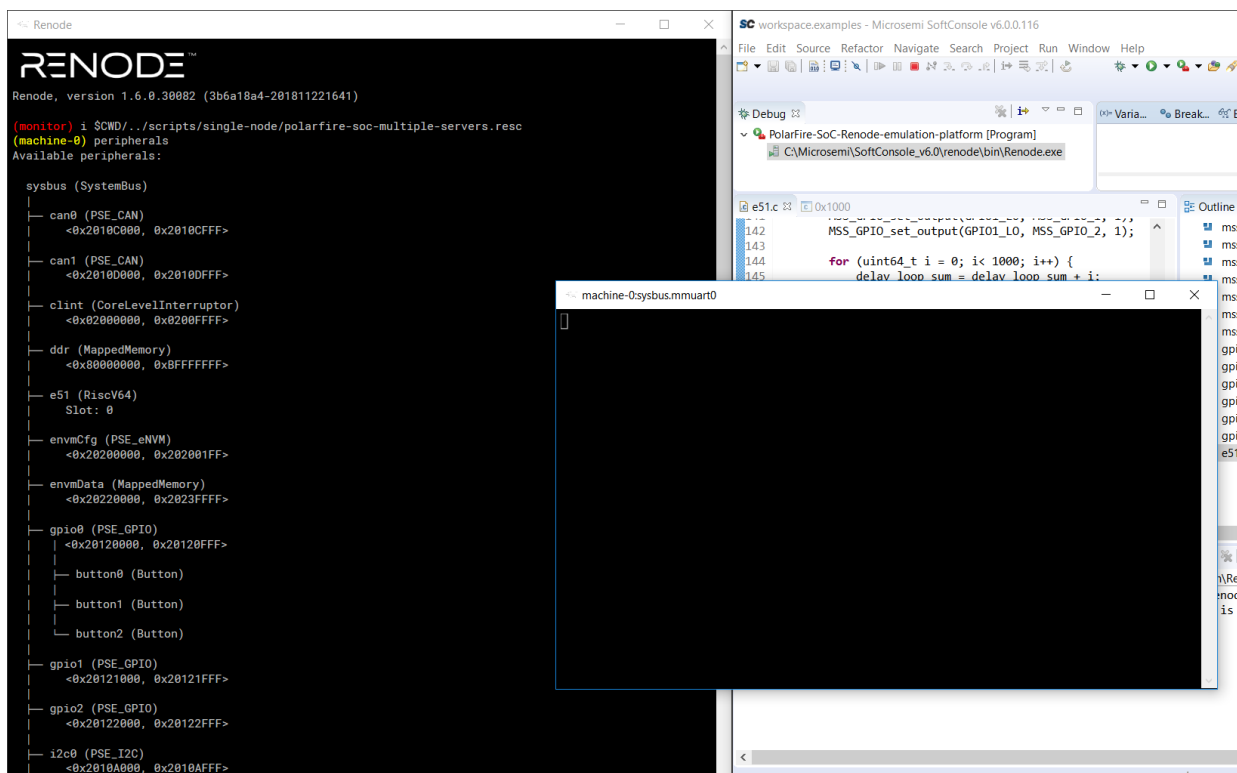
# PolarFire SoC on Renode

- **Renode Console**

- Enter commands here
- See connected peripherals / memory map

- **UART Analyzer**

- Shows UART output for a given instance





# PolarFire SoC on Renode

- **monitor**

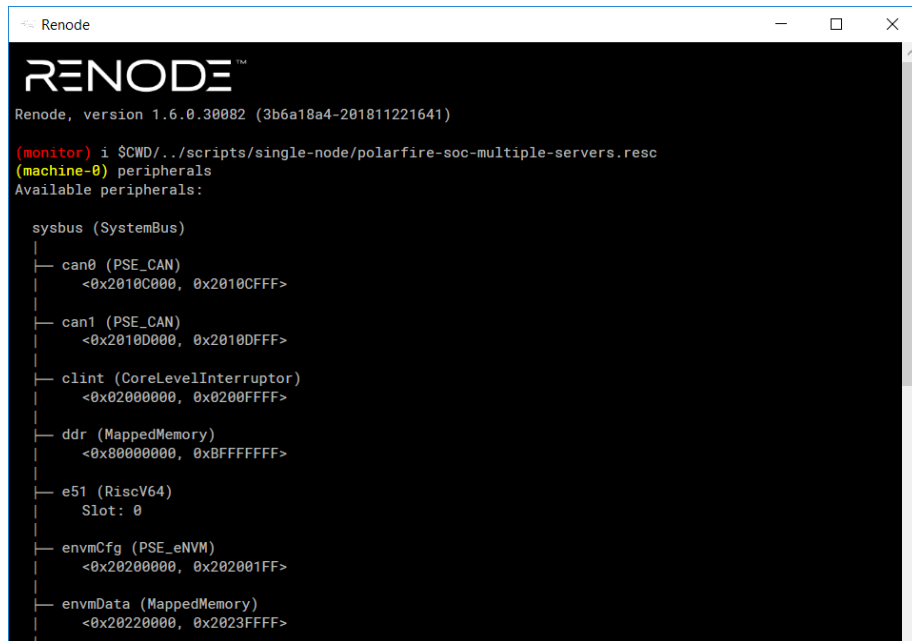
- Initial Renode launch with no system loaded

- **machine-0**

- PolarFire SoC instance
- Each machine has a sysbus (system bus)
- CPUs and peripherals are connected to sysbus

- **Sysbus**

- E51 and U54 harts are connected to sysbus
- Memory mapped peripherals are connected to sysbus



```
Renode
RENODE™
Renode, version 1.6.0.30082 (3b6a18a4-201811221641)

(monitor) i $CWD/./scripts/single-node/polarfire-soc-multiple-servers.resc
(machine-0) peripherals
Available peripherals:

sysbus (SystemBus)
├── can0 (PSE_CAN)
│   └── <0x2010C000, 0x2010CFFF>
├── can1 (PSE_CAN)
│   └── <0x2010D000, 0x2010DFFF>
├── clint (CoreLevelInterruptor)
│   └── <0x02000000, 0x0200FFFF>
├── ddr (MappedMemory)
│   └── <0x80000000, 0xBFFFFFFF>
├── e51 (RiscV64)
│   └── Slot: 0
├── envmCfg (PSE_eNVM)
│   └── <0x20200000, 0x202001FF>
└── envmData (MappedMemory)
    └── <0x20220000, 0x2023FFFF>
```

# PolarFire SoC on Renode

**RENODE™**

Renode, version 1.6.0.30082 (3b6a18a4-201811221641)

(monitor) i \$CWD/./scripts/single-node/polarfire-soc-multiple-servers.resc

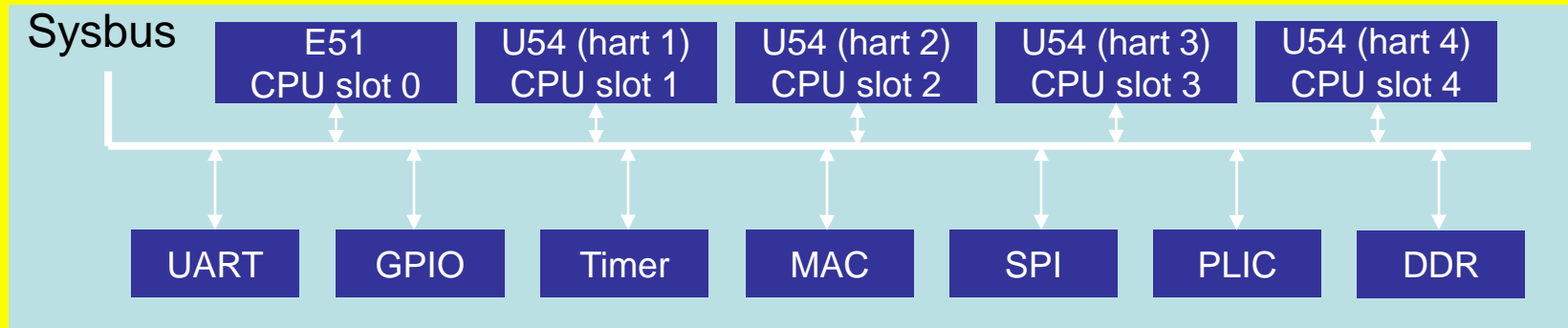
(machine-0) peripherals

Available peripherals:

sysbus (SystemBus)

Renode (monitor)

Machine-x

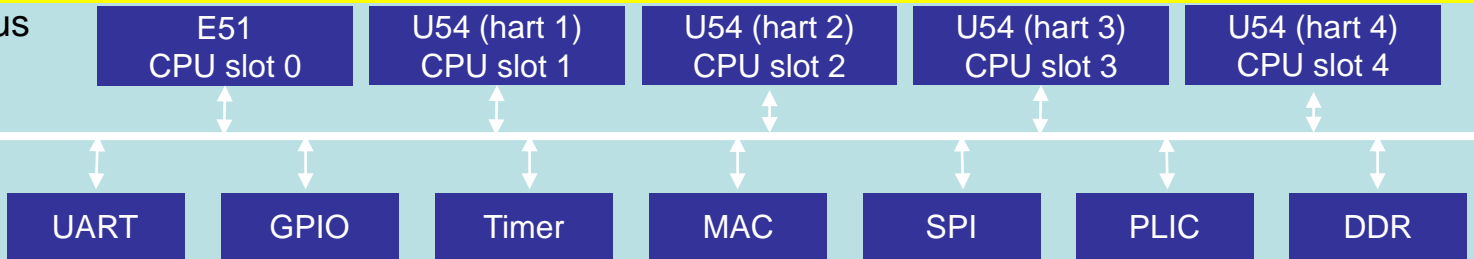


# PolarFire SoC on Renode

Renode (monitor)

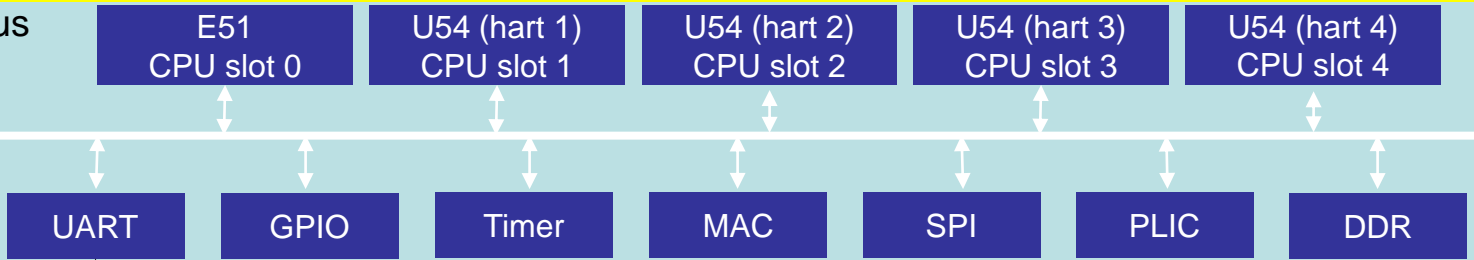
Machine-a

Sysbus



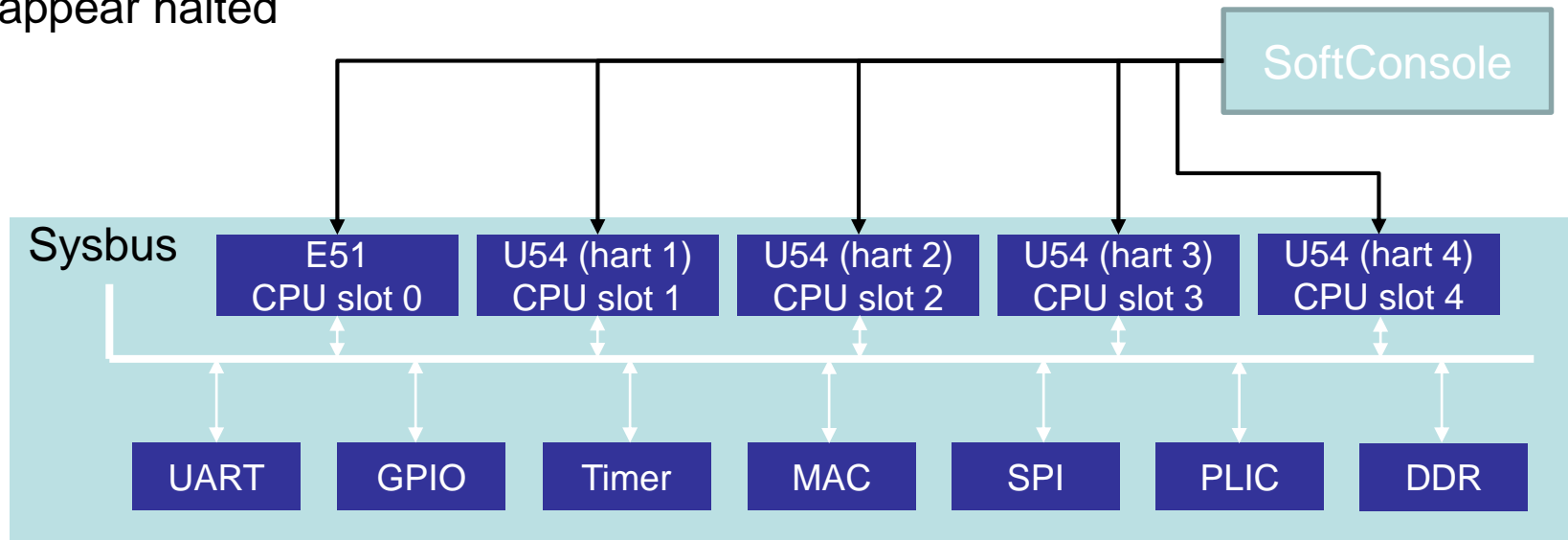
Machine-b

Sysbus



# PolarFire SoC on Renode

- If one hart is being debugged and is suspended – all harts are suspended
- I.E reaching a breakpoint and halting on the E51 will also cause the U54 harts to appear halted





# Learn to Debug a Bare-Metal PolarFire SoC Application with Renode

---

## PSE-BLINKY

# PSE-BLINKY

- Project configured to run on the PolarFire SoC Renode Emulation Platform
- Can debug a single hart or multiple
- Configured
  - Hart 0: E51
  - Hart 1: U54\_1

workspace.examples - pse-blinky/hart0/e51.c - Microsemi SoftConsole v6.0.0.116

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer

- fpga-cortex-m1-blinky
  - m1fpga-cortex-m1-blinky
  - miv-rv32im-interrupt-blinky
  - miv-rv32im-systick-blinky
  - miv-rv32imaf-mandelbrot-uart
  - miv-rv32imaf-raytracer-uart-cpp
- pse-blinky
  - Includes
  - config
  - drivers
  - hart0
    - e51.c
  - hart1
    - u54\_1.c
  - hart2
    - u54\_2.c
  - pse\_hal
    - cpu\_types.h
    - hal\_assert.h
    - hal\_irq.c
    - hal.h
    - hw\_macros.h
    - hw\_reg\_access.h
    - hw\_reg\_access.S
  - pse-blinky hart0 Attach-to-Renode.launch
  - pse-blinky hart1 Attach-to-Renode.launch
  - pse-blinky hart1 Attach-to-running-Renode.launch
  - pse-blinky Start-Renode-emulator-and-attach-launch
  - README.md
- smartfusion-cortex-m3-blinky
- smartfusion2-cortex-m3-blinky

u51.c

```

1 //*****
2 * (c) Copyright 2018 Microsemi-PR
3 *
4 * Code running on E51 hart 0
5 *
6 *****
7 #include "mss_util.h"
8 #include "mss_clint.h"
9 #include "mss_hart_ints.h"
10 #include "mss_plic.h"
11 #include "mss_gpio.h"
12 #include "mss_sysreg.h"
13 #include "mss_uart.h"
14
15 uint8_t gpio0_bit0_or_gpio2_bit13
16 MSS_UART_polled_tx_string(&g_r
17     "\r\nSetting output 0
18
19 MSS_GPIO_set_output(GPIO10_I0,
20 MSS_GPIO_clear_irq(GPIO10_I0,
21 return EXT_IRQ_KEEP_ENABLED;
22
23
24 uint8_t gpio0_bit1_or_gpio2_bit13
25 MSS_UART_polled_tx_string(&g_r
26     "\r\nSetting output 1
27
28 MSS_GPIO_set_output(GPIO10_I0,
29 MSS_GPIO_clear_irq(GPIO10_I0,
30 return EXT_IRQ_KEEP_ENABLED;
31
32
  
```

u54\_1.c

```

1 //*****
2 * (c) Copyright 2018 Microsemi-PR
3 *
4 * Code running on U54 hart 1
5 *
6 *****
7 #include "mss_util.h"
8 #include "mss_clint.h"
9 #include "mss_uart.h"
10
11 volatile uint32_t count_sw_ints_h1
12 volatile uint32_t loop_count_h1 = 0
13 volatile uint64_t dummy_h1 = 0;
14
15 void Software_h1_IRQHandler(void) {
16     uint32_t hart_id = read_csr(mha
17     if (hart_id == 1) {
18         count_sw_ints_h1++;
19     }
20 }
21
22 void u54_1(void) {
23     volatile int i;
24     uint64_t mcycle_start = 0;
25     uint64_t mcycle_end = 0;
26     uint64_t delta_mcycle = 0;
27     uint32_t num_loops = 1000000;
  
```

u54\_2.c

```

1 //*****
2 * (c) Copyright 2018 Micros
3 *
4 * code running on U54 secon
5 *
6 *****
7 #include "mss_util.h"
8
9 volatile uint32_t count_sw_i
10
11 void u54_2(void) {
12     uint32_t hartid = read_c
13     volatile uint32_t i = 0;
14     while (1) {
15         /* add code here */
16         i++; /* added some c
17         if (i == 0x1000) {
18             i = 0;
19         }
20     }
21 }
22
  
```

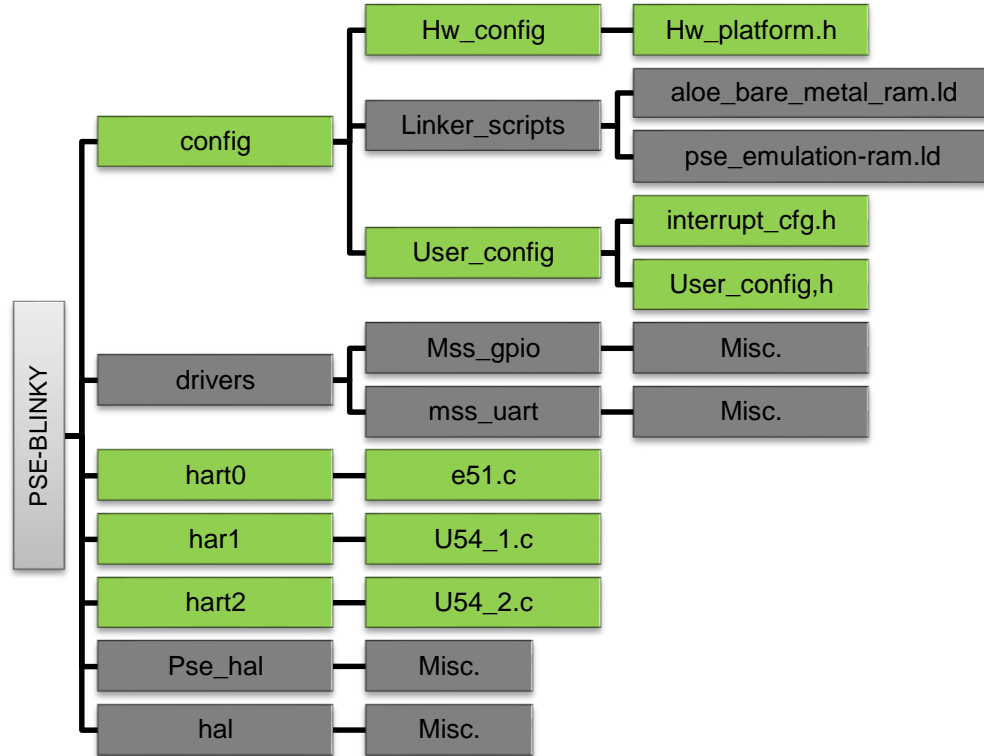
Problems Tasks Console Properties Debug

<terminated>PolarFire-SoC-Renode-emulation-platform [Program]

<terminated, exit value: 0>C:\Microsemi\SoftConsole\_v6.0\renode\bin\Renode.exe

Writable Smart Insert 78:1

# PSE-BLINKY



# PSE-BLINKY

---

- **Build the project**
- **Run the “pse-blinky Start-Renode-emulator-and-attach” launch group**
  - Starts Renode followed by a GDB debug session that connects to Renode
  - Loads ELF
  - Boots to the E51 hart and sets a breakpoint on the E51 main
  - Allow the program to raise two soft interrupts
    - These start U54 hart 0 and hart 1
  - Run the E51 program and allow it to loop
- **Run the “pse-blinky hart1 Attach-to-Renode” debug configuration**
  - This connects to the U54 hart and resets its program counter
  - Sets a breakpoint on U54 main





# Debug a Bare-Metal PolarFire SoC Application with Renode

---

- Demo



# First Thursday's

---

May 2 - Webinar 1: Discover Renode for PolarFire™ SoC Design and Debug

June 6 - Webinar 2: How to Get Started with Renode for PolarFire SoC

July 4 - Webinar 3: Learn to Debug a Bare-Metal PolarFire SoC Application with Renode

**Aug. 1 - Webinar 4: Tips and Tricks for Even Easier PolarFire SoC Debug with Renode**

**Sept. 5 - Webinar 5: Add and Debug PolarFire SoC Peripherals with Renode**

**Oct. 3 - Webinar 6: Intro to PolarFire SoC MSS Configuration and Software Flow**



**Thank You**

