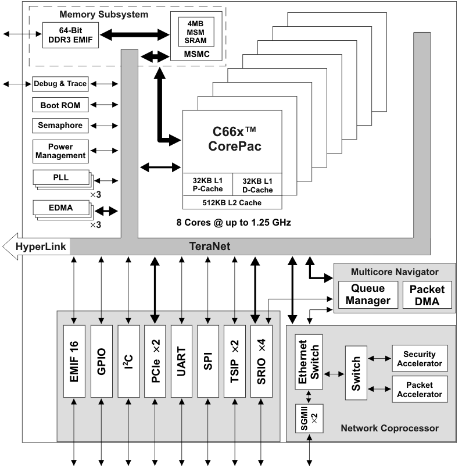
Protected mode RTOS: what does it mean?

Dr. Bernhard Sputh
bernhard.sputh@altreonic.com

Altreonic NV
Gemeentestraat 61 Bus 1
3210 Linden
Belgium

August 24, 2015

Current Trends in Embedded Systems
The RoC (Rack on a Chip)



Texas Instruments C6678

Assured Reliability Resiliance Level

Level	Definition	Measure
ARRL-0	The component might work (“use as is”).	None
ARRL-1	The component works as tested.	Testing
ARRL-2	The component meets all its specifications, if no fault occurs.	+Formal proof.
ARRL-3	+ Guarante to reach a fail-safe or reduced operational mode upon a fault.	+Fault detection, containment, and preventing error propagation.

Why Protection is needed?

- Formal checking checks only models of the software, and is only sufficent for ARRL-2.
- The industry still develops applications using C/C++.
- Humans are imperfect!
- The environment may induce faults:
  - Bit-flips due to alpha particles.
  - Power glitch induced problems.
  - Faulty components.
  - . . .
- For ARRL-3 fault detection and 'containment' are required, i.e. Protecting against unintended behaviour.

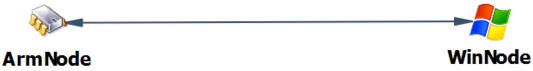
Current Approach Hypervisors

- Function:
  - Separate Applications in different Partitions.
  - Partitions cannot access the memory of other partitions.
  - Partitions get scheduled in time, i.e. time-sliced in the area of 1 – 100ms slices.
- Issues:
  - Time-slicing affects real-time behaviour.
  - Memory only protected at the partition level.

VirtuosoNext Approach

- Formally developed distributed RTOS for heterogeneous Systems;
- Virtual Single Processor (VSP) Programming Model;
- Programming with Interacting Entities, a Pragmatic Superset of CSP;
- Static allocation of Entities.
- Priority based Scheduling of Tasks.
- Tasks run separated in memory (memory protection). Currently supported on:
  - ARM-Cortex-M3 (MPU)
  - ARM-Cortex-A9 (MMU)
- Code is marked read only.
- Data is marked not-executable.

Virtual Single Processor Programming Model

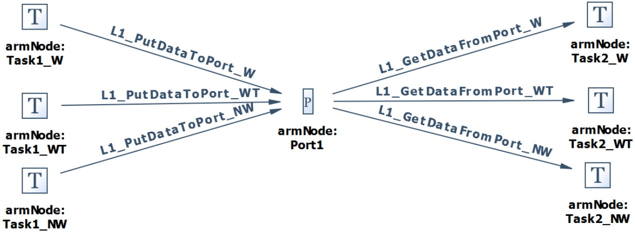


- Node: A CPU Core that executes an instance of the Kernel-Task
- Link: A communication channel between two Nodes. For instance:
  - RS-232 / 422 / 485
  - TCP-IP (Ethernet)
  - Shared Memory (in SMP systems)
  - XMOS-Links (deprecated)
  - Bongo drums (in principle)
  - ...
- Packet:
  - All Interactions / Services get represented by a Request-Packet.
  - Packets are routable throughout the System.

Interacting Entities 1/2

- Active Entities:
  - Tasks: Have their on stack and a Priority.
  - Interrupt Service Routines
- Passive Entities / Hubs:
  - Port: Acts like a Any2Any-Channel in JCSP;
  - FIFO: Acts like a Any2Any-BufferedChannel in JCSP;
  - Event: Binary Signal;
  - Semaphore: Counting Signal;
  - Resource: Lock with Priority Inheritance;
  - BlackBoard: Save System Wide Global Variable;
  - MemoryBlockQueue: Zero-Copy FIFO on a single Node;
  - DataEvent: An Event that can transport also Data;
  - PacketPool: A pool of Packets that can be allocated at runtime.
  - MemoryPool: A pool of MemoryBlocks that can be allocated at runtime.

Interacting Entities 1/2



- Interaction Semantics:
  - _W: Wait until Synchronisation can be achieved.
  - _WT: Wait until Synchronisation can be achieved, or the timeout occurs.
  - _NW: Non-Waiting.
  - _A: Asynchronous: The Task can run in parallel while the request is being processed (only locally).

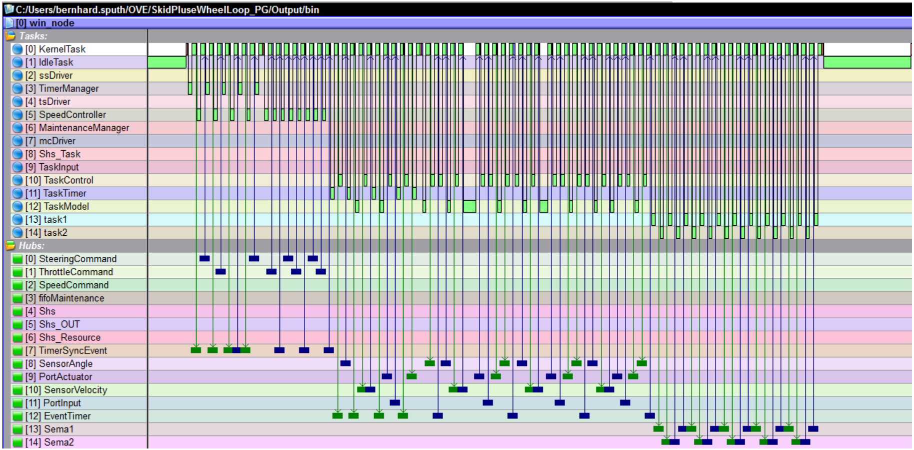
ARM-Cortex-M3 MPU Protected Mode

- Variable region size (32B, 64B, 128B, – 4GiB).
- Region alignment depends on region size.
- 8 regions in parallel.
- Context Switch had to be rewritten to reconfigure the MPU.
- The build process now performs memory mapping of Entities.

ARM-Cortex-A9 Memory Management Unit (MMU) Protected Mode

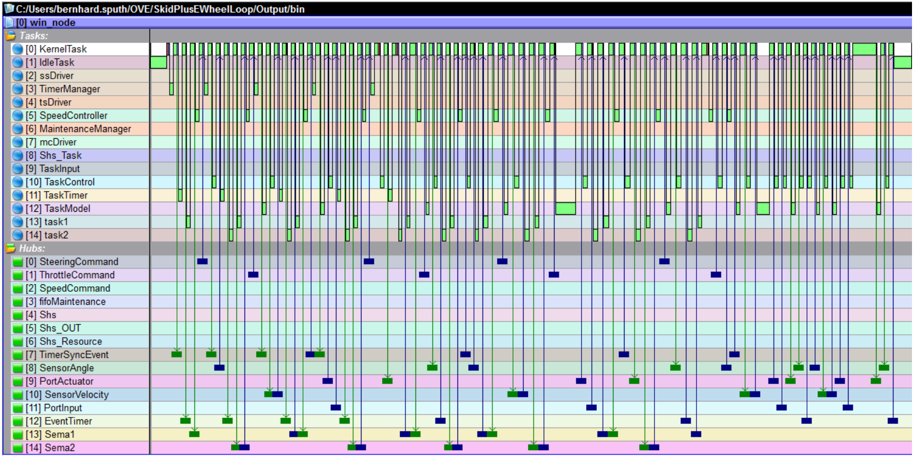
- Memory regions composed from 4kiB pages.
- Initialisation of the MMU is complex.
- Context Switch must reconfigure the MMU, impact on run-time;

Impact of Task Priorities in VirtuosoNext 1/2



Three applications at different Priorities.

Impact of Task Priorities in VirtuosoNext 2/2



C:/Users/bernhard.sputh/OVE/SkidPlusEWheelLoop/Output/bin

[0] win_node

Tasks:

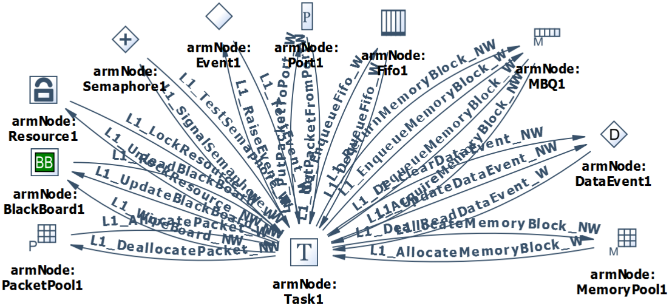
- [0] KernelTask
- [1] IdleTask
- [2] ssDriver
- [3] TimerManager
- [4] tsDriver
- [5] SpeedController
- [6] MaintenanceManager
- [7] mcDriver
- [8] Shs_Task
- [9] TaskInput
- [10] TaskControl
- [11] TaskTimer
- [12] TaskModel
- [13] task1
- [14] task2

Hubs:

- [0] SteeringCommand
- [1] ThrottleCommand
- [2] SpeedCommand
- [3] fifoMaintenance
- [4] Shs
- [5] Shs_OUT
- [6] Shs_Resource
- [7] TimerSyncEvent
- [8] SensorAngle
- [9] PortActuator
- [10] SensorVelocity
- [11] PortInput
- [12] EventTimer
- [13] Sema1
- [14] Sema2

Three applications at the same Priority.

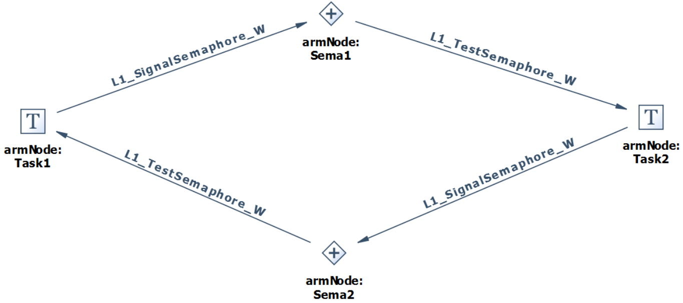
Code size differences between OpenComRTOS-1.6 and VirtuosoNext



	OCR	VN	Difference
ARM-Cortex-M3	18800 B	19060 B	+360 B
ARM-Cortex-A9	20232 B	26932 B	+6700 B

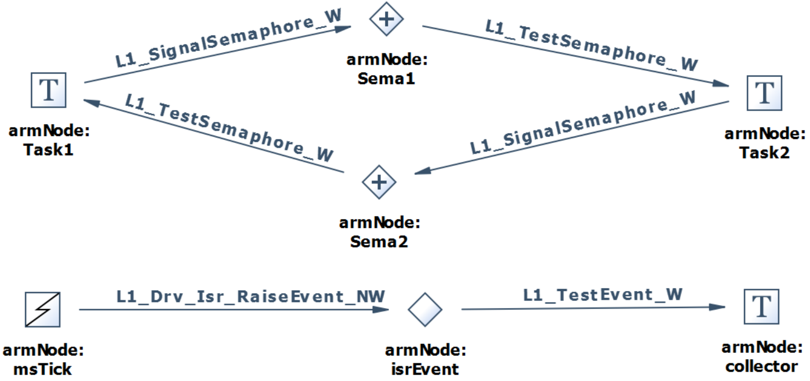
Adding memory protection has a limit impact on the code size.

Impact of Memory Protection on SemaphoreLoop Runtimes



	OCR	VN	Difference
ARM-Cortex-M3 (50MHz)	54.6μs	58.9μs	+4.3μs
ARM-Cortex-A9 (700MHz)	11.59μs	14.89μs	+3.3μs

Interrupt Latency Measurement Setup



Impact of Memory Protection on the Interrupt Latency

1. Interrupt to ISR Latency

	OCR	VN	Difference
ARM-Cortex-M3 (50MHz)	$780ns$	$780ns$	$\pm 0ns$
ARM-Cortex-A9 (700MHz)	$100ns$	$138ns$	$+38ns$

2. Interrupt to Task Latency

	OCR	VN	Difference
ARM-Cortex-M3 (50MHz)	$16\mu s$	$17\mu s$	$1\mu s$
ARM-Cortex-A9 (700MHz)	$994ns$	$1726ns$	$+732ns$

Adding memory protection has a limit impact on the run-time.

Conclusions

Comparing VirtuosoNext to a typical Hypervisor:

- Space partitioning does not require a lot of additional code.
- Lower memory consumption due to fine grain protection scheme.
- Tasks of each Application are still scheduled in order of Priority. Thus real-time behaviour is not affected by the protection.
- Hypervisors are suitable for soft-realtime applications, not for hard-realtime.

Questions?

Thank You
for Your attention

Altreonic

http://www.altreonic.com