

# Summary and Updates to the GR740 Software Ecosystem

**GR740 User Day 2022** 



# Software

- Complete ecosystem  $\bullet$
- A combination of Gaisler and 3<sup>rd</sup> party software

### **Tool chains**, **Operating systems** and compilers

- Bare-C
- Linux
- RTEMS
- VxWorks
- Zephyr

### Partner software

Time-and-Space Partitioning:

- FentISS XNG
- SYSGO PikeOS
- Wind River VxWorks RTOS

### **Boot loaders**

- GRBOOT •
- **GRBOOT-STANDBY** •
- MKPROM2 ٠

### **Development tools**

TSIM3 simulator

•

- **GRMON3** debugger
- GCC compilers 0
- LLVM/Clang compilers •







**RTEMS**"







## **GRMON3 - LEON and NOEL-V debugger**



## **GRMON3** is a hardware monitor optimized for **SPARC/LEON** and **RISC-V/NOEL-V** systems

- Provides a non-intrusive debug environment
- Graphical User Interface, Command Line Interface, and GDB
- Tcl interpreter and Tcl scripting
- Supports many debug link interfaces (JTAG, Ethernet, SpW, ..)
- Support for multiple CPUs and OS threads
  - PikeOS thread support available in next release (see picture)
- Built-in disassembler and trace buffer management
- C/C++ source level debugging (see next slide)
- Remote connection to GNU debugger (GDB)
  - IDE support via the GDB connection, see Eclipse Quick Start Guide
- SPI memory programming and Common Flash Interface (CFI) for Flash PROM programming
- Error injection for LEON3/4 systems
- Provided for Linux and Windows

<b>v</b>	G	RMON3 - 3.3.1-SNAPS	SHOT (on mars.got.gaisler.co	m)			
File Edit Run Window Help							
†r Debug ⊠ i→ 🌼 マ 🗆 🗗	E CPU (x)= Variable	es IX	🏝 🎫 🖻 🛍	1 🕸 🗢 🗖 🗖	😸 IO Registers 🔀	Ø System	
▼ @ Existing session	Name	Туре	Value		Name	Hex	
🔻 记 GR740 rev0	🔻 🕈 psp	P4_psp_descriptc	{api version=0xf8f80000	), max interrupts=0x800	₱ I4Statu		
🕨 🧬 cpu0 (Power down mode)	(×)• api versio	n P4 uint32 t	0xf8f80000		pcitrace0		
🕨 Թ cpul (Power down mode)	Ø=max inter	ri P4 uint32 t	0×80060000		▼ uart0		
🕨 🧬 cpu2 (Power down mode)	(x)=ts calibrat	tic P4 uint64 t	0x0000000800d5100		status	0000086	
🕨 🧬 cpu3 (Power down mode)	🕬 ns per ca	lil P4 time t	0x800d510080060000		▶ ctrl	8000003	
🔻 🚼 PikeOS	Ø=dyntick re	es P4 time t	0x803a000080170c90		scaler	000000a2	
Switcher#00 (Suspended)	(X)-dyntick m	na P4 time t	0xc40300008408a001		▶ uart1		
🔻 🧬 Idle#00 [cpu0] (Power down mode)	▶ ⇒ psp id	const char *			▼ gpio0		
0x80022e88 in psp_idle_workaround() at libpsp/leon-power-down.c: 54	> psp welco	on const char *			iodata	000033ff	
0x800284dc in p4_idle() at src/main.c: 1719	romheade	r drv config heade	{sig=0x00000000, versio	n=0x000000000. header c	iooutput	00000000	
0x800501d0 in p4sparc_enter_idle() at arch/sparc/src/acontext.S: 314	dev call t	al P4 devcallback t	0x02bfffa8		iodir	00000000	
0x80027f4c in enter_idle() at src/main.c: 1392	(x)=align mas		0xc423e014		irqmask	00000000	
0x80028e24 in p4_main() at src/main.c: 1240	🕨 🥭 api	P4 psp api t	{init=0x00000001. init	cpu=0x22220000. init l	irqpol	0000b46d	
0x800233cc in init_board() at src/cboot.c: 245	▶ (arch	P4 psp arch t	{ram phys base=0x80ba20		irqedge	00003ecf	
≡ 0×0000000	(X) breakcon	adt bool t	-1341120512		<ul> <li>capability</li> </ul>	0005040f	
≡ 0×00000000	⊠=num cou		AxbAAe20ff		18 pu		
0x01000000 in i() at	0x800d5100800600	00			17 ie		
Switcher#01 (Suspended)			4141741056, Oct: 0100015242002 00.0000.1000.0000.0000.0110.		16 if		
▼ m <sup>0</sup> Idle#01 [cpu1] (Power down mode)	Size: 8 bytes, Typ	e: P4_time_t	,	0000,0000,0000,0000	12:8 in		
0x80022e88 in psp_idle_workaround() at libpsp/leon-power-down.c: 54	Offset: 16, Addre	ss: 0x800d4014			4:0 nl:	0f	
0x800284dc in p4_idle() at src/main.c: 1719	≝ Disassembly 😫	Jerminal				- 0	
0x800501d0 in p4sparc_enter_idle() at arch/sparc/src/acontext.S: 314		<b>.</b>					
0x80027f4c in enter_idle() at src/main.c: 1392				Enter locatio	n here 👻  🕄	🏠 😫 😣	
Switcher#02 (Suspended)	80028df4;	mp %q1, 0	<p4 main<="" td=""><td>+1856&gt;</td><td></td><td></td></p4>	+1856>			
Idle#02 [cpu2] (Power down mode)		one,a 0x80028e2c	<p4_main< td=""><td></td><td></td><td></td></p4_main<>				
Switcher#03 (Suspended)	80028dfc:	ethi %hi(0x80059					
Idle#03 [cpu3] (Power down mode)		all 0x80038990	<p4_main< td=""><td></td><td></td><td></td></p4_main<>				
🕨 🧬 Sysgo (Suspended)		10p nov 0x20, %o2	<p4_main <p4 main<="" td=""><td></td><td></td><td></td></p4></p4_main 				
Image: Image of the second		ethi %hi(0x80059					
Image: Support of S	80028e10: 0	or %01, 0x3e0, %o					
Image:		all 0x800252d0	<p4_main< td=""><td></td><td></td><td></td></p4_main<>				
generation (Suspended)		add %o0, 0x130, %					
Icon3-uart (Suspended)		all 0x80027cd4 nov 0.%00	<p4_main <p4 main<="" td=""><td></td><td></td><td></td></p4></p4_main 				
fop_0_0 (Suspended)		all 0x80027f34	<p4 main<="" td=""><td></td><td></td><td></td></p4>				
fop_0_1 (Suspended)		nov 0, %o0	<p4_main< td=""><td></td><td></td><td></td></p4_main<>				
Interpretation of the second secon		all 0x8003a398	<p4_main< td=""><td></td><td></td><td></td></p4_main<>				
Interpretation of the second secon		or %o0, 0x230, %o nov 0, %o2	0 <p4_main <p4 main<="" td=""><td></td><td></td><td></td></p4></p4_main 				
fop_0_4 (Suspended)		add %fp, -4, %ol	<p4_main <p4_main< td=""><td></td><td></td><td></td></p4_main<></p4_main 				
u <sup>®</sup> SYS_irq_29 (Suspended)	80028e3c: 0	all 0x800350e4	<p4_main< td=""><td>+1928&gt;</td><td></td><td></td></p4_main<>	+1928>			
gP SYS_irq_30 (Suspended)		nov 0, %o0	<p4_main< td=""><td></td><td></td><td></td></p4_main<>				
In the second		:mp %o0, 0	<p4_main< td=""><td>1+1930&gt;</td><td></td><td></td></p4_main<>	1+1930>			
In thread_123456 (Suspended)							



## GRMON 3.3 released in Q4 2022



## GRMON 3.3 introduces C/C++ language debugging support in Graphical User Interface

- Based on Eclipse CDT and TCF familiar
- No dependency on GDB
- GDB still supported
- C/C++ source level debugging
  - Extended existing Views
  - New Views
  - Not in Command Level Interface
- C/C++ source code view
- Mixed Disassembly/C/C++ mode view
- C/C++ line execution control (step, step into, ..)
- Line breakpoints and watchpoint View
- C/C++ line display in Function call back trace
- C/C++ Variables View
  - Global Variables
  - Local stack frame Variables
- Symbols view listing global C/C++ symbols

Q GRMON3 - 3.3.0		- 6 ×	
Eile Edit Run Window Help			
i 🚳 👘 💌 🖳 🕪 💷 🗱 😓 🗇 😥 💼 🧱 🛅			
🗱 Debug 🛙 🏟 😒 🖓 🖓 🖓 🖓 👘 🖓 🖓 👘 🖓 🖓 🚱 System 2	🛿 🜔 Executables 🚼 O	Jutline 🗖 🗖	
~ @ GRMON ^ Context: CPU0		🔗 🗉 🖻 📴 🔻	
√			
2 @ cpu0 (Breakpoint 2 hit) 0: 00027680 0002000 00000000 Device	Informa	ation	
> 🕐 cpu1 (Power down mode) 1: 00027640 (000003e8) 00030460 f34000e2 ✓ 🖬 Basi			
/ IP cpuz (rowel down mode)	ame GR740	rev1	
	RLIB build versi 4153		
V III RIEMS	evice ID 0x740		
> @ IDLE (Suspended)	<ul> <li>Detected Freque 250 MHz</li> </ul>		
✓ IP TA1 (Suspended) 7: 00010c64 000015ec 00000000 ■ S	stem context SE0.SYS	50	
Ox0000f304 in _Thread_Do_dispatch() at threaddispatch.c: 305			
= 0x0000da0 in richis_tax_wake_arter() at sinplockil. 547		Debug Link	
≡ 0x000015ec in Test_task() at rtems-tasks.c: 252 f3400e5 00000060 00000011		V2 SpaceWire Serial Link	
0x00010c64 in _Thread_Entry_adaptor_numeric() at threadentryadaptornumeric.c 25		master interface	
= 0x0000f890 in _Thread_Handler(: 139 pc 080816690 call 0x8080c954 > 0 e	ICI EDCL m	master interface 🗸 🗸	
E 0x00001738 in _Thread_Handler() at threadget.c: 43		>	
🞬 Disassembly 😰 reters-tasks.c 🛛 🚬 🗇 👘 🐨 💷 🔯 10 Regist	ers 🛛 👔	2 📲 🖻 💰 🖬 🖗 🔻 🗖 🗖	
240 ^ Name Type Value ^ Name	Hex Deci	imal Description Address ^	
241 status = rtems_task_ident( RTEMS_SELF, RTEMS_SEARCH_ALL_NODES, &tid ); (0/ unused rtems_task_argument 0x00027680 > ) ddrsdmu	x0	ddrsdmux0	
242 assert(status == RTEMS_SUCCESSFUL); (№ tid rtems_id 0x0a010003 > mctr0		mctrl0 regi	
243 task_index = task_number(tid);		spwrtr0 reg	
244         for (;; ) {         0x900007c4         v unt0           2245         status = rtems clock get tod( &time );         0x900007c4         v unt0		uart0 regist	
246     assert(status = RTEMS_LOCK_get_COU_t;       246     assert(status = RTEMS_SUCCESSFUL);	0000086 134	UART Statu 0xFF900C	
247 if ( time second >= 3335 ) { № day uint32_t 0x000001f > ctrl	80000843 2147	7485763 UART Contr 0xFF900C	
248 puts( "*** END OF CLOCK TICK TEST ***" ); 00 hour uint32 t 0x00000009 scale	0000032d 813	UART Scale 0xFF900C	
249 exit(0); 0x minute uint32_t 0x00000000 > gpio0		gpio0 regis	
250 }		irqmp0 reg	
251 put_name(Task_name[Task_index], FALSE ); 252 print time(" - reters clock get -", ktime, "\n");	1	gptimer0 r	
253 status = rtems task wake after ( task index * 5 * rtems clock get ticks per second() );	n	gptimer1 r	
254 assert(status == RTEMS_SUCCESSFUL):	!	gptimer2 r	
> gptimer		gptimer3 r	
		antimord r	
Ø Terminal ⊠ □ K   the he all   0 the Ø = □   < > >   <		🍇 🔐 😔 🔪 🗇 🗇 🗇	
term1	~ ~		
TAL - rtems_clock_get - 09:00:00 12/31/1988			
TA1 - rtems_clock_get - 09:00:00       12/31/1988         TA2 - rtems_clock_get - 09:00:00       12/31/1988         TA3 - rtems_clock_get - 09:00:00       12/31/1988 <td></td> <td></td>			
TAL - rtems_clock_get - 09:00:00 12/31/1988			



## **TSIM3 LEON simulator**

### Accurate behavioral simulator

- Precise behavior and timing in terms of simulation time
- Algorithm performance evaluation
- Deterministic execution
- Improved debugging capabilities (infinite tracing and event logging)
- Evidence gathering for software qualification (code-coverage, error-injection)
- Significantly higher performance than a SoC VHDL/VLSI/System-C model, hardware simulator
- User extendable
- Simulates specific LEON2, LEON3, LEON4 components and generic LEON2 and LEON3 systems

### • Supports three operating modes:

- Standalone frontend with Tcl support for scripting and automation
- Used as a library to be included in larger simulator frameworks (C API is provided)
- Attached to GDB (allowing to connect Eclipse or other IDEs, see <u>quick start guide</u>)

### TSIM3 LEON4 models GR740

- Quad core LEON4 with GRPFU and MMU
- L2Cache, SDRAM, PROM
- Ethernet, CAN, SPI
- IOMMU, AHBSTAT
- Timers, UARTs, GPIO
- GRSPW2 cores as stand-in for SpaceWire router
  - SpaceWire Router IP model is under development (started 2022)
  - Prototype of the model released as part of TSIM3-GR716B

Visit the webpage for more information



## 3<sup>rd</sup> Party GR740 Simulator support



- Other vendors provide Simulation solutions for GR740
- Complements TSIM3 usage, TSIM3 does not cover real-time for GR740 and faster
- Faster simulators allows running the final application in real-time
- QEMU open-source simulator has limited support for LEON today
- Real-time simulation solutions with GR740 models,
  - TERMA TEMU
  - Wind River Simics
  - KARI LAYSIM
- Vendor Presentations at GR740 User Day 2022 for more information

## **GRBOOT - Flight Software Boot Loader**

### GRBOOT is a flight system software suite targeting LEON based systems. It provides initialization, self-test and application loading functionality to payload and on-board computers

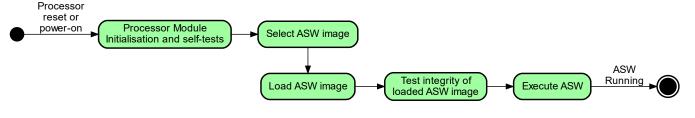
#### Features

- Implements ESA "SAVOIR Flight Computer Initialization Sequence" (SAVOIR-GS-002)
- Developed in accordance with ESA software engineering standards ECSS-E-ST-40C and ECSS-Q-ST-80C, criticality category B.
- Multiprocessor support (AMP, SMP)
- Self-testing of external memory and internal caches, register files
- Support for user extension point
  - GRBOOT-STANDBY extension point based on SAVOIR Standby mode
  - Run application from extension point (ROM resident application)
- Loader with ASW image integrity check and fallback image(s)
- Generates Boot Report for ASW or external access
- Prepares environment compatible with multiple operating system:
  - RTEMS, VxWorks, Linux, PikeOS, BCC, SMP, AMP, etc.
- Support for GR740, GR712RC and UT700 devices (support for GR765 planned)
- Supports in-flight patching by ASW (ASW image not linked to boot loader)

#### Test and validation

- Fully automated test suites
- Unit tests executing on target hardware and with TSIM3 LEON simulator.
- Code coverage captured using TSIM3 LEON
- Validation test suite executes on target hardware, checks software and system behavior













## **STANDBY – SpaceWire/PUS remote terminal**



#### **GRBOOT STANDBY**

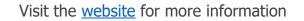
- Implements the SAVOIR Standby mode where GRBOOT implements the Boot SW Nominal Sequence Mode
- Based on JUICE GR712RC DPU Boot Software
  - Redesign with support for GR740 and GR712RC
- ECSS-E-ST-40C and ECSS-Q-ST-80C Category B. (same as GRBOOT)
  - Design Documentation
  - Unit-tests
  - Validation tests
- Maintenance operations before application is started

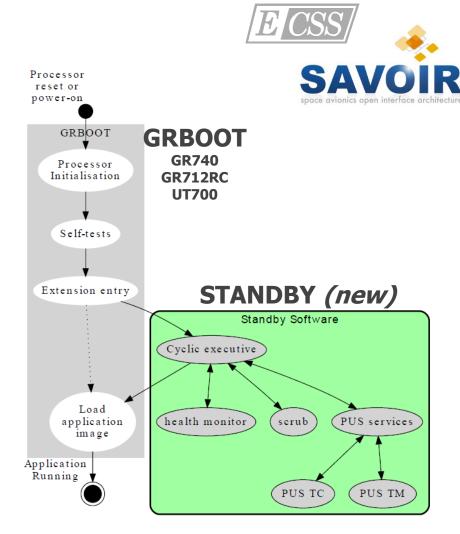
#### **Features**

- Remote terminal over SpaceWire
- PUS services according to [ECSS-E-ST-70-41C]
- PUS Layer not SpaceWire dependent
- Support for adding mission specific PUS services
- Scrubbing of read-write memories (caches, RAM)
- Monitoring of bit upsets in caches and external RAM
- Validation and Unit tests can be rerun with GRESB

#### SpaceWire/PUS Services

- House keeping
- Diagnostics, register access
- Boot Report service
- Memory commands (Read, Write, CRC)





GRBOOT-STANDBY execution sequence

## **VxWorks 7 GR740 support**



WNDR/R

#### • Real-Time Operating System provided by Wind River

- Eclipse based IDE for development, debugging, and analysis
- SMP and AMP
- MMU and non-MMU support
- Time and Space Partitioning
- C11 and C++ Support
- SPARC architectural support provided by Gaisler
  - VxWorks 7 SR0650
  - GCC and LLVM compilers
- GR740 BSP
  - · Drivers included
    - SpaceWire router
    - SpaceWire Time Distribution Protocol
    - Ethernet, MIL-STD-1553B BC/RT/BM
    - PCI, PCI DMA, CAN, SPI
    - SDRAM Memory scrubber
    - Timer, UART, GPIO
    - IOMMU, Clock-gate unit, Temperature Sensor
    - Statistical unit, Bus monitor, Level 2 Cache
  - Boot configuration for GR-CPCI-GR740
  - Documentation and Start Guide
- GRMON tools VxWorks OS awareness
  - Kernel/RTP thread support
  - TSP connection to Wind River Workbench remote connection
- We provide LEON and NOEL-V specific technical support

\77777777\	/777777	77/
\77777777	77777777	7/
\77777777\	/77777777	/
\77777777\	/77777777/	
\77777777\	\7777777/	
	\77777/	VxWorks 7 SMP 32-bit
\77777777\		
\77777777\		Release version: DEVEL-LEON-SR0650
\77777777	•	Build date: Dec 9 2022 13:27:16
\7777777		
\777777		Copyright Wind River Systems, Inc.
\77777		1984-2022
\777/		
\7/	/777\	
-		

Board: GR-CPCI-GR740 CPU Count: 4 OS Memory Size: ~1023MB ED&R Policy Mode: Permanently Deployed System Frequency: 250MHz

Adding 5440 symbols for standalone.

-> i

NAME	ENTRY	TID	PRI	STATUS	PC	SP	ERRNO	CPU #
tExcTask	fde44	1bf860		PEND	132ac8	1c39a0	0	
tLogTask	logTask	2017ea08	-	PEND	131068	20192b60	õ	-
tShell0	shellTask	201fd010	1	READY	13ef14	20213af8	Θ	1
tVxdbgTask	da948	201f8390	25	PEND	132ac8	201fbde8	Θ	-
tNet0	ipcomNetTask	20002870	50	PEND	133260	20008e50	Θ	-
tNetConf	1ed14	201dd100	50	PEND	132ac8	201e0b40	Θ	-
miiBusMoni>	124a94	2018f490	252	DELAY	13cca8	2019aeb8	Θ	-
tIdleTask0	idleTaskEnt>	1ce000	287	READY	f4054	1cded0	Θ	Θ
tIdleTask1	idleTaskEnt>	1da000	287	READY	f4054	1d9ed0	Θ	-
tIdleTask2	idleTaskEnt>	1e6000	287	READY	f4054	1e5ed0	Θ	2
tIdleTask3	idleTaskEnt>	1f2000	287	READY	f4054	1f1ed0	0	3
val <u>u</u> e = 0 =	0×0							
->								

## **Linux support for GR740**



- LEON Linux kernel releases
  - Based on 5.10 LTS kernel
  - Abandoned tracking LTSI LTS project
- GR740 Device Drivers
  - SpaceWire Router
  - CAN
  - SPI
  - PCI
  - Ethernet
  - GPIO
  - UART
- Prebuilt toolchains
  - Based on GCC-10
  - GLIBC 2.31 libraries
- MKLINUXIMG Second stage bootloader
  - OpenPROM providing services to kernel
  - Basic initialization of MMU
  - Low-level BSP settings

- LINUXBUILD custom build flow for LEON
  - Used Buildroot as subcomponent
  - Migration to Buildroot stand-alone for standard build flow
- LEON Buildroot releases
  - Based on upstream Buildroot LTS versions
  - Replaces custom LEON Linux kernel and user space build and development environment
  - Integrated support for all our LEON Linux components
  - Example configurations, including for GR740
  - SDK toolchain build with buildroot libraries
- Documentation
  - Including overview and quick start guide
  - Driver API for specific APIs, such as SpaceWire

## **RTEMS GR740 support**



### RCC-1.3

- Gaisler's RTEMS-5 distribution
- Support for SMP and AMP
- GR740 BSP
  - Build configuration
  - Drivers included
    - SpaceWire Router
    - SpaceWire Time Distribution Protocol
    - Ethernet
    - PCI and PCI DMA
    - MIL-STD-1553B BC/RT/BM
    - Timer, UART, GPIO, CAN, SPI
    - SDRAM Memory scrubber
    - IOMMU configuration, Level 2 Cache
    - Statistical unit, AHB Status Register Bus monitor
  - User Manual and Driver API documentation
- Main differences from official source:
  - Most code up-streamed, limited differences
  - Toolchain based on GNU GCC-10
  - LLVM/Clang toolchain provided
  - RCC examples for device drivers
  - Additional Documentation

### **Other News**

- RTEMS-SMP Qualification Data Pack Embedded Brains presentation
  - Available for GR740 and GR712RC
  - <u>RTEMS SMP Qualification Data Pack (https://rtems-qual.io.esa.int/)</u>
- Gaisler contributions changed to new RTEMS license
  - Update from old RTEMS open-source license to BSD-2
  - This includes all GR740 device drivers
  - Simplify integration into some space applications
- RTEMS-5.2 release candidate announced (Nov 2022)
   → pending RCC-1.3 update
- LibmCS Standard Compliant and Pre-Qualified Math Library
  - Published in December 2022, ESA funded work by GTD GmbH
  - ECSS E-ST-40 and Q-ST-80 Category B, replaces previous MLFS library
  - Comprehends all ISO C and POSIX math functions found in libm
  - CAES Gaisler to evaluate including LibmCS in RCC and/or BCC
  - <u>https://essr.esa.int/project/libmcs-mathematical-library-for-critical-systems</u>
  - <u>https://gitlab.com/gtd-gmbh/libmcs</u>
  - <u>https://data.gtd-gmbh.de/libmcs-validation</u>



### https://www.rtems.org

## **BCC2 and Zephyr RTOS**



### **BCC2** – Bare-Metal Cross Compiler

- Prebuilt C11/C++11 cross compiler
  - GNU GCC-10
  - LLVM/Clang
- Newlib C Library
- BCC run-time library for LEON
- Fast interrupt handling
- GR740 BSP
  - Linker-scripts
  - FPU, CAS, SVT, SPARCv8
- GR740 Device Drivers
  - SpaceWire AMBA DMA Port
  - MIL-1553-B BC/RT/BM
  - CAN
  - SPI
  - GPIO
  - UART
  - Timer, IRQ Controller
  - Clock-gating
  - SDRAM memory scrubber
  - AHB Status bus monitor
- Documentation
- Free and open-source

### Zephyr RTOS

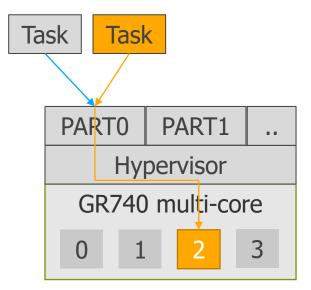
- Scalable real-time operating system
  - Small-footprint kernel
- SPARC port, SPARCv8 ABI compliant
- LEON3/4/5 supported by LEON3 BSP
- Interrupt and FPU support
- GNU GCC toolchain SDK
- Contributed by Wind River, organized under Linux Foundation
- Apache 2.0 open-source license
- Wide and active community
- Long-term-stable release planning and test-suite
- Device drivers support
  - Limited to Timer and UART
  - Reuse BCC2 drivers (single-threaded)



## **3rd Party Hypervisor and TSP support**

CRES PIONEERING ADVANCED ELECTRONICS

- Time-and-Space Partitioning a trend in space applications
- Multiple functions integrated into the same processor system
- May allow mixed-criticality applications
- Partitions describe separation
- Hypervisor scheduler, separates in time
- Hypervisor resource allocation, separates in space
  - Memory Management Unit (MMU) limit CPU address space
  - IO MMU limit DMA address space
- GR740 hardware prepared for para-virtualization
  - Multiple TIMER Cores
  - IRQ(A)MP multiple independent Interrupt controllers
  - LEON4 Memory Management Unit (MMU)
  - IOMMU for DMA separation
  - LEON4 Statistical Counters for surveillance and control
  - I/O resources sharing by 4KiB address alignment
- GR740 Solutions available today, see vendor presentations
  - FENTISS Xtratum
  - GMV AIR
  - SYSGO PikeOS
  - Wind River VxWorks Safety Profile (RTP and TSP scheduler)

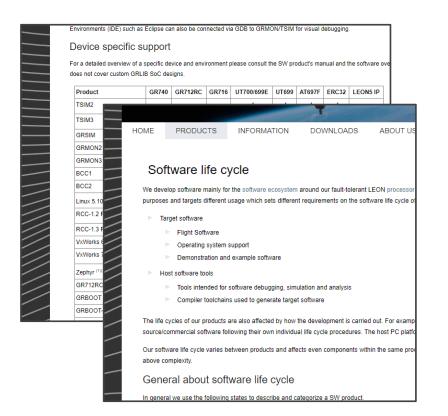


## **GR740 Software Ecosystem**



Resource overview

- Gaisler.com > Products > Software Overview
- Gaisler.com > Products > Software Lifecycle
- Gaisler.com/notes Technical Notes
- GR-CPCI-GR740 Quick Start Guide
- Newsletter for notifications on Updates and Fixes
- Other development, improvements also for GR740
  - RISC-V / NOEL-V
  - LEON5
  - GR765
- GR765 software backwards compatibility with GR740
- GR765/LEON5 SW ecosystem (GSTP 2023 SNSA funding)
  - GRBOOT update to latest SAVIOR standard
  - GRBOOT ECSS documentation and other improvements
  - GRBOOT development includes GR712RC/GR740 targets
  - Linux BSPs commonalities
  - TSIM3 simulator common improvements



## **Improving GR740 Software with Partners**



- Collaboration for improving the SW ecosystem together
- Technical help and expertise, development board and simulator access
- Collaborations in reaching out to GR740 users
- Improved visibility for partners planned through website improvements
- Typical areas involve
  - GR740 component support
  - Qualification activities we are happy to collaborate and support developments!
  - Software tools
- Community feedback welcome!
  - Please share success stories or issues with software components
  - For example, within analysis tools for WCET, MC/DC, static analysis, CPU interference analysis, etc.

#### CRES PIONEERING ADVANCED ELECTRONICS

### **GR740 Software** Summary

- Major new items from CAES Gaisler
  - GRMON-3.3 C/C++ language debugging GUI integration
  - TSIM-3 TLIB for integration and improved GR740 models
  - RCC-1.3 RTEMS-5.1 stable release
  - VxWorks SR0650 support with improved GR740 driver support
  - New Linux 5.10 LTS kernel, GLIBC, toolchain and stand-alone buildroot
  - Common GCC-10 toolchain with DWARF4 support
  - STANDBY remote SpaceWire/PUS terminal
  - Zephyr RTOS support
- Ecosystem for GR740 evolving with new software becoming available
- Thanks to partners and 3<sup>rd</sup> parties for the added value
- Sign up to <u>CAES Gaisler Newletter</u> for updates





