

## Software Design Challenges for heterogenic SOC's

René Janssen, Product manager Logic Technology

**DESIGN AUTOMATION  
& EMBEDDED SYSTEMS**

FPGA - SECURITY - EMBEDDED - INTERNET OF THINGS - PCB TECHNOLOGIEËN - BLUETOOTH LE - ELECTRONIC DESIGN & PRODUCTION

**11 OKT** ←  
TECHNOPOLIS, BELGIË

**12 OKT** ←  
**BRABANTHALLEN, DEN BOSCH**

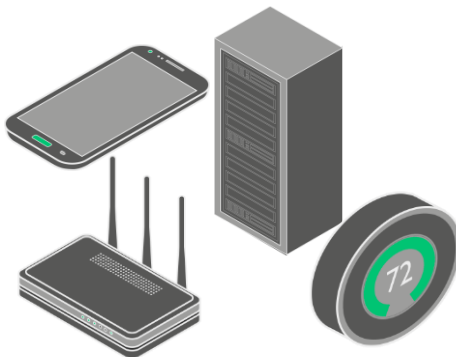
**D&E**  
event  
**2017**

1. Advantages of heterogenous devices
2. How to manage inter-processor communication
3. Example Hardware setup demo snapshots
4. Summary

## Cortex-A

Highest performance

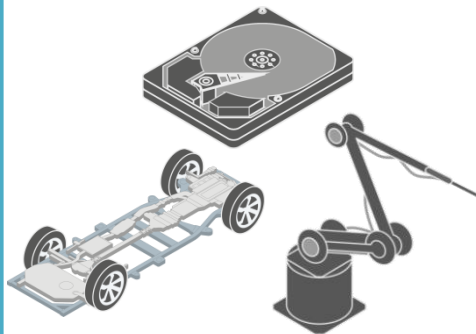
Optimized for  
rich operating systems



## Cortex-R

Fast response

Optimized for  
high performance,  
hard real-time applications



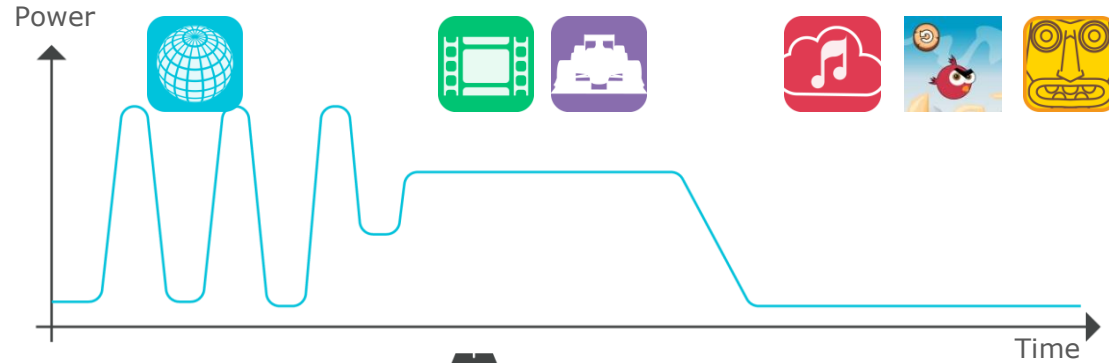
## Cortex-M

Smallest/lowest power

Optimized for  
discrete processing and  
microcontrollers



# Modern compute systems have diverse workloads



Ambient mode



Interactive mode



Sleep mode

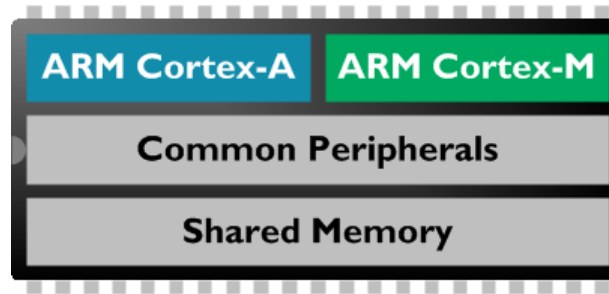


# Why heterogeneous devices?

Combines best of both worlds: feature-rich Linux and MCU with deterministic RTOS

## Cortex-A benefits

- Feature rich OS
- Complex applications
- Sophisticated HMI



## Cortex-M benefits

- Low I/O latency
- Low power standby
- Fast system start-up time

Linux application



Inter-processor communication

RTOS application



# Use cases of HMP systems in embedded



Industrial



Consumer



Medical

Cortex-A	Rich UI and OS, high performance		
Cortex-M	Real-time control and monitoring	Deterministic sensor control	Real-time monitoring



**Task  
partitioning**

How to optimally  
partition tasks?



**Data  
sharing**

Is coherency  
necessary?

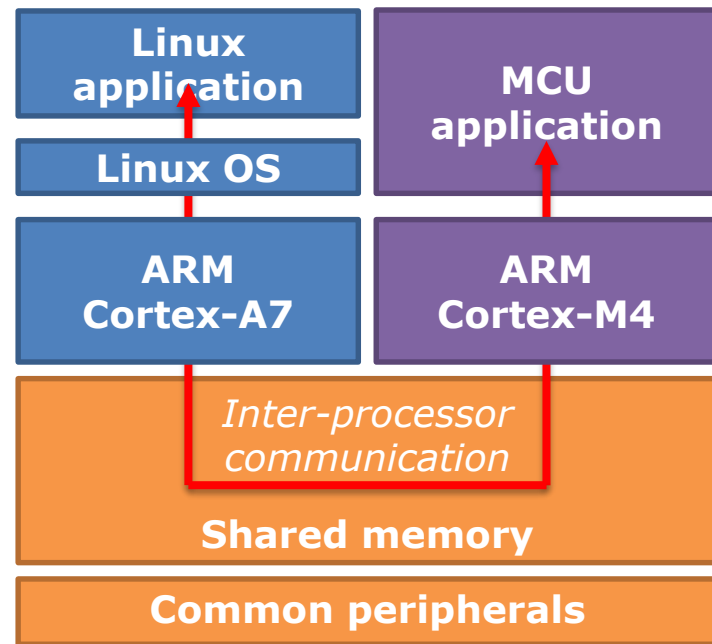


**Developer  
productivity**

Usability,  
portability,  
debugging

## Complexity increases with heterogeneous devices

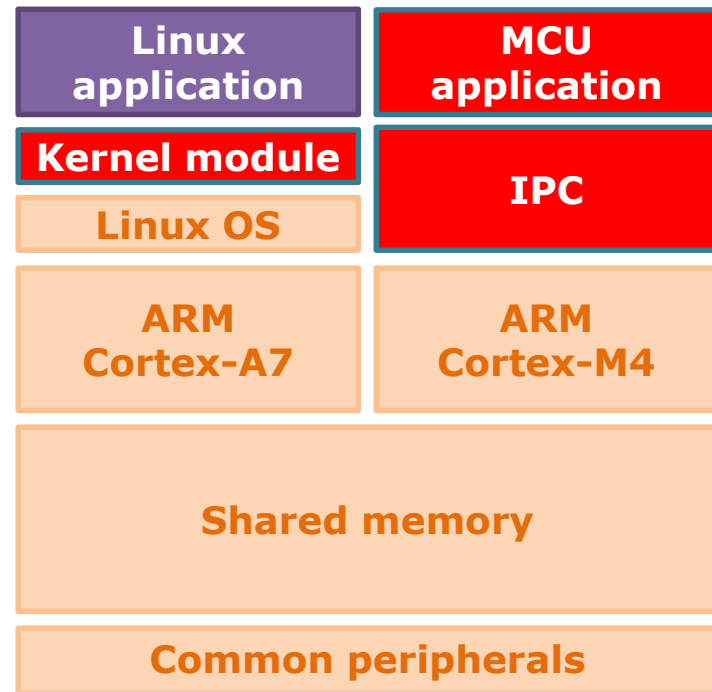
- Running multiple operating systems
- Debugging bare-metal code and Linux applications at the same time
- Controlling communication between Linux application and bare-metal application





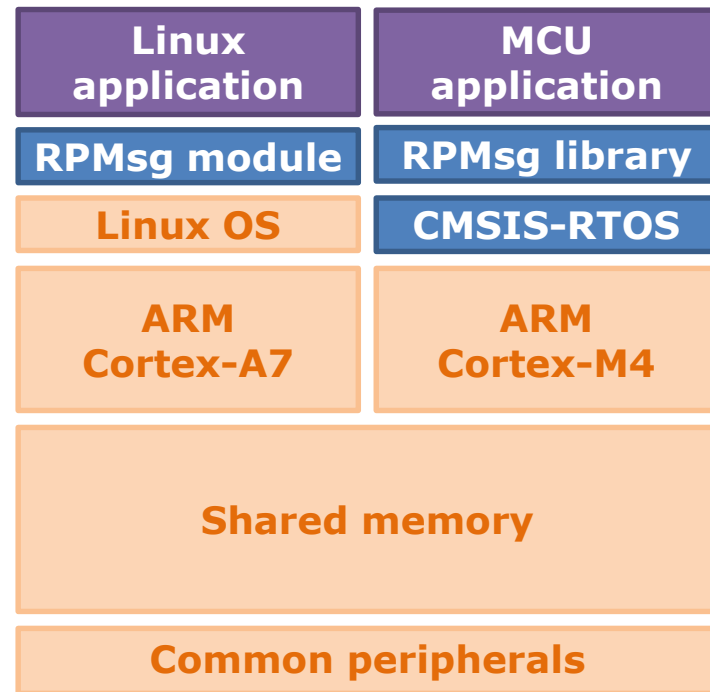
## Manually manage communication between Cortex-A and Cortex-M

- Requires to write a Linux kernel module
  - » Memory segmentation
  - » Manage concurrency
  - » Kernel mode to User mode communication
- Bare-metal
  - » Manage concurrency
  - » Integration with RTOS
  - » Handle interrupts and memory management



## Standardization of software interfaces

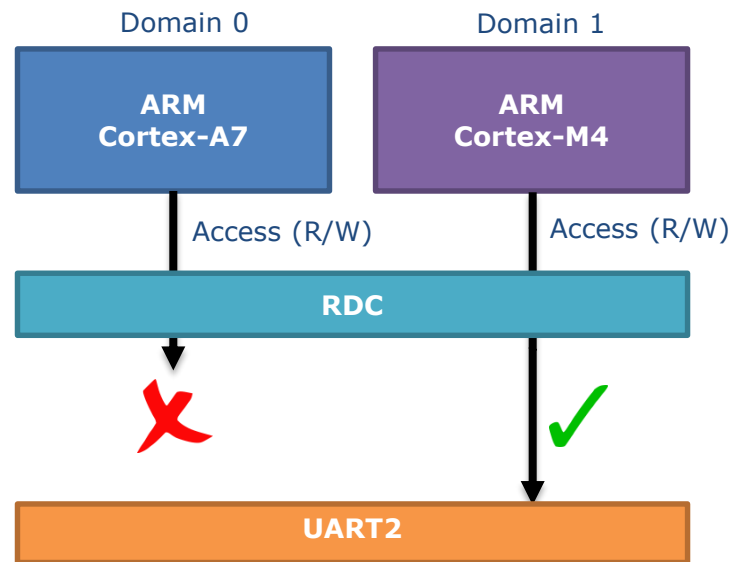
- CMSIS adopting OpenAMP
  - » CMSIS - Cortex Microcontroller Software Interface Standard
  - » Open source on Github
- OS support for HMP systems
  - » Remote Processor Messaging (RPMsg) for inter-processor communication
  - » Management framework using remoteproc



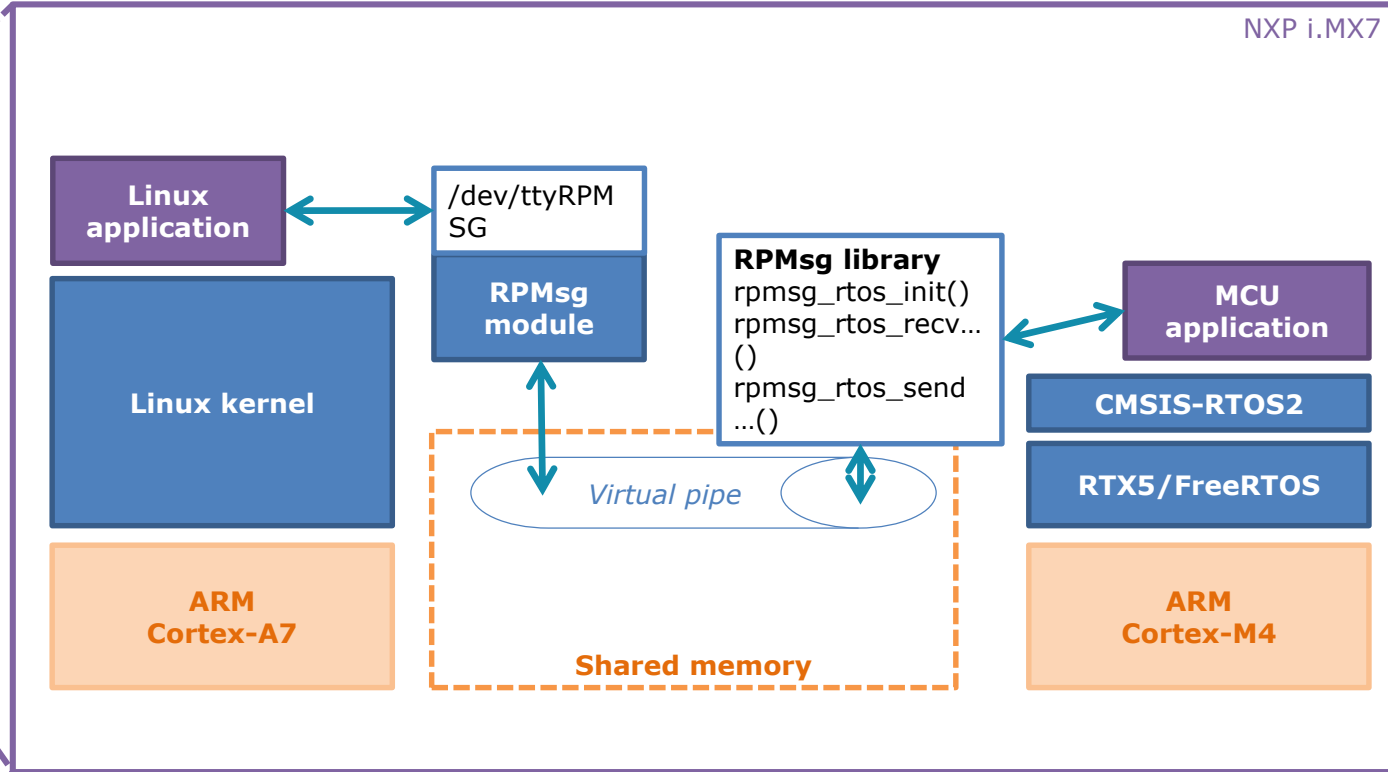
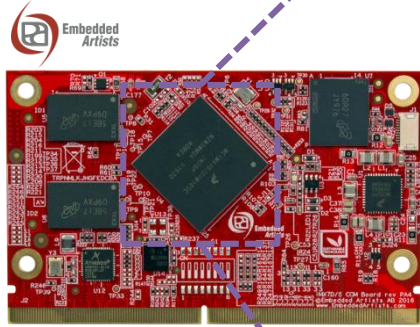
- Improve security by assigning peripherals to different domains
- Reduce risk of concurrent access to peripherals

## Example code

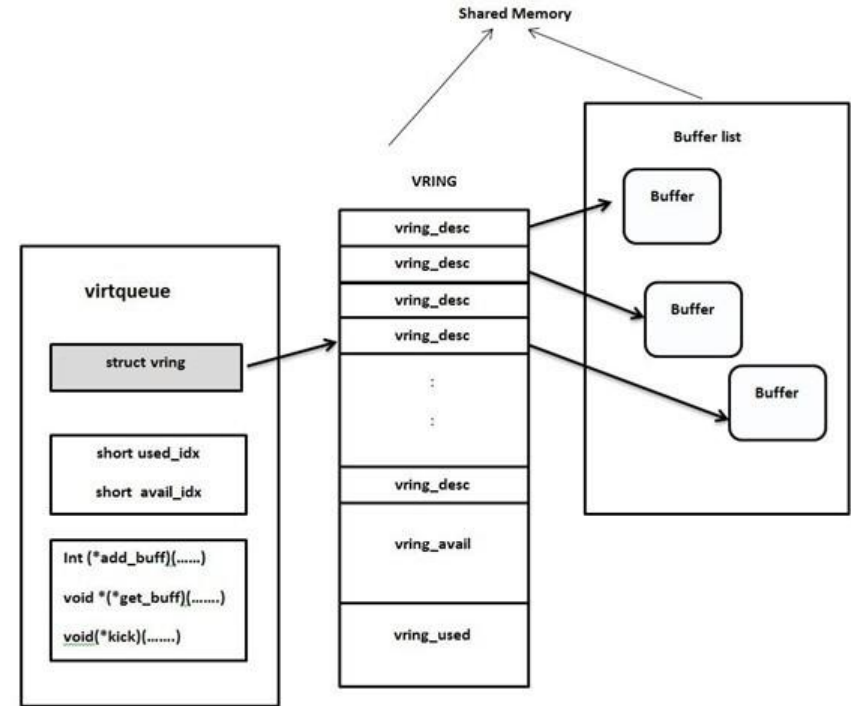
```
#define BOARD_DEBUG_UART_RDC_PDAP rdcPdapUart2
#define M4_DOMAIN_ID 1
...
/* Set UART2 for M4 core domain access only */
RDC_SetPdapAccess(RDC,
    BOARD_DEBUG_UART_RDC_PDAP,
    3 << (M4_DOMAIN_ID * 2),
    false, false);
```



# Inter-processor communication



- Virtio provides **virtqueue** API that allows user drivers to transmit and receive data with the communicating counterpart using the **vring** structure.
- **Vring** is a buffer management component
  - » Ring data structure to manage buffer descriptors located in shared memory.
- Inter-Processor Interrupts is used for notifications



Ref: <https://github.com/OpenAMP/open-amp/wiki/OpenAMP-RPMsg-Virtio-Implementation>

## Performance

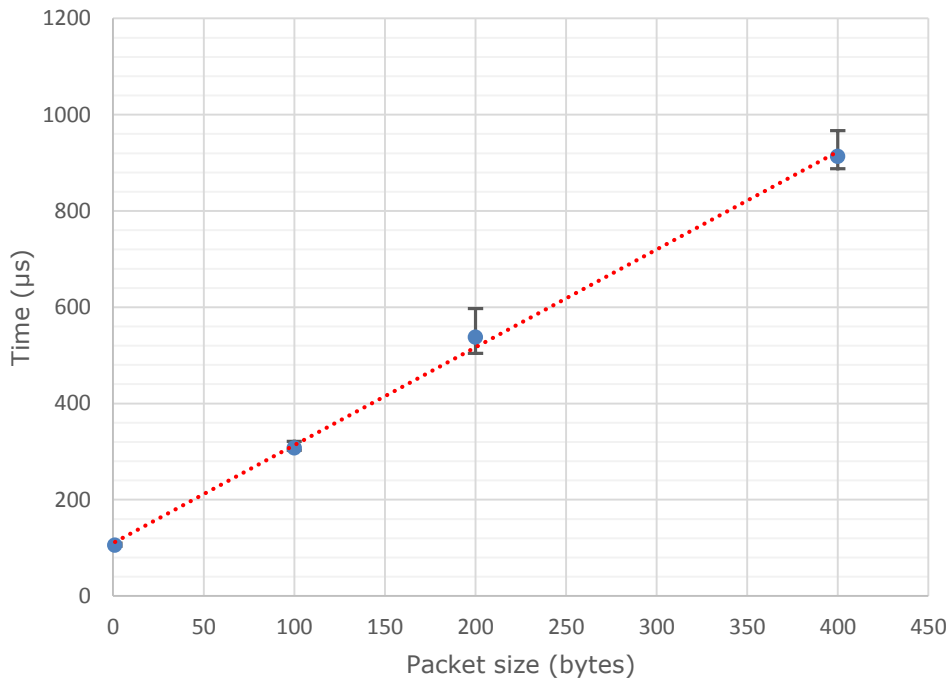
### Measurement of round-trip time



**50  $\mu$ s**  
Setup time

**$\sim 1\text{Mb/s}$**   
Bandwidth

RPMsg round-trip time



# Typical Hardware setup

## Embedded Artists in a Nutshell

- NXP, ARM based Computer-on-Module solutions.
- A family of pin-compatible boards.
- Industrial focus, with 10+ years longevity.
- A Proven Partner to NXP.
- High quality technical support directly from our engineers.
- The Art of Embedded Systems Development – made EASY™





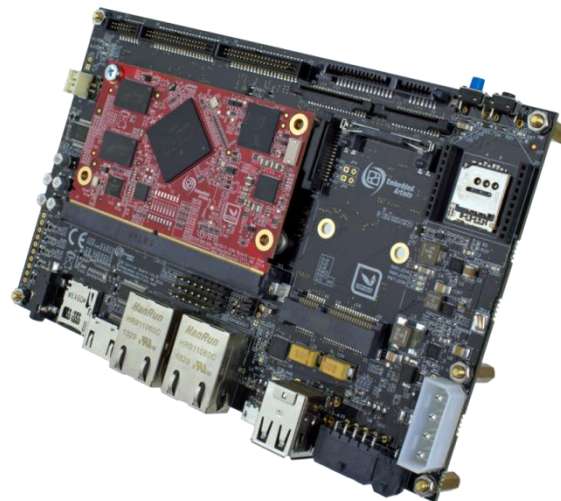
## iMX Developer's Kits

- Flexible hardware platforms..
- Documentation and guides...
- High quality support...

...to get up-and-running quickly!  
...for Prototypes and Proof-of-concepts!

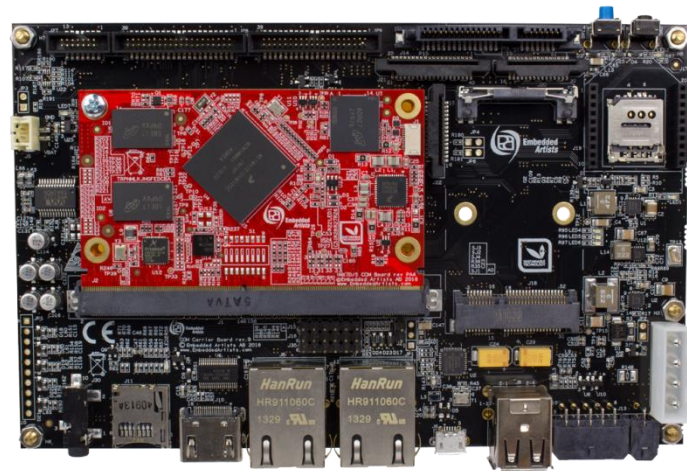
### Documentation / guides

- Getting Started with the iMX Developer's Kit
- Working with Yocto to Build Linux
- Working with Cortex-M4
- Getting started with Android (v5.1.1)
- Interface Testing on iMX Developer's Kit
- EACOM Selection Guide
- Display Solutions for COM Boards
- Adding Displays to iMX Developer's Kits
- Wireless Communication on iMX Developer's Kit
- Developing with Python
- Developing with Qt5
- Developing with C
- Developing with Node.js

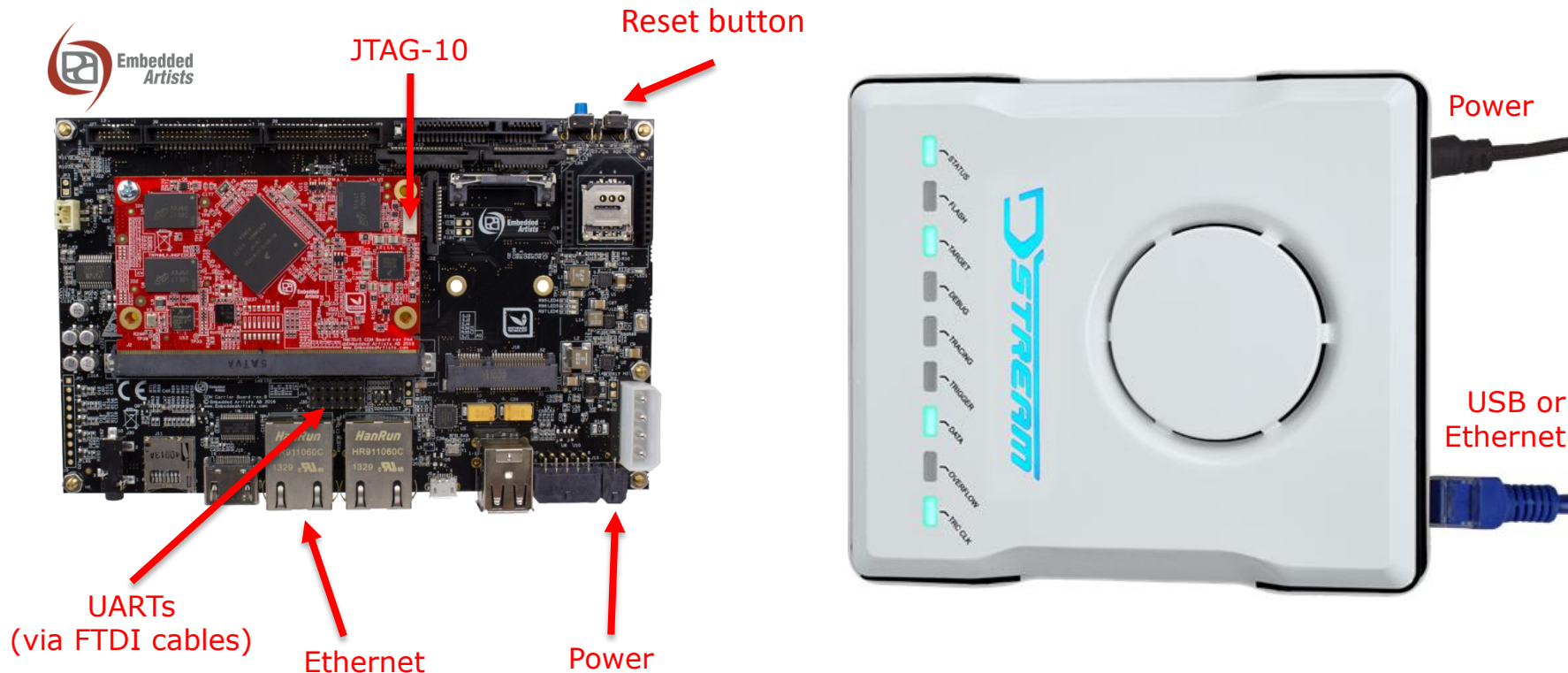


## Development board

- iMX7 Dual Developer's Kit
  - » 1 GHz dual-core Cortex-A7 (and 200 MHz Cortex-M4)
  - » 1 GByte DDR3L RAM
  - » 4 Gbyte eMMC Flash
  - » One Gigabit Ethernet interface
- To get started with an iMX Developer's Kit
  - » [http://www.embeddedartists.com/com\\_getting\\_started](http://www.embeddedartists.com/com_getting_started)



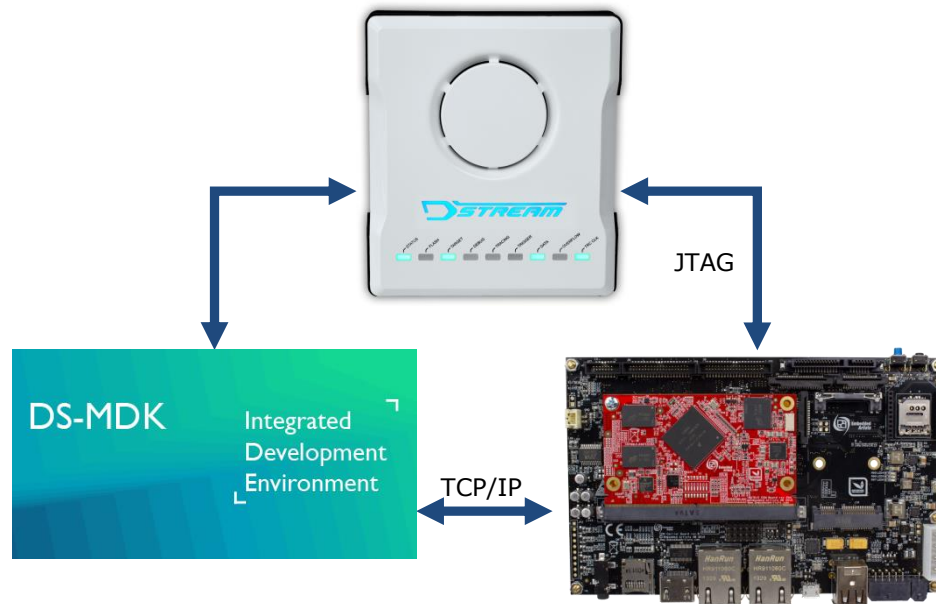
## Embedded Artists IMX7D Developer Kit + Arm DSTREAM-ST



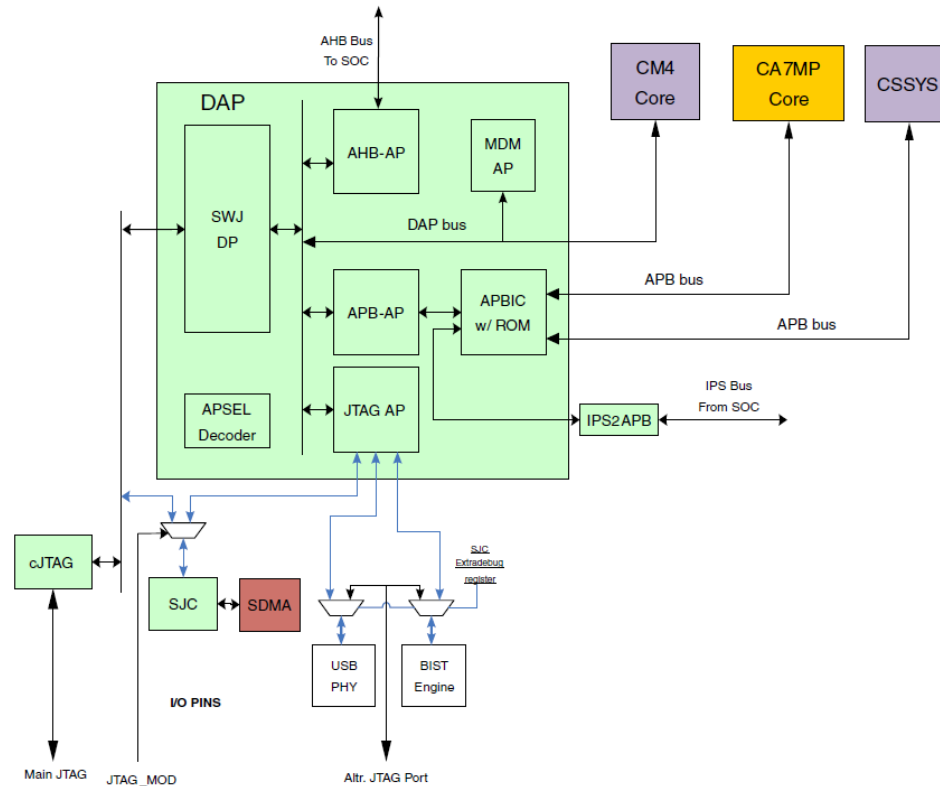
# Multiple simultaneous debug connections

## Complete visibility to all software applications in the heterogeneous system

- The Cortex-M application via DSTREAM and analyse with RTOS awareness and peripheral views.
- The Cortex-A Linux kernel and Linux kernel modules via DSTREAM and list kernel threads and processes.
- The Cortex-A Linux application via gdbserver on the running Linux system using Remote System Explorer.



# Coresight debug interface



**The i.MX7D debug is based on ARM's CoreSight "HUGO" platform, with support for Cortex-A7 and Cortex-M4 core from ARM.**

- Support 5-pins (JTAG), 2-pins (cJTAG, ARM SWD) interface
- Support both non-intrusive and halt-mode trace / debug options
- MDM-AP registers for debugger to control multi-core halt / resume cores
- Trace Memory Controller (TMC) is used to enable capturing trace
  - » 4 KB in SOC trace block
  - » ETR (4 G memory range, 64-bit wide at 266 MHz) is used to allow routing tracedata to system memory
- Support ARM real time trace interface: TPIU (16-bit x 133 MHz)
- Support cross trigger between CA7 and CM4
- Four JTAG security levels, via SJC security functions together with e-Fuse(challenge response, field return, intrusive detection)

# Demo

The screenshot displays the DS-5 Debug IDE interface for a Cortex-M4 target. The main window shows the source code of `ttt_rtx.c` with a breakpoint set at line 117. The `Debug Control` pane indicates that the application has terminated and the thread is stopped. The `Commands` pane shows the execution flow, including the `break -p` command and the `break-script` command. The `Tasks` pane lists the tasks running on the target, including `osIdle_demon`, `osTimerThread`, `main`, and `TTYThread`. The `Trace` pane shows the execution trace, including the `Buffer Used: 4.0 KB` and the `Trace discontinuity (trace buffer has wrapped)` message.

**Source Code (ttt\_rtx.c):**

```

113 result = rpsmg_rtos_send_nocopy(app_chnl->rp_ept, tx_buf, 14, sr
114 }
115 }
116 }
117 }
118 int main(void)
119 {
120     hardware_init();
121 }
122 SystemCoreClockUpdate();
123 //InitRetargetIOUSART();
124 //InitRetargetIOUSART();
125 /* Initialize CMSIS-RTOS */
126 osKernelInitialize();
127 /* Create new thread */
128 osThreadCreate(osThread(TTYThread), NULL);
129 /* Start thread execution */
130 osKernelStart();
131 /* Infinite loop */
132 while(true);
133 }
134 }
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }
144 }
145 }
146 }
147 }
148 }
149 }
150 }
151 }
152 }
153 }
154 }
155 }
156 }
157 }
158 }
159 }
160 }
161 }
162 }
163 }
164 }
165 }
166 }
167 }
168 }
169 }
170 }
171 }
172 }
173 }
174 }
175 }
176 }
177 }
178 }
179 }
180 }
181 }
182 }
183 }
184 }
185 }
186 }
187 }
188 }
189 }
190 }
191 }
192 }
193 }
194 }
195 }
196 }
197 }
198 }
199 }
200 }
201 }
202 }
203 }
204 }
205 }
206 }
207 }
208 }
209 }
210 }
211 }
212 }
213 }
214 }
215 }
216 }
217 }
218 }
219 }
220 }
221 }
222 }
223 }
224 }
225 }
226 }
227 }
228 }
229 }
230 }
231 }
232 }
233 }
234 }
235 }
236 }
237 }
238 }
239 }
240 }
241 }
242 }
243 }
244 }
245 }
246 }
247 }
248 }
249 }
250 }
251 }
252 }
253 }
254 }
255 }
256 }
257 }
258 }
259 }
260 }
261 }
262 }
263 }
264 }
265 }
266 }
267 }
268 }
269 }
270 }
271 }
272 }
273 }
274 }
275 }
276 }
277 }
278 }
279 }
280 }
281 }
282 }
283 }
284 }
285 }
286 }
287 }
288 }
289 }
290 }
291 }
292 }
293 }
294 }
295 }
296 }
297 }
298 }
299 }
300 }
301 }
302 }
303 }
304 }
305 }
306 }
307 }
308 }
309 }
310 }
311 }
312 }
313 }
314 }
315 }
316 }
317 }
318 }
319 }
320 }
321 }
322 }
323 }
324 }
325 }
326 }
327 }
328 }
329 }
330 }
331 }
332 }
333 }
334 }
335 }
336 }
337 }
338 }
339 }
340 }
341 }
342 }
343 }
344 }
345 }
346 }
347 }
348 }
349 }
350 }
351 }
352 }
353 }
354 }
355 }
356 }
357 }
358 }
359 }
360 }
361 }
362 }
363 }
364 }
365 }
366 }
367 }
368 }
369 }
370 }
371 }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 }
381 }
382 }
383 }
384 }
385 }
386 }
387 }
388 }
389 }
390 }
391 }
392 }
393 }
394 }
395 }
396 }
397 }
398 }
399 }
400 }
401 }
402 }
403 }
404 }
405 }
406 }
407 }
408 }
409 }
410 }
411 }
412 }
413 }
414 }
415 }
416 }
417 }
418 }
419 }
420 }
421 }
422 }
423 }
424 }
425 }
426 }
427 }
428 }
429 }
430 }
431 }
432 }
433 }
434 }
435 }
436 }
437 }
438 }
439 }
440 }
441 }
442 }
443 }
444 }
445 }
446 }
447 }
448 }
449 }
450 }
451 }
452 }
453 }
454 }
455 }
456 }
457 }
458 }
459 }
460 }
461 }
462 }
463 }
464 }
465 }
466 }
467 }
468 }
469 }
470 }
471 }
472 }
473 }
474 }
475 }
476 }
477 }
478 }
479 }
480 }
481 }
482 }
483 }
484 }
485 }
486 }
487 }
488 }
489 }
490 }
491 }
492 }
493 }
494 }
495 }
496 }
497 }
498 }
499 }
500 }
501 }
502 }
503 }
504 }
505 }
506 }
507 }
508 }
509 }
510 }
511 }
512 }
513 }
514 }
515 }
516 }
517 }
518 }
519 }
520 }
521 }
522 }
523 }
524 }
525 }
526 }
527 }
528 }
529 }
530 }
531 }
532 }
533 }
534 }
535 }
536 }
537 }
538 }
539 }
540 }
541 }
542 }
543 }
544 }
545 }
546 }
547 }
548 }
549 }
550 }
551 }
552 }
553 }
554 }
555 }
556 }
557 }
558 }
559 }
560 }
561 }
562 }
563 }
564 }
565 }
566 }
567 }
568 }
569 }
570 }
571 }
572 }
573 }
574 }
575 }
576 }
577 }
578 }
579 }
580 }
581 }
582 }
583 }
584 }
585 }
586 }
587 }
588 }
589 }
590 }
591 }
592 }
593 }
594 }
595 }
596 }
597 }
598 }
599 }
600 }
601 }
602 }
603 }
604 }
605 }
606 }
607 }
608 }
609 }
610 }
611 }
612 }
613 }
614 }
615 }
616 }
617 }
618 }
619 }
620 }
621 }
622 }
623 }
624 }
625 }
626 }
627 }
628 }
629 }
630 }
631 }
632 }
633 }
634 }
635 }
636 }
637 }
638 }
639 }
640 }
641 }
642 }
643 }
644 }
645 }
646 }
647 }
648 }
649 }
650 }
651 }
652 }
653 }
654 }
655 }
656 }
657 }
658 }
659 }
660 }
661 }
662 }
663 }
664 }
665 }
666 }
667 }
668 }
669 }
670 }
671 }
672 }
673 }
674 }
675 }
676 }
677 }
678 }
679 }
680 }
681 }
682 }
683 }
684 }
685 }
686 }
687 }
688 }
689 }
690 }
691 }
692 }
693 }
694 }
695 }
696 }
697 }
698 }
699 }
700 }
701 }
702 }
703 }
704 }
705 }
706 }
707 }
708 }
709 }
710 }
711 }
712 }
713 }
714 }
715 }
716 }
717 }
718 }
719 }
720 }
721 }
722 }
723 }
724 }
725 }
726 }
727 }
728 }
729 }
730 }
731 }
732 }
733 }
734 }
735 }
736 }
737 }
738 }
739 }
740 }
741 }
742 }
743 }
744 }
745 }
746 }
747 }
748 }
749 }
750 }
751 }
752 }
753 }
754 }
755 }
756 }
757 }
758 }
759 }
760 }
761 }
762 }
763 }
764 }
765 }
766 }
767 }
768 }
769 }
770 }
771 }
772 }
773 }
774 }
775 }
776 }
777 }
778 }
779 }
780 }
781 }
782 }
783 }
784 }
785 }
786 }
787 }
788 }
789 }
790 }
791 }
792 }
793 }
794 }
795 }
796 }
797 }
798 }
799 }
800 }
801 }
802 }
803 }
804 }
805 }
806 }
807 }
808 }
809 }
810 }
811 }
812 }
813 }
814 }
815 }
816 }
817 }
818 }
819 }
820 }
821 }
822 }
823 }
824 }
825 }
826 }
827 }
828 }
829 }
830 }
831 }
832 }
833 }
834 }
835 }
836 }
837 }
838 }
839 }
840 }
841 }
842 }
843 }
844 }
845 }
846 }
847 }
848 }
849 }
850 }
851 }
852 }
853 }
854 }
855 }
856 }
857 }
858 }
859 }
860 }
861 }
862 }
863 }
864 }
865 }
866 }
867 }
868 }
869 }
870 }
871 }
872 }
873 }
874 }
875 }
876 }
877 }
878 }
879 }
880 }
881 }
882 }
883 }
884 }
885 }
886 }
887 }
888 }
889 }
890 }
891 }
892 }
893 }
894 }
895 }
896 }
897 }
898 }
899 }
900 }
901 }
902 }
903 }
904 }
905 }
906 }
907 }
908 }
909 }
910 }
911 }
912 }
913 }
914 }
915 }
916 }
917 }
918 }
919 }
920 }
921 }
922 }
923 }
924 }
925 }
926 }
927 }
928 }
929 }
930 }
931 }
932 }
933 }
934 }
935 }
936 }
937 }
938 }
939 }
940 }
941 }
942 }
943 }
944 }
945 }
946 }
947 }
948 }
949 }
950 }
951 }
952 }
953 }
954 }
955 }
956 }
957 }
958 }
959 }
960 }
961 }
962 }
963 }
964 }
965 }
966 }
967 }
968 }
969 }
970 }
971 }
972 }
973 }
974 }
975 }
976 }
977 }
978 }
979 }
980 }
981 }
982 }
983 }
984 }
985 }
986 }
987 }
988 }
989 }
990 }
991 }
992 }
993 }
994 }
995 }
996 }
997 }
998 }
999 }
1000 }

```

**Commands:**

```

break -p "C:\Users\stcad01\Documents\DS-5\MDK_Workspace_EmbeddedArtists\RPMSG_TTY_1
Breakpoint 4 at 0x1FFFB9C
on file ttt_rtx.c, line 79
condition 4
break-script 4 ""
ignore 4 0
break-stop-on-cores 4
unsilence 4
Breakpoint

```

**Tasks:**

Task	Name	Priority	State
255	osIdle_demon	0	READY
1	osTimerThread	6	WAIT_MBX
2	main	4	RUNNING
3	TTYThread	4	WAIT_MBX

**Trace:**

```

Trace Capture Device Source Ranges
Buffer Used: 4.0 KB
Index Address Opcode Detail
Trace discontinuity (trace buffer has wrapped)
Trace buffer contains no trace for this core.

```

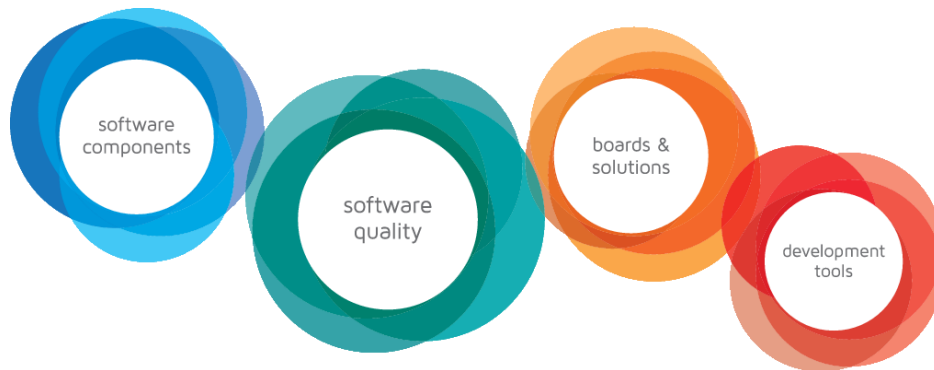
**App Console:**

```

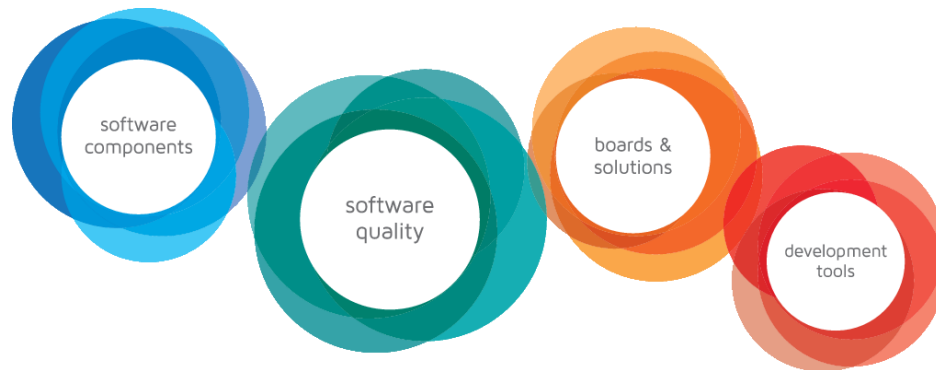
Serial: (COM5, 115200, 8, 1, None, None - CONNECTED) - Encoding (ISO-8859-1)
/dev/tty17 /dev/tty28 /dev/tty39 /dev/tty5 /dev/tty60
/dev/tty18 /dev/tty29 /dev/tty4 /dev/tty50 /dev/tty61
root@imx7dea-com:~#
root@imx7dea-com:~#
root@imx7dea-com:~# virtio_rpmsg_bus virtio0: creating channel rpmsg-openamp-demo-channel addr 0x0
imx_rpmsg_tty rpmsg0: new channel: 0x400 -> 0x0
Install rpmsg tty driver!
root@imx7dea-com:~#
root@imx7dea-com:~# ls /dev/ttyRPMSG
/dev/ttyRPMSG

```





## Do you have any questions?



**Thank you for your attention**

See you at the Logic Technology stand