

First Thursdays

- 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: Add and Debug and Pre-Existing Peripheral in PolarFire SoC
- Nov. 7 Webinar 7: How to write custom models filters, offloading, acceleration etc
- Dec. 5 Webinar 8: Handling Binaries

Contd.



Second Thursdays

Jan. 9 - Webinar 9: Run Linux on Renode (PolarFire SoC Model as a Quad-core SMP) – this is not a Linux / Buildroot tutorial

- Feb. 13 Webinar 10: Build applications for Linux on PolarFire SoC
- Mar. 12 Webinar 11: Introduction to PolarFire SoC MSS Configuration and Software Flow
- Apr. 9 Webinar 12: Two baremetal Applications on PolarFire SoC
- May 14 Webinar 13: Linux + Real-Time (AMP Mode) on PolarFire SoC





Getting Started with the RISC-V Based PolarFire[®] SoC FPGA Webinar Series Session 5: "Add and Debug PolarFire SoC Peripherals with Renode"

> Hugh Breslin, Embedded Linux Engineer Thursday Sep. 5, 2019



Supporting Content

→ C △ ≜ https://www.microsemi.c	om/product-directory/fpga-soc/5210-mi-v-embedded-ecosystem#renode-webinar-series	
C Microsemi. 8 S Microcelle com	Search Microsemi.com	
Products & Services Applications	Ordering Company Partners Support	
FPGA & SOC	Libero SoC Design Suite v12.0	
Antifuse FPGAs FPGA & SoC Design Tools FPGA Applications	• 60% nurtime reduction for Timing and 20% runtime reduction for Power • 25% runtime improvement for Place and Route	
FPGA Boards, Kits, and Solutions FPGAs	Home / Products & Services / FPGA & SoC / MI-V RISC-V Ecosystem	
Mi-V RISC-V Ecosystem	Mi-V RISC-V Ecosystem	
 Design Tools Operating Systems RISC-V CPUs 	Overview Mi-V Partners Tutorials Renode Webinar Series Articles and News	
 Rad-Tolerant FPGAs SoC FPGAs Technology Solutions 	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 alonside the linux® creating 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.	



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 Renode¹⁰¹ 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 PolarFire¹⁰² SoC architecture and how to use Renode to develop your application.

www.microsemi.com/Mi-V "Renode Webinar Series"



Add and Debug PolarFire[®] SoC Peripherals with Renode

- The files that make a Renode system
- The launch script
- CPU file
- Board file
- Summary
- Debugging a peripheral



Learn to Debug a Bare-Metal PolarFire[®] SoC Application with Renode

The Files That Make a Renode System



• We have two pre-configured systems:

<u>Mi-V Renode emulation platform (Mi-V system)</u>:

Configured to emulate the Mi-V soft CPUs (RV32G core with UART, GPIOs and Timers)

<u>PolarFire SoC Renode emulation platform (PolarFire SoC system)</u>: Configured to emulate PolarFire Soc and its peripherals



- 3 files that can make up a Renode system
 - Only the script file is passed as an argument when launching Renode
- Script file: Tells Renode to create a machine, load the board file and run commands
- Board file: Tells Renode to load the CPU file and describes the hardware connections for the system (e.g. GPIO connecting to an LED)
- CPU file: Describes the system (e.g. CPU and peripherals)









• Easiest way to deal with the files and folder structure is to create a "Renode" project in SoftConsole







 Platforms: Contains configuration files for platforms (i.e. CPU and board files)

Renode

 Scripts: Contains launch files for different platforms (i.e. launch scripts)





- miv.resc: Launches the Mi-V system
- polarfire-soc-multiple-servers.resc: Launches the PolarFire SoC system

		🗸 🖂 singi
SC External Tools Configurations	– 🗆 X	🗎 co
Create, manage, and run configurations	©	🗎 ef
Run a program		🗎 hi
		🗎 i3
	Name: PolarFire-SoC-Renode-emulation-platform	🗎 lit
type filter text	📄 Main 🤣 Refresh 🚮 Build 🌌 Environment 🔲 Common	🗎 m
v 🤷 Program	Location:	□ m
Hughs_test_platform	\$(env_var:RENODE_BINARY_LAUNCHER)	E
PolarFire-SoC-Renode-emulation-platform	Browse Workspace Browse File System Variables	E 11
♀ renode_system_builder	Working Directory	≣ pi
	(farlinea home) /renode/hin	E po
		🗎 po
	Browse Workspace Browse File System Variables	
	Arguments:	E 44
	\$(env_var:RENODE_BINARY_ARGUMENT)/scripts/single-node/polarfire-soc-multiple-servers.resc	E SI
	v	📄 st
	Variables	🗎 st
		_





✓ ⇒ Renode > 🗁 bin Dicenses v >> platforms 🗸 🗁 boards 🤇 > > > silabs arduino_101-shield.repl colibri-vf61.repl mars zx3-externals.repl mars_zx3.repl miv-board-additional-uarts.repl miv-board.repl guark_c1000-cc2520.repl stm32f4 discovery-additional gpios.repl stm32f4 discovery-bb.repl stm32f4_discovery-kit.repl stm32f4_discovery.repl stm32f7_discovery-bb.repl tegra_externals.repl tegra2.repl

Board files

- PF SoC system: n/a
- Mi-V system: miv-board.repl



- 🗸 🗁 Renode
 - 🗵 🗁 bin
 - > 🗁 licenses
 - ✓ ➢ platforms
 - > 🗁 boards
 - v 🗁 cpus 🧲
 - 💈 🗁 silabs
 - 🖹 a20.repl
 - at91rm9200.repl
 - cc2538.repl
 - 📄 i386.repl
 - litex_vexriscv.repl
 - 🗎 miv.repl
 - mpc5567.repl
 - picosoc.repl
 - polarfire.repl
 - quark-c1000.repl
 - sifive-fe310.repl
 - ····

CPU files

PolarFire SoC system: polarfire.repl

• Mi-V system: miv.repl







The Launch Script



The Launch Script

SC workspace.examples - Renode/scripts/single-node/polarfire-soc-multiple-servers.resc - Microsemi SoftConsole v6.0.0.116 Eile Edit Source Refactor Navigate Search Project Git Run Window Help 🐘 🔨 ㅋ 🔍 🕪 🗉 🗉 그 그 그 또 예 🗁 🌫 수 🌮 🖕 수 🕫 🚸 ㅋ 🗘 ㅋ 🖉 ㅋ 🦉 ㅋ 🌽 🏷 수 ㅋ 수 ㅋ polarfire-soc-multiple-servers.resc XX Project Explorer 🛛 ✓ i Renode 1:name: PolarFire SoC 2:description: This is a sample script prepared to create a PolarFire SoC platform > 🗁 bin Description Description Description 4 logLevel 3 > > > platforms v b scripts 6 using sysbus 1. Create a new machine > > > multi-node 7 mach create > > > pydev 9 machine LoadPlatformDescription @platforms/cpus/polarfire.repl ✓ isingle-node 2. Load the platform 10 machine LoadPlatformDescriptionFromString cc2538.resc 11 """ efr32ma.resc 12 button0: Miscellaneous.Button @ gpio0 hifive_unleashed.resc -> gpio0@0 i386.resc 14 litex vexriscv.resc 15 button1: Miscellaneous.Button @ gpio0 Connect buttons to GPIOs * Could be done in the board file * miv-basic.resc -> gpio0@1 miv.resc 18 button2: Miscellaneous.Button @ gpio0 mpc5567.resc 19 -> gpio0@2 picosoc.resc 20 """ polarfire-soc-multiple-se 21 polarfire.resc 22 logLevel 3 sysbus.e51 guark_c1000.resc 23 logLevel 3 sysbus.u54 1 sifive fe310.resc 24 logLevel 3 sysbus.u54 2 4. Configure logging stm32f103.resc 25 logLevel 3 sysbus.u54 3 26 logLevel 3 sysbus.u54 4 stm32f4 discovery.resc stm32f746.resc 5. Show UART analyser 28 showAnalyzer mmuart0 tegra3.resc 29 versatile.resc 30 e51 StartGdbServer 3333 true vexpress.resc 31 u54 1 StartGdbServer 3334 true 6. Start a GDB server on each hart 32 u54 2 StartGdbServer 3335 true vvbrid.resc 33 u54 3 StartGdbServer 3336 true zedboard.resc 34 u54 4 StartGdbServer 3337 true monitor.py 35 beta ³⁶ log "Renode has been started successfully and is ready for a gdb connection. (This is not an error)" ³ 7. Print message to allow SC to start GDB 🛸 smartfusion-cortex-m3-blinky





ctm22w108 reni

<pre>\$C workspace.examples - Renode/platforms/cpus/polarfire.repl - Microsemi SoftConsole v6.0.0.115 File Edit Source Relacion Navigate Search Project Sit Bun Window Help \$C relation a start of the start of the</pre>			
<pre>Elle Edit Source Refactor Navigate Search Project Git Run Window Help Segendary Search Project Git Run Window Help Project Explorer 3 P</pre>	SC workspace.examples - Renode/platform	ns/cpus/polarfire.repl - Microsemi SoftConsole v6.0.0.116	
<pre>Project Explorer % @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @ @</pre>	<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate	Se <u>a</u> rch <u>P</u> roject <u>G</u> it <u>R</u> un <u>W</u> indow <u>H</u> elp	
<pre>Project Explorer % Project Explorer % Project</pre>	🕼 🔨 🕶 🖎 🕨 II 🔳 3. 73. 18 ->[i	☞ 코 수 윤│ 읖 퀵 椽 ▾ ◑ ▾ ▾ ⊿ ♡ 수 ▾ ↔ ▾	
<pre>> See pointsy > Encode = Encode =</pre>	🍋 Project Explorer 🛛 🕒 🔄 🌫 🗖 🗖	Delarfire.repl	
Picosoc.repl 18 19 u54_3: CPU.RiscV64 @ sysbus 20 cpuType: "rv64gc" 21 hartId: 3 22 privilegeArchitecture: PrivilegeArchitecture.Priv1_10 23 sifive-fe310.repl 24 clint: clint 25 u54_4: CPU.RiscV64 @ sysbus 26 cpuType: "rv64gc" 27 hartId: 4 28 stm32f429.repl 26 cpuType: "rv64gc" 27 hartId: 4 28 privilegeArchitecture: PrivilegeArchitecture.Priv1_10 29 clint: clint	<pre>> Sepe-blinky ^ > Penode > Penode</pre>	<pre>1e51: CPU.RiscV64 @ sysbus 2</pre>	ADD CPU to the system Connecting a CPU: [name]: [path_to_cpu] @ sysbus cpu_Type: "[cpu_type]" hartId: [ID] privilegeArchitecture: PrivilegeArchitecture.[version] clint: clint



SC workspace.examples - Renode/platform	S workspace.examples - Renode/platforms/cpus/polarfire.repl - Microsemi SoftConsole v6.0.0.116				
<u>File E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate S	Se <u>a</u> rch <u>P</u> roject <u>G</u> it <u>R</u> un <u>W</u> indow <u>H</u> elp				
© 🦠 ♥ i 🔌 I 🕨 🗉 ス ଓ .R ♥ i+	> ℤ ♀ ∲ - ₽ ☜ ね ▼ 0 ▼ 9 ▼ 🦀 ▼ 🥬 🖌 🖉 ७ 여 ▼ ↔ ▼				
Project Explorer 🛛 🖻 😫 🔻 🗖 🗖	Delarfire.repl				
 > 🖆 pse-blinky > 🖆 Renode > > bin > > licenses > > platforms > > boards > > cpus > > silabs 	<pre>53 mmuart0: UART.NS16550 @ sysbus 0x2000000 54 wideRegisters: true 55 IRQ -> plic@90 56 57 mmuart1: UART.NS16550 @ sysbus 0x20100000 58 wideRegisters: true 59 IRQ -> plic@91 60 61 mmuart2: UART.NS16550 @ sysbus 0x20102000</pre>	 ADD UART to the system 			
a20.repl at91rm9200.repl cc2538.repl i386.repl i386.repl ilitex_vexriscv.repl	<pre>62 wideRegisters: true 63 IRQ -> plic@92 64 65 mmuart3: UART.NS16550 @ sysbus 0x20104000 66 wideRegisters: true 67 IRO -> plic@93</pre>	Connecting a peripheral: It depends on the peripheral!			
 miv.repl mpc5567.repl picosoc.repl polarfire.repl quark-c1000.repl sifive-fc310.repl sifive-fu540.repl 	68 69 mmuart4: UART.NS16550 @ sysbus 0x20106000 70 wideRegisters: true 71 IRQ -> plic@94 72 73 mmc: SD.PSE_SDController @ sysbus 0x20008000 74 IRQ -> plic@88 75 WidepurEPC -> plic@89	[<mark>name</mark>]: [<mark>path_to_peripheral</mark>] @ sysbus [<mark>address</mark>] [<mark>parameters0</mark>] [parameters1]			
 stm32f103.repl stm32f4.repl stm32f429.repl stm32f746.repl stm32l751.repl 	76 77 spi0: SPI.PSE_SPI @ sysbus 0x20108000 78 IRQ -> plic@54 79 80 spi1: SPI.PSE_SPI @ sysbus 0x20109000 81 IRQ -> plic@55	etc.			



SC workspace.examples - Renode/platform	s/cpus/miv.repl - Microsemi SoftConsole v6.0.0.116	
<u>File Edit Source Refactor Navigate S</u>	Se <u>a</u> rch <u>P</u> roject <u>G</u> it <u>R</u> un <u>W</u> indow <u>H</u> elp	
💿 🍕 🕶 🗽 🍽 🗉 🗉 D. O. R. 🗐 H	> ≳ ♀ ₽ = 4 ね ★ ▼ 0 ▼ 9 ▼ 9 ▼ 8 √ ▼ ⊿ ♥ ♀ → ↓ ♥ ♀ ▼ ↔ ▼	
陷 Project Explorer 🛛 🕒 😫 🔻 🗖 🗖	miv.rept 🕴	
✓ I Renode > ≥> bin	1 flash: Memory.MappedMemory @ sysbus 0x60000000 2 size: 0x40000	1
 > Icenses > platforms > boards 	4 ddr: Memory.MappedMemory @ sysbus 0x80000000 5 size: 0x4000000	2
✓ ⇐ cpus > ⇐ silabs	7 uart: UART.MiV_CoreUART @ sysbus 0x70001000 8 clockFrequency: 66000000	3
iii a20.repl iiii at91rm9200.repl iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	9 10cpu: CPU.RiscV32 @ sysbus 11 cpuType: "rv32g" 12 privilegeArchitecture: PrivilegeArchitecture.Priv1_09 13 clint: clint	4
 Intex_vexnscv.repl miv.repl mpc5567.repl picosoc.repl quark-c1000.repl sifive-fe310.repl sifive-fu540.repl stm32f103.repl stm32f429.repl stm32f429.repl stm32f429.repl stm32f151.repl 	14 15 plic: IRQControllers.PlatformLevelInterruptController @ sysbus 0x40000000 16 [0-3] -> cpu@[8-11] 17 numberOfSources: 31 18 prioritiesEnabled : false 10	5
	<pre>20 // Power/Reset/Clock/Interrupt 21 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 22 frequency: 66000000 23 [0, 1] -> cpu@[3, 7]</pre>	6
	24 25gpioInputs: GPIOPort.MiV_CoreGPIO @ sysbus 0x70002000 26 -> plic@29 27	7
stm32w108.repl	28 gpioOutputs: GPIOPort.MiV_CoreGPIO @ sysbus 0x70005000 29 30 timer0: Timers.MiV_CoreTimer @ sysbus 0x70003000 31	8 Q
i versatile.repl	32 clockFrequency: 6600000	3
 vexpress.repl vybrid.repl zynq-7000.repl 	33 34 timer1: Timers.MiV_CoreTimer @ sysbus 0x70004000 35 -> plic@31	10
	36 ClockFrequency: 66000000	

<pre>clint (CoreLeveInterruptor)</pre>	sysbus (SystemBus) I	
<pre> </pre>	<pre>clint (CoreLevelInterruptor)</pre>	6
- ddr (MappedMemory) 2 - flash (MappedMemory) 1 - ds0000000, 0x6003FFF> 1 - gpioInputs (MiV_CoreGPIO) 7 - user_switch.0 (Button) Address: 0 - user_switch.1 (Button) Address: 1 - user_switch.2 (Button) Address: 2 - gpioOutputs (MiV_CoreGPIO) 8 - user_switch.2 (Button) Address: 3 - user_switch.3 (Button) Address: 1 - user_switch.2 (Button) Address: 3 - led0 (LED) Address: 6 - led1 (LED) Address: 1 - led1 (LED) Address: 1 - work000000, 0x70005000, 0x70005003> 5 - timer0 (Miv_CoreTimer) 9 - timer1 (Miv_CoreTimer) 9 - timer1 (Miv_CoreTimer) 10 - 0x70004000, 0x700040105 3	— cpu (RiscV32) Slot: 0	4
<pre>- flash (MappedMemory)</pre>	├── ddr (MappedMemory) 	2
gpioInputs (MiV_CoreGPI0) 7 - user_switch_0 (Button) Address: 0 - user_switch_1 (Button) Address: 2 - user_switch_2 (Button) Address: 2 - user_switch_2 (Button) Address: 2 - user_switch_2 (Button) Address: 1 - led0 (LED) Address: 1 - led1 (LED) Address: 1 - led1 (LED) Address: 1 - wide00000, 0x43FFFFF> 5 - timer0 (MiV_CoreTimer) 9 - timer1 (MiV_CoreTimer) 10 - warto0004000, 0x70004010> 3	 — flash (MappedMemory) 	1
→ user_switch_0 (Button) Address: 0 → user_switch_1 (Button) Address: 1 → user_switch_2 (Button) Address: 2 → gpioOutputs (Miv_CoreGPIO) Address: 2 → led0 (LED) Address: 1 → led0 (LED) Address: 1 → led1 (LED) Address: 1 → user (Miv_CoreTimer) <0x70004000, 0x70004010> <0x70004000, 0x70004010> uart (Miv_CoreUART) <0x70001000, 0x70001017>	 — gpioInputs (MiV_CoreGPIO) 	7
user_switch_1 (Button) Address: 1 user_switch_2 (Button) Address: 2 gpioOutputs (Miy_CoreGPIO) <pre></pre>	 → user_switch_0 (Button) Address: 0	
user_switch_2 (Button) Address: 2 gpioOutputs (MiV_CoreGPIO) 8 <0x70095000, 0x700050A3> B Address: 0 led0 (LED) Address: 1 plic (PlatformLeveIInterruptController) <0x70000000, 0x43FFFFF> 5 timer0 (MiV_CoreTimer) <0x700040000, 0x7000401B> 0 uart (MiV_CoreUART) <0x70001000, 0x70001017> 3 3		
gpioOutputs (MiV_CoreGPIO) 8 <8x78085808, 8x780858A3> 8 - led8 (LED) - Address: 8 - led1 (LED) Address: 1 - plic (PlatformLevelInterruptController) <8x78080808, 8x7808787FFFF> <8x780803808, 8x78083818> + timer8 (MiV_CoreTimer) <8x780848080, 8x78084818> uart (MiV_CoreUART) <8x78081808, 8x78081817>	└── user_switch_2 (Button) Address: 2	
<pre>led0 (LED) Address: 0 led1 (LED) Address: 1 led1 (LED Address: 1 led1 (LED) Address: 1 led1 (LED Address: 1 led1 (LE</pre>	 — gpioOutputs (MiV_CoreGPIO) 	8
Led1 (LED) Address: 1 plic (PlatformLeveInterruptController) 5 <8x4000000, 8x43FFFFF> 5 + timer0 (MiV_CoreTimer) 9 <8x70004000, 8x7000401B> 10 <8x70004400, 8x7000401B> 30 uart (MiV_CoreUART) <8x70001000, 8x70001017> 3		
plic (PlatformLevelInterruptController) 5 <8x4000000, 0x43FFFFF> 5 timer0 (Miy_CoreTimer) 9 <8x70003000, 0x70004010> 10 <8x70004000, 0x70004010> 3 uart (Miy_CoreUART) 3	│ └── led1 (LED) │ Address: 1	
	├─ plic (PlatformLevelInterruptControll │ <0x400000000, 0x43FFFFFF>	^{er)} 5
Limer1 (MiV_CoreTimer) 10 <0x70004000, 0x70004010> 10 uart (MiV_CoreUART) <0x70001000, 0x70001017> 3	├─ timer0 (MiV_CoreTimer) <0x70003000, 0x7000301B>	9
l uart (MiV_CoreUART) <0x70001000, 0x70001017> 3	└── timer1 (MiV_CoreTimer) └── <0x70004000, 0x7000401B>	10
	 └── uart (MiV_CoreUART) <0x70001000, 0x70001017>	3



SC workspace.examples - Renode/platforms/cpus/miv.repl - Microsemi SoftConsole v6.0.0.116

<u>File Edit Source Refactor Navigate Sea</u>	rch <u>P</u> roject <u>G</u> it <u>R</u> un <u>W</u> indow <u>H</u> elp
💼 🍕 🖬 🗽 💷 💷 🔍 📲 🖬 🗃	े � @ ₽ 41 ₩ ▼ 0 ▼ 9 ▼ 9 ▼ Ø ▼ Ø ▼ / / ♥ ♥ ♥ ♥ ▼
🍋 Project Explorer 🛛 🕒 🗟 🗵 🗖 🗎	∎ miv.repl 🖾
▶ Project Explorer 22 ▶ ● ● ●	<pre>Nivrepl X Iflash: Memory.MappedMemory @ sysbus 0x60000000 Size: 0x40000 dustriation of the sysbus 0x8000000 full the sysbus 0x8000000 full the sysbus 0x8000000 full the sysbus 0x70001000 full the sysbus 0x70002000 full the sysbus 0x70002000 full the sysbus 0x70002000 full the sysbus 0x70002000 full the sysbus 0x70003000 full the sysbus 0x70004000 full the sysbus 0x70004000</pre>
< >	39 clockFrequency: 66000000

Add a second UART

Current UART: uart: UART.MiV_CoreUART @ sysbus 0x70001000 clockFrequency: 66000000

Second UART: uart2: UART.MiV_CoreUART @ sysbus 0x70006000 clockFrequency: 66000000



Consideration and the Description	and an and the second	
 workspace.examples - Kenode/platform Edit Source Refactor Navigate 	ns/cpus/miv.repi - microsemi Soriconsole vo.u.u i to	
e <u>cuit source</u> kerac <u>tor</u> Navigate	segici rioject git run Window Help	
	◎ ∞ ♀ ₽ ♀ ☜ ♀ ♥ ▼ ♥ ► ♥ ∧ ▼ ♪ ♥ ♥ ▼ ♥ ♥	
占 Project Explorer 🛛 🕒 😒 🔍 🖿 🖯	imiv.repl 🛛	
✓ ➢ Renode ^	1 flash: Memory.MappedMemory @ sysbus 0x60000000	1
> 🗁 DIN	3	
Platforms	4 ddr: Memory.MappedMemory @ sysbus 0x80000000	- 2 -
> 🗁 boards	5 size: 0x4000000	-
✓ ➢ cpus	7uart: UART.MiV_CoreUART @ sysbus 0x70001000	3
> 🗁 silabs	8 clockFrequency: 66000000	U
a20.repl	9 10 cmu: CDIL RiscV32 @ syshus	
at91rm9200.repl	11 cpuType: "rv32g"	A
i386 reni	12 privilegeArchitecture: PrivilegeArchitecture.Priv1_09	4
litex vexriscy.repl	13 clint: clint	-
miv.repl	14 15 plic: IROControllers.PlatformLevelInterruptController @ sysbus 0x40000000	_
mpc5567.repl	16 [0-3] -> cpu@[8-11]	5
picosoc.repl	17 numberOfSources: 31	U
polarfire.repl	18 prioritiesenabled : Taise	
i quark-c1000.repl	20// Power/Reset/Clock/Interrupt	
sifive-fu540 repl	21 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000	6
stm32f103.repl	22 frequency: 66000000	U
stm32f4.repl	24	
stm32f429.repl	25 gpioInputs: GPIOPort.MiV_CoreGPIO @ sysbus 0x70002000	7
stm32f746.repl	26 -> plic@29 27	1
stm32I151.repl	28 gpioOutputs: GPIOPort.MiV_CoreGPIO @ sysbus 0x70005000	8
stm32w108.repl	29	0
egra2.repl	30 timer0: limers.MiV_Corelimer @ sysbus 0x70003000	0
versatile.repl	32 clockFrequency: 6600000	Э
vexpress.repl	33	
vybrid.repl	34 timer1: limers.MiV_CoreTimer @ sysbus 0x70004000	10
zynq-7000.repl	36 clockFrequency: 6600000	10
> 🗁 scripts	37	
> 🗁 tests 🗸 🗸	38 uart2: UART.MiV_CoreUART @ sysbus 0x70006000	11
	70 CLOCKT Equency. 0000000	





SC workspace.examples - Renode/platforms/boards/miv-board.repl - Microsemi SoftConsole v6.0.0.116 File Edit Source Refactor Navigate Search Project Git Run Window Help 👘 🔨 🕶 🕵 👟 😵 🗸 🚱 🖕 🚱 🖕 🖓 🗣 🕼 🖓 🗸 🖓 😓 🖓 🗸 陷 Project Explorer 🛛 🕒 😫 🔻 🖻 🗖 imiv-board.repl ¹using "platforms/cpus/miv.repl" 1. Load the CPU file ✓ I⇒ Renode 2 > 🗁 bin 3 gpioOutputs: Description in the second s 2. Connect GPIO outputs to LEDs platforms 1 -> led1@0 boards > > > silabs 7 led0: Miscellaneous.LED @ gpioOutputs 0 arduino 101-shield.repl invert: true 8 3. Create LEDs 9 colibri-vf61.repl 10 led1: Miscellaneous.LED @ gpioOutputs 1 mars zx3-externals.repl 11 invert: true mars_zx3.repl 12 miv-board-additional-ua 13 user switch 0: Miscellaneous.Button @ gpioInputs 0 📄 miv-board.repl invert: true 14 guark c1000-cc2520.repl 15 -> gpioInputs@0 stm32f4 discovery-addit 16 17 user switch 1: Miscellaneous.Button @ gpioInputs stm32f4_discovery-bb.re 4. Create switches invert: true 18 stm32f4_discovery-kit.re 19 -> gpioInputs@1 stm32f4_discovery.repl 20 stm32f7_discovery-bb.re 21 user_switch_2: Miscellaneous.Button @ gpioInputs 2 tegra_externals.repl 22 invert: true -> gpioInputs@2 tegra2.repl 23 tegra3.repl 24 5. Connect switch to GPIO 25 versatile.repl 26 vexpress-externals.repl 27



SC workspace.examples - Renode/platforms/boards/miv-board.repl - Microsemi SoftConsole v6.0.0.116

<u>File Edit Source Refactor Navigate Search Project Git Run Window Help</u>

📓 🔦 🔹 🔌 🗈 🗉 🗉 그는 데 🖬 🤧 🖓 🕼 🖓 🖓 🖓 👙 레 🎋 국 🔘 국 🥵 🖋 국 🌶 🏷 🗢 국 하 국

Project Explorer 🛛 🕞 😵 🍷 🗖	miy rent 23	mix-board repl 22
v 😂 Renode 🧳	1 flash, Marani Marand Marani A sushus 0.60000000	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
> 🗁 bin	1 TTASH: Hemory.Happedremory @ Sysbus 0X00000000	2
> 🗁 licenses	3	3 gnioOutputs:
✓ ➢ platforms	4 ddr: Memory.MappedMemory @ sysbus 0x80000000	4 0 -> led0@0
🗸 🗁 boards	5 size: 0x4000000	5 1 -> led1@0
> 🗁 silabs	6	6
arduino_101-shield.repl	7 uart: UART.MiV_CoreUART @ sysbus 0x70001000	7 led0: Miscellaneous.LED @ gpioOutputs 0
colibri-vf61.repl	8 clockFrequency: 6600000	8 invert: true
mars_zx3-externals.repl	9 10 cpu: CPU RiccV22 @ sychus	10 lod1: Missellaneous LED @ gnioOutputs 1
mars_zx3.repl	11 chullybe: "ry32g"	11 invert: true
miv-board-additional-uarts.repl	12 privilegeArchitecture: PrivilegeArchitecture.Priv1 09	12
miv-board.repl	13 clint: clint	13 user_switch_0: Miscellaneous.Button @ gpioInputs 0
guark c1000-cc2520.repl	14	14 invert: true
stm32f4 discovery-additional gpios.repl	15 plic: IRQControllers.PlatformLevelInterruptController @ sysbus 0x40000000	15 -> gpioInputs@0
stm32f4 discovery-bb.repl	16 [0-3] -> cpu@[8-11]	
stm32f4 discovery-kit.repl	1/ numberUtSources: 31	1/user_switch_1: Miscellaneous.Button @ gpioInputs 1
stm32f4 discovery.repl	19	19 -> gnioTnouts@1
stm32f7 discovery-bb.repl	20 // Power/Reset/Clock/Interrupt	20
tegra externals.repl	21 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000	21 user_switch_2: Miscellaneous.Button @ gpioInputs 2
tegra2.repl	22 frequency: 6600000	22 invert: true
egra3.repl	23 [0, 1] -> cpu@[3, 7]	23 -> gpioInputs@2
ersatile.repl		24
vexpress-externals.repl	25 gp101nputs: GP10Port.M1V_CoreGP10 @ sysbus 0x/0002000	25
express.repl	20 -> piic@29	20
zedboard-externals.repl	28 gpioOutputs: GPIOPort.MiV CoreGPIO @ sysbus 0x70005000	28
zedboard.repl	29	29
	30 timer0: Timers.MiV_CoreTimer @ sysbus 0x70003000	30
> 🔁 silabs	31 -> plic@30	31
a20 reni	32 clockFrequency: 66000000	32
at91rm9200 reni	33 24 timer 1. Timer NiV Constinue & suchus 0.70004000	33
CC2538 repl	34 timeri: iimers.miv_coreiimer @ sysous 0x/0004000	34
 i386 rent 	36 clockErequency: 6600000	36
litev vevrisov reni	37	37
mivren	38 uart2: UART.MiV_CoreUART @ sysbus 0x70006000	38
mnc5567 reni	39 clockFrequency: 66000000	39

sysbus (SystemBus) ├ clint (CoreLevelInterruptor) <0x44000000, 0x4400FFFF> cpu (RiscV32) ddr (MappedMemory) <0x80000000, 0x83FFFFFF> flash (MappedMemory) <0x60000000, 0x6003FFFF> gpioInputs (MiV_CoreGPIO) <0x70002000, 0x700020A3> └── user_switch_0 (Button) Address: 0 — user_switch_1 (Button) Address: 1 — user_switch_2 (Button) Address: 2 <0x70005000, 0x700050A3> - led0 (LED) led1 (LED) plic (PlatformLevelInterruptController) <0x40000000, 0x43FFFFFF> timer0 (MiV_CoreTimer) <0x70003000, 0x7000301B> timer1 (MiV_CoreTimer) <0x70004000, 0x7000401B> uart (MiV_CoreUART) <0x70001000, 0x70001017>



SC workspace.examples - Renode/platforms/boards/miv-board.repl - Microsemi SoftConsole v6.0.0.116

sysbus (SystemBus)
├─ clint (CoreLevelInterruptor)
<0x44000000, 0x4400FFFF>
(Disp)(20)
- cpu (Riscv32)
1 3101. 0
⊢ ddr (MappedMemory)
<0x80000000, 0x83FFFFFF>
├─ flash (MappedMemory)
<0x60000000, 0x6003FFFF>
$=$ gptothputs (MIV_corecpto)
user_switch_0 (Button)
Address: 0
Address: 1
user_switcn_2 (Button)
Add(6551 2
gpioOutputs (MiV_CoreGPIO)
<0x70005000, 0x700050A3>
Address: 0
Led1 (LED)
Address: 1
<pre> plic (PlatformLevelInterruptController) </pre>
<0x40000000, 0x43FFFFFF>
└── timer1 (MiV_CoreTimer)
<0x70004000, 0x7000401B>
└─ uart (MiV_CoreUART)
242/4441444 42/6616175

🖹 miv-board .repl 🛛
<pre>1using "platforms/cpus/miv.repl"</pre>
2
3 gpioOutputs:
4 0 -> Led0@0
$6 2 \rightarrow 1ed2@0$
7
8 led0: Miscellaneous.LED @ gpioOutputs 0
9 invert: true
11 led1: Miscellaneous.LED @ gnioOutputs 1
12 invert: true
13
14 led2: Miscellaneous.LED @ gpioOutputs 2
16
17 user_switch_0: Miscellaneous.Button @ gpioInputs 0
18 invert: true
19 -> gpioInputs@0
20 21user switch 1: Miscellaneous Button @ gnioInputs 1
22 invert: true
23 -> gpioInputs@1
24
25 user_switch_2: Miscellaneous.Button @ gpioinputs 2
27 -> gpioInputs@2
28
29
31
32
33
34
36
37
38
39
40
42

- 0 ×

sysbus	(SystemBus)		
- cli	nt (CoreLeve 0x44000000,	lInterruptor 0x4400FFFF>	
 → cpu	(RiscV32)		
S	lot: 0		
ddr <	(MappedMemo 0x80000000,	ory) 0x83FFFFFF>	
— fla <	sh (MappedMe 0x60000000,	mory) 0x6003FFFF>	
_ gpi < 	oInputs (MiV 0x70002000,	_CoreGPI0) 0x700020A3>	
	user_switch Address:	_0 (Button) 0	
	user_switch Address:	_1 (Button) 1	
	user_switch Address:	_2 (Button) 2	
⊢ gpi <	00utputs (Mi 0x70005000,	V_CoreGPIO) 0x700050A3>	
	led0 (LED) Address:		
	led1 (LED) Address:		
	Address:	2	
	c (PlatformL 0x400000000,	0x43FFFFFF>	tControlle
tim <	er0 (MiV_Cor 0x70003000,	eTimer) 0x7000301B>	
— tim <	er1 (MiV_Cor 0x70004000,	eTimer) 0x7000401B>	
uar var	t (MiV_CoreU 0x70001000,	ART) 0x70001017>	



Summary















Summary

- 1. Launch the Renode external tool configuration
- 2. Renode starts
 - A. Loads the script file passed to it as an argument
 - B. Script creates a machine
 - c. Script tells Renode to load a board file
 - D. Board file tells Renode to load a CPU file
 - E. Renode loads the CPU file
 - F. Renode loads the board file
 - G. Script tells Renode to start the GDB server
 - н. Script tells Renode to print the started message in the console
- 3. SoftConsole starts GDB

<pre>(monitor) i \$CWD//scripts/single-node/miv-basic.resc Available peripherale;</pre>			
sysbus (SystemBus)	_aunch Script		
 clint (CorelevelInterruntor)			
<0x44000000, 0x4400FFFF>	CPU File		
cpu (RiscV32) Slot: θ			
ddr (MappedMemory) <0x80000000, 0x83FFFFF>			
flash (MappedMemory) <0x60000000, 0x6003FFFF>			
gpioInputs (MiV_CoreGPIO)	_		
	Board File		
user_switch_1 (Button) Address: 1			
user_switch_2 (Button) Address: 2			
☐ gpioOutputs (MiV_CoreGPIO) <0x70005000, 0x700050A3>	_		
led0 (LED) Address: 0			
⊨ led1 (LED) Address: 1			
led2 (LED) Address: 2			



Summary: Available Peripherals Mi-V system

Memory	Cores	Peripherals
DDR	RV32G	CoreUART
Flash		PLIC
		CLINT
		CoreGPIO
		CoreTimer



Summary: Available Peripherals PolarFire SoC system

Cores	Memory	Peripherals	Peripherals contd.
RV64IMAC	DDR	CLINT	Cadence GEM (MAC)
RV64GC	Flash	PLIC	PSE GPIO
	PSE eNVM	NS16550 UART	PSE RTC
		PSE SD Controller	PSE Timer
		PSE SPI	PSE USB
		PSE I2C	PSE PCI
		PSE CAN	





🔞 🍕 🔺 🗽 🗈 🗉 🗉 🗵 👁 12 🗐 🗗	32 수 윤 음 성 袮 ▾ O ▾ O ▾ 🖉 🖋 ▾ 🥖 🏷 수 ▾ ୦ ▾	Quick J
💫 Project Explorer 🛛 🗖 🗖	Projugal (1)	Projectored real 9
₽ 😵 🗸		a hiv-board.tepi ⊗
v 😂 Renode 📃 \land	1 dar: memory.mappedmemory @ sysbus 0x80000000	1 using "platforms/cpus/miv.repl"
> 🗁 bin	2 5126. 0X4000000	
> 🗁 licenses	4 uart: UART.MiV CoreUART @ sysbus 0x70001000	4
✓ ▷ platforms	5 clockFrequency: 6600000	5
boards	6	6 Renode
> 🗁 silabs	7 cpu: CPU.RiscV32 @ sysbus	7
arduino_101-shield.rep	8 cpuType: "rv32g"	
colibri-vf61.repl	9 privilegeArchitecture: PrivilegeArchitecture.Priv1_09	
mars zx3-externals.rep	10 CHINE CHINE	
mars zx3.repl	12 clint: TROControllers.CorelevelInterruptor @ sysbus 0x44000000	Renode, version 1.6.0.30082 (3b6a18a4-201811221641)
miv-board_repl	13 frequency: 66000000	13
miv-board-additional-	14 [0, 1] -> cpu@[3, 7]	¹⁴ (monitor) i \$CWD//scripts/single-node/miv-basic.
miv-board.repl	15	¹⁵ Available peripherals:
guark c1000-cc2520.re	16	
stm32f4 discovery-add	1/	1/ 19 avabua (SvatamBua)
stm32f4 discovery-bb.	10	19 I Systembus (
stm32f4_discovery-kit.	20	
stm32f4_discovery.repl	21	\sim
stm32f7 discovery-bb.	22	22 <0x44000000, 0x4400FFFF>
tegra_externals.repl	23	23
tegra2.repl	24	24 ⊢ cpu (RiscV32)
tegra3.repl	23	25 26 Slot: 0
versatile.repl	27	27
vexpress-externals.repl	28	28 ├── ddr (MappedMemory)
vexpress.repl	29	29 <0x80000000, 0x83FFFFFF>
zedboard-externals.re;	30	30
zedboard.repl	31	Jan └── uart (MiV_CoreUART)
🗸 🗁 cpus	32	<0x70001000, 0x70001017>
> 🗁 silabs	34	34
a20.repl	35	35 (Mi-V)
at91rm9200.repl	36	36
cc2538.repl	37	37
i386.repl	38	38
litex_vexriscv.repl	39	39
miv.repl	40	40
mpc5567 repl	41	41







- There was an error executing command 'machine LoadPlatformDescription C:\Microsemi\SoftConsole_v6.0.0.0\renode\platforms\boards\miv-board.repl'
- 2. Error E25: Could not find suitable constructor for type 'Antmicro.Renode.Peripherals.Timers.MiV_CoreTimer'.
 - A. Constructor selection report:
 - B. Considering ctor Antmicro.Renode.Peripherals.Timers.MiV_CoreTimer with the following parameters: [Antmicro.Renode.Core.Machine, System.Int64].
 - c. Parameter 'machine' of type 'Antmicro.Renode.Core.Machine' filled with default value = 'Mi-V'.
 - D. Could not find corresponding attribute for parameter 'clockFrequency' of type 'System.Int64' and it is not a default parameter. Rejecting constructor.
 - E. At C:\Microsemi\SoftConsole_v6.0.0.0\renode\platforms\cpus\miv.repl:17:9:
 - F. timer0: Timers.MiV_CoreTimer @ sysbus 0x70003000

📓 🔦 ㅋ 🔍 🗈 🗉 🗉 그는 데 🗄 😥 🗇 🌮 🚔 🖓 ㅋ 🖉 🎋 ㅋ 🖉 ㅋ 🥵 🛷 ㅋ 🍠 🏷 ㅋ ㅎ ㅋ

SC workspace.examples - Renode/platforms/cpus/miv_repl - Microsemi SoftConsole v6.0.0.116

<u>File Edit Source Refactor Navigate Search Project Git Run Window Help</u>

Project Explorer 🛛 🖹 miv .repl 🛛 📄 miv.repl 🛛 🖻 🔩 1 ddr: Memory.MappedMemory @ sysbus 0x80000000 1 ddr: Memory.MappedMemory @ sysbus 0x80000000 ✓ ⇒ Renode size: 0x4000000 size: 0x4000000 2 2 > 🗁 bin Dicenses 4 uart: UART.MiV CoreUART @ sysbus 0x70001000 4 uart: UART.MiV CoreUART @ sysbus 0x70001000 clockFrequency: 66000000 clockFrequency: 66000000 v blatforms 5 6 > > > boards 7 cpu: CPU.RiscV32 @ sysbus 7 cpu: CPU.RiscV32 @ sysbus 🗸 🗁 cpus cpuTvpe: "rv32g" cpuTvpe: "rv32g" 8 > > > silabs privilegeArchitecture: PrivilegeArchitecture.Priv1 09 privilegeArchitecture: PrivilegeArchitecture.Priv1 09 9 9 a20.repl 10 clint: clint 10 clint: clint at91rm9200.repl 11 11 cc2538.repl 12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 📄 i386.repl 13 frequency: 66000000 13 frequency: 66000000 14 [0, 1] -> cpu@[3, 7] 14 [0, 1] -> cpu@[3, 7] litex vexriscv.repl 15 miv_.repl 16 timer0: Timers.MiV CoreTimer @ sysbus 0x70003000 16 timer0: Timers.MiV CoreTimer @ sysbus 0x70003000 miv.repl 17 17 clockFrequency: 66000000 mpc5567.repl 18 18

Quick Access 🔡 🔡 躍



[X] ▶ 01 ≡ 3. ③ .k ⊲] [⇒ 🕱 ♀ ₽] ⊕ 43 卷 ▼ 0 ▼ 💁 ♥ 🤌 ▼ 🥖 ♥ ♀	▼	Quick Access 🕴 😰 🖡
t Explorer 🛛 📃 📄 miv.repl 😒	i miv-board.repl ≅	
Node Imixrepi I3 Node iddr: Memory.MappedMemory @ sysbus 0x8000 2 size: 0x4000000 3 duart: UART.Miv_CoreUART @ sysbus 0x70001 > boards clockFrequency: 66000000 > boards 6 > boards 7 cpu: CPU.RiscV32 @ sysbus mars_zx3repl 8 mars_zx3repl 12 clint: clint mars_zx3repl 12 clint: IRQControllers.CoreLevelInterrupt miv-board_repl 13 miv-board_additional- 16 stm32t4_discovery-ktu 20 zedboard-externals.repl 23 vexpress-externals.repl 23 zedboard-externals.repl 24 zedboard-externals.repl 23 zedboard-externals.repl 24 zedboard-externals.repl	imit-board.rep! 83 1using "platfo 3 4 5 6 7 8 9 9 10 11 12 13 16 17 18 19 20 21 22 23 24 25 26 27 28 31 32 33 34 35 36 37 38 39 40	<pre>rms/cpus/miv.repl" Renode RECOODE Ode, version 1.6.0.30082 (3b6a18a4-201811221641) nitor) i \$CWD//scripts/single-node/miv-basic.resc ilable peripherals: ysbus (SystemBus) - clint (CoreLevelInterruptor) <0x44000000, 0x4400FFFF> - cpu (RiscV32) Slot: 0 - ddr (MappedMemory) <0x80000000, 0x83FFFFFF> - timer0 (Miv_CoreTimer) <0x70003000, 0x7000301B> - uart (Miv_CoreUART) <0x70001000, 0x70001017> </pre>



SC workspace.examples - Renode/platforms/board	ls/miv-board.repl - Microsemi SoftConsole v6.0.0.116	- 0
Eile Edit Source Refactor Navigate Search	Project <u>Git Run Window H</u> elp	
	◇ ∅ ♀ ℓ3 ね ▼ ♥ ▼ ♥ ▼ Ø ⋪ ▼ Ø や ▼ ◇ ▼ \	Quick Access [:] 🖻
v ≥ Renode	E mivrepl ⊠ 1ddp: Managy ManagdManagy € systems 9x20202020	inv-board.repl ⊗
> 🔄 bin > 🔁 licenses > 🗠 platforms	2 size: 0x4000000 3 4uart: UART.MiV_CoreUART @ sysbus 0x70001000 5 clockErgenery: 6600000	2 7 Renode X
bails bails arduino.101-shield.repl collbri-vf61.repl mars_zx3-externals.repl mars_zx3-ardenals.repl mars_zx3-ardenals.repl	6 6 7 cpu: CPU.RiscV32 @ sysbus 8 cpuType: "rv32g" 9 privilegeArchitecture: PrivilegeArchitecture.Priv1_09 10 clint: clint	Renode, version 1.6.0.30082 (3b6a18a4-201811221641)
 miv-board_repl miv-board-additional-uarts.repl miv-board.repl quark_c1000-cc2520.repl 	12 clint: IRQControllers.CorelevelInterruptor @ sysbus 0x44000000 13 frequency: 66000000 14 [0, 1] → cpu@[3, 7] 15 16 timer@: Timers MiV CoreTimer @ sysbus 0x70001000	(monitor) i \$CWD//scripts/single-node/miv-basic.resc There was an error executing command 'machine LoadPlatformDescription C:\Mic recemi)SoffConcele v6 0 0 0 remode)alatforms\beards\miv.beard repl'
istm32f4_discovery-additional_gp istm32f4_discovery-bbrepl istm32f4_discovery-kitrepl istm32f4_discovery-repl istm32f4_discovery-repl istm32f7_discovery-bbrepl istm32f7_discovery-bbrepl istgra_externals.repl	17 clockFrequency: 66000000 18 19 20 21 22 22	Error E39: Exception was thrown during registration of 'timer0 in 'sysbus': Could not register Given address <0x70001000, 0x7000101B> for peripheral Ant micro.Renode.Peripherals.Timers.MiV_CoreTimer conflicts with address <0x7000 1000, 0x70001017> of peripheral Antmicro.Renode.Peripherals.UART.MiV_CoreUAR
tegra2.repl tegra3.repl tegra3.repl tegra3.repl tegra3.repl tegra5.repl tegra5.repl tegra5.repl tegra5.repl	23 24 25 26 27 28	At C:\Microsemi\SoftConsole_v6.0.0.0\renode\platforms\cpus\miv.repl:17:32: timer0: Timers.MiV_CoreTimer @ sysbus 0x70001000
 i zedboard-externals.repl ii zedboard.repl ✓ i pus > i pus > i albs > a20 repl 	29 30 31 32 33	
adv.repi ad	34 35 36 37 38 38	
mpc5567.repl	40 41	40 41



- There was an error executing command 'machine LoadPlatformDescription C:\Microsemi\SoftConsole_v6.0.0.0\renode\platforms\boards\miv-board.repl'
- 2. Error E39: Exception was thrown during registration of 'timer0 in 'sysbus':
 - 1. Could not register Given address <0x70001000, 0x7000101B> for peripheral Antmicro.Renode.Peripherals.Timers.MiV_CoreTimer conflicts with address <0x70001000, 0x70001017> of peripheral Antmicro.Renode.Peripherals.UART.MiV_CoreUART in address.
 - 2. At C:\Microsemi\SoftConsole_v6.0.0.0\renode\platforms\cpus\miv.repl:17:32:
 - 3. timer0: Timers.MiV_CoreTimer @ sysbus 0x70001000

SC workspace.examples - Renode/platforms/cpus/miv.repl - Microsemi SoftConsole v6.0.0.116

 $\underline{F}ile \quad \underline{E}dit \quad \underline{S}ource \quad Refact\underline{t}or \quad \underline{N}avigate \quad Se\underline{a}rch \quad \underline{P}roject \quad \underline{G}it \quad \underline{R}un \quad \underline{W}indow \quad \underline{H}elp$

[6] 🔦 ▾ ! ♥ | ▶ = = 3. ۞ .요 팩| 바 🕱 수 윤| 음 팩 🕸 ▾ 🔾 ▾ 🗛 ▾ 🦉 ▾ 🍠 🛷 ▾ 스 ▾

Imivrepl ≅ Imivrepl ≅ Imivrepl ≅ Imivrepl ≅ Iddr: Memory.MapedMemory @ sysbus 0x8000000 Size: 0x4000000 Iddr: Memory.MapedMemory @ sysbus 0x8000000 Size: 0x4000000 Iddr: Memory.MapedMemory @ sysbus 0x8000000 Iddr: Memory.MapedMemory @ sysbus 0x8000000 Iddr: Memory.MapedMemory @ sysbus 0x70001000	Project Explorer 32			
<pre>> S miv-rv32imaf-raytracer-uart-cpp > S miv-rv32imaf-raytracer-uart-cpp > S pse-blinky > S pse-blinky = spe-blinky =</pre>		🖹 miv.repl 🛛	📄 miv_repl 🛛	
 > is pse-blinky > is pse-bl	> 😂 miv-rv32imaf-raytracer-uart-cpp 🔷	1 ddg: Mamony Mannad Mamony & syshus @v90000000	1 ddp: Mamany ManadMamany & syshus Av8000000	
 > ≥ Renode > ≥ bin > ≥ bin > ≥ bin > ≥ bin > ≤ clockFrequency: 6600000 > ≥ boards > ≥ boards > ≥ cpus > privilegeArchitecture: PrivilegeArchitecture.Priv1_09 > ≥ silbs > clockFrequency: 6600000 > ≥ silbs > clint: clint > a20repl 11 2 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 2 clint: clint 3 frequency: 66000000 3 clint: clint 3 frequency: 66000000 3 frequency: 66000000 3 frequency: 66000000 4 [0, 1] -> cpu@[3, 7] 10 timer0: Timers.Miv_CoreTimer @ sysbus 0x70001000 11 fitz, verif 12 clint: Timers.Miv_CoreTimer @ sysbus 0x70001000 13 frequency: 66000000 14 [0, 1] -> cpu@[3, 7] 15 16 timer0: Timers.Miv_CoreTimer @ sysbus 0x70001000 17 clockFrequency: 66000000 17 clockFrequency: 66000000 	> 😂 pse-blinky	citati Av400000	citate endorse and a state and	
<pre>> bin 4uart: UART.MiV_CoreUART @ sysbus 0x70001000 4uart: UART.MiV_CoreUART @ sysbus 0x70001000 5 clockFrequency: 66000000 6 6 7 cpu: CPU.RiscV32 @ sysbus 8 cpuType: "rv32g" 9 privilegeArchitecture: PrivilegeArchite</pre>	✓ ⇒ Renode	3	3	
> bicenses 5 clockFrequency: 6600000 5 clockFrequency: 6600000 > bidforms 6 6 6 > bids 7 cpu: CPU.RiscV32 @ sysbus 7 cpu: CPU.RiscV32 @ sysbus 7 cpu: CPU.RiscV32 @ sysbus > bids 8 cpuType: "rv32g" 9 9 privilegeArchitecture: PrivilegeArchit	> 🗁 bin	4uart: UART.MiV_CoreUART @ sysbus 0x70001000	4uart: UART.MiV_CoreUART @ sysbus 0x70001000	
 ✓ P platforms ✓ P cpus ✓ P privilegeArchitecture: PrivilegeArch	> 🗁 licenses	5 clockFrequency: 6600000	5 clockFrequency: 66000000	
> ≥ boards 7 cpu: CPU.RiscV32 @ sysbus 7 cpu: CPU.RiscV32 @ sysbus > ≥ cpus 8 cpuType: "rv32g" 8 cpuType: "rv32g" > ≥ silabs 10 clint: clint 8 cpuType: "rv32g" a a20repl 11 12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 a c2538.repl 13 frequency: 66000000 13 frequency: 66000000 13 frequency: 66000000 a i386.repl 14 [0, 1] -> cpu@[3, 7] 14 [0, 1] -> cpu@[3, 7] 15 a litx_vexiscurepl 15 15 16 timer0: Timers.MiV_CoreTimer @ sysbus 0x70001000 14 [0, 1] -> cpu@[3, 7] miv_repl 17 clockFrequency: 66000000 18 16 timer0: Timers.MiV_CoreTimer @ sysbus 0x70003000 17 clockFrequency: 66000000	platforms	6	6	
8 cpuType: "rv32g" 8 cpuType: "rv32g" 9 privilegeArchitecture: PrivilegeArchitect	> 🗁 boards	7 cpu: CPU.RiscV32 @ sysbus	7 cpu: CPU.RiscV32 @ sysbus	
9 privilegeArchitecture: PrivilegeArchitectu		8 cpuType: "rv32g"	8 cpuType: "rv32g"	
10 clint: clint 10 clint: clint a 20repl 11 12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 c c2538.repl 13 frequency: 66000000 13 frequency: 66000000 i386.repl 14 [0, 1] -> cpu@[3, 7] 14 [0, 1] -> cpu@[3, 7] litx_vexiscurepl 15 15 15 mix_repl 16 timer0: Timers.MiV_CoreTimer @ sysbus 0x7001000 16 timer0: Timers.MiV_CoreTimer @ sysbus 0x70003000 mix_repl 17 clockFrequency: 66000000 17 mix_repl 18 18	> C cilaba	9 privilegeArchitecture: PrivilegeArchitecture.Priv1_09	9 privilegeArchitecture: PrivilegeArchitecture.Priv1_09	
a20/repl 11 11 advalue 12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 advalue 13 frequency: 66000000 13 frequency: 66000000 14 [0, 1] -> cpu@[3, 7] isis6.repl 15 15 inter.verpl 16 timer0: Timers.MiV_CoreTimer @ sysbus 0x70001000 14 [0, 1] -> cpu@[3, 7] inter.verpl 16 timer0: Timers.MiV_CoreTimer @ sysbus 0x70001000 17 clockFrequency: 66000000 inter.verpl 18 10	> Silabs	10 clint: clint	10 clint: clint	
at91m9200.repl 12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000 cc2538.repl 13 frequency: 6600000 13 frequency: 6600000 i386.repl 14 [0, 1] -> cpu@[3, 7] litx_version.repl 15 miv_repl 16 timer0: Timers.MiV_CoreTimer @ sysbus 0x70001000 miv_repl 18	a20.repl	11	11	
¹³ frequency: 66000000 ¹³ frequency: 66000000 ¹³ frequency: 66000000 ¹⁴ [0, 1] -> cpu@[3, 7] ¹⁵ ¹⁵ ¹⁵ ¹⁶ timer0: Timers.MiV_CoreTimer @ sysbus 0x70001000 ¹⁷ clockFrequency: 66000000 ¹⁷ clockFrequency: 66000000 ¹⁷ clockFrequency: 66000000 ¹⁸ ¹³ frequency: 66000000 ¹⁴ [0, 1] -> cpu@[3, 7] ¹⁵ ¹⁵ ¹⁵ ¹⁶ ¹⁶ ¹⁶ ¹⁷ clockFrequency: 66000000 ¹⁷ clockFrequency: 66000000 ¹⁸ ¹⁹	at91rm9200.repl	12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000	12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000	
i386.repl i4 [0, 1] → cpu@[3, 7] 15 i5 i5 i6 timer0: Timers.MiV_CoreTimer @ sysbus 0x70001000 inv_repl i7 clockFrequency: 66000000 17 clockFrequency: 66000000 18 14 [0, 1] → cpu@[3, 7] 15 15 15	cc2538.repl	13 frequency: 66000000	13 frequency: 66000000	
Its_vexriscv.repl 15 Inter: 16 timer0: Timers.MiV_CoreTimer @ sysbus 0x70001000 Inter: 10 timer0: Timers.MiV_CoreTimer @ sysbus 0x70003000 Inter: 10 timer0: Timers.MiV_CoreTimer @ sysbus 0x70003000 </td <td>i386.repl</td> <td>14 [0, 1] -> cpu@[3, 7]</td> <td>14 [0, 1] -> cpu@[3, 7]</td> <td></td>	i386.repl	14 [0, 1] -> cpu@[3, 7]	14 [0, 1] -> cpu@[3, 7]	
Imiv_repl 16 timer0: Timers.Miv_CoreTimer @ sysbus 0x70001000 16 timer0: Timers.Miv_CoreTimer @ sysbus 0x70003000 Imiv_repl 17 clockFrequency: 66000000 17 clockFrequency: 66000000 18 18 18	litex vexriscy.repl	15	15	
Informed 17 clockFrequency: 66000000 17 clockFrequency: 66000000 18	miv repl	16timer0: Timers.MiV_CoreTimer @ sysbus <mark>0x70001000</mark>	16timer0: Timers.MiV_CoreTimer @ sysbu <mark>s 0x70003000</mark>	
18 18		17 clockFrequency: 6600000	17 clockFrequency: 66000000	
	i miv.repi	18	18	
i mpc556/.repl 19 19	mpc5567.repl	19	19	

Quick Access 🗄 😭 🔡 👪





SC workspace.examples - Renode/platforms/board	ls/miv-board.repl - Microsemi SoftConsole v6.0.0.116		-	o ×	
Eile Edit Source Refactor Navigate Search	Project <u>Git Run W</u> indow <u>H</u> elp				
🐚 🔦 🕶 i 🗙 🕪 💷 🔳 🗵 👁 .e. 利 i+ 🕱 🌳	@ ⊕ -\$1 荐 ▼ () ▼ 9⊾ ▼ 19 🖋 ▼ ⊿ ♥ ♀ ♀ ⇒ ▼		Quick Access	🕴 😰 🖬 🔯	
Project Explorer 🛛 🕒 😒 🗢 🗖	mivrent X	mix-board repl X			
v 🗁 Renode 🔷	1 ddg: Memory MannedNemory @ syshus @x8000000	1 using "platforms/cous/	niv renl"	^	
> 🗁 bin	2 size: 0x4000000	2	iter opt		
> 🗁 licenses	3	3 gpioOutputs:			
v 🗁 platforms	4 uart: UART.MiV_CoreUART @ sysbus 0x70001000	4 0 -> led0@0			
🗸 🗁 boards	5 clockFrequency: 66000000	5	Renode		- 🗆 X
> 🗁 silabs	7 cpus CPU BiccV22 & suchus	7			
arduino_101-shield.repl	8 cnuTyne: "ry32g"	8			^
colibri-vf61.repl	9 privilegeArchitecture: PrivilegeArchitecture.Priv1 09	9			
mars_zx3-externals.repl	10 clint: clint	10			
mars_zx3.repl	11	11		1 001011001(11)	
miv-board_repi	12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000	12 Rei	node, version 1.6.0.30082 (306a188	a4-201811221641)	
miv-board-additional-uarts.repi	14 [0 1] -> cou0[3 7]	14			
miv-board.repi	15	15 (m	<mark>onitor)</mark> i \$CWD//scripts/single-r	node/miv-basic.resc	
quark_crooo-cc2520.repi ctm22f4_discovery_additional_gr	16 gpioOutputs: GPIOPort.MiV_CoreGPIO @ sysbus 0x70005000	16 Th e	ere was an error executing command	d 'machine LoadPlatf	ormDescription C:\Mic
stm32f4_discovery-additional_gp	17	17	semi\SoftConsole_v6.0.0.0.0.vrenode	e\platforms\boards\m	iv-board.repl'
stm22f4_discovery-bb.repi	18	18	ror Ell: Ira destination 'led0' do	oes not exist	
stm32f4_discovery-repl	20	20	Orl Missesseril OrftOrmeele w(0.0.	0 0) menode) aletterme	Version de Version de served anne
stm32f7_discovery-bb repl	21	20 21	C:\MICFOSEMI\SOTTCONSOle_V6.0.0.	0.0\renode\piatrorms	\boards\miv-board.rep
tegra externals repl	22	22	4:10:		
tegra2.repl	23	23	0 -> led000		
tegra3.repl	24	24	~ ~ ^ ~		
versatile.repl	25	25 26	i-V)		
vexpress-externals.repl	27	27			
vexpress.repl	28	28			
zedboard-externals.repl	29	29			
zedboard.repl	30	30			
🗸 🗁 cpus	32	32			
> 🗁 silabs	33	33			
a20.repl	34	34			
at91rm9200.repl	35	35			
Cc2538.repl	36	36			
i386.repl	38	38			
Ittex_vexriscv.repl	39	39			
miv.repl	40	40			
mpc5567.repi	41	41			\sim



- There was an error executing command 'machine LoadPlatformDescription C:\Microsemi\SoftConsole_v6.0.0.0\renode\platforms\boards\miv-board.repl'
- 2. Error E11: Irq destination 'led0' does not exist.
 - 1. At C:\Microsemi\SoftConsole_v6.0.0.0\renode\platforms\boards\miv-board.repl:4:10:
 - 2. 0 -> led0@0
 - 3.

SC workspace.examples - Renode/platforms/boards/miv-board_.repl - Microsemi SoftConsole v6.0.0.116

<u>File Edit Source Refactor Navigate Search Project Git Run Window Help</u>

ⓑ 🗞 ▾ ! ◙ 卧 ఐ ■ આ ۞ ! 안 팩 ↦ 🕱 今 ⊕ ♀ ☜ ‡ ▼ 🖸 ▼ 🤮 🖋 ▾ ⊿ や Ϙ ▾ ⇔ ▾					
Project Explorer 🛛 🕞 🔽 🗖 🗖	miv-hoard rent X	miv-board repl X			
 E Renode bin licenses platforms boards silabs arduino_101-shield.repl colipri-vf61.repl 	<pre>Inv-board.repl to 1 using "platforms/cpus/miv.repl" 2 3 gpioOutputs: 4 0 -> led0@0 5 6 7 8</pre>	<pre>Inv-board_repl ☆ 1 using "platforms/cpus/miv.repl" 2 3 gpioOutputs: 4 0 -> led0@0 5 6 led0: Miscellaneous.LED @ gpioOutputs 0 7 invert: true 8</pre>			
 mars_zx3-externals.repl mars_zx3.repl miv-board_repl miv-board-additional-uarts.repl miv-board.repl miv-board.repl quark_c1000-cc2520.repl stm32f4_discovery-additional_gp 	9 10 11 12 13 14 15 16 17	9 10 11 12 13 14 15 16 17			



SC workspace.examples - Renode/platforms/cpus/miv.repl - Microsemi SoftConsole v6.0.0.116

File Edit Source Refactor Navigate Search Project Git Run Window Help

[6] 🔨 ▾ [▲] 🕨 🗉 🔳 그 ۞ 고 · · ·] 🗁 🕱 ۞ 윤] 음 · 성 🕸 ▾ 🚺 ▾ 💁 🔗 ▾ ⊿ 🏷 쓴 ▾ ↔ ▾

Project Explorer 🛛 🛛 🕒 🖄 🔍 🗖 🗖	mivrent X	miv-board repl 💥	🖘 Renode
∽ 🗁 Renode 🔨	1 ddau Mamany ManaedMarrany A. cychus Av9000000	1using "plotforms/onus/mix_popl"	
> 🗁 bin	2 size: 0x/000000	2	
> 🗁 licenses	3	3 gpioOutputs:	
✓ ➢ platforms	4 uart: UART.MiV_CoreUART @ sysbus 0x70001000	4 0 -> led0@0	
🗸 🗁 boards	5 clockFrequency: 6600000	5	Penode version 1 6 0 30082 (366-18-4-201811221641)
> 🗁 silabs	6	6 led0: Miscellaneous.LED @ gpioOutputs 0	Renode, Version 1.0.0.30002 (300a10a4-201011221041)
arduino_101-shield.repl	7 cpu: CPU.RiscV32 @ sysbus	7 invert: true	
colibri-vf61.repl	8 cpulype: "rv32g" 9 pnivilagaAnchitactuma: DnivilagaAnchitactuma Dniv1 00	8	(monitor) i \$CWD//scripts/single-node/miv-basic.resc
mars_zx3-externals.repl	10 clint: clint	10	Available peripherals:
mars_zx3.repl	11	11	
miv-boardrepl	12 clint: IRQControllers.CoreLevelInterruptor @ sysbus 0x44000000	12	sysbus (SystemBus)
miv-board-additional-uarts.repl	13 frequency: 66000000	13	
imiv-board.repl	14 [0, 1] -> cpu@[3, 7]	14	clipt (CorelevelInterruptor)
quark_c1000-cc2520.repl	15 16 milliontenten CDTODent Mill ConsCDTO A suchus A.70005000	15	
stm32f4_discovery-additional_gp	16 gpiooutputs: GPioPort.Miv_CoreGPi0 @ Sysbus 0x70005000	10	<0X44000000, 0X4400FFFF>
stm32f4_discovery-bb.repl	17	18	
stm32f4_discovery-kit.repl	19	19	├─ cpu (RiscV32)
stm32f4_discovery.repl	20	20	Slot: 0
stm32f7_discovery-bb.repl	21	21	
tegra_externals.repl	22	22	⊢ ddr (MappedMemory)
tegra2.repl	23	23	<pre></pre>
tegra3.repl	24	24	
versatile.repl	26	26	
vexpress-externals.repl	27	27	☐ gpioUutputs (MiV_CoreGPIU)
vexpress.repl	28	28	
zedboard-externals.repl	29	29	
zedboard.repl	30	30	└── led0 (LED)
🗸 🗁 cpus	31	31	Address: 0
> 🗁 silabs	33	33	
a20.repl	34	34	uart (MiV CorellADT)
at91rm9200.repl	35	35	- 0.070001000 - 0.070001017
cc2538.repl	36	36	<0x/0001000, 0x/000101/>
i386.repl	37	37	
litex_vexriscv.repl	38	38	(Mi-V)
miv.repl	33	10	
mpc5567.repl	41	41	



First Thursdays

- 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: Add and Debug and Pre-Existing Peripheral in PolarFire SoC
- Nov. 7 Webinar 7: How to write custom models filters, offloading, acceleration etc
- Dec. 5 Webinar 8: Handling Binaries

Contd.



Second Thursdays

Jan. 9 - Webinar 9: Run Linux on Renode (PolarFire SoC Model as a Quad-core SMP) – this is not a Linux / Buildroot tutorial

- Feb. 13 Webinar 10: Build applications for Linux on PolarFire SoC
- Mar. 12 Webinar 11: Introduction to PolarFire SoC MSS Configuration and Software Flow
- Apr. 9 Webinar 12: Two baremetal Applications on PolarFire SoC
- May 14 Webinar 13: Linux + Real-Time (AMP Mode) on PolarFire SoC



Thank You