The new Microcontroller giant Renesas and the challenges for embedded internet.

2nd International Workshop

on

Embedded Systems, Internet Programming and Industrial IT

Renesas Technology Europe

Content

- Some facts about Renesas
- MCU portfolio and Applications
- Ubiquitous Computing Network Society
- Java on embedded MCU
- Optimisation of Embedded Internet application
- MCU Weblearning System

Hello Renesas...

A new era is upon us

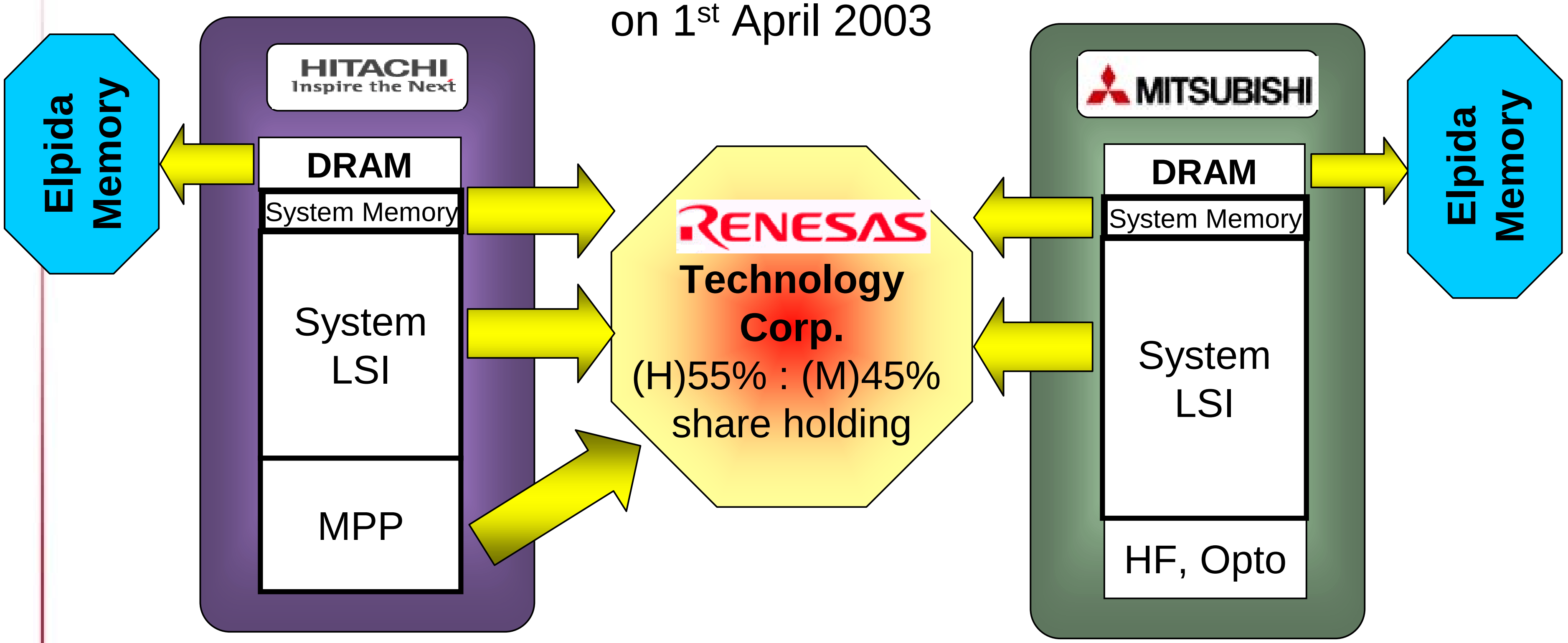
No longer do we simply
sit in front of computers.
Computing power is all around us.

Everywhere you imagine.

At the forefront of this new era is Renesas Technology, a new semiconductor company from the best of Hitachi and the best of Mitsubishi Electric

Creating Renesas Technology Corp

Formed from the semiconductor operations of Mitsubishi and Hitachi on 1st April 2003



- **Renesas Technology Europe**
  - – operated as a consolidated European Group
  - – organised as separate legal companies for tax, employment, admin reasons.

Renesas vision and focus

- **Corporate Vision:**
  - Aiming to realise a ubiquitous networked society

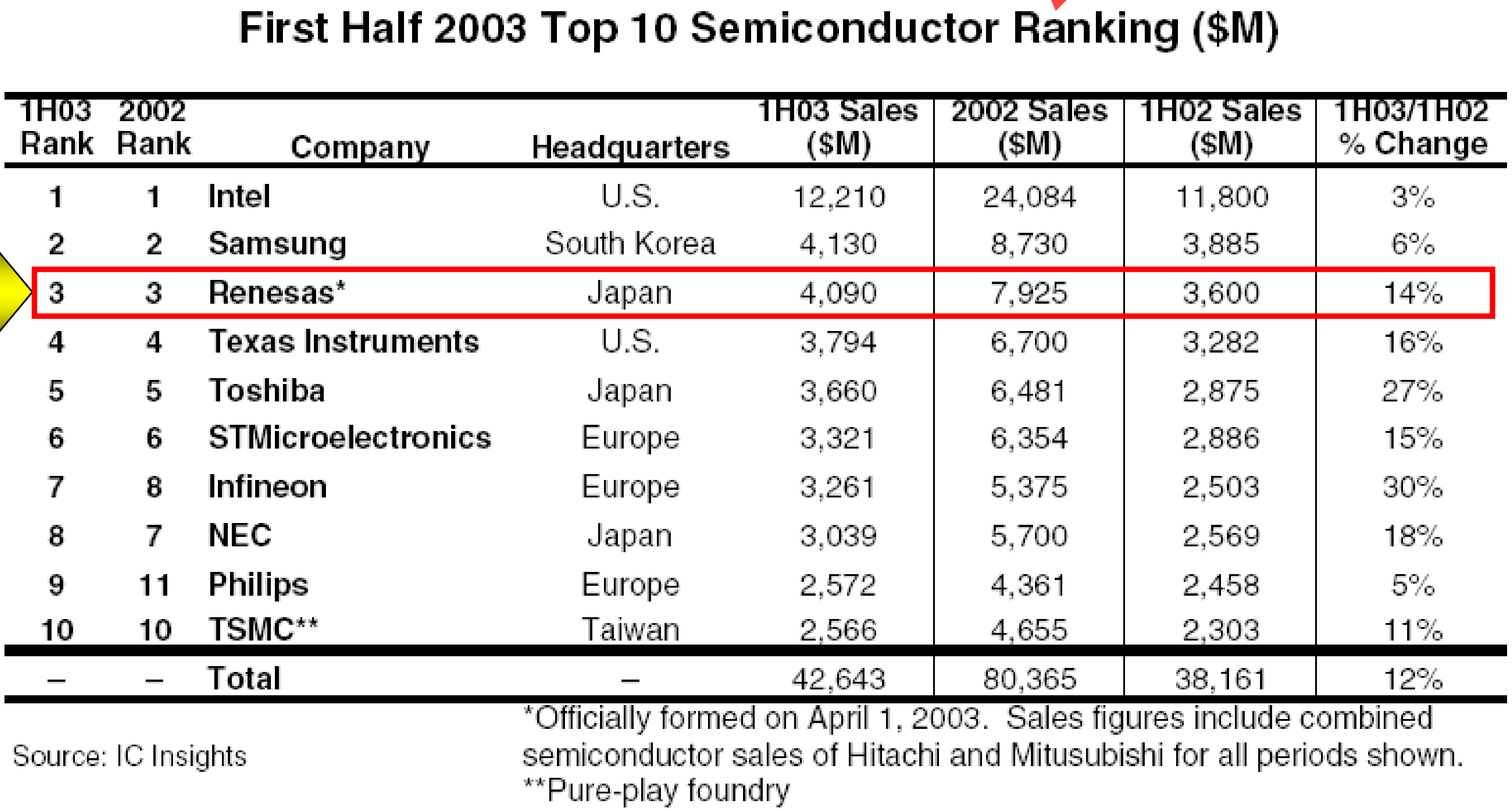
    Renesas offers intelligent microchip solutions to customers all over the world through continued technological innovations.
  - As a leading microcomputer company

    Renesas will continue to offer superior microcomputers for a broad range of applications.
  - As a trustworthy company

    We will continue to grow as a through trustworthy company
- **Company Focus:**
  - microcontrollers as flagship product
  - analog/flash/discrete development
  - SoC activities … & key markets focus.

WW Ranking List

No. 3 World Wide !!!



First Half 2003 Top 10 Semiconductor Ranking ($M)

1H03 Rank	2002 Rank	Company	Headquarters	1H03 Sales ($M)	2002 Sales ($M)	1H02 Sales ($M)	1H03/1H02 % Change
1	1	Intel	U.S.	12,210	24,084	11,800	3%
2	2	Samsung	South Korea	4,130	8,730	3,885	6%
3	3	Renesas*	Japan	4,090	7,925	3,600	14%
4	4	Texas Instruments	U.S.	3,794	6,700	3,282	16%
5	5	Toshiba	Japan	3,660	6,481	2,875	27%
6	6	STMicroelectronics	Europe	3,321	6,354	2,886	15%
7	8	Infineon	Europe	3,261	5,375	2,503	30%
8	7	NEC	Japan	3,039	5,700	2,569	18%
9	11	Philips	Europe	2,572	4,361	2,458	5%
10	10	TSMC**	Taiwan	2,566	4,655	2,303	11%
–	–	Total	–	42,643	80,365	38,161	12%

Source: IC Insights

*Officially formed on April 1, 2003. Sales figures include combined semiconductor sales of Hitachi and Mitusubishi for all periods shown.
**Pure-play foundry

RENESAS

Some facts about Renesas

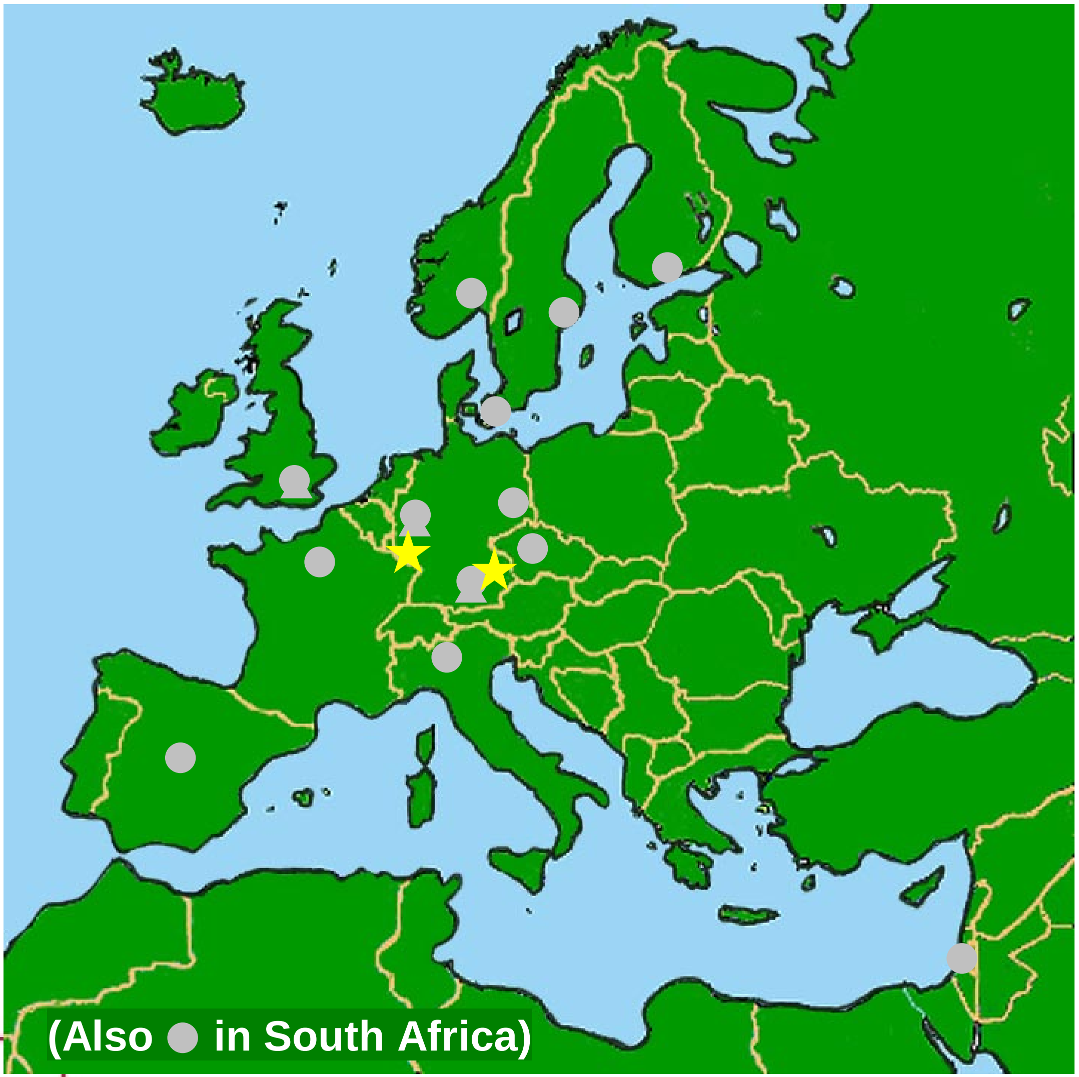
Worldwide

- #3 worldwide semiconductor maker
- #1 microcontroller supplier
- #2 smart card supplier
- World’s leading supplier of components for CD/DVD
- 25% w/w share of TV microcontroller market
- #1 World-wide for RF devices in GSM mobile applications
- 80% share of Japanese Mobile Applications Processor market
- World’s largest supplier of flash microcontrollers, having shipped more than 500 million embedded flash MCUs since 1990

In Europe

- > 70% of Europe’s industrial drives companies using Renesas MCUs
- MCU supplier of choice to 3 of Europe’s top 5 white goods producers
- #1 supplier of microcontrollers to European Utility Metering market
- 60% share of European In Car Multimedia / Navigation market

Facilities in Europe

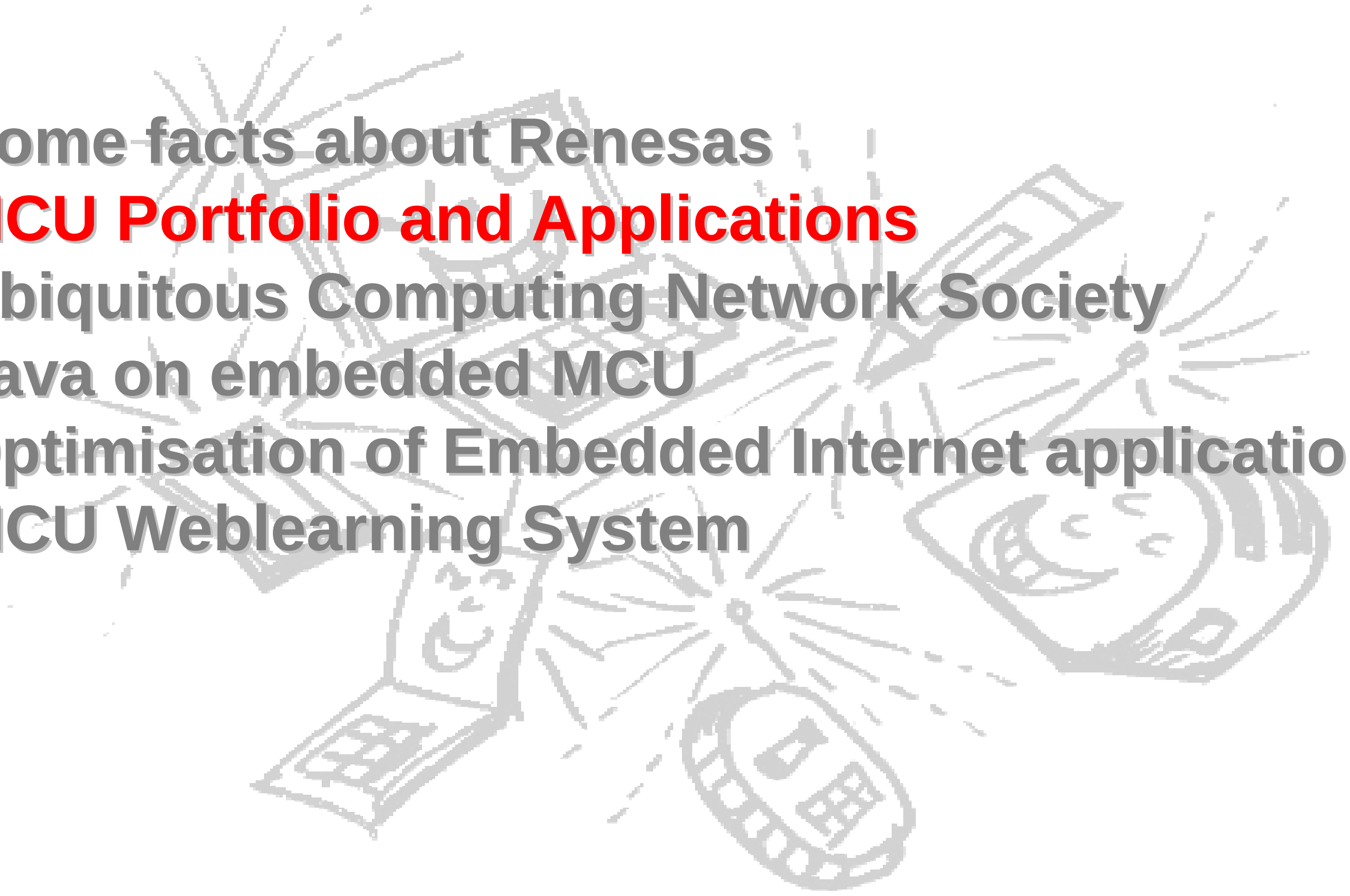


- Engineering, Marketing, Sales & Support more than 500 people
- Global support for smart card, mobile, automotive products
- European HQ in Bourne End, UK
- Major centres in Ratingen & Feldkirchen, Germany
- Offices in 12 countries

● Sales / Marketing
▲ Engineering
★ Manufacturing

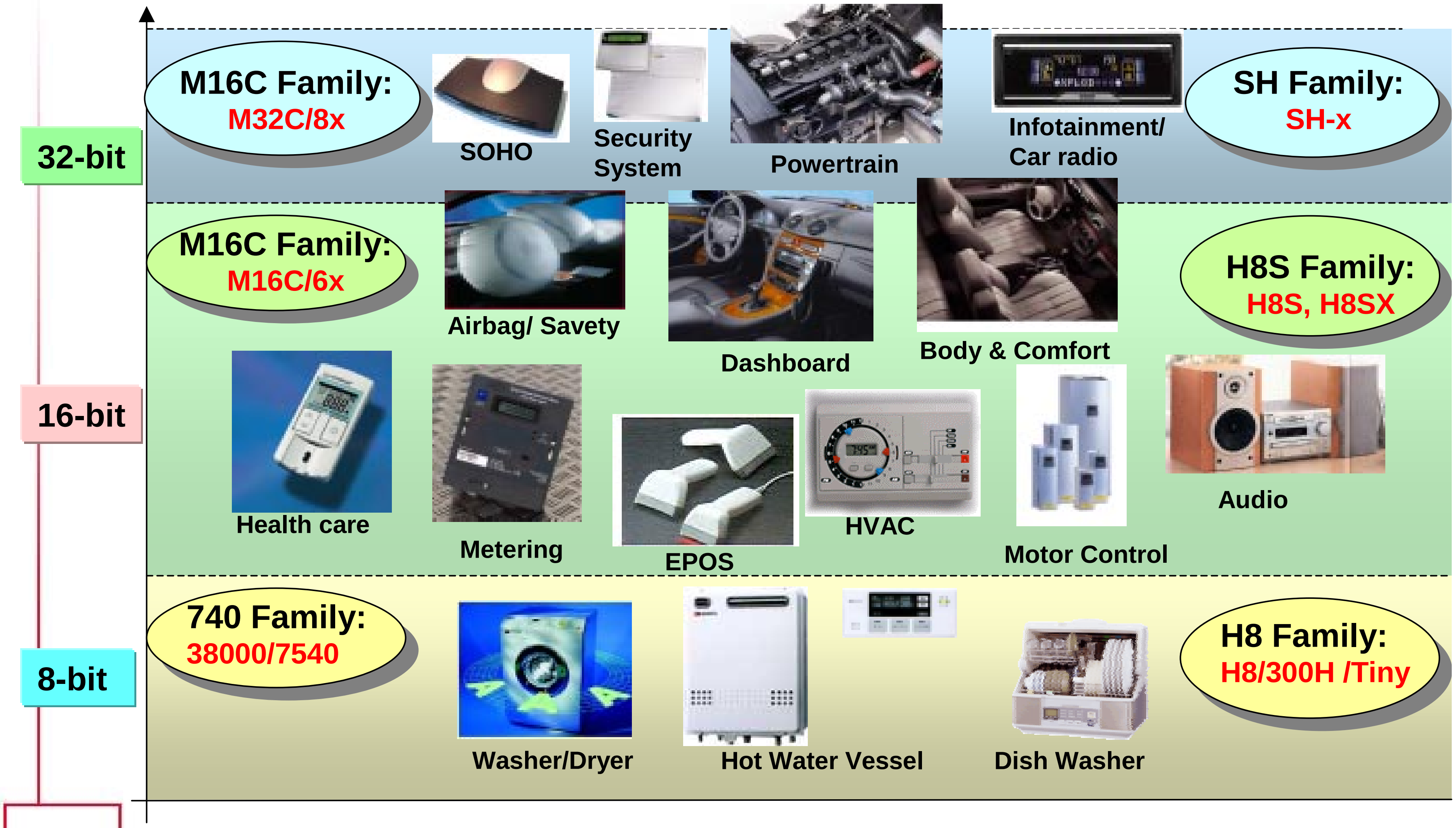
Associated Semiconductor manufacturing in Alsdorf and Landshut, Germany

Content



- Some facts about Renesas
- MCU Portfolio and Applications
- Ubiquitous Computing Network Society
- Java on embedded MCU
- Optimisation of Embedded Internet application
- MCU Weblearning System

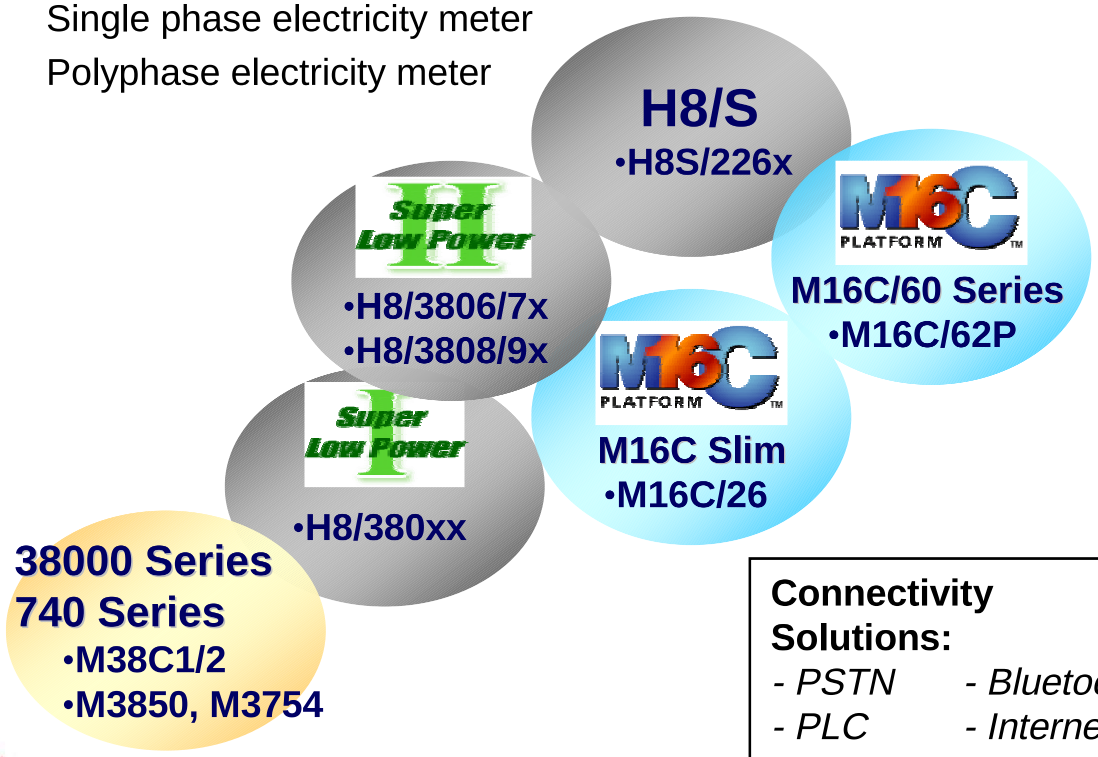
Application examples for Renesas MCU



Application Focus: Utility Metering

Widest range of Solutions for all metering applications:

- Gas / Water flow meter
- Heat power meter
- Single phase electricity meter
- Polyphase electricity meter



Offering attractive benefits

- Scaleable architectures
- High integration
- HW / SW LCDC + RTC
- Low Power (1uA in Standby)
- Embedded FLASH Microcontroller

Reference Solution for single phase electricity meter:

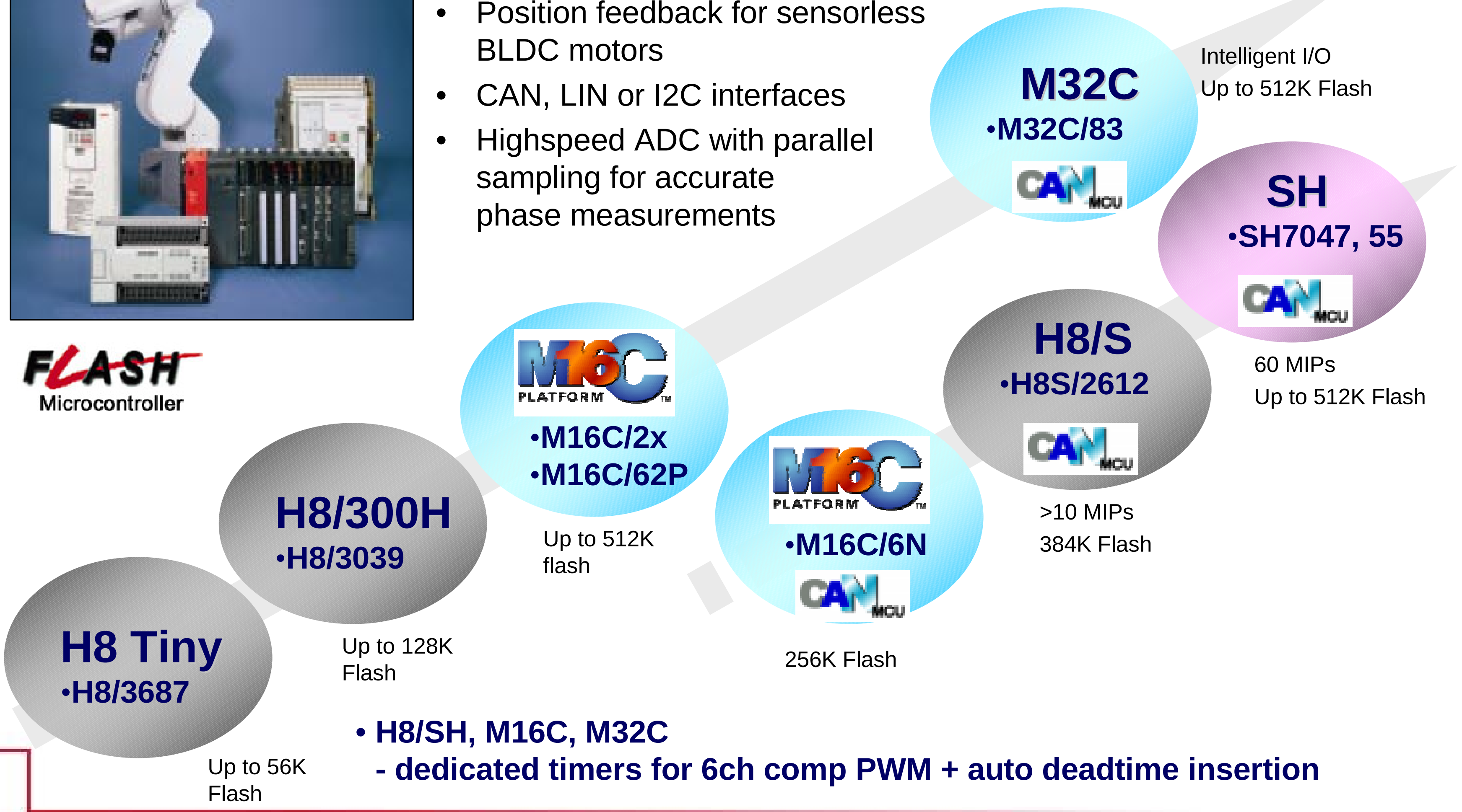
Connectivity Solutions:

- PSTN	*- Bluetooth*
- PLC	*- Internet*
- IrDa	

Application Focus: Industrial Automation

FLASH Microcontroller

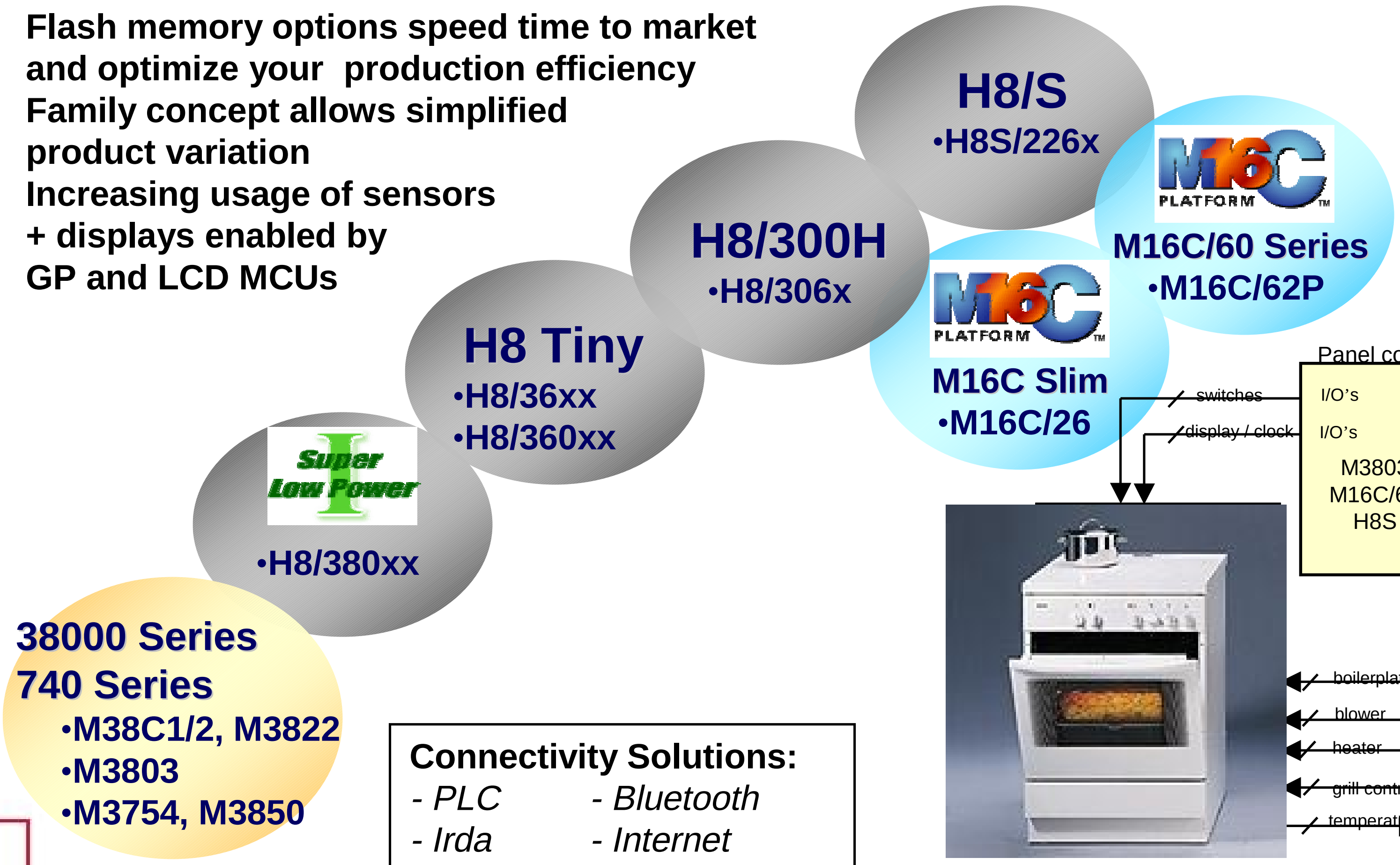
- High performance for closed loop control
- Position feedback for sensorless BLDC motors
- CAN, LIN or I2C interfaces
- Highspeed ADC with parallel sampling for accurate phase measurements



• H8/SH, M16C, M32C
- dedicated timers for 6ch comp PWM + auto deadtime insertion

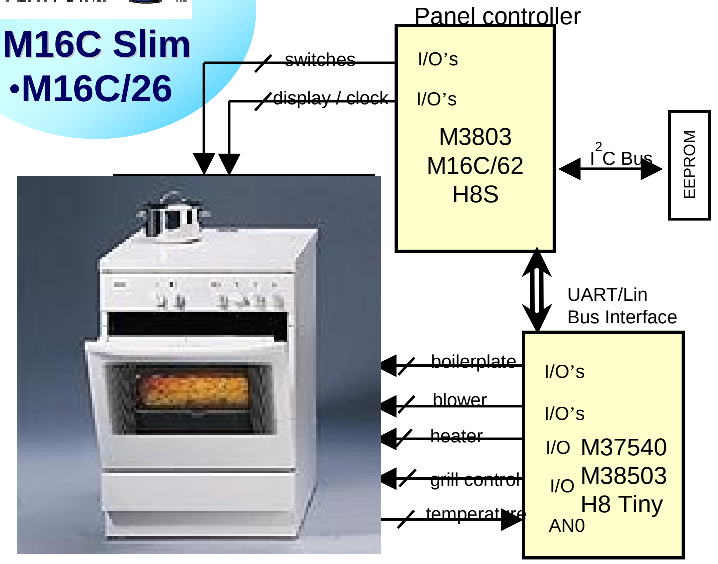
Application Focus: White Goods

- Scaleable architectures protect your software investment
- Flash memory options speed time to market and optimize your production efficiency
- Family concept allows simplified product variation
- Increasing usage of sensors + displays enabled by GP and LCD MCUs



Connectivity Solutions:

- PLC	*- Bluetooth*
- Irda	*- Internet*

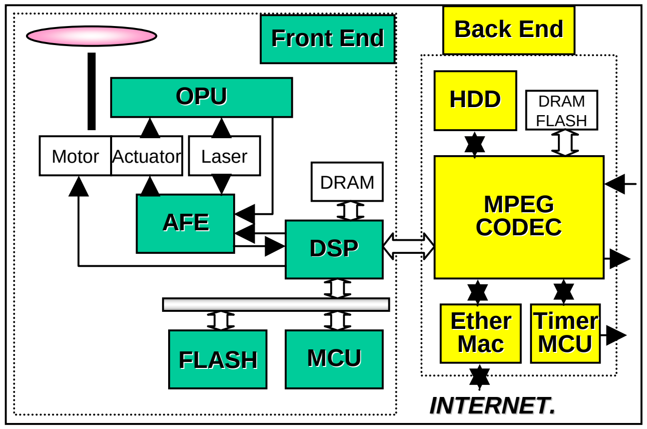


Application Focus: Audio/Video

- Broad range of Digital Audio solutions including MP3, DAB, HD Audio...
- Optical storage systems expertise for recordable DVD
- MCU Solutions for Flat TV applications
- Scalable MCU architectures for the Audio MMI
- Wide range of motor driver for CD/DVD drives
- Memory solutions including embedded Flash, PCMCIA, MMC and CF cards up to 1GByte

CD-RW + DVD Player RPF:

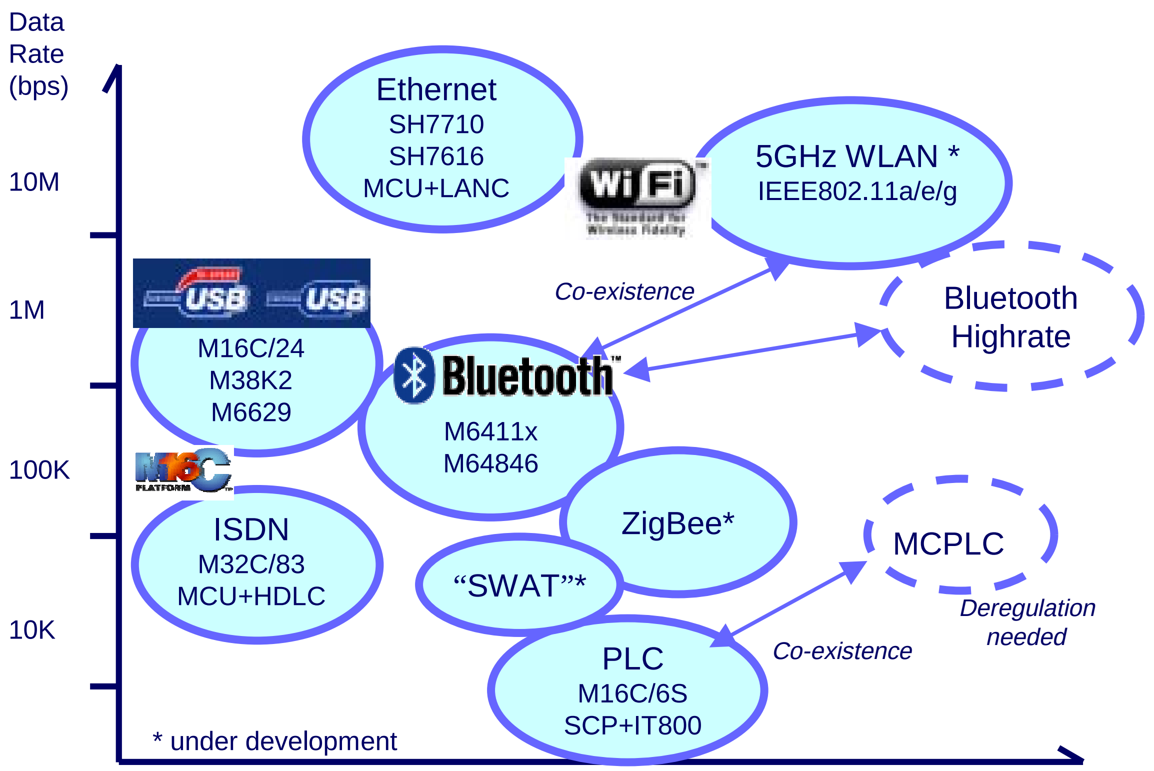
DVD Recorder Solution:



MP3 RPF using SH "Solution Engine"

Bluetooth/MP3 RPF using M32C and M6411x

Home Network Solutions



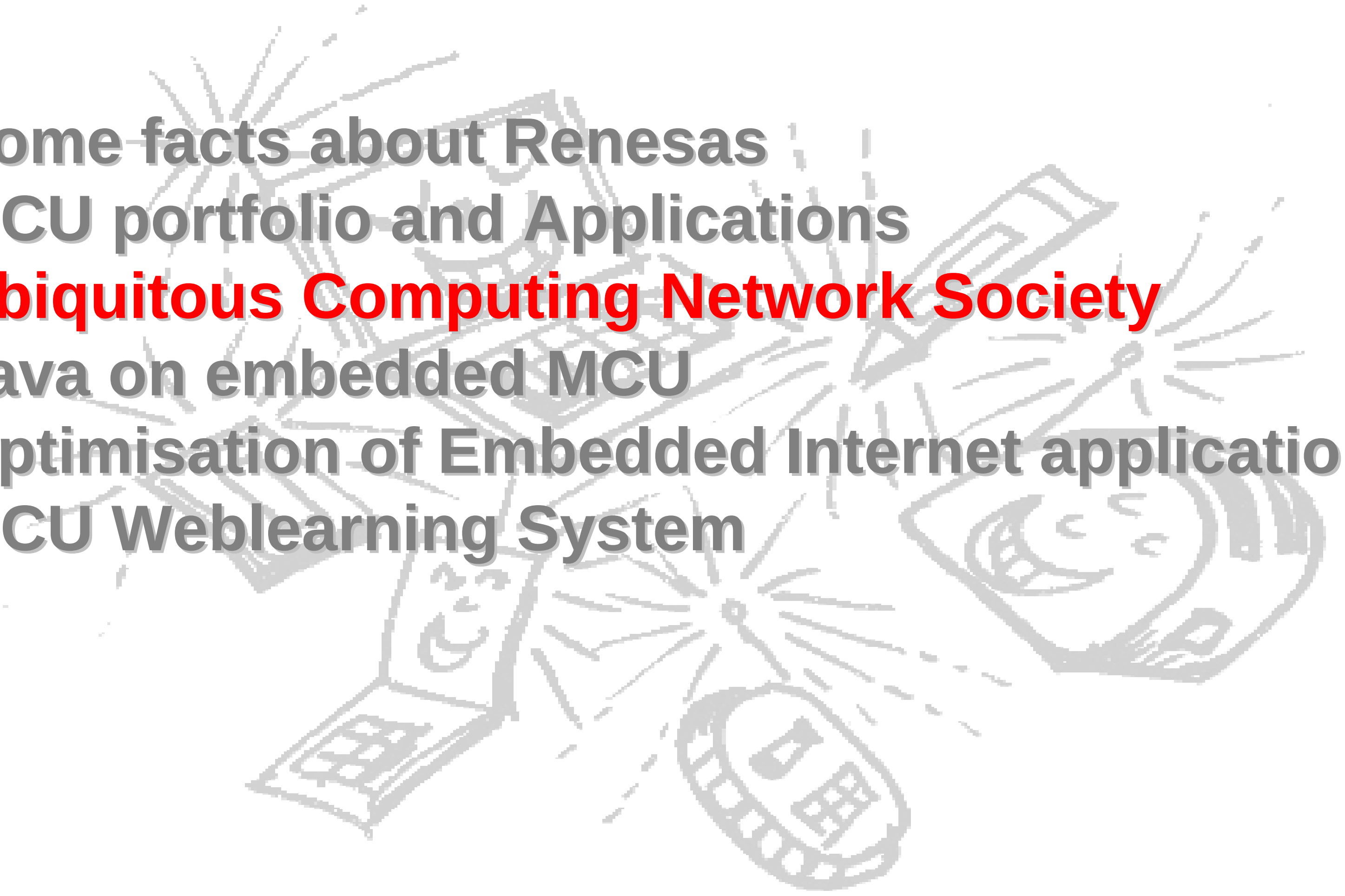
Renesas has expertise in a wide range of connectivity solutions including:

- •Embedded Ethernet 10/100 MAC on SH 32bit RISC devices
- •WLAN - 802.11 a/e/g complete solutions (under development)
- •Internet terminal solutions with reference designs on SH and M32C platforms
- •Strong capability in RF covering DAB front-end, Bluetooth solutions etc.
- •Software IP stack + software modems for H8 and M16C families
- •PLC solutions

In summary, Renesas will provide you:

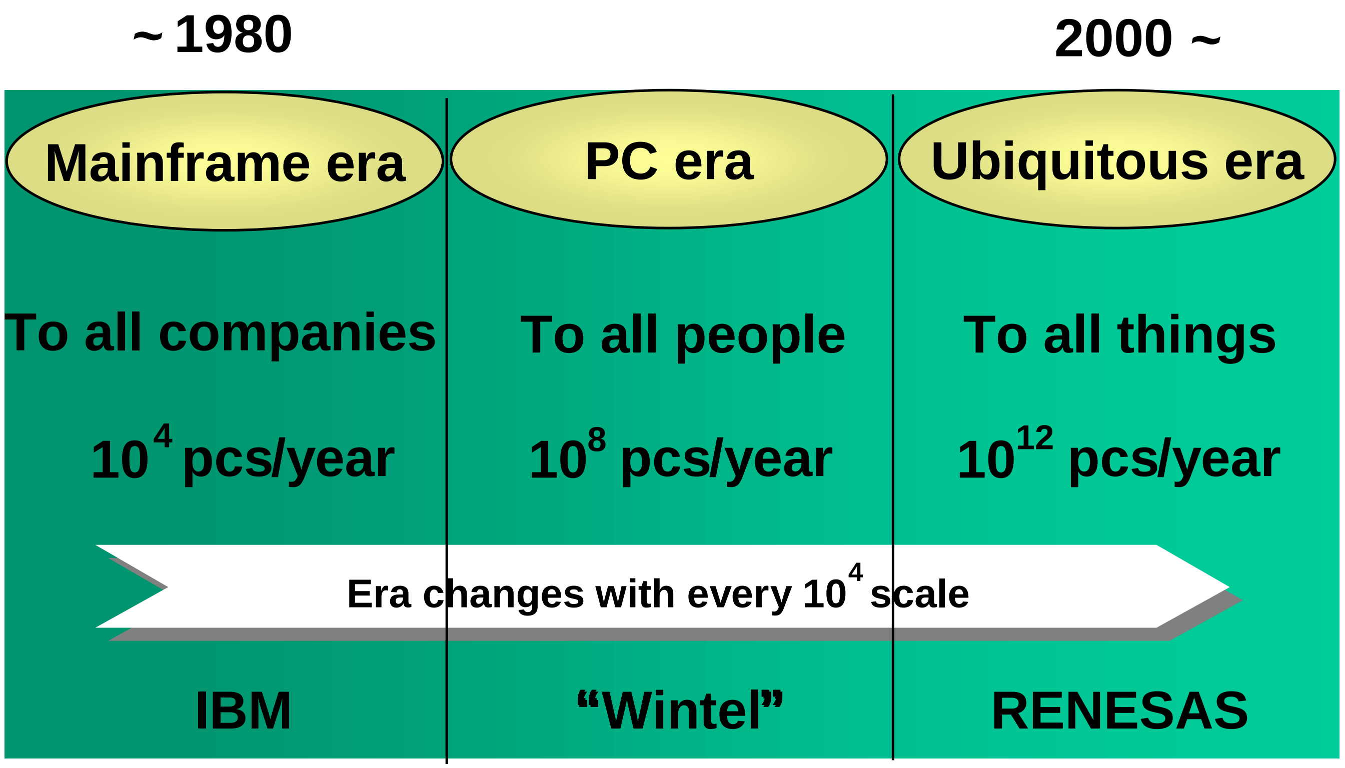
- World leading Technology, Process and Production capability
- Best in class standard product and system solution know how with particular strength in microcontrollers, RISC core architectures, SOC/ASSPs and connectivity
- A wide range of European designed reference solutions for the Consumer Industrial segment with applications notes, software libraries…..and specialist engineering support
- All back up by a strong European sales, marketing and applications support infrastructure that is second to none.

Content



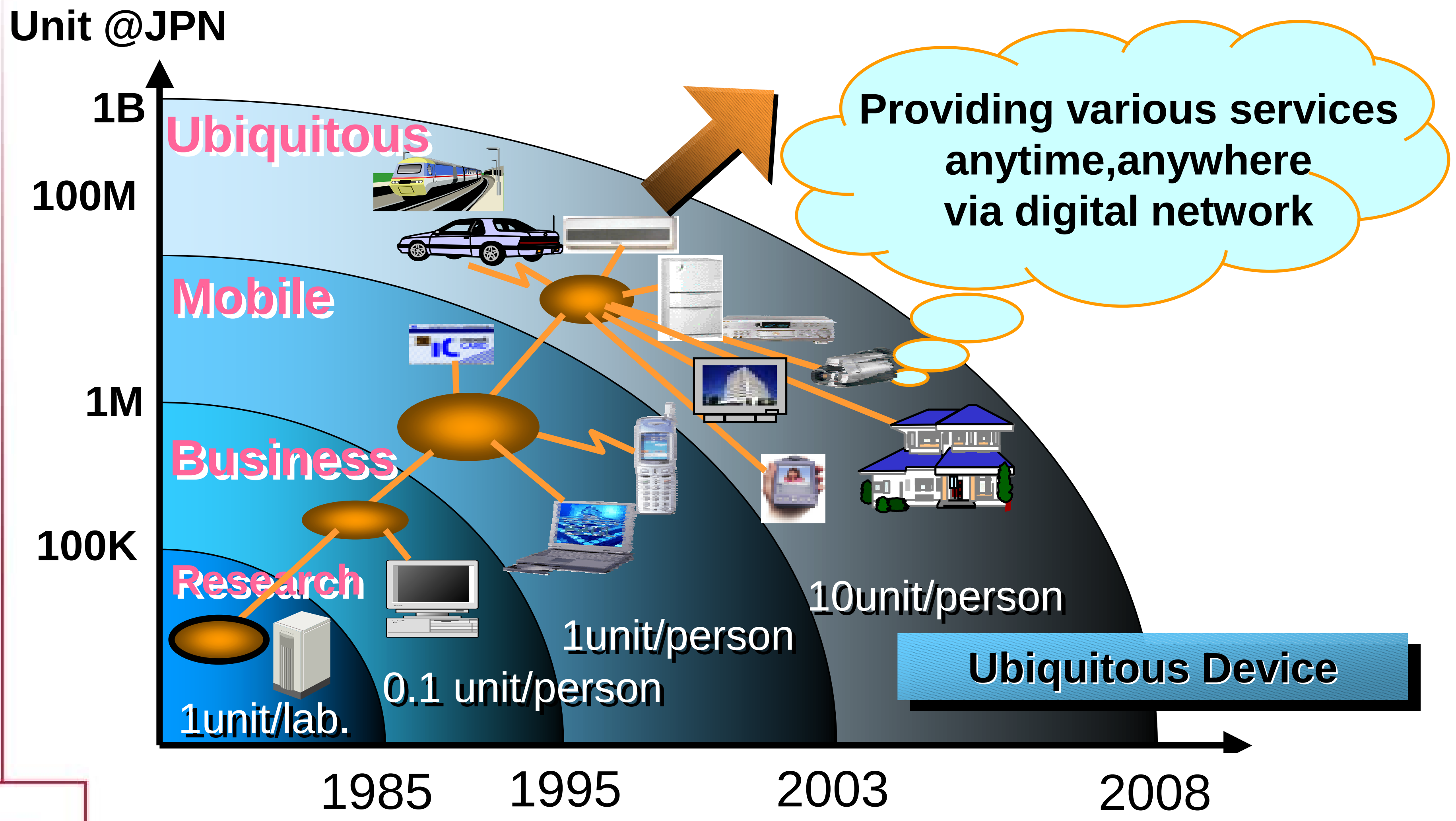
- Some facts about Renesas
- MCU portfolio and Applications
- Ubiquitous Computing Network Society
- Java on embedded MCU
- Optimisation of Embedded Internet application
- MCU Weblearning System

New era will start



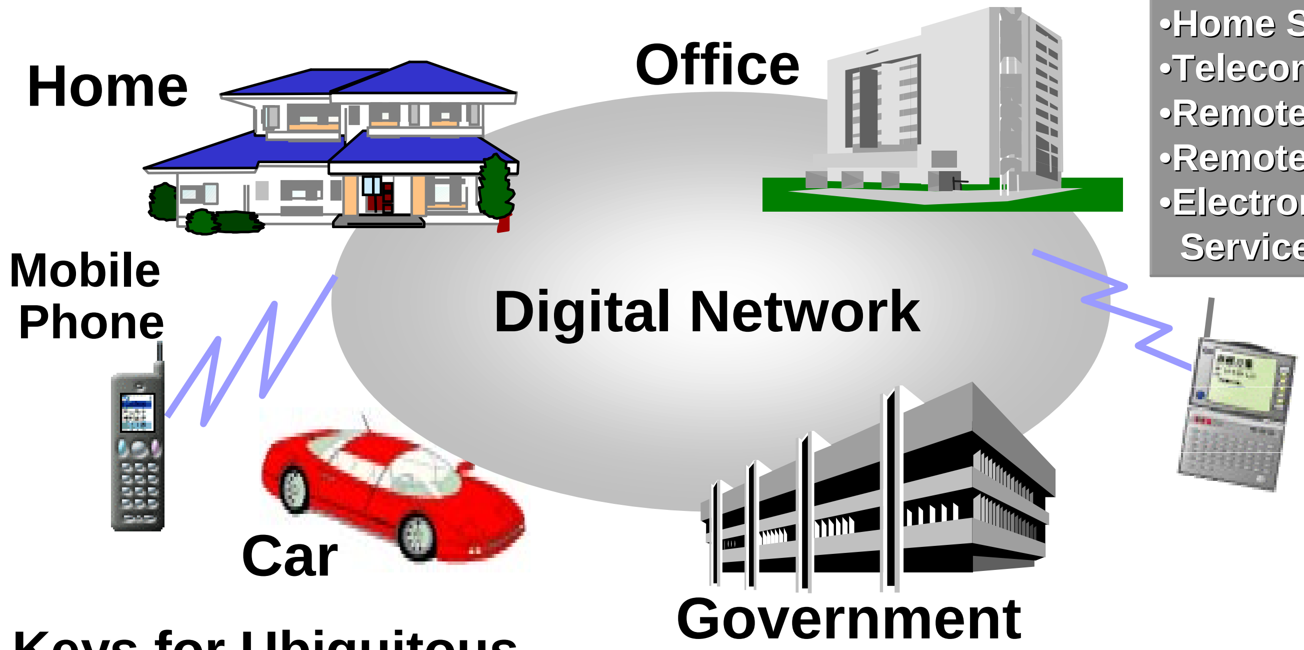
Renesas will… continue to be a reliable and trustworthy semiconductor manufacturer, providing intelligent chip solutions to create a more prosperous ‘Ubiquitous Society’ and giving global support to our system partners.

New Phase of Network toward Ubiquitous Era



Ubiquitous Computing Network Society

- Providing various services anytime, anywhere via digital network



Service Contents

- Home Shopping
- Telecommuting
- Remote Education
- Remote Medical Care
- Electronic Government Service etc.

- Keys for Ubiquitous

Amusement | Barrier-free | Reliability & Safety

From Cellular to Mobile Universal Terminal

Cellular driving the technologies for ubiquitous computing

Anywhere

- Human I/F technology (ten-key character translation)
- Universal remote (bridge with I-net between home appliance)
- Positional information(GPS)

Amusement

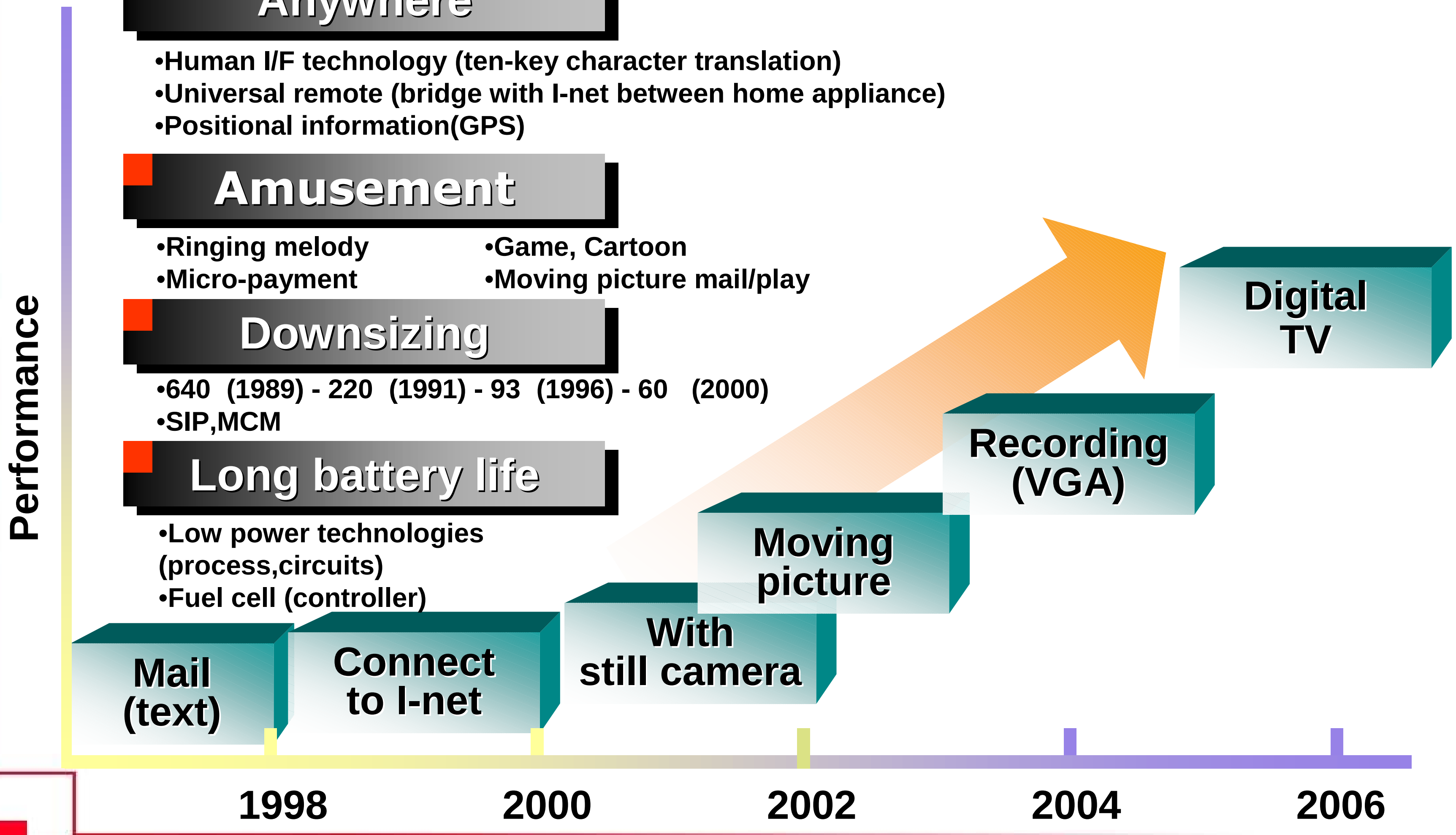
- Ringing melody
- Micro-payment
- Game, Cartoon
- Moving picture mail/play

Downsizing

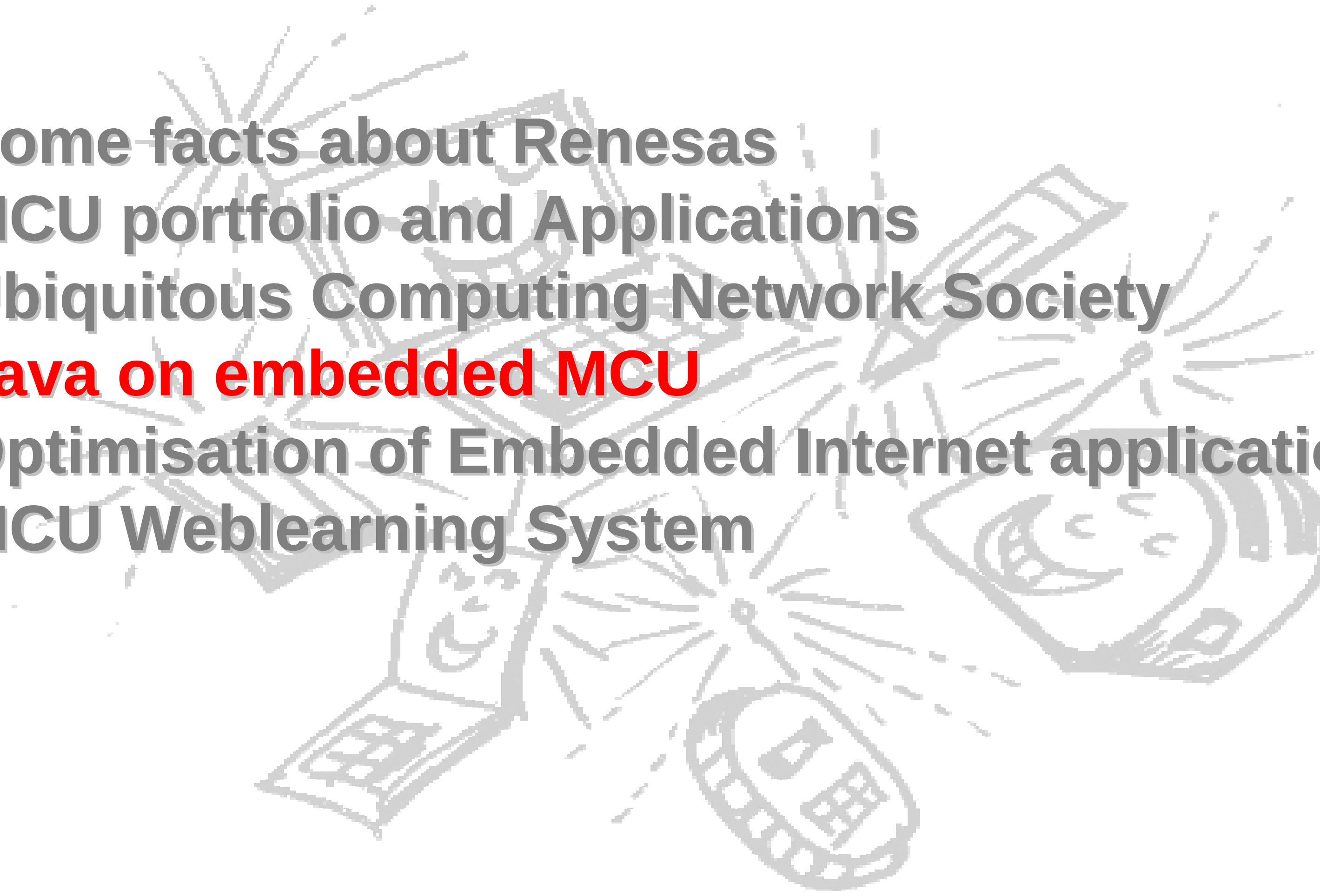
- 640 (1989) - 220 (1991) - 93 (1996) - 60 (2000)
- SIP,MCM

Long battery life

- Low power technologies (process,circuits)
- Fuel cell (controller)



Content



- Some facts about Renesas
- MCU portfolio and Applications
- Ubiquitous Computing Network Society
- Java on embedded MCU
- Optimisation of Embedded Internet application
- MCU Weblearning System

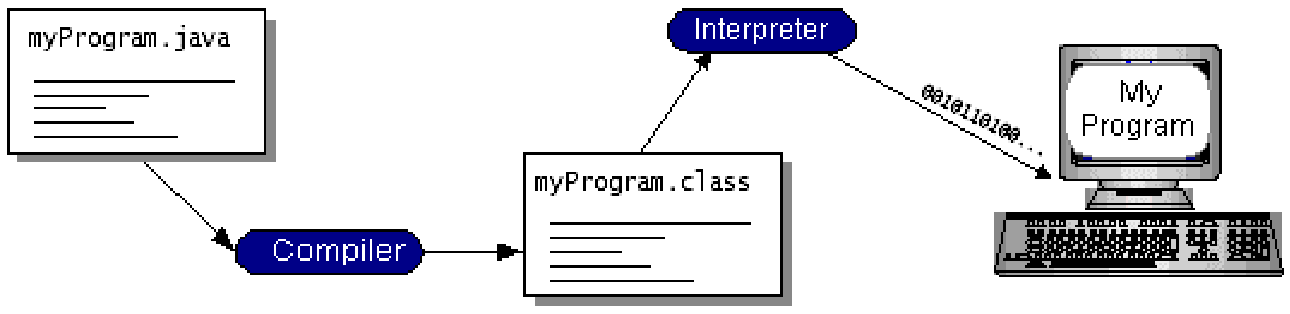
Java programming language

The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

- Simple
- Architecture neutral
- Object oriented
- Portable
- Distributed
- High performance
- Interpreted
- Multithreaded
- Robust
- Dynamic
- Secure

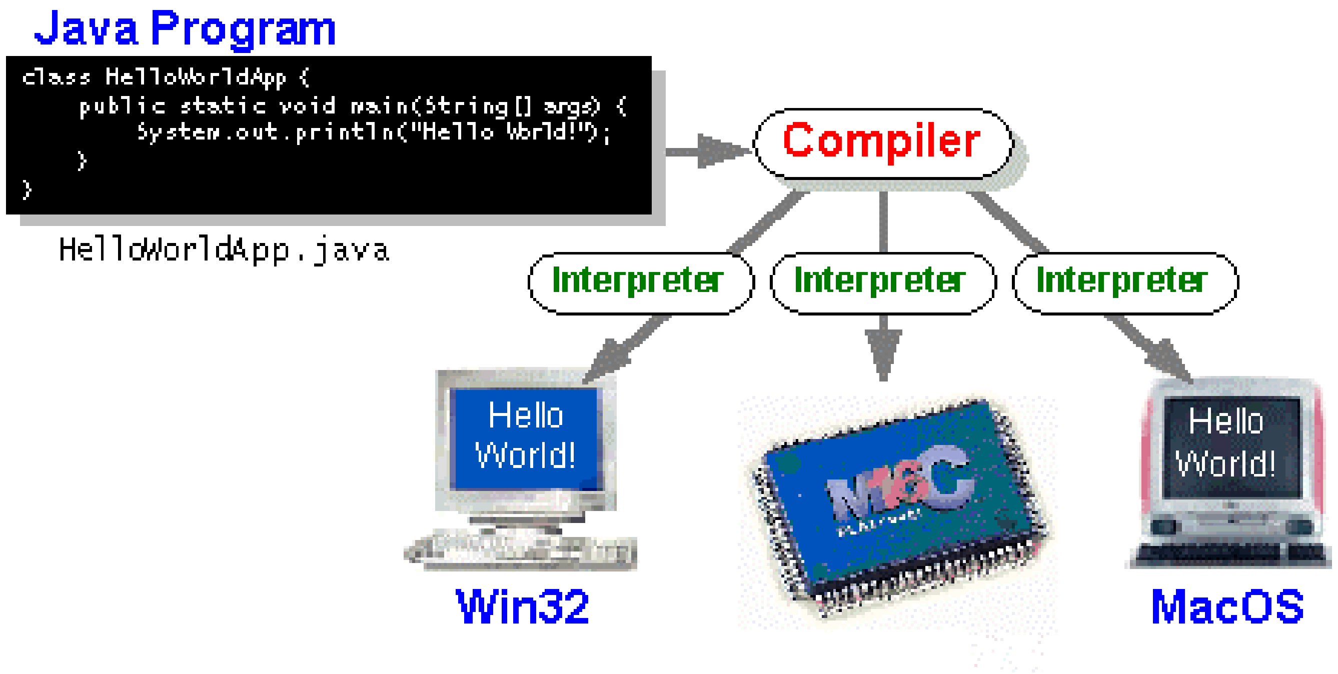
How does it work

- **Java source myProgram.java code will be translated into intermediate language**
- **The generated myProgram.class Java bytecode is platform independent.**
- **Java bytecodes is like the machine code instructions for the *Java Virtual Machine* (JVM).**
- **The interpreter on the platform parses and runs each Java bytecode instruction on the computer**
- **Compilation happens just once; interpretation occurs each time the program is executed**



Java compatibility

- Java bytecode supports "write once, run anywhere“.
- Compiling can be done on any platform with Java compiler.
- The bytecode itself can run on every platform with a JVM running on.



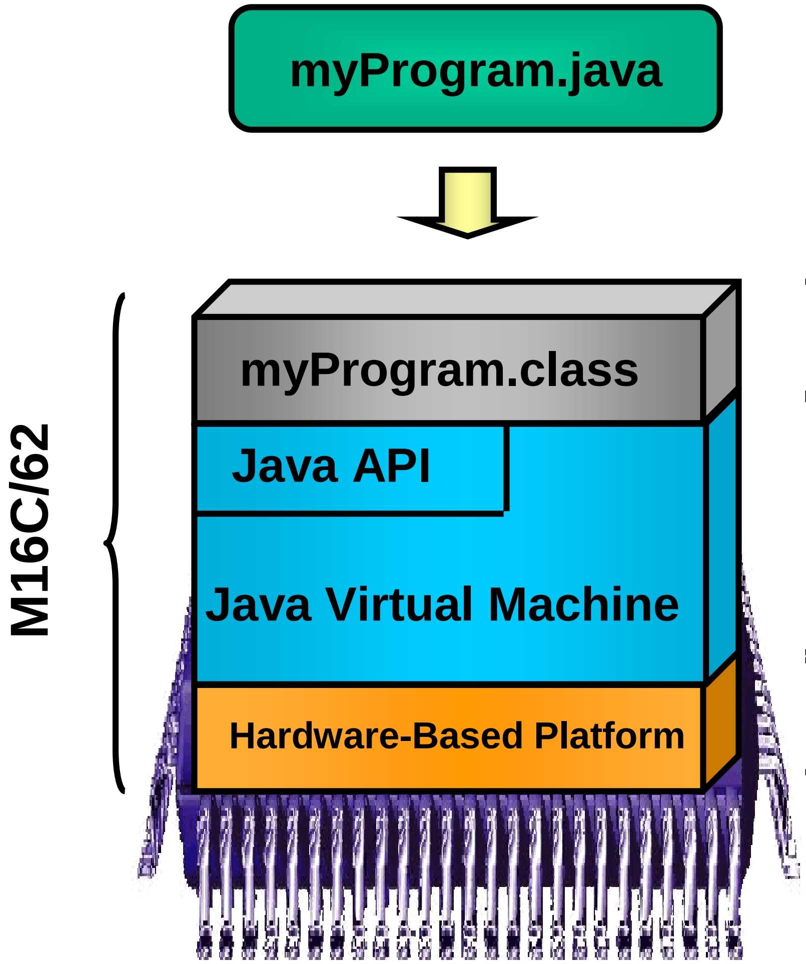
```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

There are huge differences between a server, a desktop, and a consumer device.

Java 2 Platform, Micro Edition

- **Java 2 Platform, Micro Edition (J2ME) addresses the vast consumer area.**
- **Technologies, libraries, and tools are included under the J2ME branding framework.**
- **Cell phones, pagers and personal organizers are examples of devices, which fit in the J2ME sub category CLDC (Connected, Limited Device Configuration).**
- **Specific for the CLDC category are following characteristics:**
  - **Devices have very simple user interfaces**
  - **Minimum memory budgets**
  - **Low bandwidth,**
  - **Intermittent network access**
- **The KVM is a compact, portable Java virtual machine specifically designed from the ground up for small, resource-constrained devices.**
- **The KVM is so named because its size is measured in kilobytes, around 80-100 Kbytes, fitting in the tiniest device**

Java on M16C/62



JDK - v.1.3.1_02 for Bytecode generation

Bytecode located inside Flash

Java Platform
J2ME_CLDC - v.1.0.3
with KVM

Pysical layer
C-code based, done with
IAR EWM16C V2.10a

Status of M16C Java Port

- M16C Java port is currently done by Prof. Jerker Delsing of the Lulea University of Technology.

- KVM operates on M16C
- Communication over serial port to terminal as standard IO
- Capable to run simple Java programs like e.g. HELLO WORLD
- Memory usage:
  - 233 kB Flash
  - 75 kB RAM

M16C Java Demoboard

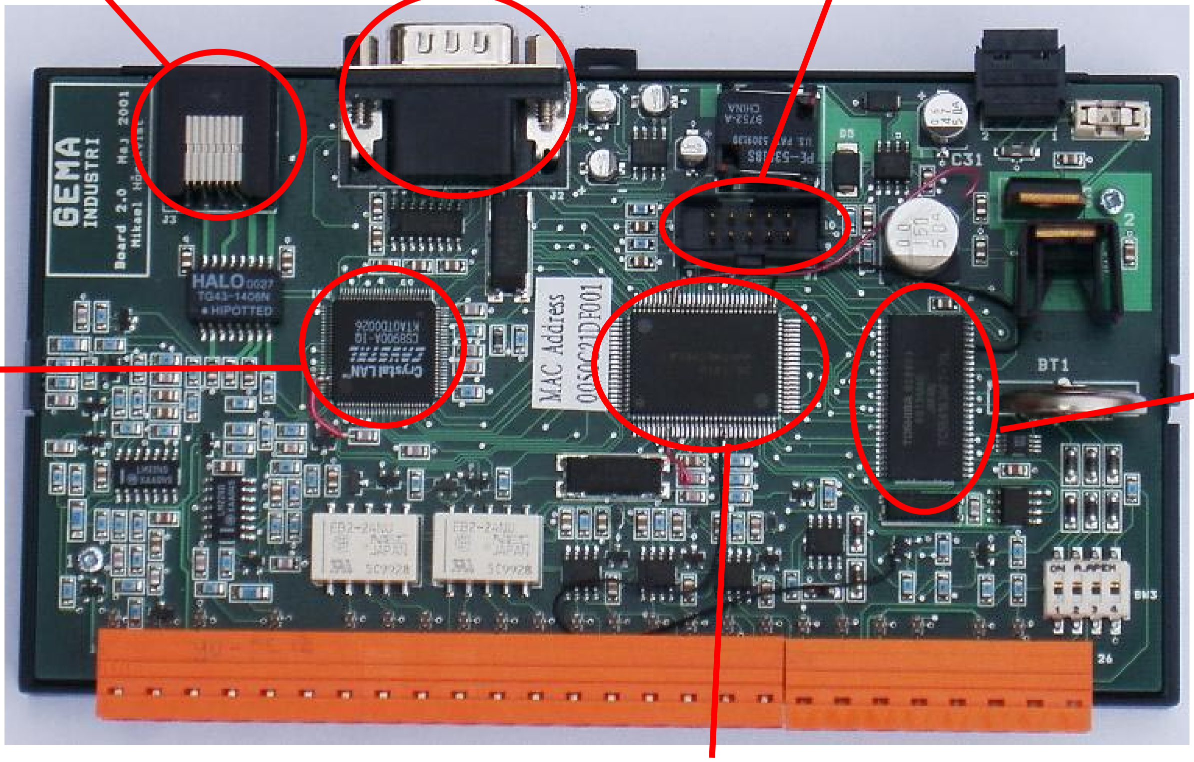
RJ45 Ethernet I/F

Terminal SubD I/F

Debug I/F

Ethernet chip

128 kB RAM

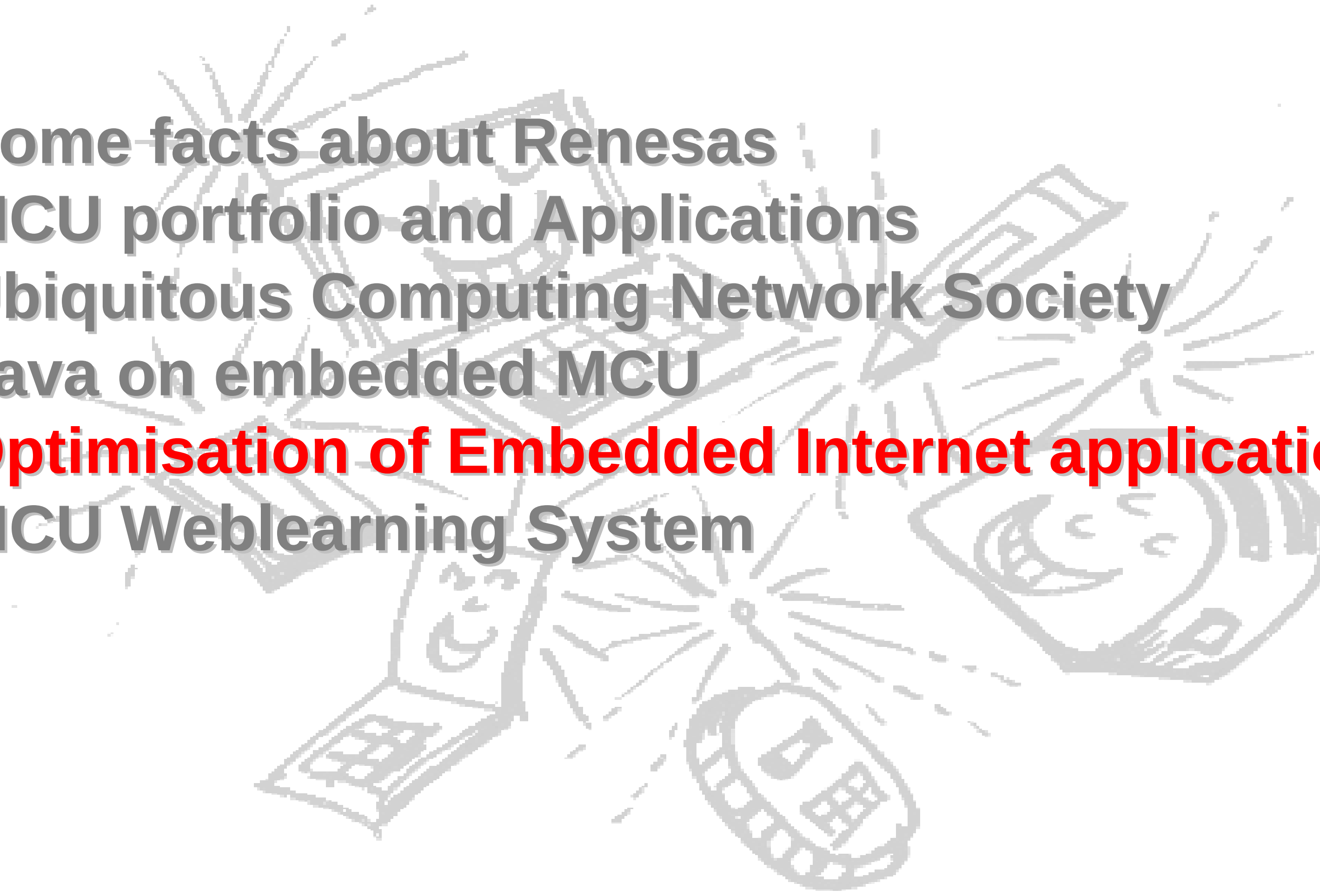


M16C/62A

–256 kB Flash

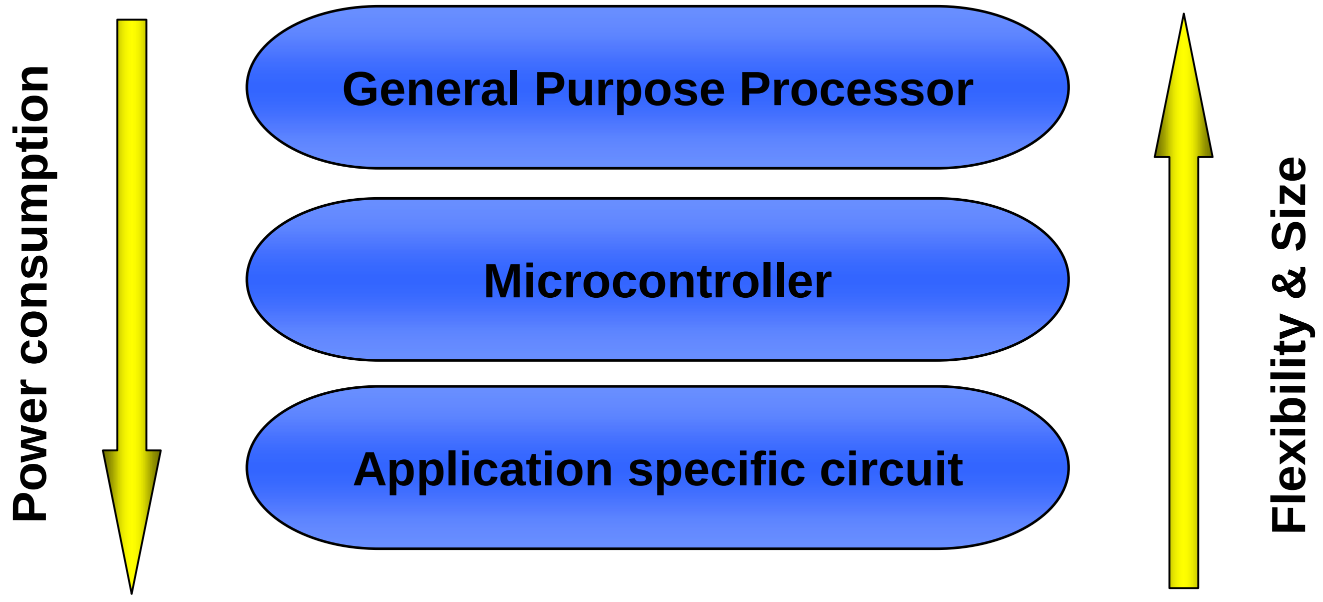
–20 kB RAM

Content



- Some facts about Renesas
- MCU portfolio and Applications
- Ubiquitous Computing Network Society
- Java on embedded MCU
- Optimisation of Embedded Internet application
- MCU Weblearning System

Positioning of MCU‘s



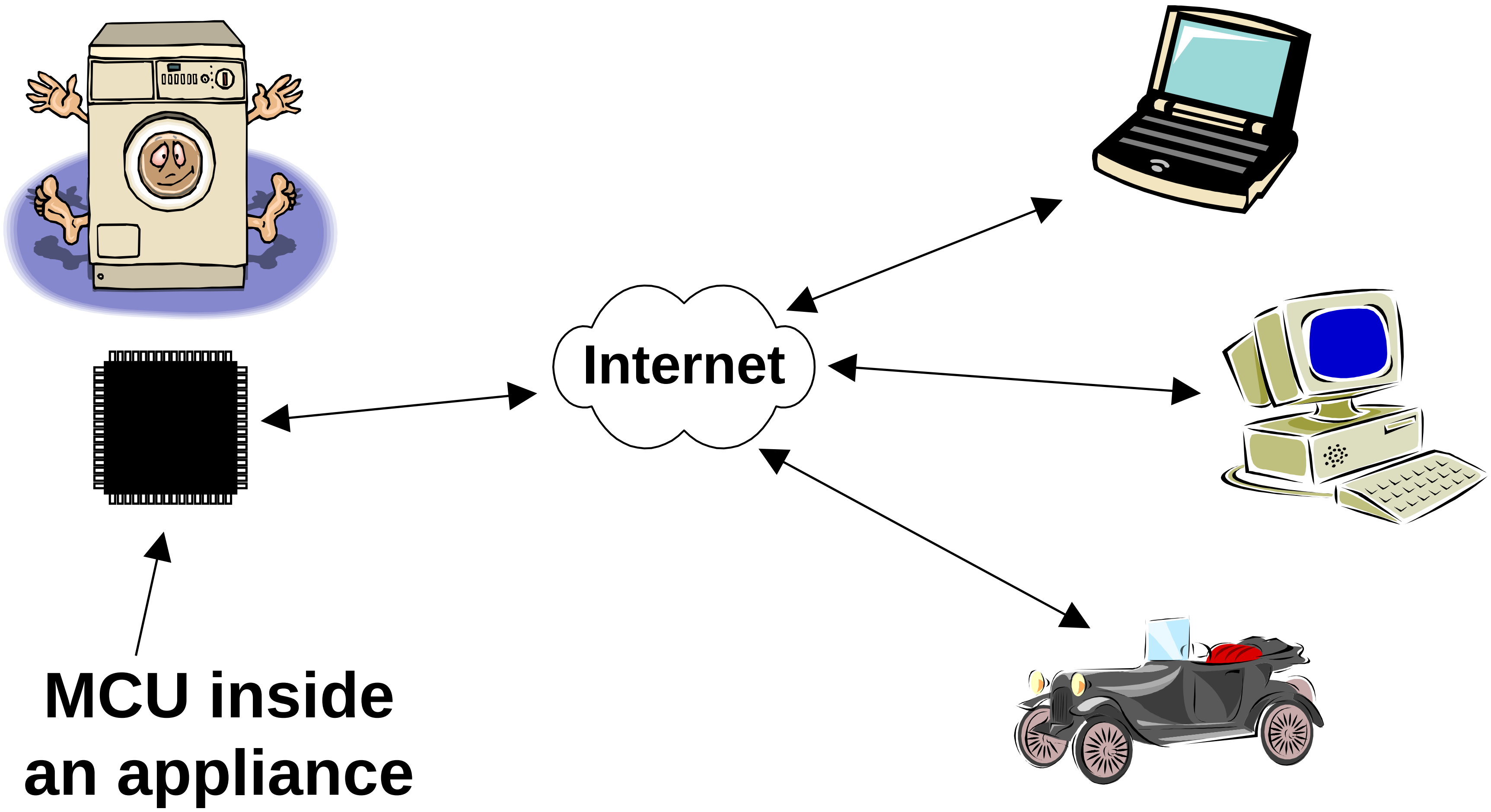
Capabilities of MCU

- Reduced processing power
- Small or no display, LEDs
- Lower energy consumption

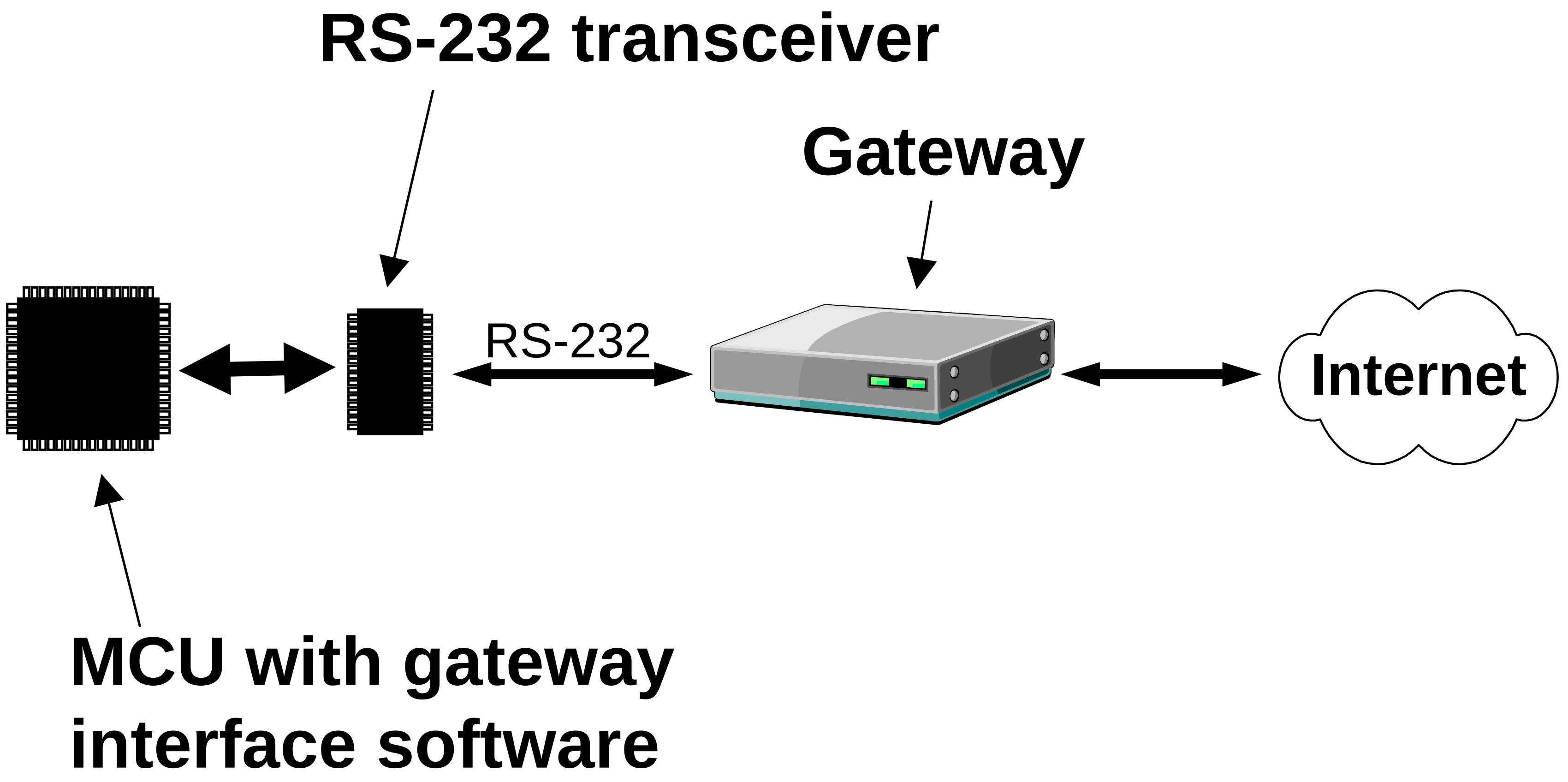
Advantages of MCU

- Small size
- SW fits to task
- Predictable behaviour
- Simple interaction with environment, due to local awareness

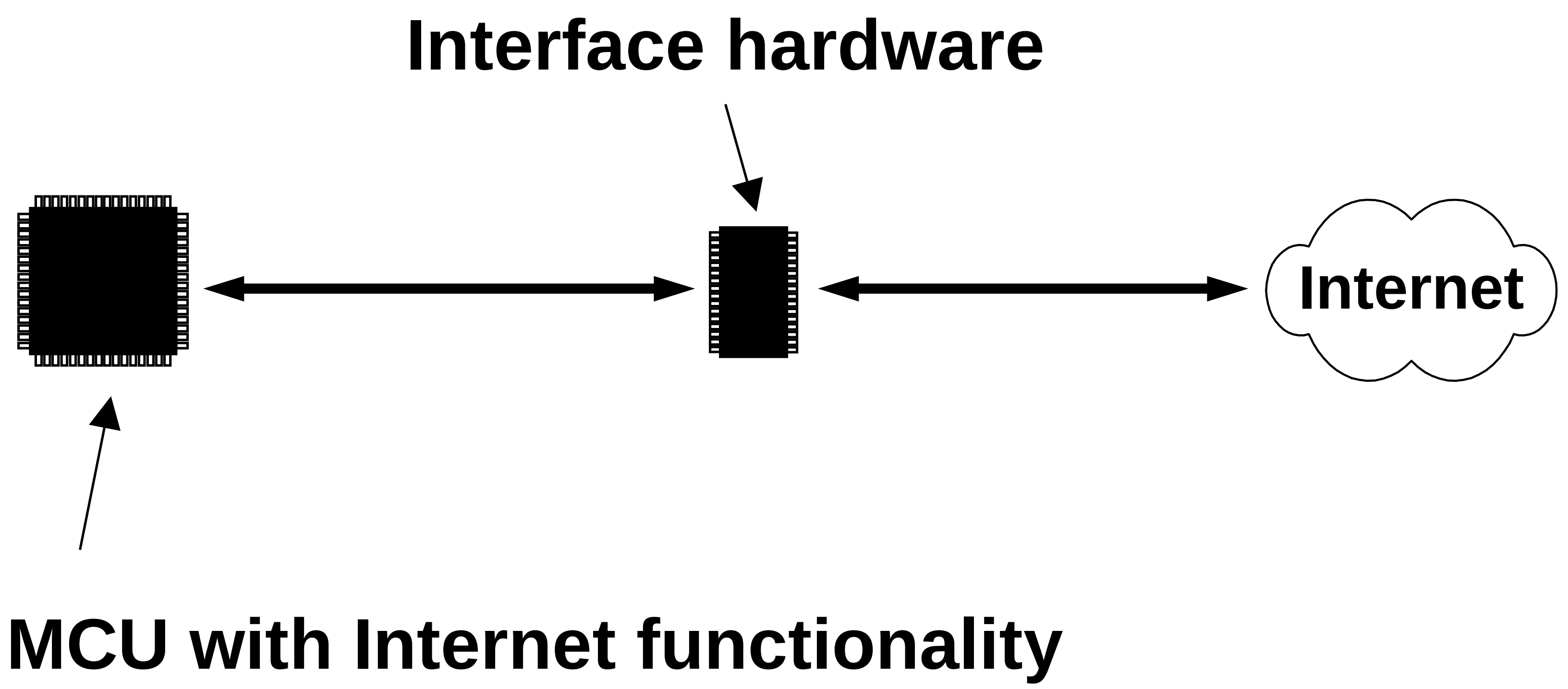
Why the internet?



Hardware gateway



Software solution

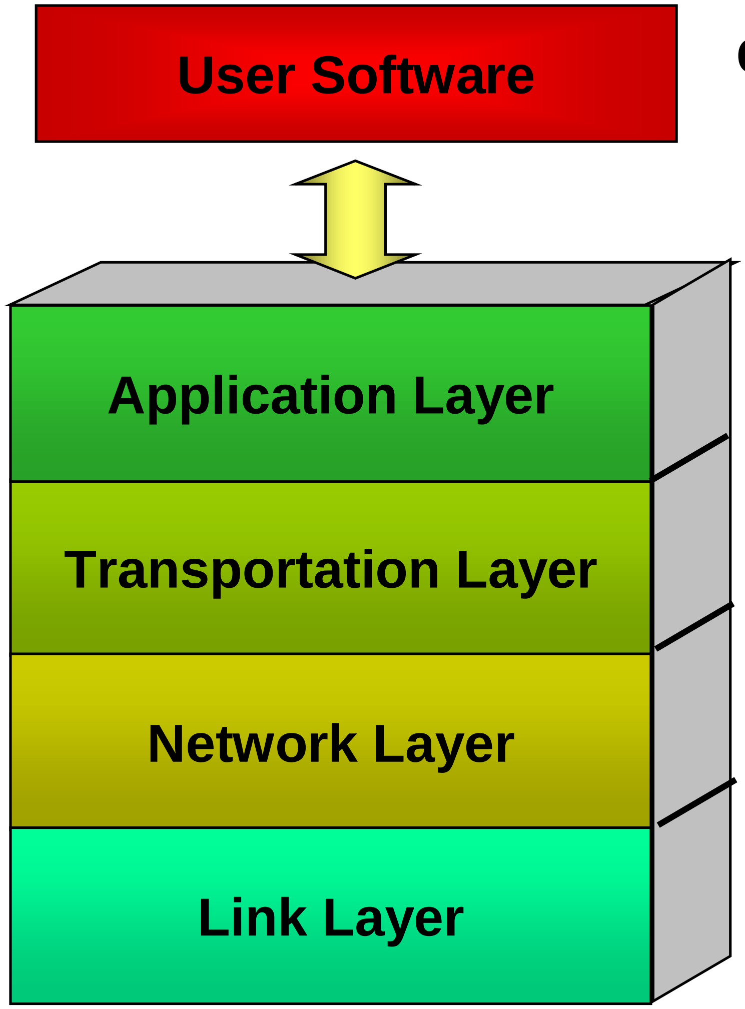


Significant ways of usage

Embedded internet functionality can be utilized for:

- **In-circuit development.**
- **Downloading new code or data.**
- **Configuration.**
- **Uploading stored data.**
- **Diagnostics.**

OSI Internet Protocol stack



e.g. Web-Server, Email Client

FTP, SMTP, POP3, HTTP

TCP, UDP

IP

PPP, ARP, RARP

Checking your needs

Due to limited resources of an Embedded Internet application, following points should be considered:

(example Email appl.)

- **Is sending and receiving of Emails necessary?**
- **What is the use of the mail, messaging or writing?**
- **Which language should be supported?**
- **Will binary attachements be used?**
- **How much system resources are available?**
- **What does additional software require?**

Hardware considerations

- <u>Use a MCU which can access multi-byte values at byte boundaries</u>
  - Network protocol headers often are byte- or word aligned.
  - MCU should natively be able to store e.g. word values at an odd address.
  - No additional C code needed to access those addresses.
  - To save ROM size, a processor should be chosen which can access multi-byte values at any address, eg. M16C family.

- <u>Store multi-byte values in Big Endian order</u>
  - Computer networks are Big Endian
  - If Little Endian MCU are going to pass integers over the network (IP checksums for example), they need to convert them to network byte order.
  - If Little Endian MCU are going to receive integer values over the network, they need to convert them back to their own native format.
  - To avoid this conversation, values should be stored as they come over the network.

Hardware considerations

- <u>**Exploit all hardware features**</u>
  - **Most MCUs contain advanced peripheral that can make life easier.**
  - **M16C family has a CRC engine on board.**
  - **M32C family has a additional HDLC block onboard.**
  - **This reduce code size and speed up performance.**

Protocol considerations

- <u>**Simplify the protocols and write them from scratch**</u>
  - **Most network protocol, due to their mature code base, are very huge.**
  - **Optional feature are included**
  - **Huge amount of memory is needed**
  - **A multitasking OS is needed**
  - **To run this protocols on a MCU functionality and size have to be stripped down**
  - **Best way to do is to reinvent the wheel from scratch and do not follow the RFC completely.**
  - **It's only relevant that your implementation generates what the other side expects.**

Protocol considerations

- **Simplify the protocols and write them from scratch (Part 2)**
  - **A good example of such simplification is the ICMP echo request/reply**.

~~- Received an ICMP echo request~~

~~- Construct a new echo reply~~

~~- Send echo back to the addresser~~

~~This procedure is needs a lot of performance due to:~~

~~- IP headers creation~~

~~- Checksums creation~~

~~- ICMP message assembly~~

~~- Maybe data must be copied from the request to the new reply message before it can be sent.~~

- Received an ICMP echo request
- Change the type field of the received message to ICMP_ECHO_REPLY
- Swap IP addresses of the IP header.
- Adjust the ICMP checksum field
- Send back the modified packet.

The result is the same but took place much faster – and less code is needed

Protocol considerations

- <u>**Reuse memory of different protocol states**</u>
  - **Protocols likes PPP usually walk through different states**
  - **After negotiation PPP use the IP to transmit TCP/IP**
  - **PPP never does link negotiation during an open connection**
  - **PPP is either in link negotiation or any other state**
  - **This behaviour makes it possible to use the same memory for all variables of different states.**
  - **Just create a union which members are structures that hold the variables for one state.**

```
union
{
        struct
        {
                int m1;
                int m2;
                ....
        } state1;
        struct
        {
                int m1;
                int m2;
                ....
        } state2;
        ...
        ...
} all_states;
```

Protocol considerations

- 'On the fly' PPP frame de/encoding
  - **PPP uses HDLC framing to let packets travel over serial lines**
  - **HDLC frames consist of start/end flag, payload and CRC value, as well data en/stuffing**
  - **Incoming frames must be recognized and decoded**
  - **Outgoing data must be transformed into PPP frames**

~~**Typical handling:**~~

~~A memory wasting framer read packets into a buffer and then decode them.~~

~~Outgoing frames will be stored into buffers, before they will be send~~

Better way:

A state machine should be used which handles the incoming frame byte by byte in a streaming manner.

On transmission, a state machine should generate single bytes which are immediately delivered to the UART.

This method requires only a few variables instead of several 100 of bytes.

Protocol considerations

- <u>**How to avoid Webserver input buffering**</u>
  - **A webserver starts transaction after receiving a complete HTTP request.**
  - **HTTP requests can be quite long, e.g. 0.5kBytes are not seldom**
  - **Therefore collecting the complete frame into memory is not useful**
  - **A special state machine which processes the stream byte by byte direct can save memory**
  - **The state machine have to set some variable and extract strings from the stream**
  - **So the webserver has all information but the HTTP request frame itself is gone**

Protocol considerations

- **<u>Minimize memory usage on Webserver output</u>**
  - **The reply frame of a webserver may be very long as well**
  - **Depending on the HTTP request, it consists of:**
    - **Different HTTP headers**
    - **Different HTML pages, bitmaps, applets and so on.**

~~**Typical handling:**~~
~~Usually, when a webserver builds a HTTP frame, it assemble all parts using a buffer and then delivers the complete frame to TCP for transmission.~~
~~This needs a lot of RAM.~~

Better way:

-Static HTTP header fragments are stored in ROM.

-Content (e.g. HTML page) is largely stored in ROM

-Webserver builds the variable HTTP header fragments and also the variable site content (e.g. for dynamic HTTP).

-Webserver generates a pointer/size list.

-The list contains pointers to fragments of the HTTP header and to the parts of the content (static and variable).

-When the webserver has finished building the list, the pointer list is (element-by-element) delivered to TCP.

-At last the TCP connection is closed.

Considerations for buffer handling

- **Buffer sharing**
  - Normally each protocol layer do have it's own receive and transmit buffer.
  - You can save a lot of memory, if you do share the buffers between all layers
  - Lowest network layer software (e.g. PPP or an Ethernet driver) manages the receive and transmit buffers
  - If higher level protocol wants to send something, it writes the data directly into the transmit buffer and tells the lower layer to send it off to the network
  - On reception the higher level get a pointer delivered to the receive buffer and a size information

Considerations for buffer handling

- **The moderate way - 2 shortened buffers**
  - **According to PPP standard, PPP should always be prepared to receive frames of 1500 bytes in length**
  - **In fact, such large packets never appear in reality, because higher level protocols can be told to generate packets which are much smaller**
  - **Use two small frame buffers (one for TX and one for RX) which size is the maximum size of a packet you really expect.**

Receive handling:
A TCP, on connection establishment, can tell the other station which MSS (maximum segment size) it accepts. If this is 512 bytes per segment, then received TCP packets aren't greater than this value (plus overhead from the lower layer).

Transmit handling:

On transmission, you pull the strings by yourself. It's up to you how big transmitted packets are, as long as they match the minimal sizes and don't interfere with something like MSS (see above).

Considerations for buffer handling

- **The naughty way – 1.5 buffers**
  - A single processor without a task-switching operating system, cannot do transmission and reception at frame level simultaneously.
  - This allows for the use of a shared buffer.
  - Use a single shared frame buffer for transmit and receive
  - Additionally a very small modem receive buffer will be used
  - Data may arrive on the hardware at any time.
  - These bytes must be stored by an interrupt routine into a very small buffer modem buffer.
  - The size of this buffer depends mostly on the transmission speed and processor speed.
  - On reception, data is transferred from the modem buffer to the shared buffer until a packet is complete.
  - The disadvantage of having only a single buffer is that transmissions must wait and see if the buffer is available.

Considerations for buffer handling

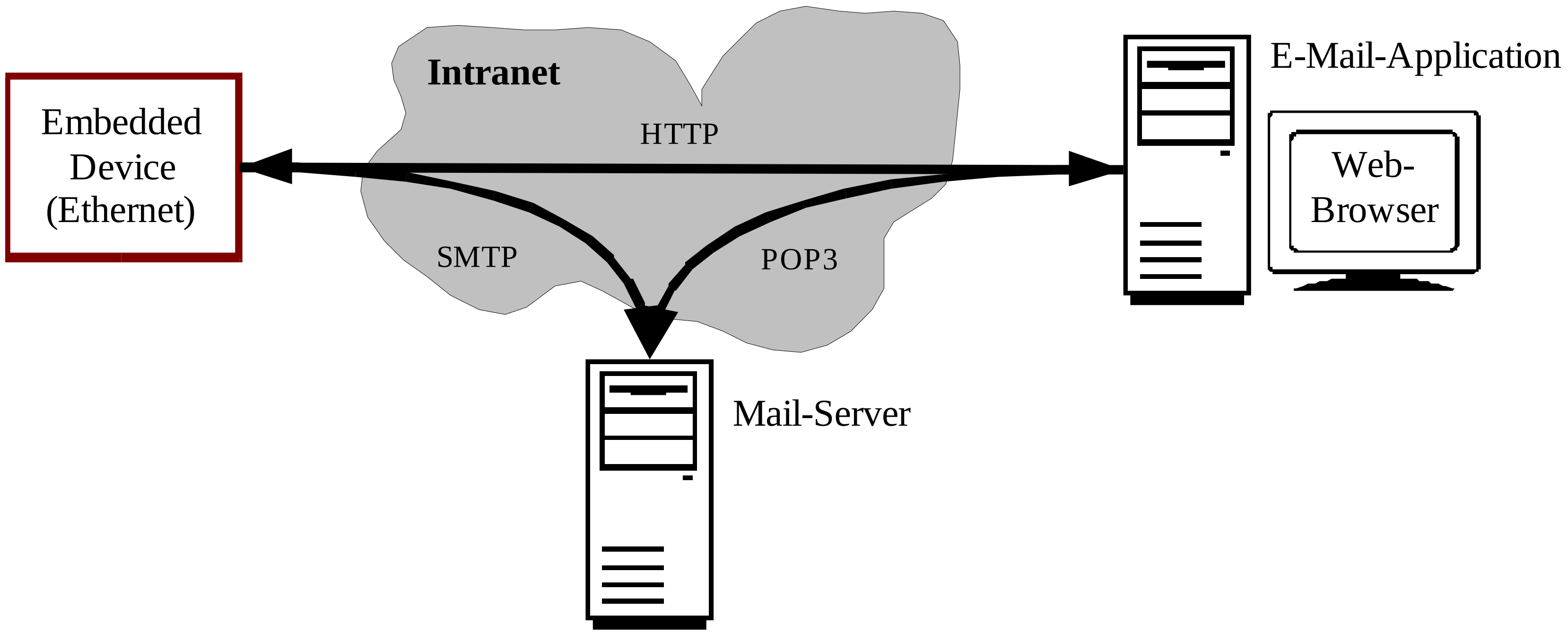
- The dodgy way – 0 buffers
  - Many Ethernet Controller Chips have on-chip memory which is used for buffering RX and TX frames.
  - Ethernet-enabled devices have a real chance to do TCP/IP without using buffer memory from the MCU's internal (or external) RAM.
  - An example is the SMSC LAN91C96.
  - It has about 6k bytes on-chip RAM and a convenient MMU (memory management unit).
  - The problem here is that Ethernet Controller's on-chip memory often must be accessed in a sequential manner.
  - One cannot read/write bytes from/to a random address.
  - So the protocols must be designed to deal with that restriction.

Requirements for a Embedded stack

- **Minimal system resources**
  - designed for embedded systems (8..32 bit)
  - stacks re-designed from bottom up new
  - minimal RAM/ROM resources
  - minimized CPU load
- **Easy to use API**
  - portable ANSI-C code
  - no OS required
  - callback function interface
  - reduced API function set

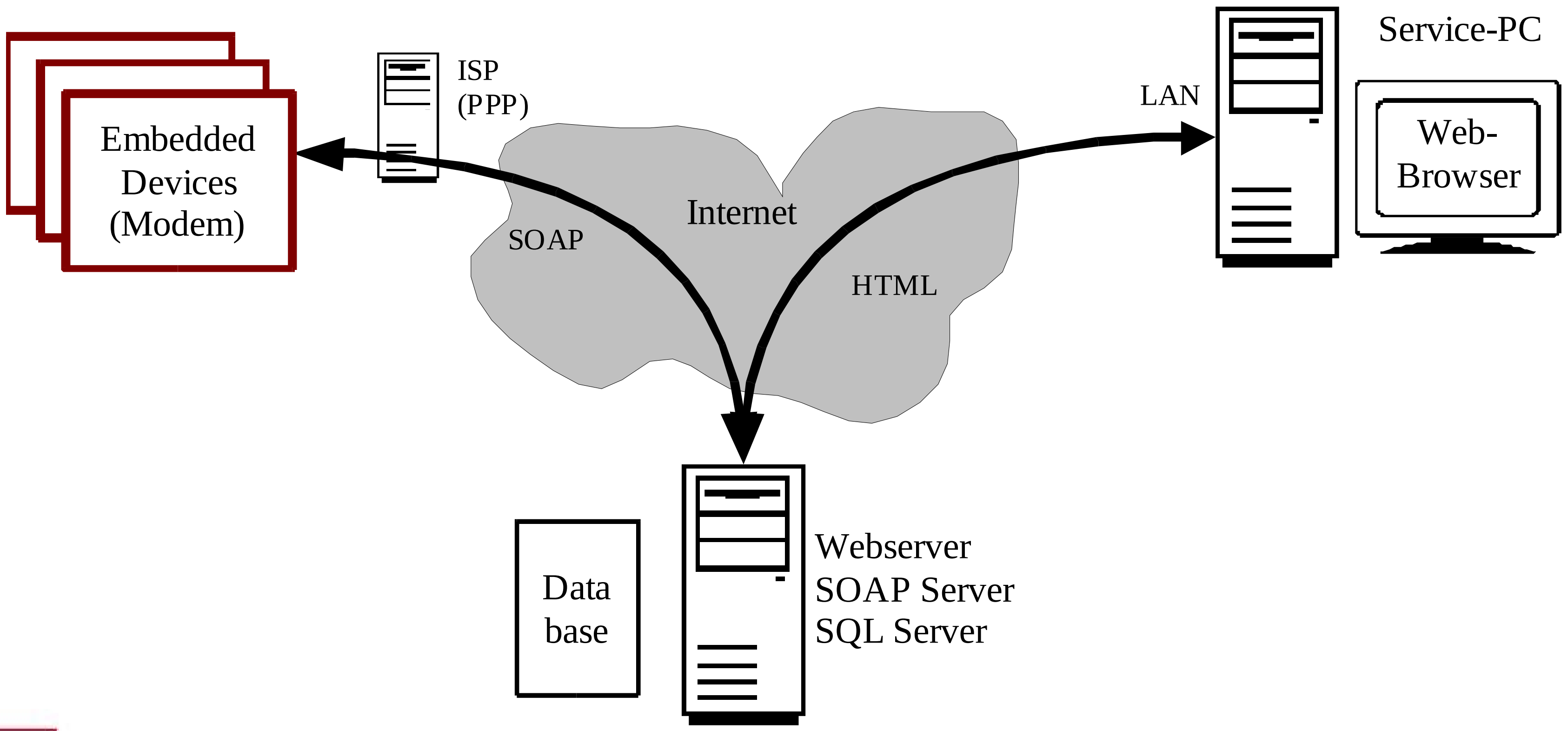
Example A

An ethernet network device
with embedded Webserver and SMTP



Example B

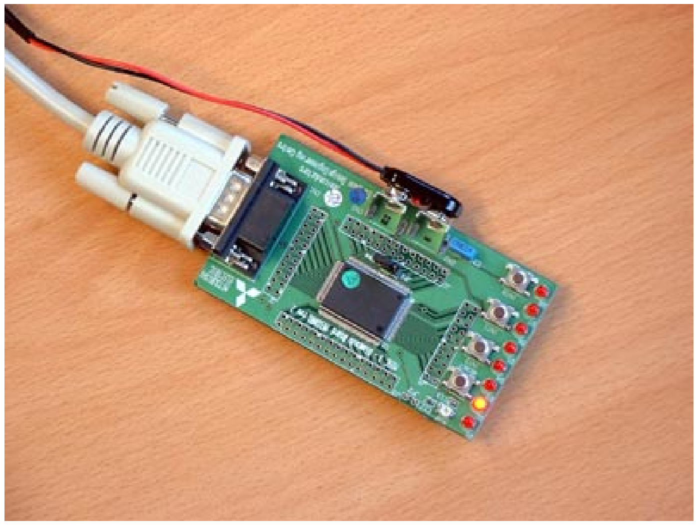
Multiple modem network devices with SOAP / XML communication



Demonstration

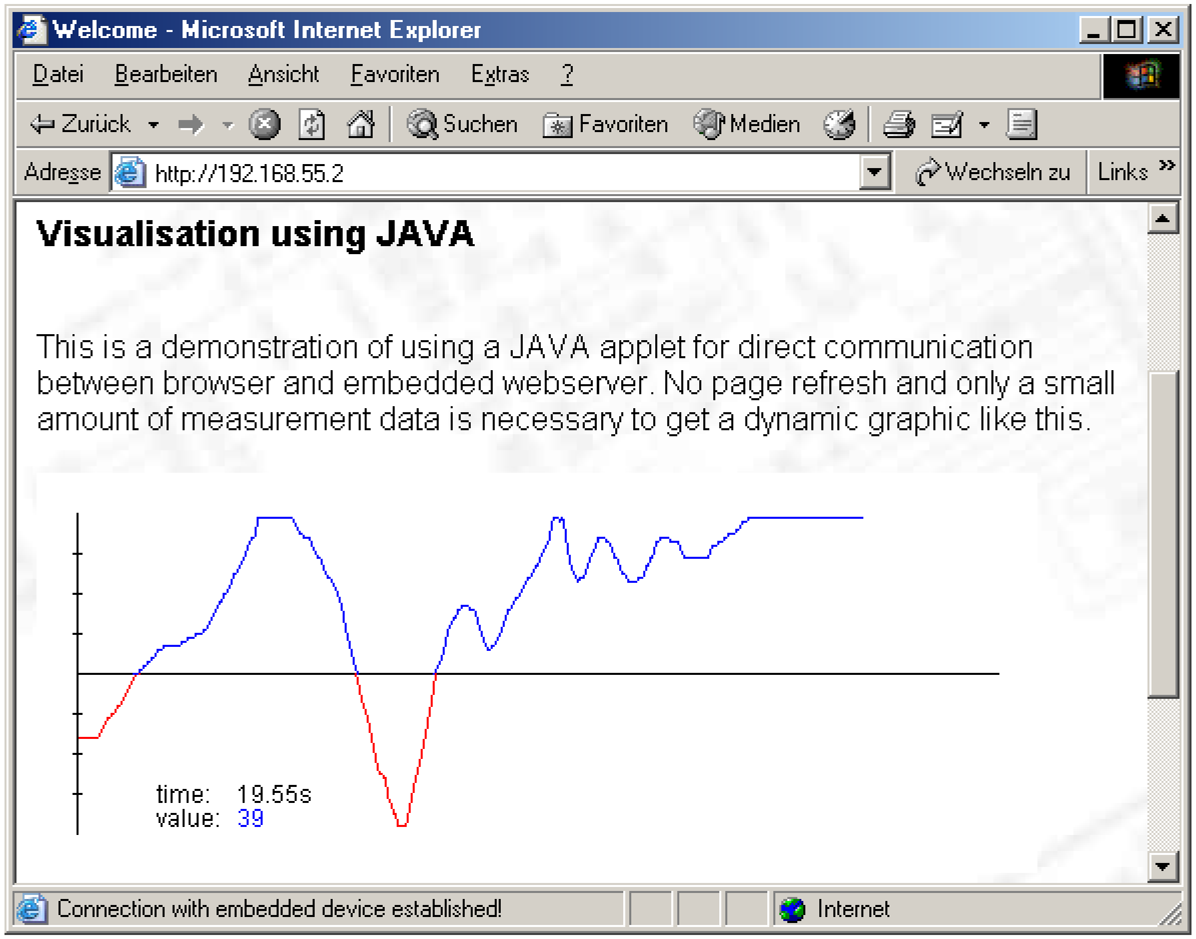
- Demo
  - 3D StarterKit with M16C A-Type (16 MHz)
  - Windows PC with serial 'Direct Connection' (38.4 kbs)
  - Web Browser

- Embedded Resources from sevenstax
  - sevenstaxPPP
  - sevenstaxTCP
  - sevenstaxWebserver



- Features
  - uses 11 kB RAM, 41 kB ROM Const, 23 kB ROM Code
  - 1st TCP connection: Embedded Webserver
  - 2nd TCP connection: Java applet for ADC data

Demonstration Output


Welcome - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Medien
Adresse http://192.168.55.2
Wechseln zu Links

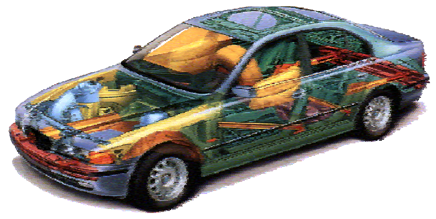
Visualisation using JAVA

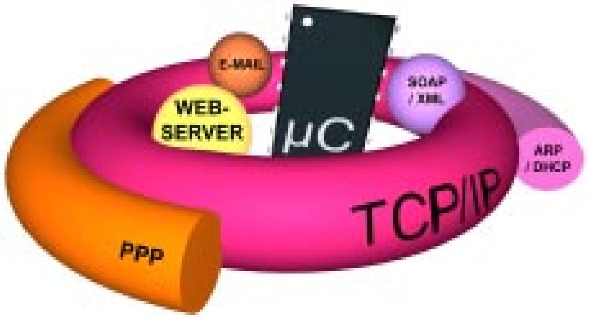
This is a demonstration of using a JAVA applet for direct communication between browser and embedded webserver. No page refresh and only a small amount of measurement data is necessary to get a dynamic graphic like this.

```
time:   19.55s
value:  39
```

Connection with embedded device established!

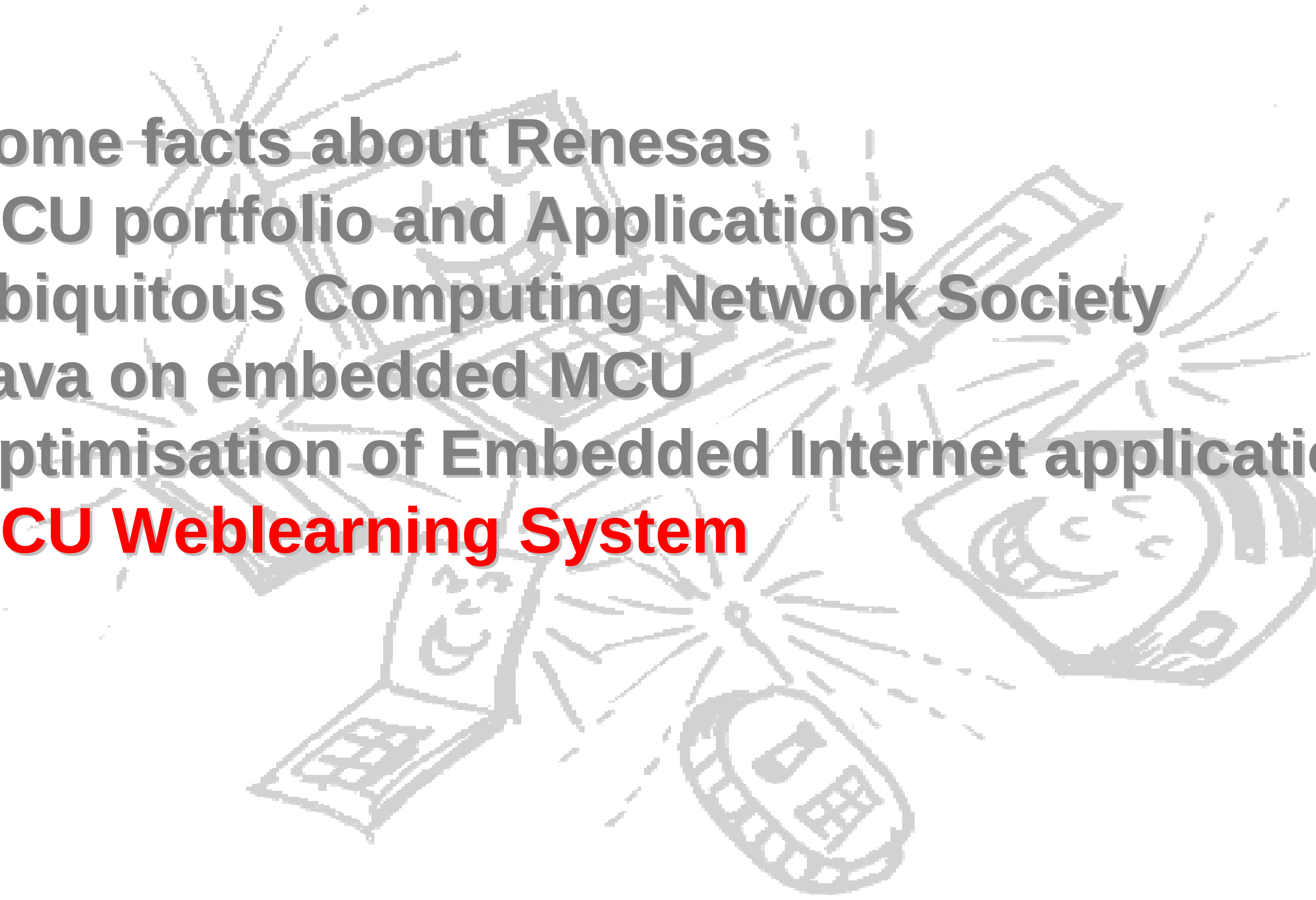
Field of Application





- •Intelligent Home, eHome
- •Facility Management
- •In-Car communication
- •Sensor/actuator information
- •Location-based information
  - -Tourist information
  - - Museum guides

Content



- Some facts about Renesas
- MCU portfolio and Applications
- Ubiquitous Computing Network Society
- Java on embedded MCU
- Optimisation of Embedded Internet application
- MCU Weblearning System

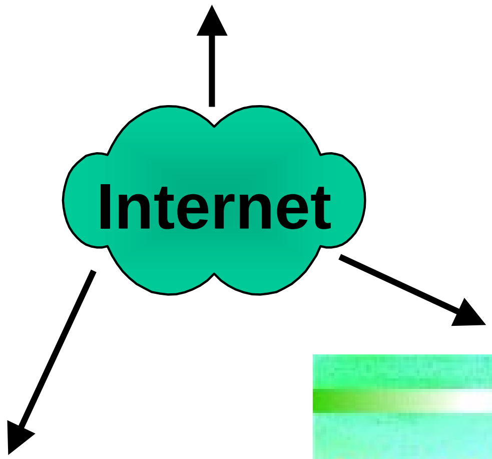
Renesas Weblearning Systems

Renesas Training Center

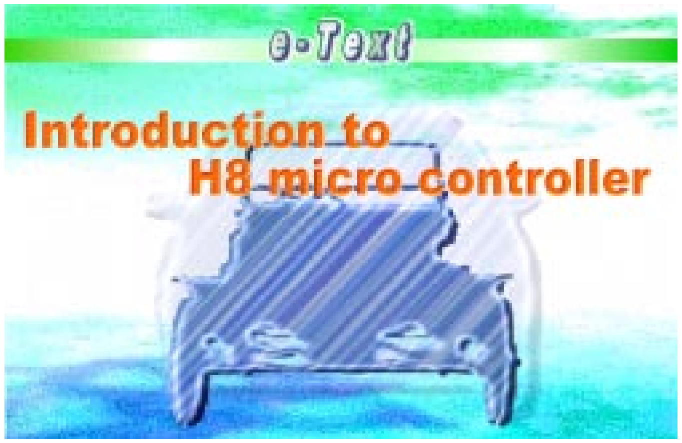
RENESAS Quick Learning

Multi-Function and High Performance 16-bit Microprocessor M16C

Welcome to the Web short time learning service for specific functions and RENESAS microprocessor M16C usages.



- 24/7 available
- Free of charge
- Interactive presentations
- World Wide access
- Trainings about MCU, Tools and SW



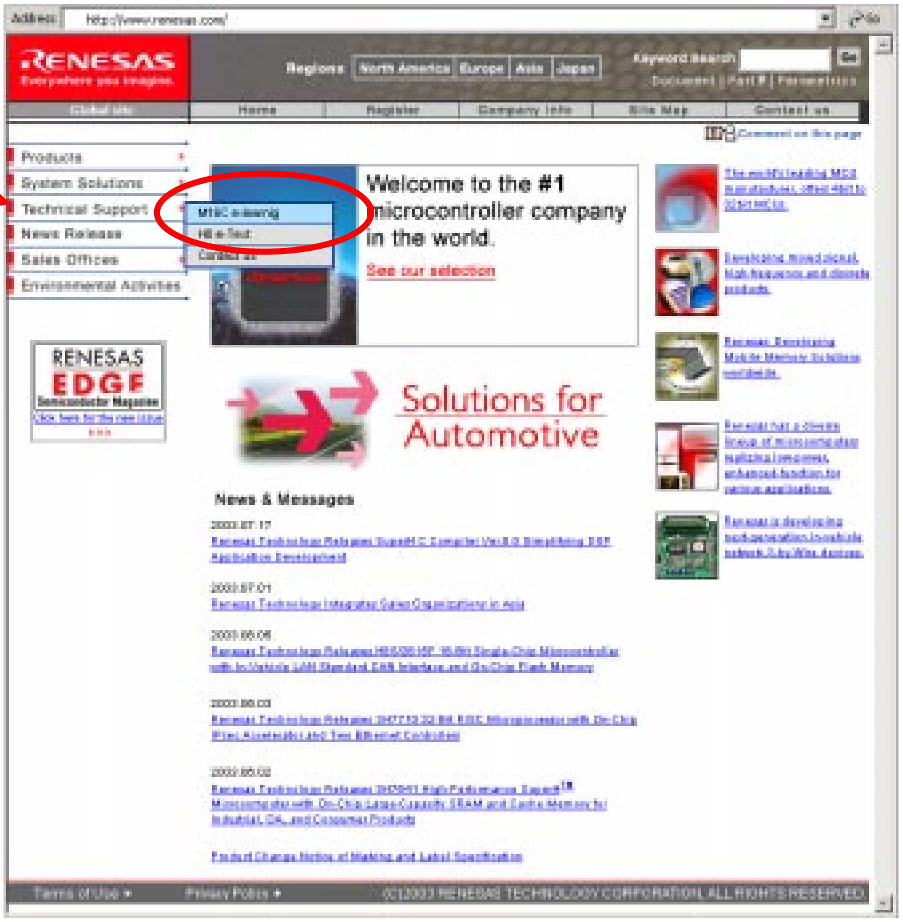
Introducing Renesas Quick learning

Access to the Quick Learning portal via:

http://www.renesas.com

Follow the „Technical Support“ entry
at the left menu.

Chose M16C e-Learning from the popup menu.



Welcome to the #1 microcontroller company in the world.

See our selection

News & Messages

2003.07.17
[illegible]

2003.07.01
Renesas Technology Integrates Sales Organizations in Asia

2003.06.06
[illegible]

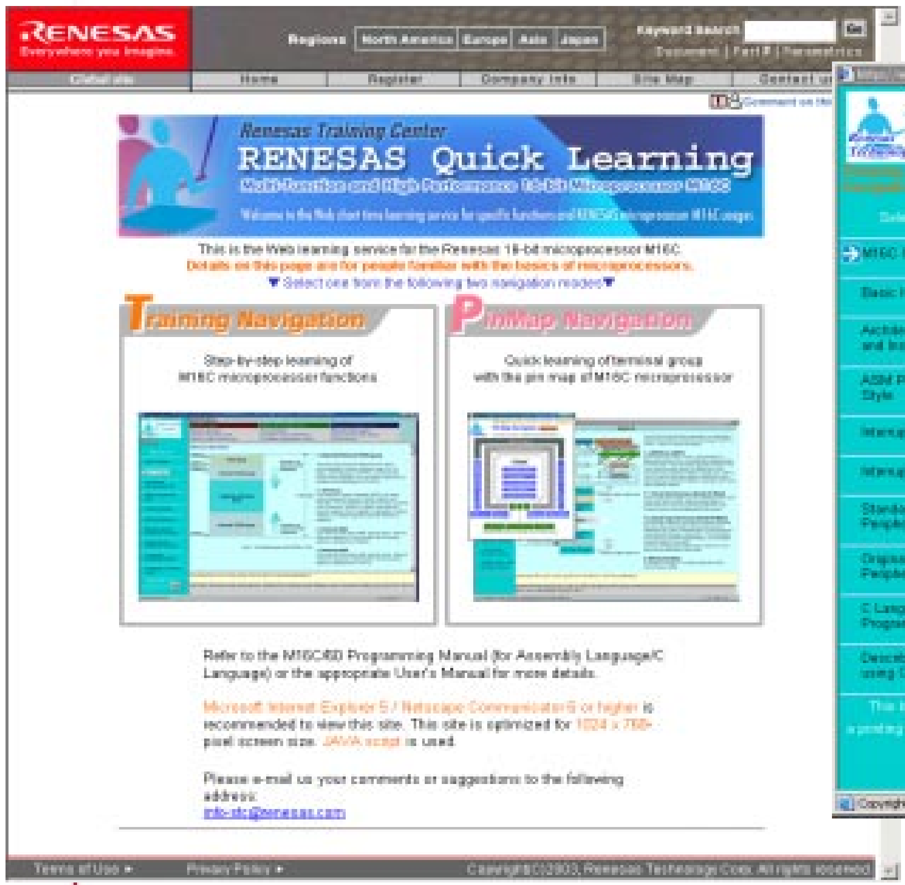
2003.06.03
[illegible]

2003.06.02
[illegible]

Product Change Notice of Marking and Label Specification

(C)2003 RENESAS TECHNOLOGY CORPORATION, ALL RIGHTS RESERVED.

Renesas Quick learning



Renesas Training Center

RENESAS Quick Learning

Multi-function and High Performance 16-bit Microprocessor M16C

This is the Web learning service for the Renesas 16-bit microprocessor M16C.

Details on this page are for people familiar with the basics of microprocessors.

▼Select one from the following two navigation modes!▼

Training Navigation

Step-by-step learning of M16C microprocessor functions

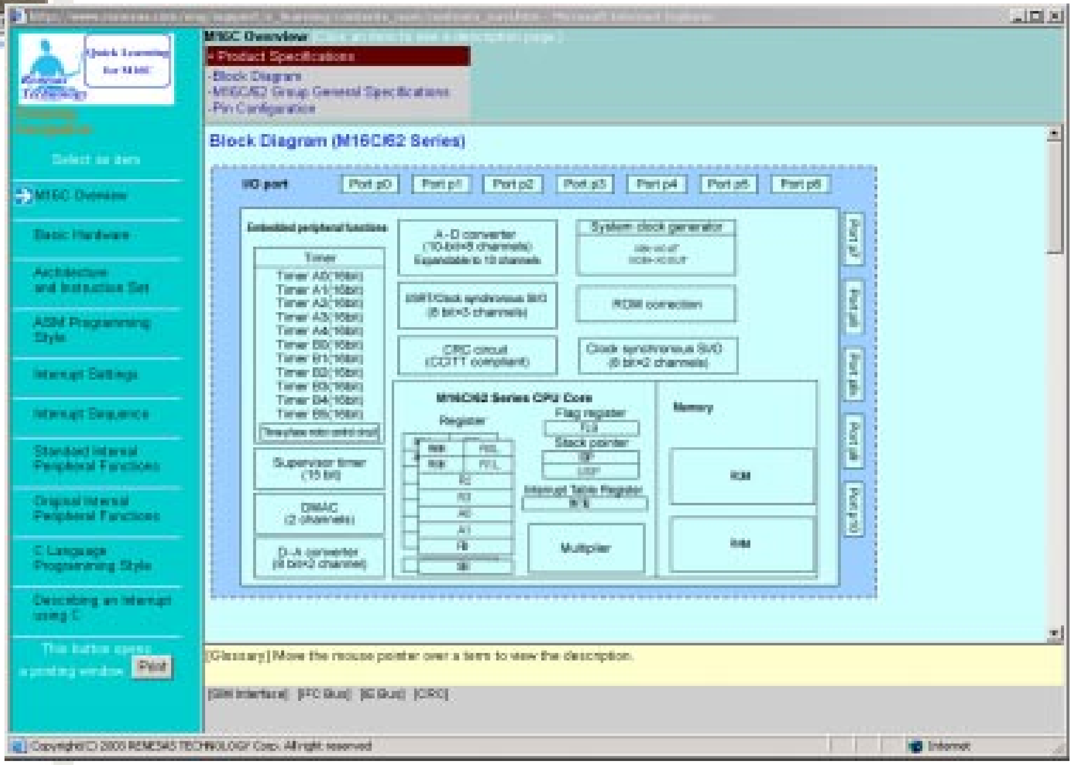
PinMap Navigation

Quick learning of terminal group with the pin map of M16C microprocessor

Refer to the M16C/60 Programming Manual (for Assembly Language/C Language) or the appropriate User's Manual for more details.

Microsoft Internet Explorer 5 / Netscape Communicator 6 or higher is recommended to view this site. This site is optimized for 1024 x 768-pixel screen size. JAVA script is used.

Please e-mail us your comments or suggestions to the following address:
info-stc@renesas.com



M16C Overview

- Product Specifications
- Block Diagram
- M16C/62 Group General Specifications
- Pin Configuration

Block Diagram (M16C/62 Series)

[Glossary] Move the mouse pointer over a term to view the description.

Copyright(C) 2003 RENESAS TECHNOLOGY Corp. All right reserved

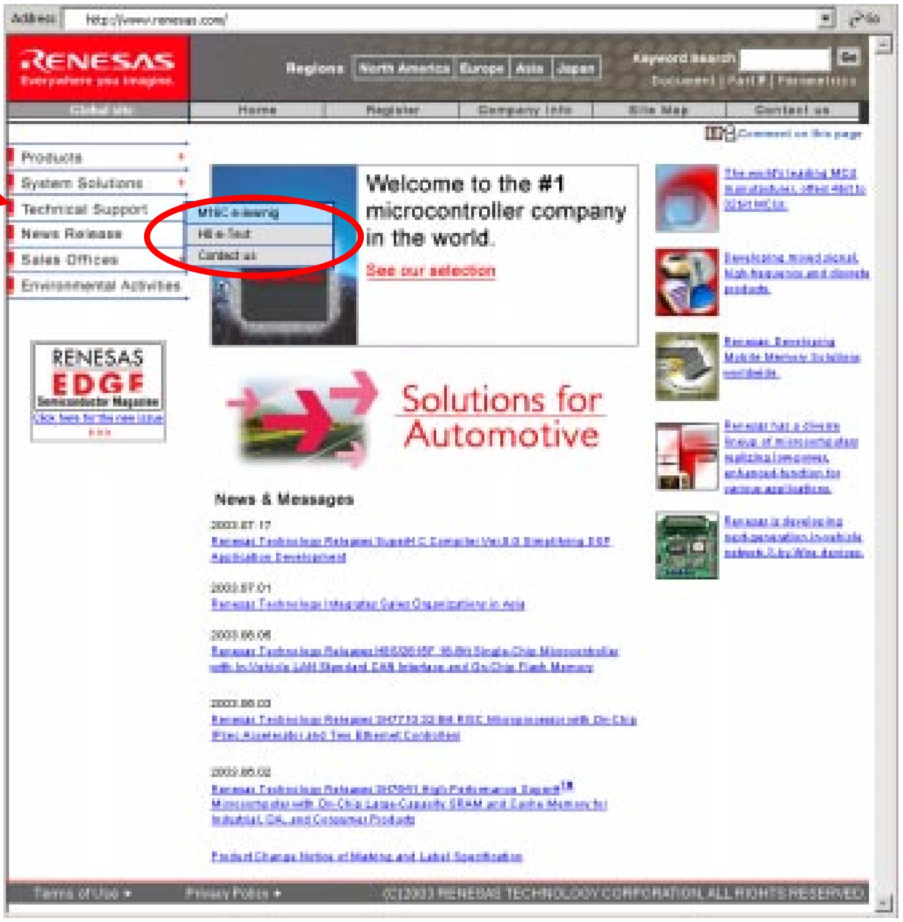
Introducing Renesas eText

Access to the Quick Learning portal via:

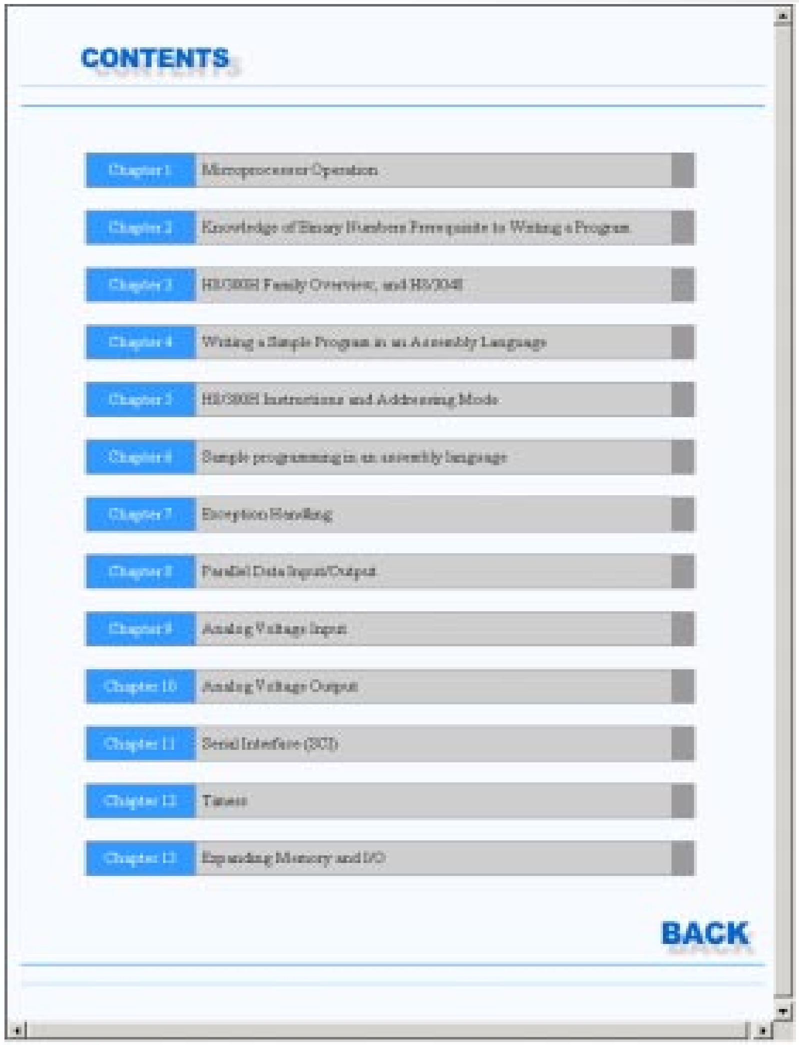
http://www.renesas.com

Follow the „Technical Support“ entry at the left menu.

Chose H8 e-Text from the popup menu.

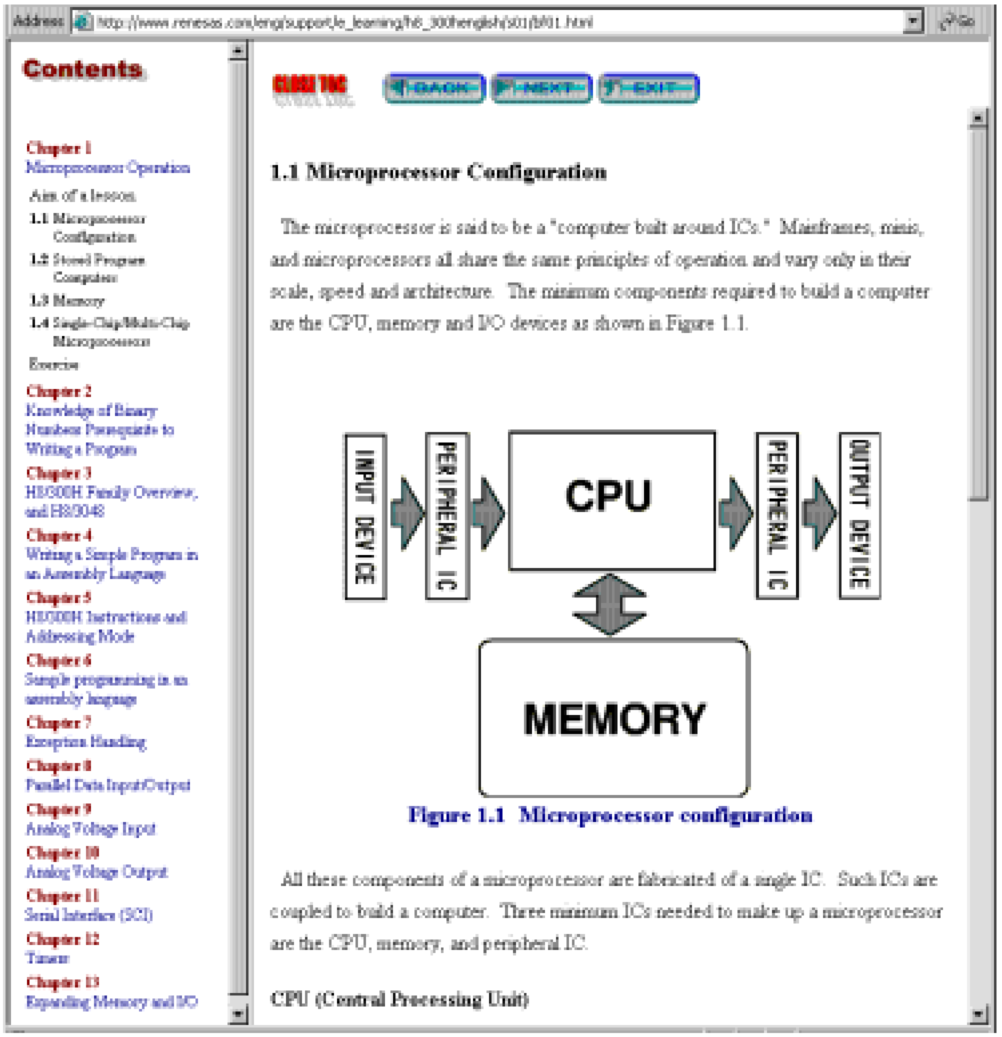


Renesas eText



CONTENTS

BACK



Address http://www.renesas.com/eng/support/e_learning/h8_300henglish/s01/b901.html

Contents

1.1 Microprocessor Configuration

The microprocessor is said to be a "computer built around ICs." Mainframes, minis, and microprocessors all share the same principles of operation and vary only in their scale, speed and architecture. The minimum components required to build a computer are the CPU, memory and I/O devices as shown in Figure 1.1.

Figure 1.1 Microprocessor configuration

All these components of a microprocessor are fabricated of a single IC. Such ICs are coupled to build a computer. Three minimum ICs needed to make up a microprocessor are the CPU, memory, and peripheral IC.

CPU (Central Processing Unit)

Introducing Renesas Interactive

Access to the Renesas Interactive portal via:

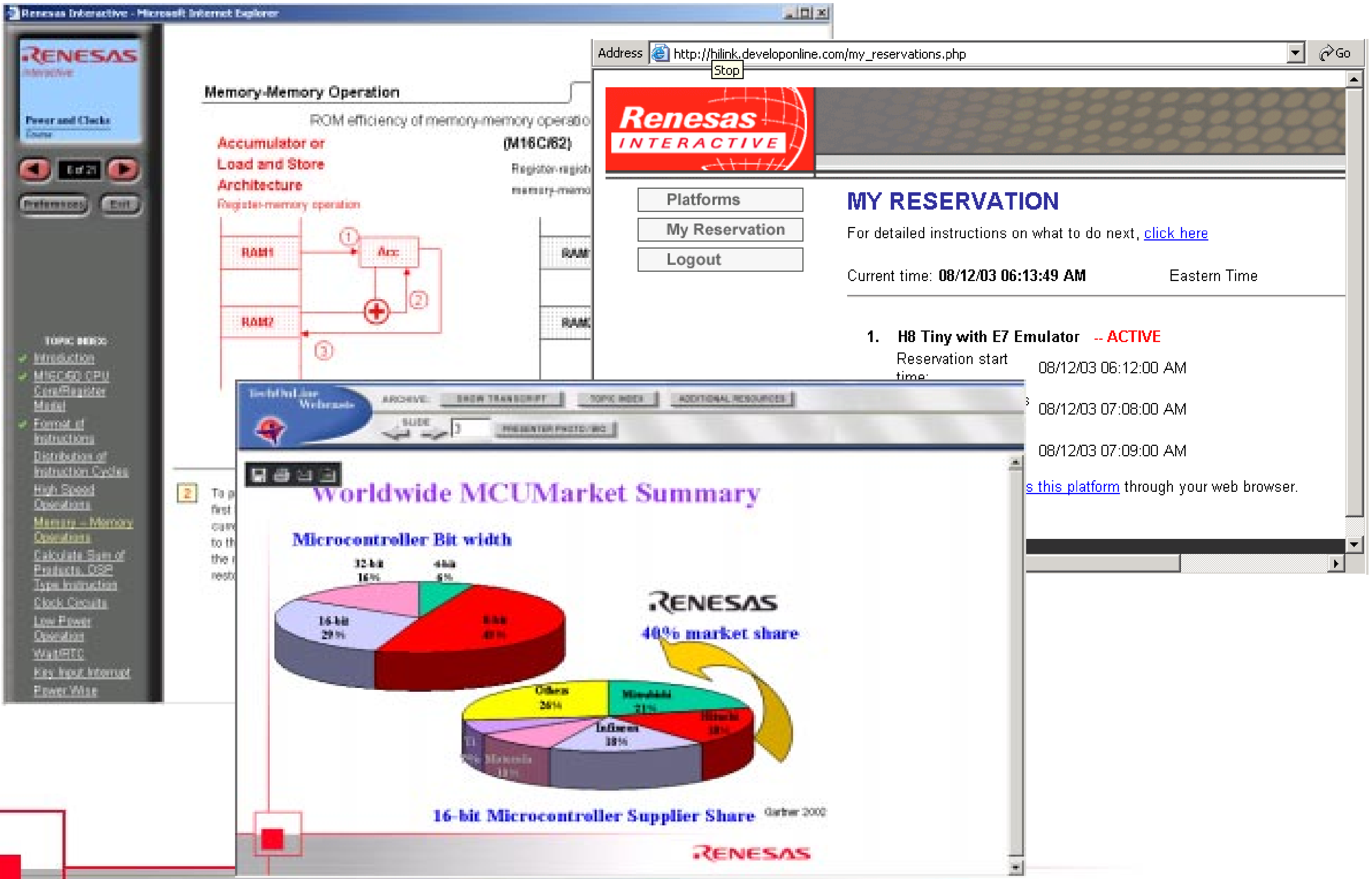
http://www.renesasinteractive.com

Renesas Interactive provides many ways in which you will be able to learn about and use Renesas microcontrollers and microprocessors:

- **Courses** provide easy access to self-paced, in-depth education on Renesas platforms and applications
- **Webcasts** are "live" or archived broadcasts conducted by Renesas subject matter experts
- **VirtuaLabs** allow you to evaluate actual Renesas platforms online

Please note, that a free of charge registration is necessary

Renesas Interactive



MY RESERVATION

For detailed instructions on what to do next, click here

Current time: **08/12/03 06:13:49 AM** Eastern Time

1. **H8 Tiny with E7 Emulator** -- ACTIVE

Reservation start time:	08/12/03 06:12:00 AM
	08/12/03 07:08:00 AM
	08/12/03 07:09:00 AM

s this platform through your web browser.

References

- sevenstax
  http://www.sevenstax.com

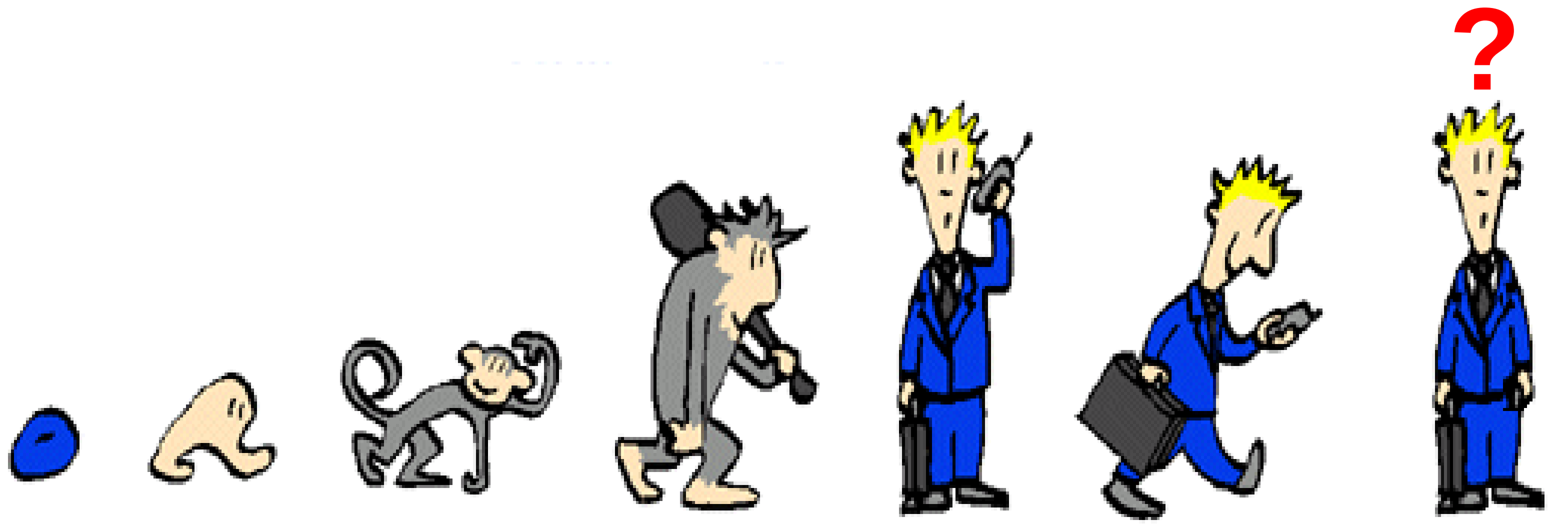
- LULEÅ TEKNISKA UNIVERSITET
  Prof. Jerker Delsing
  Lulea University of Technology.

  http://www.luth.se

- RENESAS Everywhere you imagine.

  http://www.renesas.com

Any Question?



RENESAS

Renesas Technology Europe