Impact Case Studies Real-Time Systems

Robert Davis

Real-Time Systems Research Group, University of York

rob.davis@york.ac.uk
http://www-users.cs.york.ac.uk/~robdavis/

Overview

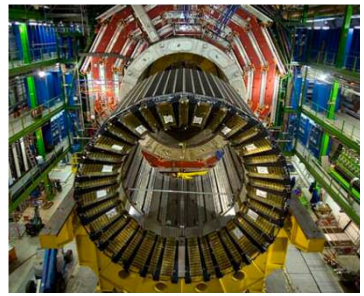
- Background on Real-Time Systems
- Case studies
  - 1. Guaranteeing the real-time performance of in-vehicle networks
  - 2. The world's smallest automotive real-time operating system
  - 3. How long does your real-time software take to run?
- Questions?

What is a Real-Time System?

- Real-Time System is any system which has to respond to externally generated input stimuli within a specified time
  - Functional correctness – the right computations
  - Timing correctness – completed within predefined time constraints
  - Time constraints typically expressed in terms of deadlines

- Hard Real-Time
  - Failure to meet a deadline constitutes a failure of the application (e.g. flight control system)
- Soft Real-Time
  - Latency in excess of the deadline leads to degraded quality of service (e.g. data acquisition, video playback)

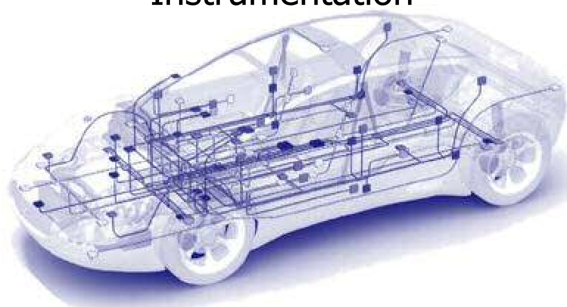
Examples of Real-Time Systems



Instrumentation

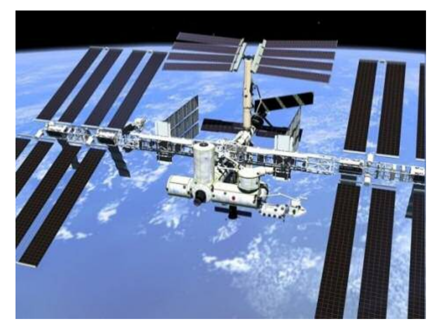
Robotics and Factory Automation

Avionics



Automotive Electronics

Telecommunications

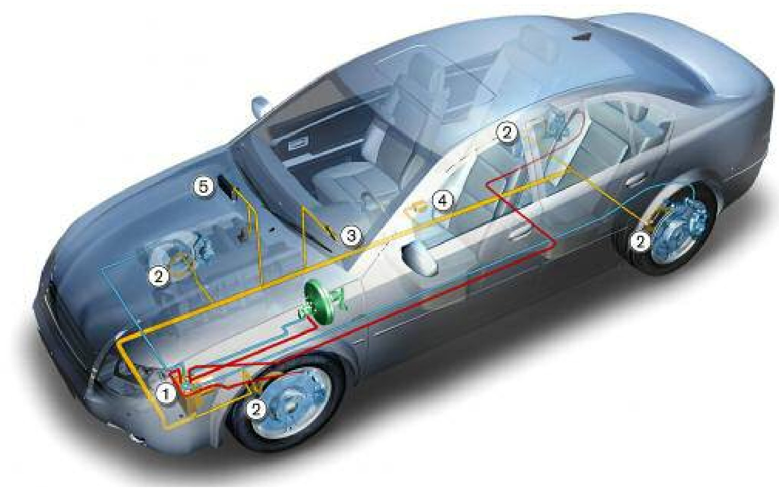


Space

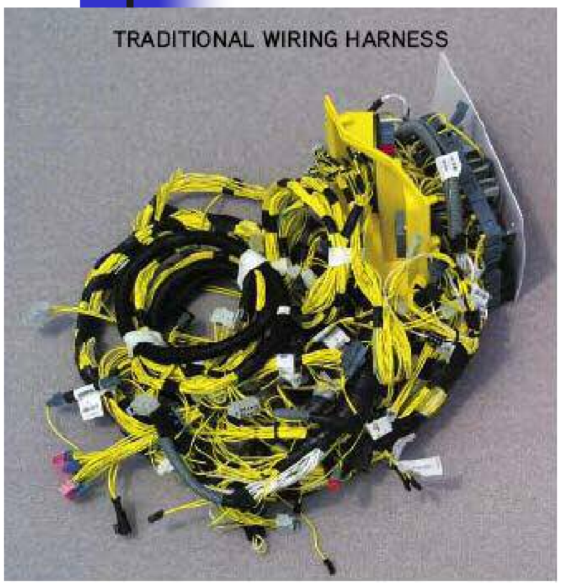
Medical Systems

Case Study 1: Guaranteeing the real-time performance of in-vehicle networks

- Controller Area Network (CAN)
  - Simple, robust and efficient, in-vehicle digital communications network
  - Originally developed by BOSCH in the 1980s
  - First used in a production car in the 1991 Mercedes S-Class



Multiplex v. Point-to-point Wiring



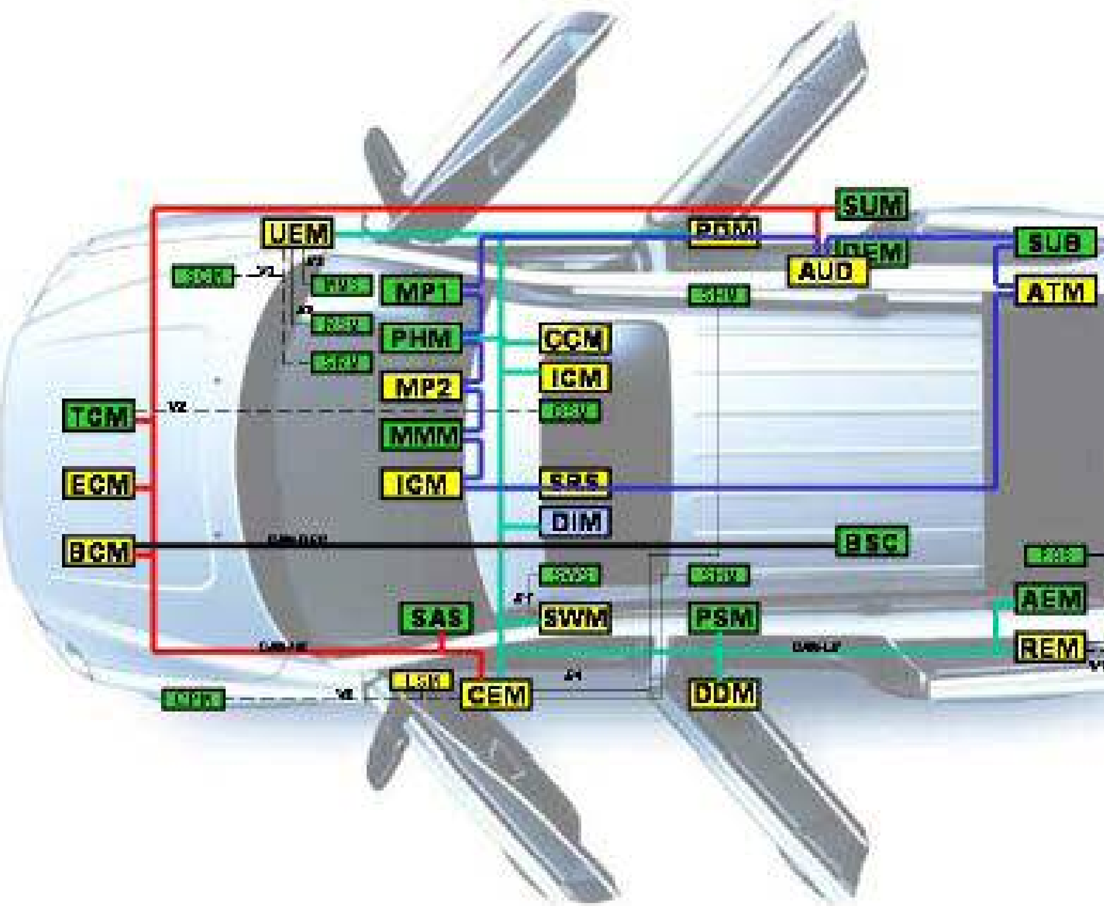
- Traditional point-to-point wiring
  - Early 1990s an average luxury car had:
    - 30Kg wiring harness
    - > 1km of copper wire
    - > 300 connectors, 2000 terminals, 1500 wires
  - Expensive to manufacture, install and maintain
    - Example: Door system with 50+ wires

- Multiplex approach (e.g. CAN)
  - Massive reduction in wiring costs
    - Example: Door system reduced to just 4 wires
  - Small added cost of CAN controllers, transceivers etc.
    - Reduced as CAN devices became on-chip peripherals

Messages on CAN

- CAN used to communicate *signals* between Electronic Control Units (ECUs)
  - Typically 25-35 ECUs in a modern car
  - Signals include:
    - wheel speeds, oil and water temperature, battery voltage, engine rpm, gear selection, accelerator position, dashboard switch positions, climate control settings, window switch positions, fault codes, diagnostic information etc.
  - > 2,500 signals in a high-end vehicle
  - Multiple signals piggybacked into CAN messages to reduce overhead, but still 100's of CAN messages
- Real-time constraints on signal transmission
  - End-to-end deadlines in the range 10ms – 1sec
  - Example LED brake lights

Volvo XC90 Network Architecture



Volvo XC90

500 Kbit/sec CAN bus for power train

125 Kbit/sec CAN bus for body electronics

MOST (infotainment system)

Schedulability Analysis for CAN

- Research from RTSRG* in 1994
  - Mathematical analysis to compute offline the longest time that each message can take to be transmitted over the network (including time in queues)
  - Used to prove if all messages are guaranteed to meet their deadlines

- Schedulability Analysis
  - Message Length $$C_m = \left(g + 8s_m + 13 + \left\lfloor \frac{g + 8s_m - 1}{4} \right\rfloor \right) \tau_{bit}$$
  - Queuing delay $$w_m^{n+1} = B^{MAX} + \sum_{\forall k \in hp(m)} \left\lceil \frac{w_m^n + J_k + \tau_{bit}}{T_k} \right\rceil C_k$$
  - Response time $$R_m = J_m + w_m + C_m \leq D_m$$

*by Ken Tindell, Alan Burns, Andy Wellings

Start-up Company

- Origins
  - In 1994 research presented at a Conference on CAN was picked up by Volvo Car Corporation
- Start-up company **Northern Real-Time Applications Ltd**
  - Founded by Ken Tindell and Rob Davis in 1995
- Products developed
  - Analysis tools "Volcano Network Architect" with Swedish company Kimble AB.
  - CAN device drivers, communications software layer, and configuration tools called "Volcano Target Package"
- Rights transfer
  - Rights to Volcano technology transferred to Swedish company Volcano Communications Technologies AB in 1997
  - Acquired by Mentor Graphics in 2005

Exploitation

- Today Volcano technology is available for more than 30 different microprocessors used in automotive
  - Fujitsu 16LX, FR Series; Hitachi H8S, SH7055, SH7058; Infineon C16x, TC179x, TC176x, XC800, XC2000; Renesas M16C, R32C/M32C; Freescale HC08, HC12, MC683xx, MPC5xx, MAC71xx; S12, S12X, MPC55xx, MPC 56xx; Mitsubishi M32R, MC32C; PowerPC; National CR16; NEC V85x, 78K0; ST Microelectronics ST9, ST10; Texas Instruments TMS470; Toshiba TMP92/TMP94.

Impact

- Volcano Technology
  - First used in Volvo S80 in 1997

  - Subsequently in XC90, S80, S/V/XC70, S60, S40, and V50
  - Approx **4.5 million** Volvo cars since 1997
  - Ford bought Volvo in 1999: Volcano adopted by Jaguar, Land Rover, Aston Martin
  - Used by SAIC since 2007 and Mazda since 2012

Impact

- Benefits for car manufacturers and suppliers
  - Possible to configure networks using 70-80% of bandwidth and still guarantee that all messages meet their deadlines, compares favourably with approx. 30% previously possible
  - Enables more ECUs to be connected to the same network, supporting more functionality at lower cost
  - Fewer wires and connectors, lower network speeds needed, increases reliability
  - Guaranteed performance greatly reduces the time and cost spent testing: No intermittent timing faults due to network reduces warranty costs

- Benefits to wider society
  - More reliable cars with better functionality at lower cost

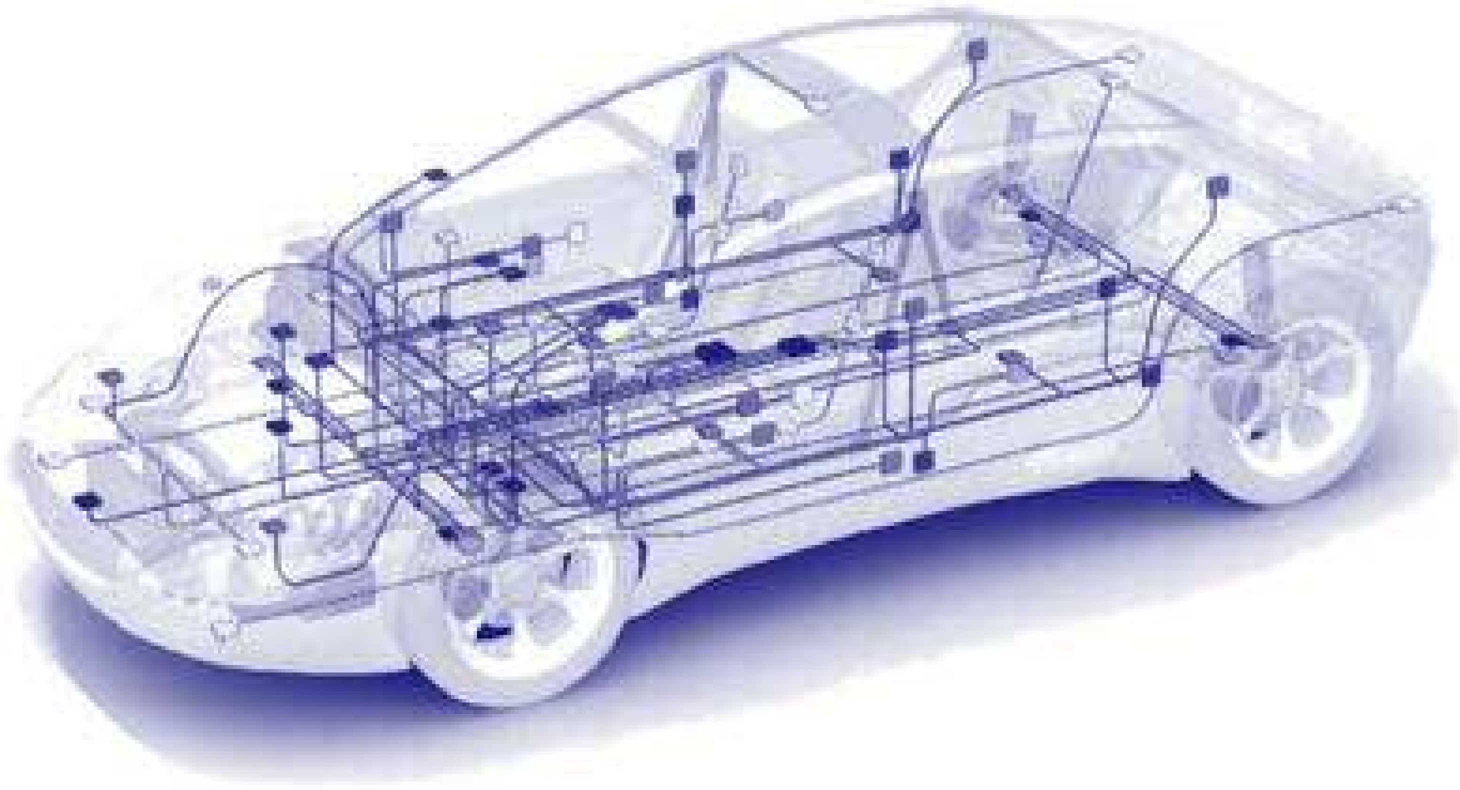
Impact

- Quotes
  - "*The advantages to Volvo of the development and application of Volcano include: Production cost benefits due to high bus efficiency (four times as many signals can be transmitted at half the baud rate). Development cost benefits (in the form of a single, proven implementation which is much cheaper than multiple implementations by suppliers and conformance testing by Volvo). Improved network reliability, resulting in higher product quality. Reduction in Volvo´s test load. Reduction in supplier´s test load.*"

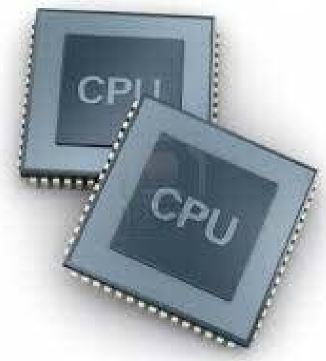
    Volvo 1998
  - "*By using Volcano, network design is made easy and predictable, guaranteeing data communication, which reduces the verification effort to almost zero and eliminates warranty and change costs caused by networking issues.*"

    SAIC 2006

Case Study 2: The world's smallest automotive real-time operating system



Automotive Electronics



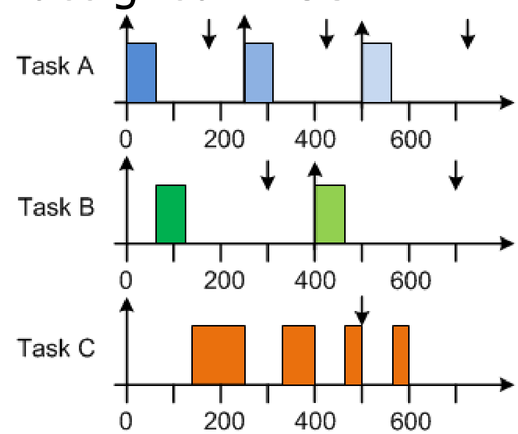
- Typical family car (e.g. VW Golf)
  - 25-35 Electronic Control Units connected via two or more communications networks
  - Relatively simple low cost microprocessors (single CPU)
- System functionality
  - Multiple software tasks running on each ECU
  - Time constraints on each task
    - e.g. read and process data, output results by a specified deadline
- Real-Time Operating System
  - Needed to schedule when each task can run so that all tasks meet their deadlines
  - Essential that all deadlines are met otherwise the system will suffer intermittent timing faults and poor reliability

Schedulability Analysis for Processors

- Research from RTSRG* in early 1990s
  - Response Time Analysis for fixed priority scheduling
  - Determines longest response time for each task from becoming ready to completing execution
  - Accounts for the complexities of timing behaviour for tasks in automotive systems
  - Accounts for the overheads of a well designed RTOS

$$w_{i,q}^{m+1} = B_i + (q+1)C_i + \sum_{\forall j \in hp(i)} \left\lceil \frac{w_{i,q}^{m} + J_j}{T_j} \right\rceil C_j$$

$$R_i = \max_{\forall q = 0,1,2 \ldots Q_i - 1} (w_{i,q} - qT_i + J_i)$$



*by Neil Audsley, Alan Burns, Ken Tindell, Andy Wellings

Start-up Company

- Origins
  - In 1997, following work for Volvo on Volcano Ken Tindell and Rob Davis founded **Northern Real-Time Applications Ltd**.
  - Purpose: to develop a RTOS for automotive applications

- Aims for the RTOS
  - Systems built using the RTOS must be analysable using schedulability analysis tools
  - RTOS overheads and memory footprint must be much smaller than any other automotive RTOS

Start-up Company

- Development
  - **Real-Time Architect** schedulability analysis tools
  - **RTA-OSEK** real-time operating system
- Standards
  - Influenced OSEK automotive operating system standard to allow single stack execution (enabled low memory use)
- Funding
  - £1M venture capital funding in 1999
  - £9.2M venture capital funding in 2000
- Jobs
  - Grew from <10 to more than 30 employees by 2001

Start-up Company

- Trade Sale
  - ETAS (a subsidiary of Bosch) benchmarked RTA-OSEK against their in-house RTOS and found it was much more efficient – faced with the option to start from scratch or buy the company
  - ETAS bought the company in 2003

Exploitation

- Further development
  - ETAS adapted the operating system to meets the new AUTOSAR standard (RTA-OS)

- RTA-OSEK / RTA-OS Available for over 50microprocessor families including:

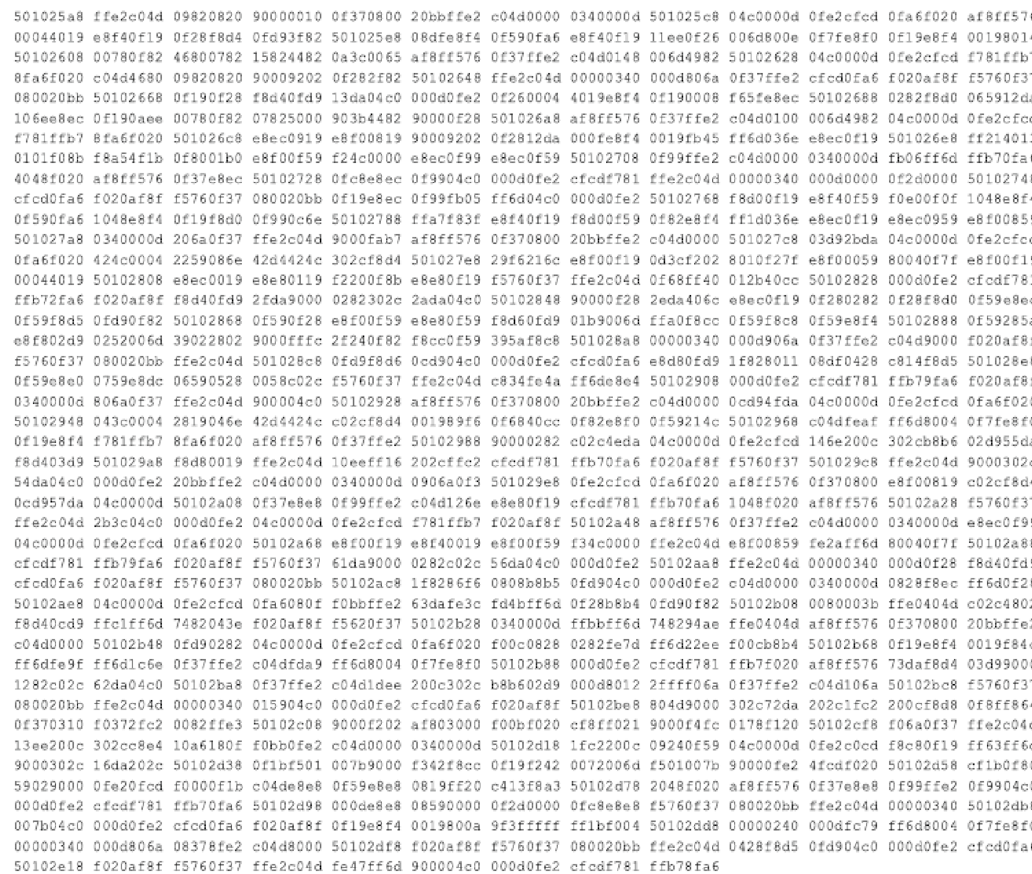
  Renesas: V850E, SH2, SH2A, H8S, H8SX, M16C, Xilinx Microblaze, PPC405 Core; Texas Instruments TMS470P, TMS570P; Infineon Tricore TC17x6, C166, XC2000; Freescale Star12, MPC555, MPC55xx, S12X, MPC56x, HC12X16, HC08, HCS12; Fujitsu 16LX; Analog Devices Blackfin, STMicroelectronics ST30, ST7, ST10

Advantages: Low memory usage

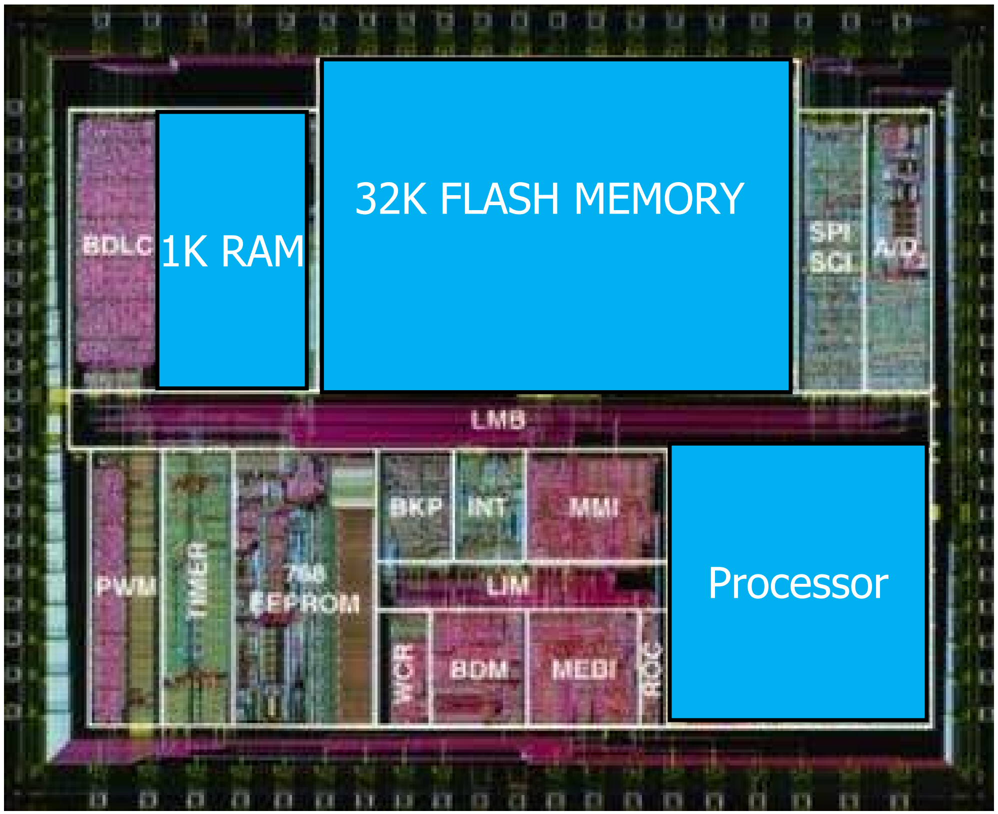
RTA-OS:

World’s smallest and fastest AUTOSAR compliant RTOS

Approx. 1K to 1.5K Bytes (depends on processor)



Advantages:
Low memory usage



Different microprocessor variants available with more / less memory at higher / lower cost

Save a few cents per chip by having less memory

x multiple ECUs per car

x millions of cars

= $$$

Advantages and Benefits

- Low memory footprint
  - Cheaper microprocessor variants reduce unit costs in production
- Low execution time overheads
  - Can include more useful functionality without upgrading to more expensive processors
- Analysable behaviour
  - Can guarantee timing behaviour leading to more reliable systems
  - Reduces time spent debugging intermittent timing problems
- Benefits
  - Reduced development, production and warranty costs for car companies and suppliers
  - Competitive market place hence benefits passed to consumers via less expensive and more reliable cars

Impact

- RTOS Deployment
  - RTOS used by most of the world's leading car companies and suppliers
  - In 2015, the total number of deployed copies of the RTOS was over

1 billion

  - This number is increasing at the quite astonishing rate of between 1 and 2 million new ECUs *per week*
  - Profitable business with substantial revenues
  - Sustained a large number of high tech jobs in York for over 15 years

Case Study 3: How long does your real-time software take to run?

- The Worst-Case Execution Time (WCET) problem
  - Finding the longest time that software components can take to run on a microprocessor is an important issue in embedded systems development
  - Overrunning execution time budgets can cause operational and reliability problems or worse
  - WCET needs to be tightly bounded to avoid the need for overprovisioned hardware

Measurement-based WCET analysis

- Research in RTSRG* from early 2000s
  - Set of hybrid and probabilistic techniques developed for WCET analysis
- RapiTime technology
  - Recognises that the best possible model of a processor is the processor itself – hence uses online testing to obtain measurements
  - Recognises that the best way to determine the overall structure of the code is offline analysis
  - Combines static analysis of the structure of the code and measurements of short sub-paths obtained via testing to obtain tight WCET estimates

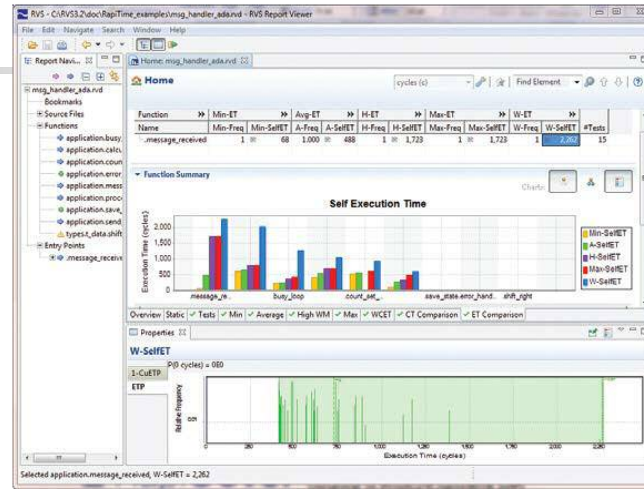
*By Guiem Bernat, Antoine Colin, and Alan Burns

Spin-out Company

- Origins
  - Guiem Bernat, Ian Broster, Antoine Colin and Rob Davis founded **Rapita Systems Ltd** in 2004.
  - Purpose: to develop **RapiTime** WCET analysis technology for aerospace and automotive applications
- Further developments
  - RapiTime: Extended to support programs written in C++, C, and Ada
  - RapiCover: Code coverage tool
  - RapiTask: Scheduling visualisation
  - Together form RVS (Rapita Verification Suite)
- Funding
  - £200k of investment from Viking Fund + Business Angels in 2005

Exploitation

- Initial prototypes
  - Technology demonstrated on an Audi drive-by-wire system – as part of an EU project
- BAE Systems
  - In 2006 RapiTime used on Hawk Jet Trainer project
  - Identified the 1% of many 100,000s of lines of code that contributed to nearly 1/3rd of the WCET
  - Optimisations to this 1% reduced the WCET by 23%
  - Received BAE Chairman's award for innovation



Impact

- Technology
  - RapiTime Technology deployed on major aerospace and automotive projects in the UK, Europe, Brazil, India, China, and the USA
  - Key customers include leading aerospace companies as well as major automotive suppliers as well as the European Space Agency

- Company
  - Rapita Systems Ltd. is a successful and profitable business that now employs more than 25 people in York

Impact

- Quotes

"The biggest benefit that RapiTime brought to our development process was just how quickly we could get comprehensive timing measurements from our tests. Not only did we reduce our effort requirements for the testing, but we could use our results in ways that were infeasible before."

Engineering Fellow at a major aerospace supplier 2009

"the main advantage [of using RapiTime] is the possibility to identify software bottlenecks that can be subject to optimisation. Without RapiTime the mandatory code optimisation would have been done without the knowledge of where to concentrate the efforts."

Alenia Aermacchi

Summary

- Three Case Studies - one common thread
  - World-class Research from RTSRG
    - Many of the research papers produced are recognised as the seminal ones in the field
    - Some cited more than 500 times
  - Exploitation via a start-up company
  - World-wide impact
    - Products have been adopted and standardised upon by many large companies in the automotive and aerospace industries
    - Created and sustained large numbers of high technology jobs

Questions?

- More info
  - White papers about the Real-Time Systems group Impact Case Studies can be found on my webpage
    http://www-users.cs.york.ac.uk/~robdavis/
  - Videos available soon!