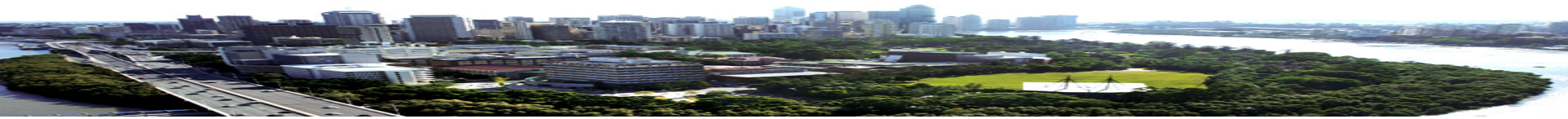
ENB 350 Real-time Computer based systems

Lecture 1

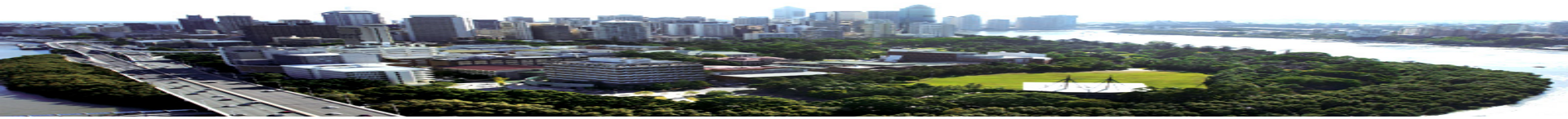
Embedded and Real-time Systems

V. Chandran



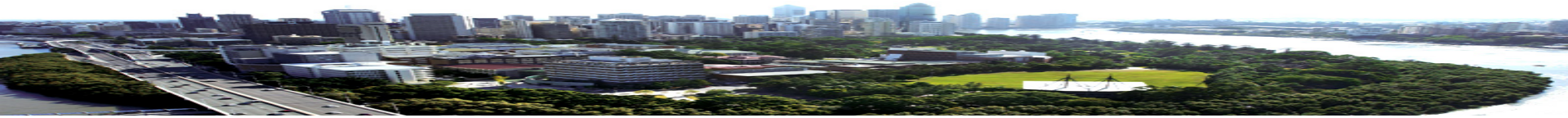
Contents

- What is this unit about?
- Embedded and Real-time
- Embedded Systems Development
- The laboratory – hardware / software



Teaching team

- Prof. Vinod Chandran – Unit coordinator and Lecturer
- Mr. Kyran Findlater – Tutor
- Mr. Erik Klokman – Technical staff for Laboratory S1129 maintenance
- Guest Lecturer: Mr. Stefan Slomka



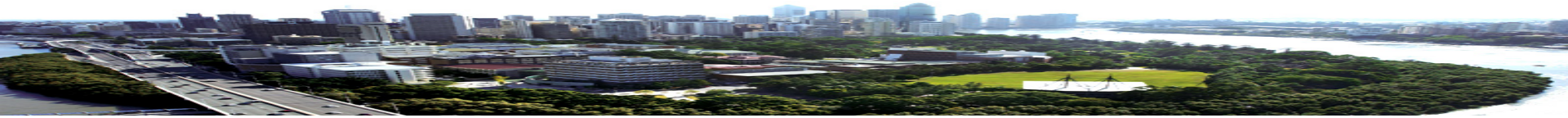
Formal contact

Lecture in M303

- Tue 10am-12pm (All students)

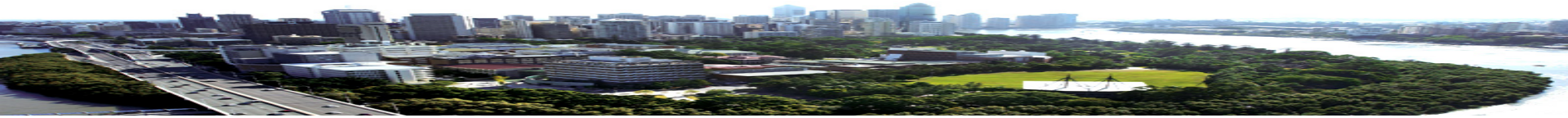
Tutorial / Lab session in S1129

- Tue 2-4pm OR
- Wed 10am-12 OR
- Wed 2-4pm (one session as allocated)



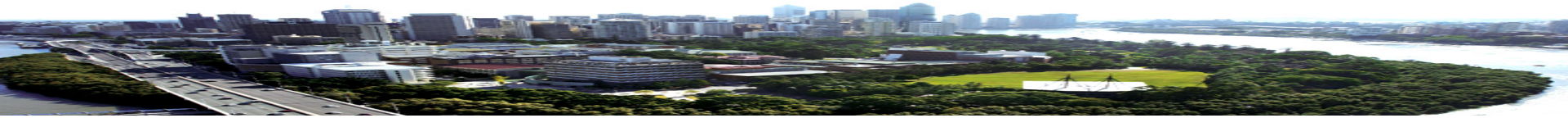
Other requirements

- Health and safety induction
  - general induction online for laboratories (certificate/sticker on id)
  - Compressed air safety in week 2 in S1129
- Lab exercises: (group of 2)
  - Demonstrate tasks as instructed in each specification sheet
- Assignment : (group of 4)
  - Demonstrate in week 13
  - Report in week 14
- Final examination (individual)



Assessment

- 6 Laboratory exercises – 30%
- Best 2 of 3 short answer tests – 10%
- Problem based learning project (Assignment) – 30%
  - Presentation (5%)
  - Demonstration (15%)
  - Report (10%)
- Final examination (multiple choice) – 30%



Resources

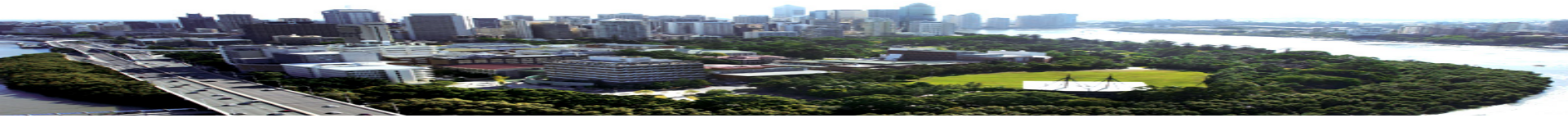
Reference books

1. D.W. Lewis, *Fundamentals of Embedded Software with the ARM Cortex-M3*, Pearson 2012
2. K. Hyder and B. Perrin, *Embedded Systems Design using the Rabbit 3000 microprocessor – Interfacing, Networking and Application Development*, Newnes 2005
3. D.W. Lewis, *Fundamentals of Embedded Software – where C and assembly meet*, Prentice Hall 2002 (out of print)
4. J. Labrosse, *Micro C OS/II The Real-time Kernel,* CMP Books, 2002

Other

1. Software –sample programs installed with Dynamic C
2. Vendor supplied documents in pdf format placed on Blackboard
3. References available on the world wide web; QUT library

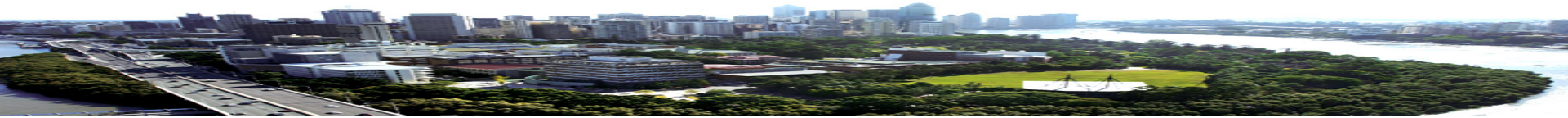
For unit related information – read the week by week study guide



Prerequisites for the unit

- Fundamentals of digital systems
- Basic knowledge of Microprocessors
- Programming skills - C Language

Formal – ENB244 Digital systems or equivalent that covers gates, Boolean algebra, number systems, flip flops, counters, logic families



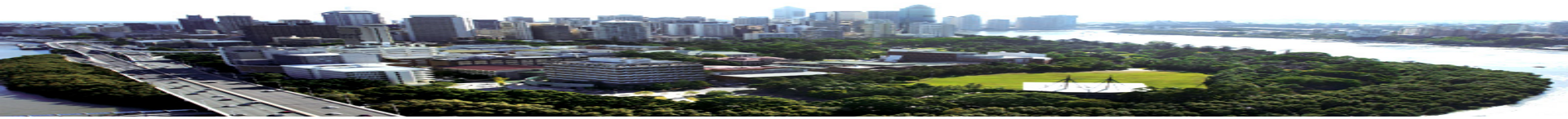
Prerequisite knowledge Test

Duration: 15 minutes

Please answer on the sheets provided.

No need to write your name or id.

Return the sheets to the lecturer.



What is an Embedded System?

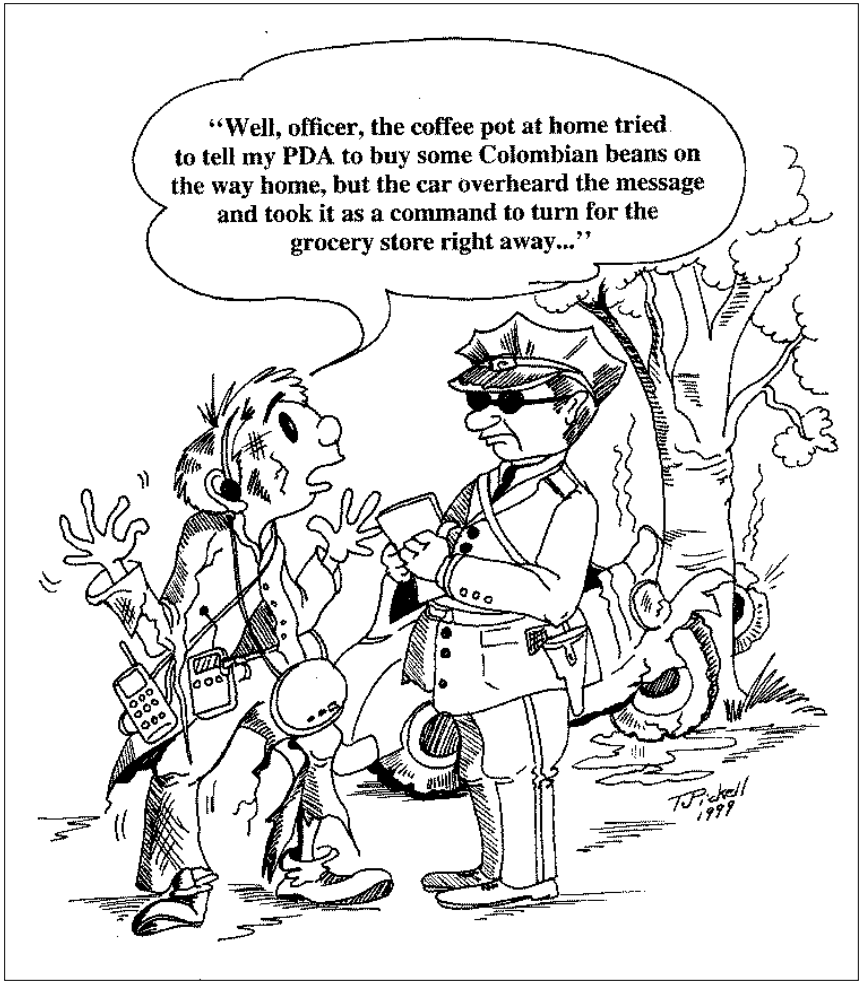
- Electronic devices that incorporate a computer (usually a microprocessor) within their implementation.
- A computer is used in such devices primarily as a means to simplify the system design and to provide flexibility.
- Often the user of the device is not even aware that a computer is present.

Embedded Rules!

- There are more than 50 embedded devices in a typical home!
- Total #ARM processors sold exceeded 37 billion in 2010.
- More than 40 companies make microprocessors/microcontrollers.

Design Goal: Reliability

- Mission Critical
- Life-Threatening
- 24/7/365
- Can’t reboot!



Design Goal: Performance

- Multitasking and Scheduling
- Optimized I/O ➔ Assembly Language
- Limits, Inaccuracies of Fixed Precision

Design Goal: Cost

- Consumer Market: Minimize Manufacturing Cost.
- Fast Time to Market Required
- No chance for future modification.

What is a Real-Time System?

- Real-time systems process events.
- Events occurring on external inputs cause other events to occur as outputs.
- Meeting deadlines is usually a primary objective, or otherwise the entire system may fail to operate properly.
- Minimizing response times is important

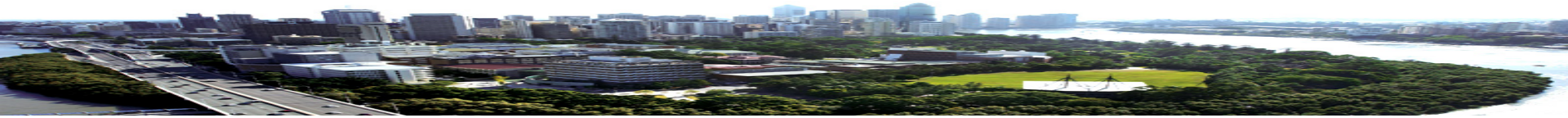
Hard/Soft Real-Time Systems

- Soft Real-Time System
  - Compute output response as fast as possible, but no specific deadlines that must be met.

- Hard Real-Time System
  - Output response must be computed by specified deadline or system fails.

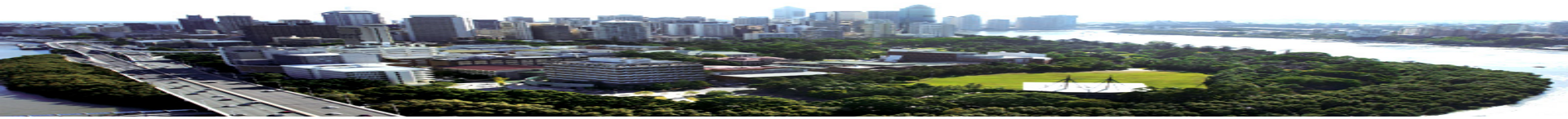
Hard and soft real-time systems

- Hard = deadline missed -> catastrophic
- Soft = deadline missed -> less valuable



Discussion Example 1

Activity will be done in class.



Multi-Tasking and Concurrency

- Most real-time systems are also embedded systems w/several inputs and outputs and multiple events occurring independently.
- Separating tasks simplifies programming, but requires somehow switching back and forth among the different threads of computation (*multi-tasking*).
- *Concurrency* is the appearance of simultaneous execution of multiple tasks.
- Avoids idle waiting times and utilizes CPU better

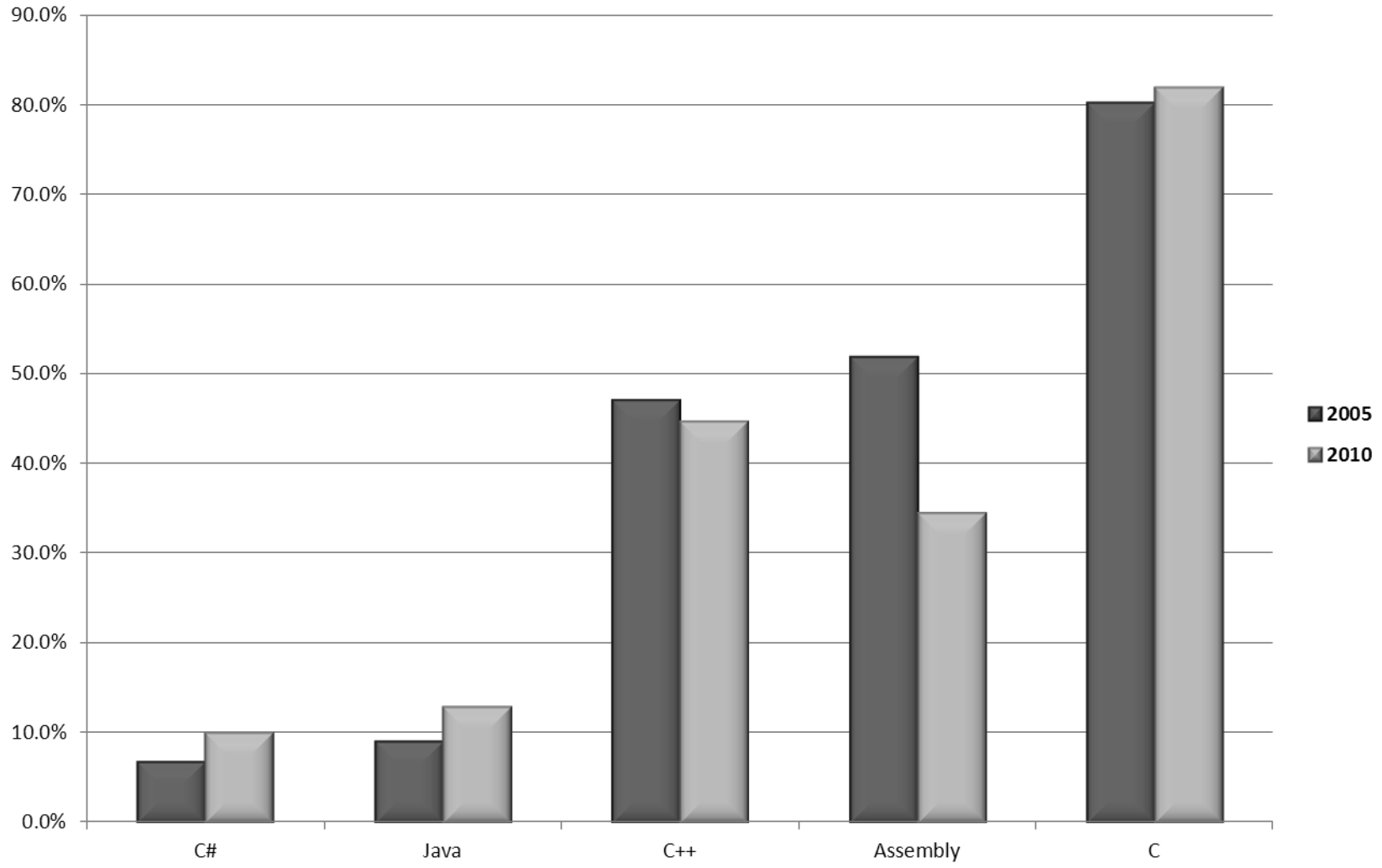
Three Concurrent Tasks Within a Programmable Thermostat

```
/* Monitor Temperature */
do forever {
        measure temp ;
        if (temp < setting)
                start furnace ;
        else if (temp >
                setting + delta)
                stop furnace ;
}
```

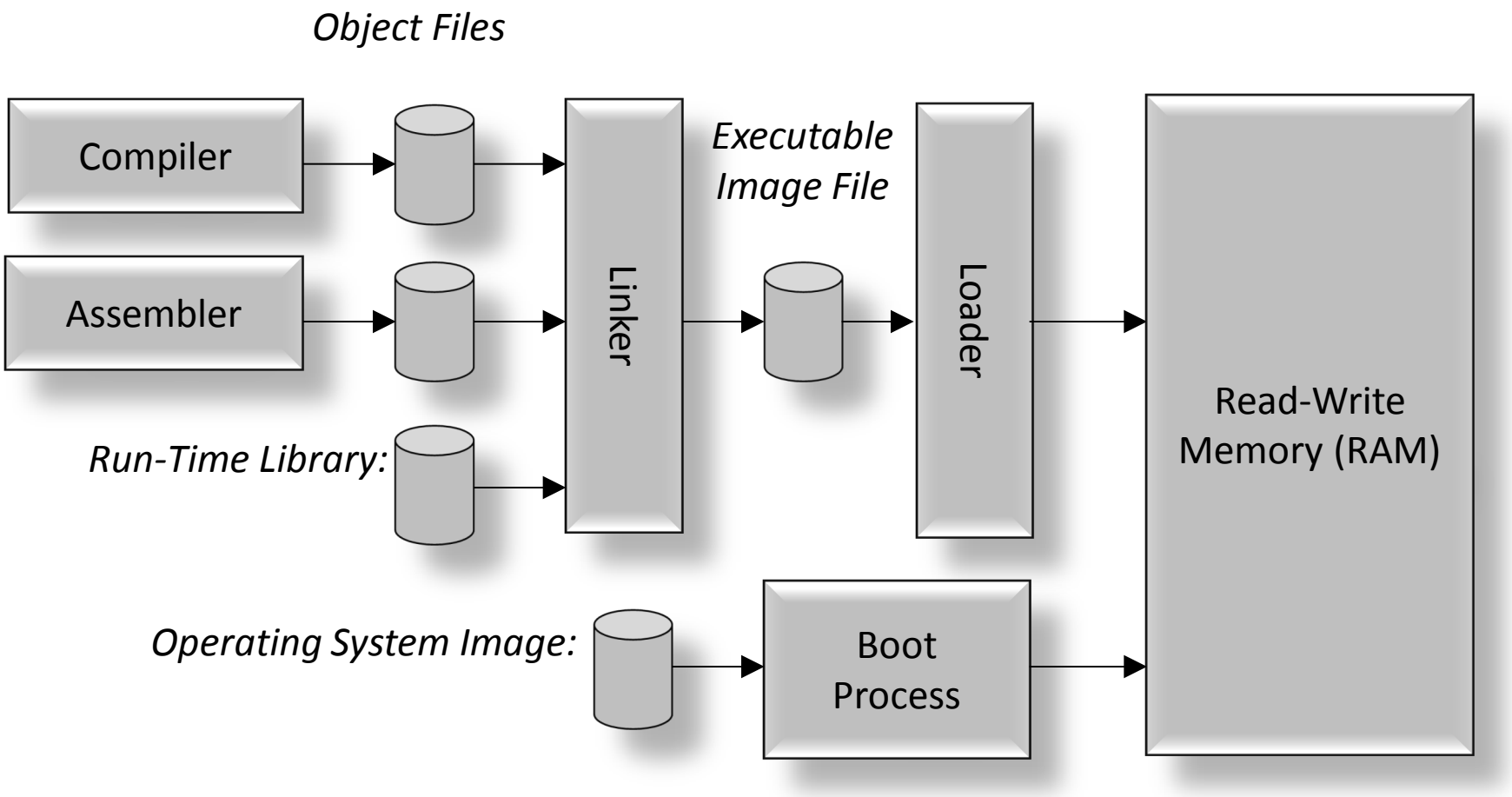
```
/* Monitor Time of Day */
do forever {
        measure time ;
        if (6:00am)
                setting = 72°F ;
        else if (11:00pm)
                setting = 60°F ;
}
```

```
/* Monitor Keypad */
do forever {
        check keypad ;
        if (raise temp)
                setting++ ;
        else if (lower temp)
                setting-- ;
}
```

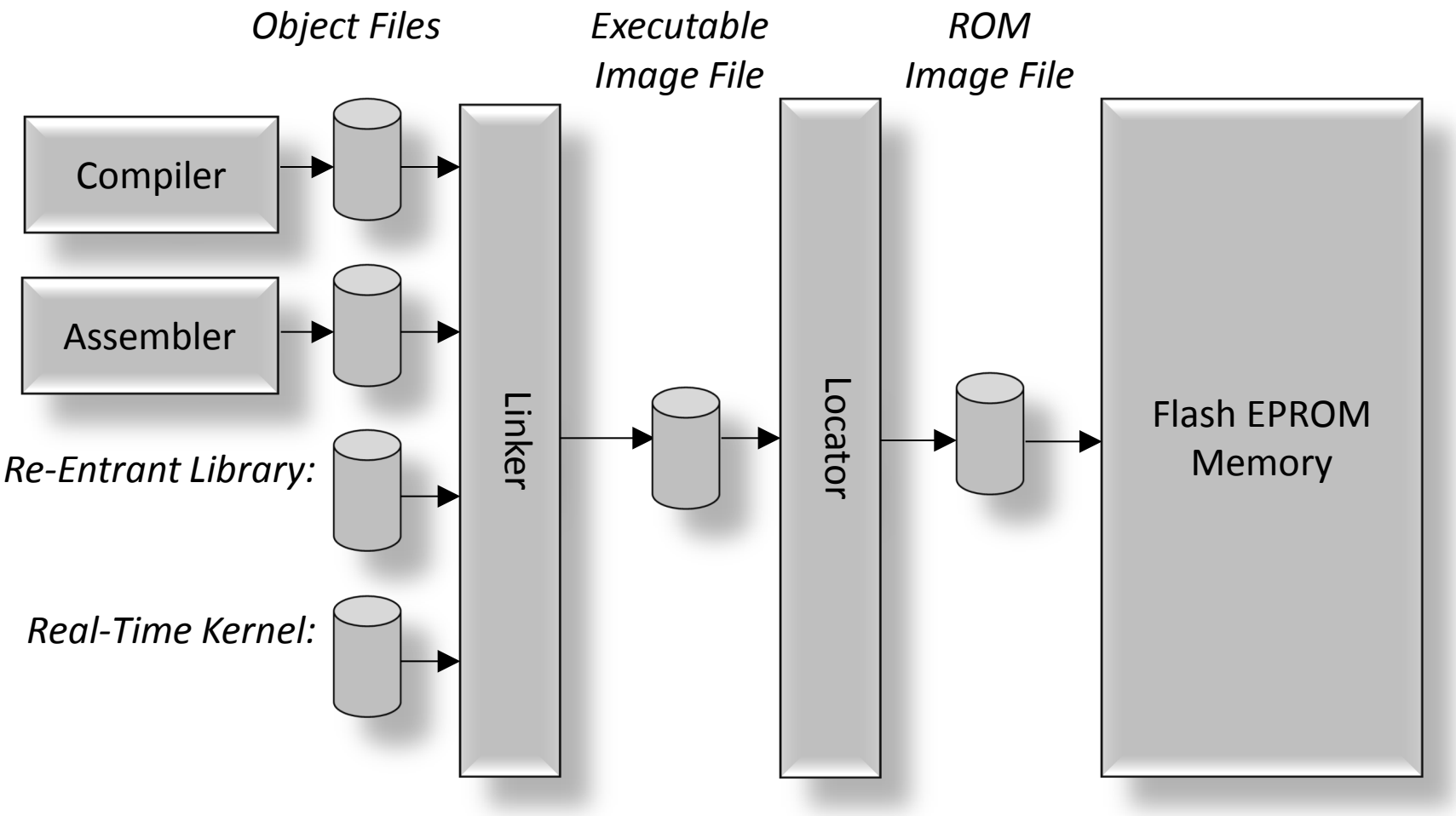
Programming Language Use in Embedded Designs



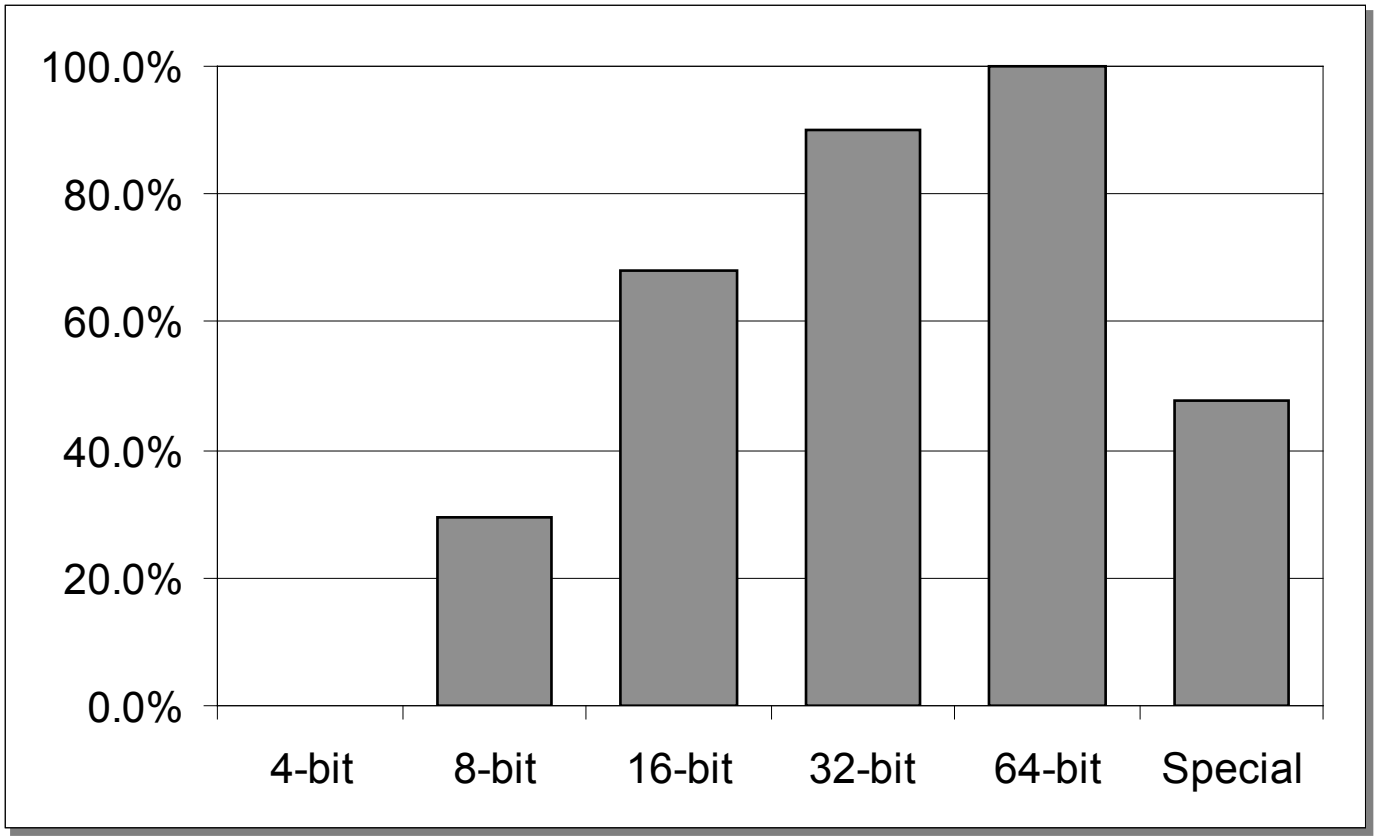
Desktop Application Development



Embedded Application Development



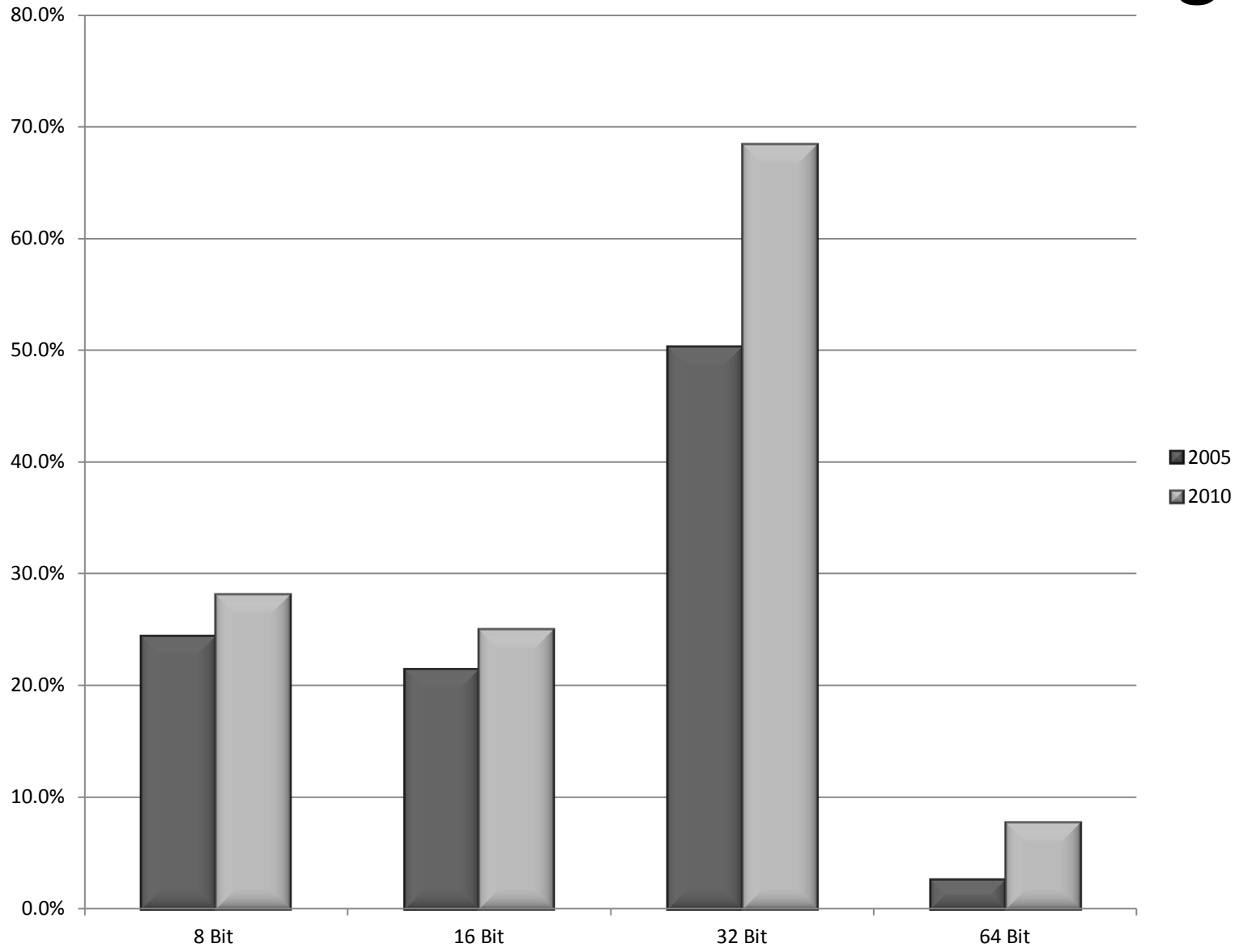
Use of Real-Time Kernels in New Embedded Designs.



Examples of Embedded Real-Time Software.

Property	*FAX Machine*	*CD Player*
Microprocessor:	16-bit	8-bit
Number of Threads:	6	9
Read-Write Memory (RAM):	2048 Bytes	512 Bytes
Total RAM Actually Used:	1346 Bytes (66%)	384 Bytes (75%)
Amount Used by Kernel:	250 Bytes (19%)	146 Bytes (38%)
Read-Only Memory (ROM):	32.0 KB	32.0 KB
Total ROM Actually Used:	28.8 KB (90%)	17.8 KB (56%)
Amount Used by Kernel:	2.5 KB (8.7%)	2.3 KB (13%)

Processor Use in Embedded Designs



ARM Powered Products

Product: Hunter Programmable Digital Thermostat.

Microprocessor: 4-bit

The tiny ATMEL 8-bit picoPower AVR processor in Vitality's GlowCap™ helps people remember to take their medication on time. It can sense when the bottle is opened, transmit that information wirelessly to a Vitality server, flash its LED, and play a ring-tone.

The Vendo Vue40 vending machine uses a 16-bit Hitachi H8/3007 processor.

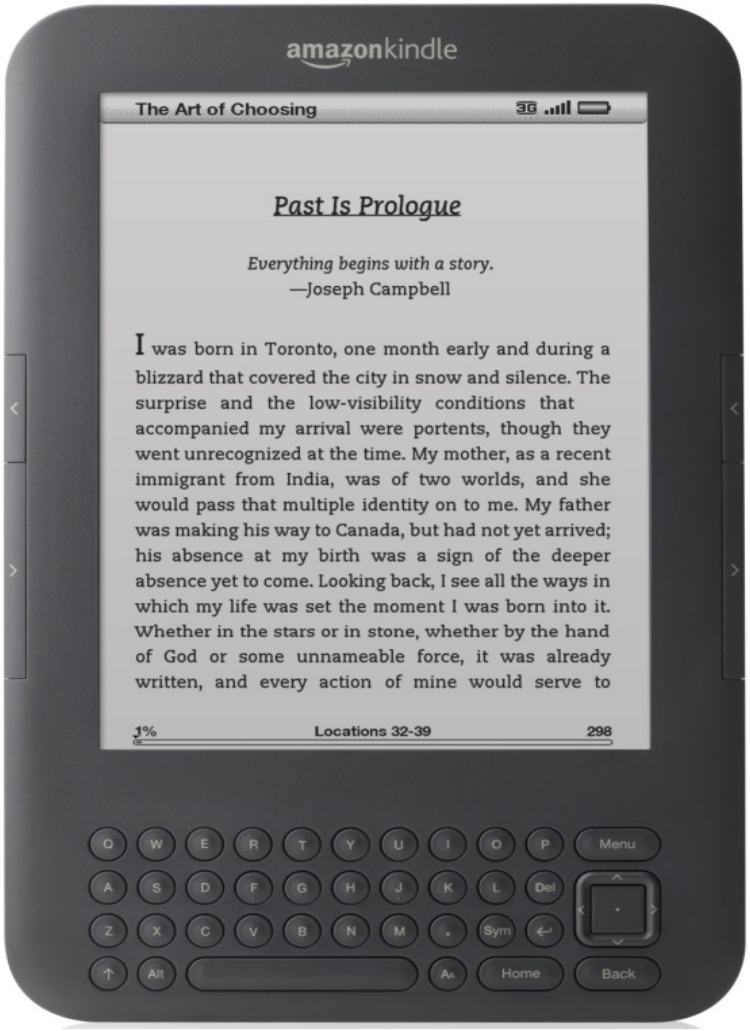
The Sonicare DiamondClean toothbrush uses an 8-bit PIC microprocessor.

Product: Miele dishwashers.

Microprocessor: 8-bit Motorola 68HC05.

NASA's 2003 Mars Exploration Rover used an BAE Systems RAD6000 32-bit RISC cpu and Wind River Systems' VxWorks embedded real-time operating system

The Seagate Barracuda XT disk drive incorporates two ARM Cortex-R4 processors – one to control the servos, and another to handle the command and data flow.



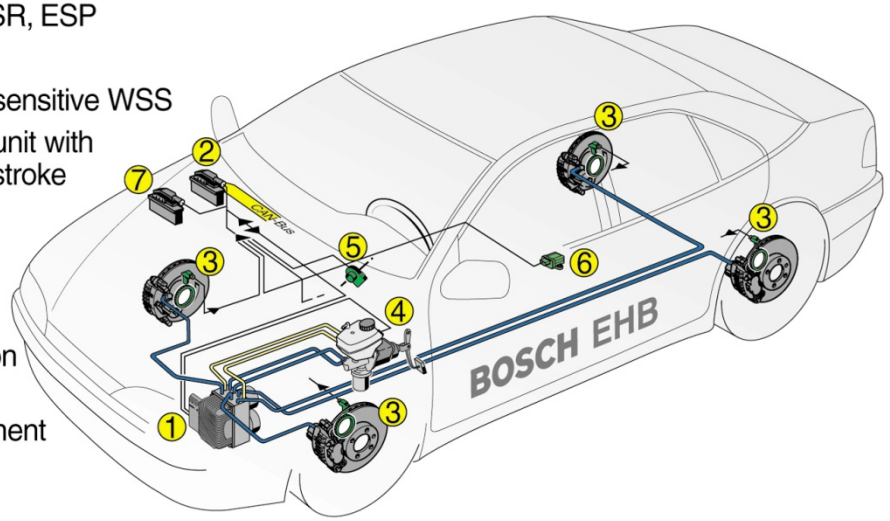
The Amazon Kindle 2 uses a 32-bit ARM processor.

Product: Sony Aibo ERS-110 Robotic Dog.

Microprocessor: 64-bit MIPS RISC.

Bosch Electrohydraulic Brake EHB

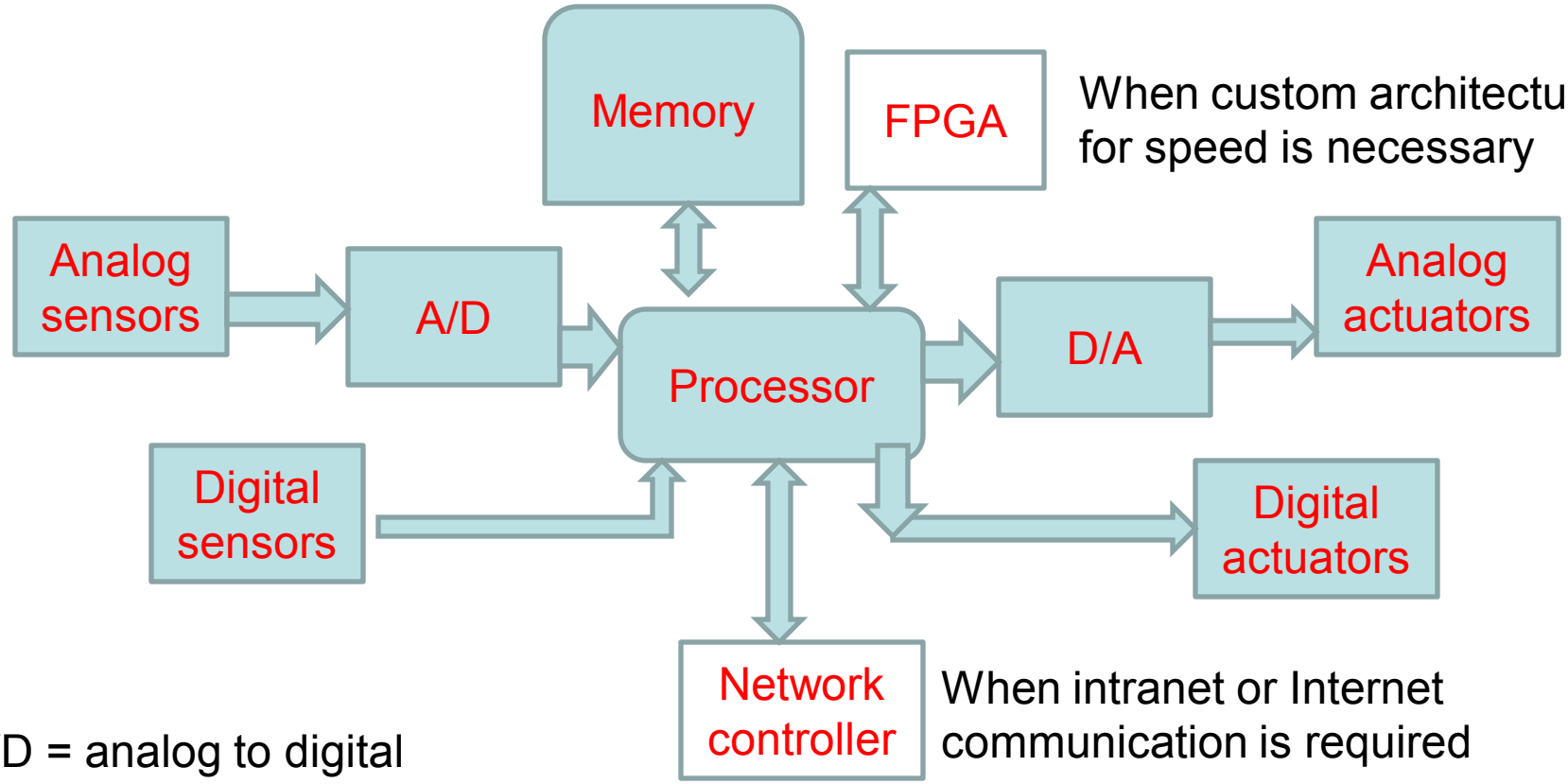
① Electrohydraulic actuator for EHB, ABS, ASR, ESP
② EHB - ECU
③ Active, direction-sensitive WSS
④ Brake operation unit with integrated pedal stroke sensor
⑤ Steering wheel angle sensor
⑥ Yaw rate and lateral acceleration sensor
⑦ Engine management ECU



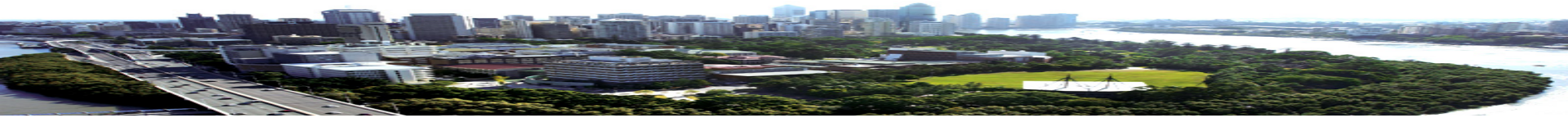
BOSCH

Press photo No. 1-K1-10583

Typical building blocks

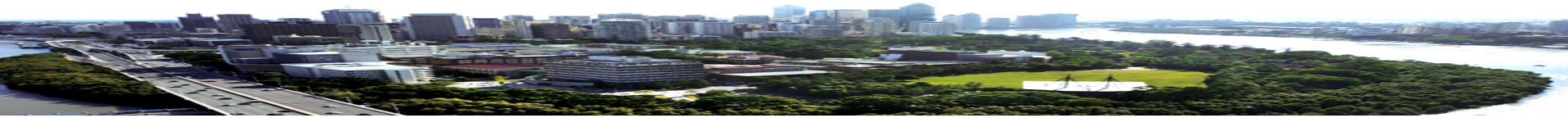


A/D = analog to digital
D/A = digital to analog



What is a real-time system?

- Real-time systems process events
- Correctness depends not only on logic but also on timeliness of the output
- Deadline-driven
- Examples – engine control unit of a car, video streaming decoder, flight control or airplane autopilot



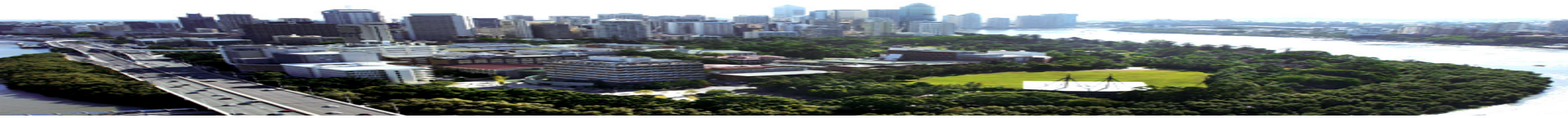
What are Microcontrollers?

Computer on a chip (microprocessor) for control

CPU, memory, peripherals all in one chip

- timers,
- pulse width modulation,
- analog to digital convertors,
- many ports

Low interrupt latency to support real-time tasks



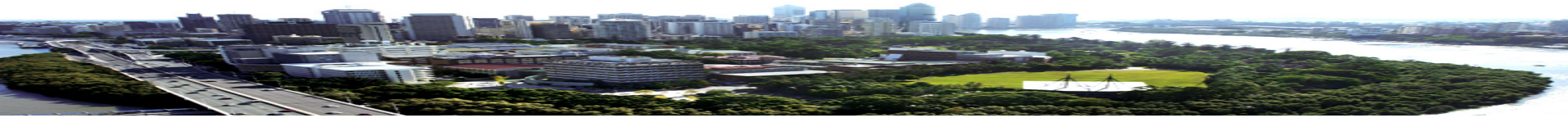
Development tools

- What is a compiler?

  Software that translates higher level language source code to assembly language code (and then using an assembler to machine code in binary form)

- What is a real-time operating system?

  Software that manages a real-time system providing the means for creating tasks, scheduling tasks, input-output, memory management and other functions of an operating system. RTOS or real-time kernel



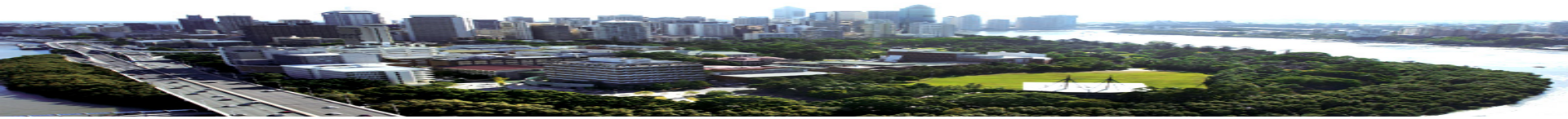
Development tools we're using

- Which compiler?

  Dynamic C v 10.66 for the Rabbit microprocessors
  We are using the Rabbit Core Module RCM4000 which has the Rabbit 4000 processor.

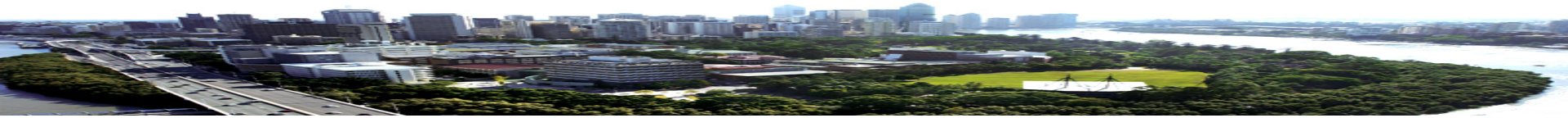
- Which real-time operating system?

  Micro C OS/II which links to Dynamic C as a library



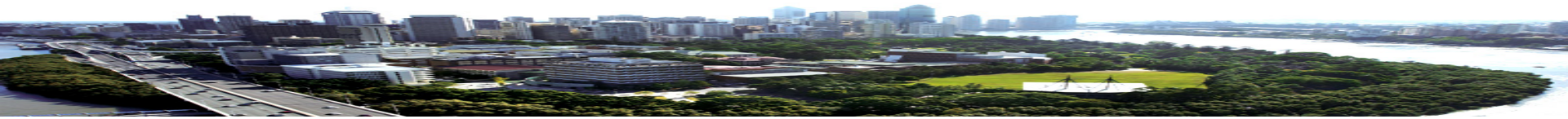
Which microcontroller are we using?

- Rabbit Semiconductor R4000
- Data bus width is 8 bits. Load/Store operations on 16 bits and 32 bits take more clock cycles
- Registers and operations for 8, 16 and 32
- Address bus width is 16 bits (64K) extended to 24 bits (16M) using memory management registers
- Has many on-board peripherals such as timers, serial ports, Pulse width modulator, Input capture module



Laboratory S1129

- Rabbit core modules (RCM4000) extended with connectors, LCD/keypad and housed in protective box
- Programming Environment for these is Dynamic C
- Windows PCs with Dynamic C is installed and connected via serial port to the RCM4000 boards
- 10 workstations x 2 students per session
- A miniaturized production unit (a station where work pieces can get tested for colour, material and height) available for the assignment



Resources

Consult documents available on Blackboard

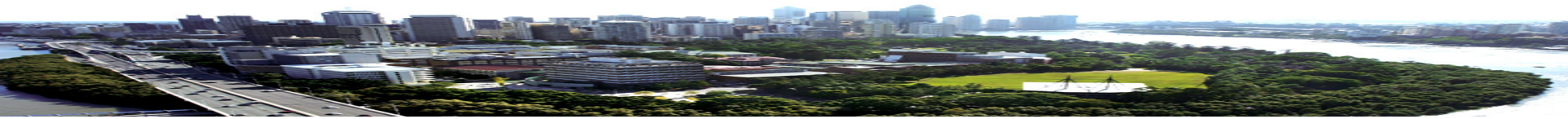
The poster (rab40_ref_poster.pdf)

Dynamic C user manual DCPUM.pdf

R4000UM.pdf user manual

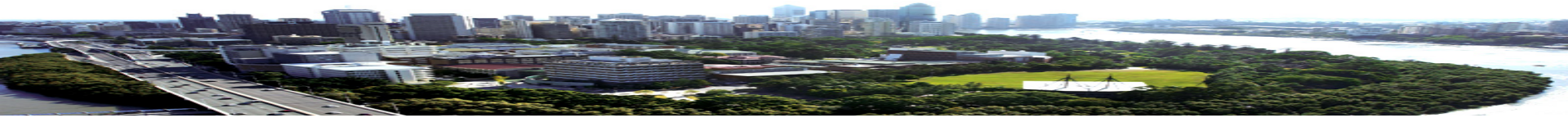
Hardware reference R4000DH.pdf

Instruction set manual



Lab 1

- Download and read the Lab 1 specification sheet from Blackboard.
- In the lab, get familiar with the hardware and programming environments. Examine the Sample programs.
- Refer the poster and manuals and find out about the architecture of R4000.



To do

- Consult the RCM4000 manual

- Find out the details about and the role of
  - the data bus
  - the address bus
  - The system clock
  - The real-time clock