EMBEDDED SYSTEMS

BASED ON CORTEX-M4 AND THE RENESAS SYNERGY PLATFORM

2020
PROF. DOUGLAS RENAUX, PHD
PROF. ROBSON LINHARES, DR.
UTFPR / ESYSTECH

RENESAS ELECTRONICS CORPORATION

BIG IDEAS FOR EVERY SPACE RENESAS

1 – INTRODUCTION

1. What are Embedded Systems
2. Characteristics
3. Market Segments
4. The IoT Era

WHAT ARE EMBEDDED SYSTEMS?

An Embedded System (a.k.a. Embedded Computing System) is a computing system that is built-into (i.e. embedded) a larger device, such as an equipment, a system, or a vehicle.

Embedded Systems (ES) are usually application-specific and have real-time constraints; thus, many ES are also real-time systems. Often, ES are used in control loops: reading sensors, processing data, and generating outputs that control the device they are embedded into. Finite State Machines are commonly used to model the behavior of ES.

CHARACTERISTICS OF EMBEDDED SYSTEMS (1/4)

Typical characteristics of an Embedded System are:

1. **Microcontroller based system** consisting of a processor,
   non-volatile memory (Flash), volatile memory (RAM), and a large number of inputs and output interfaces as well as communication channels.
2. **Cost effective implementations** as many device architectures are cost-driven.
3. **Energy efficient solutions** as many devices are battery powered. Current trend is to develop battery-less devices that harvest energy from the environment.

CHARACTERISTICS OF EMBEDDED SYSTEMS (2/4)

4. **Heterogeneous.** While desktop computers are based on standard platforms (Windows PC, Apple IOS, ...) there is a large variety of hardware and software for embedded systems.
5. **Variety of restrictions** to the design solutions, such as:
   a) Physical: Size, Weight, Temperature Range, Vibration, Dust, Spills, Water;
   b) Computational resources: processing speed, non-volatile memory, RAM, available I/O;
   c) Response time.
6. **Interconnected.** Embedded devices and systems are ever more interconnected to each other. Trend is to increase the interconnection rate (IoT).

CHARACTERISTICS OF EMBEDDED SYSTEMS (3/4)

7. **Reliability**

   The ability of a system or component to function under stated conditions for a specified period of time.

8. **Availability**

   The ability of a system or component to function at a specific moment of interval of time.

9. **Maintainability**

   Measures how easily and how fast a system can be restored to operational status after a failure.

10. **Testability**

    The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met.

CHARACTERISTICS OF EMBEDDED SYSTEMS (4/4)

11. **Scalability**

    The ability of a system to handle increased workload by repeatedly and cost-effectively adding components to extend the system´s capacity.

12. **Safety**

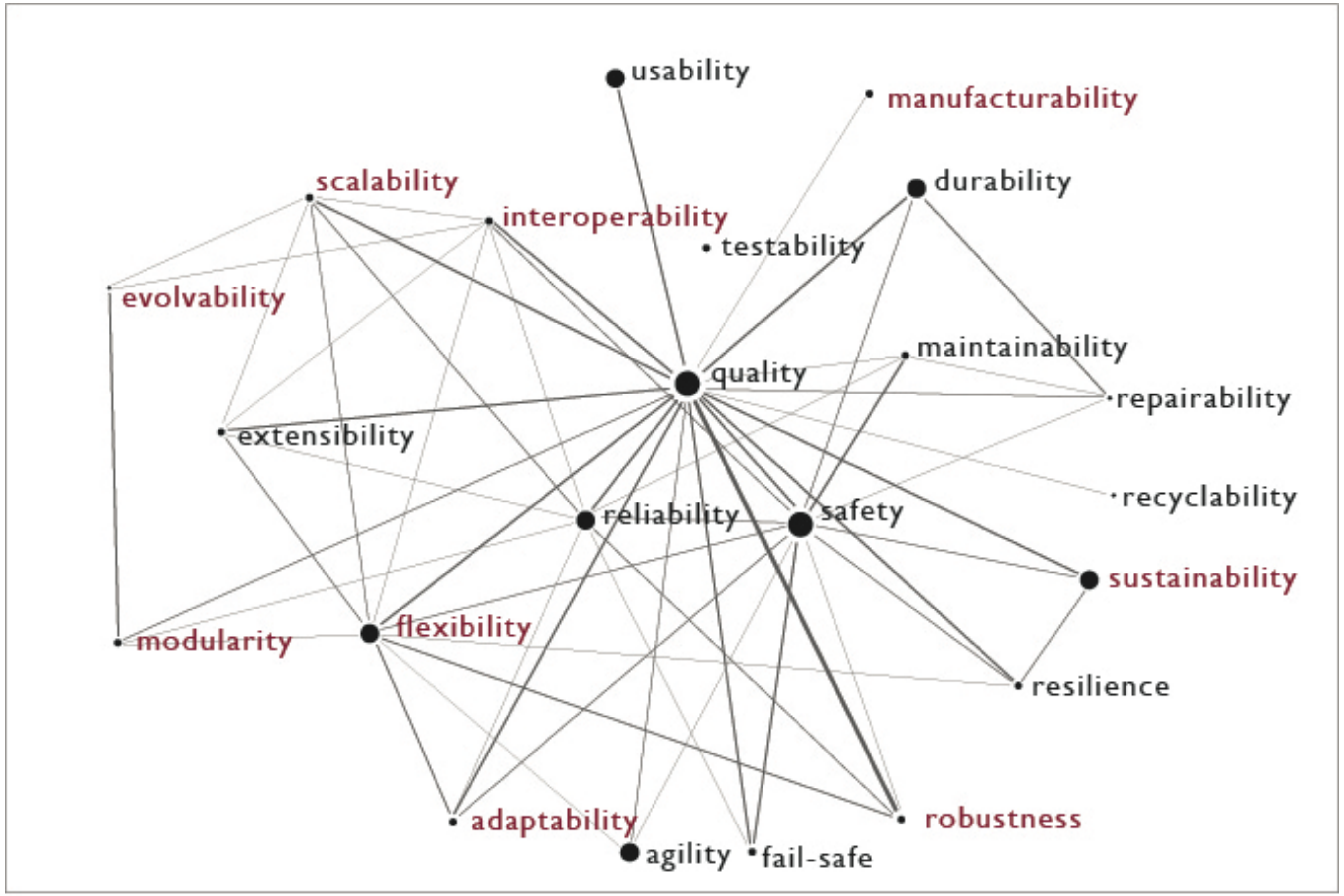
    Concerns the requirement: not to harm people, the environment or other assets.

13. **Security**

    The ability of a system to protect information and system resources with respect to integrity and confidentiality.

CHARACTERISTICS OF EMBEDDED SYSTEMS

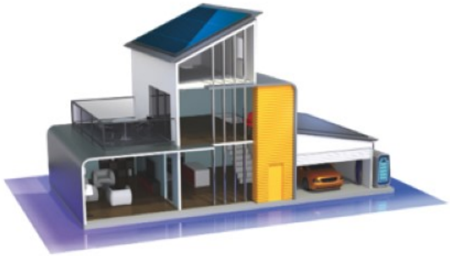
Relationship among 22 of the most common ilities:



to read about this graph: Book Chapter about the Ilities: Chapter 4 from "Engineering Systems: Meeting Human Needs in a Complex Technological World" by de Weck O., Roos D. and Magee C, MIT Press, January 2012 (http://strategic.mit.edu)

SAMPLE MARKET SEGMENTS

- Consumer Electronics
- Telecommunications
- Home Automation
- Industrial Automation
- Transportation
  - Avionics
  - Navigation
  - Electric Vehicles
- Defense
- Medical Equipment
- ??? (many new areas to come)


Smart House

Smart Factory


Drone


Connected Care


Cool Gadget


Robots

source: Renesas DevCon2015

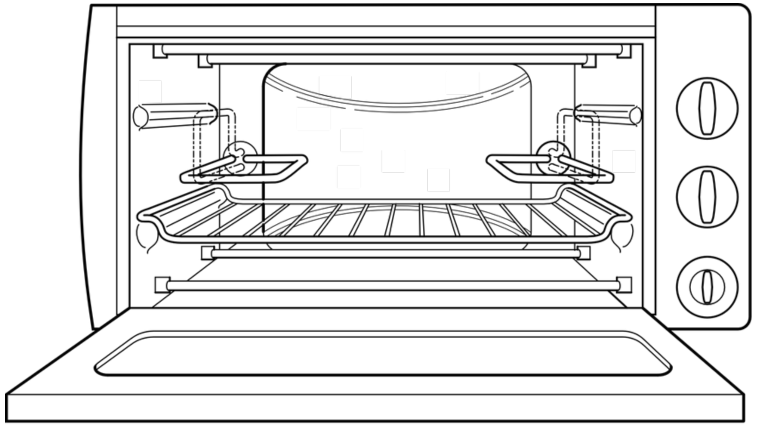
CONSUMER ELECTRONICS

- Phones
- Videogame consoles
- Printers
- Digital cameras
- Audio/Video:
  - Television
  - Music Players
  - Home Entertainment Systems
  - BRD players

source: pixabay.com (CC)

HOUSEHOLD APPLIANCES

- Washing Machines
- Dishwasher
- Air Conditioners
- Microwave Oven



source: pixabay.com (CC)

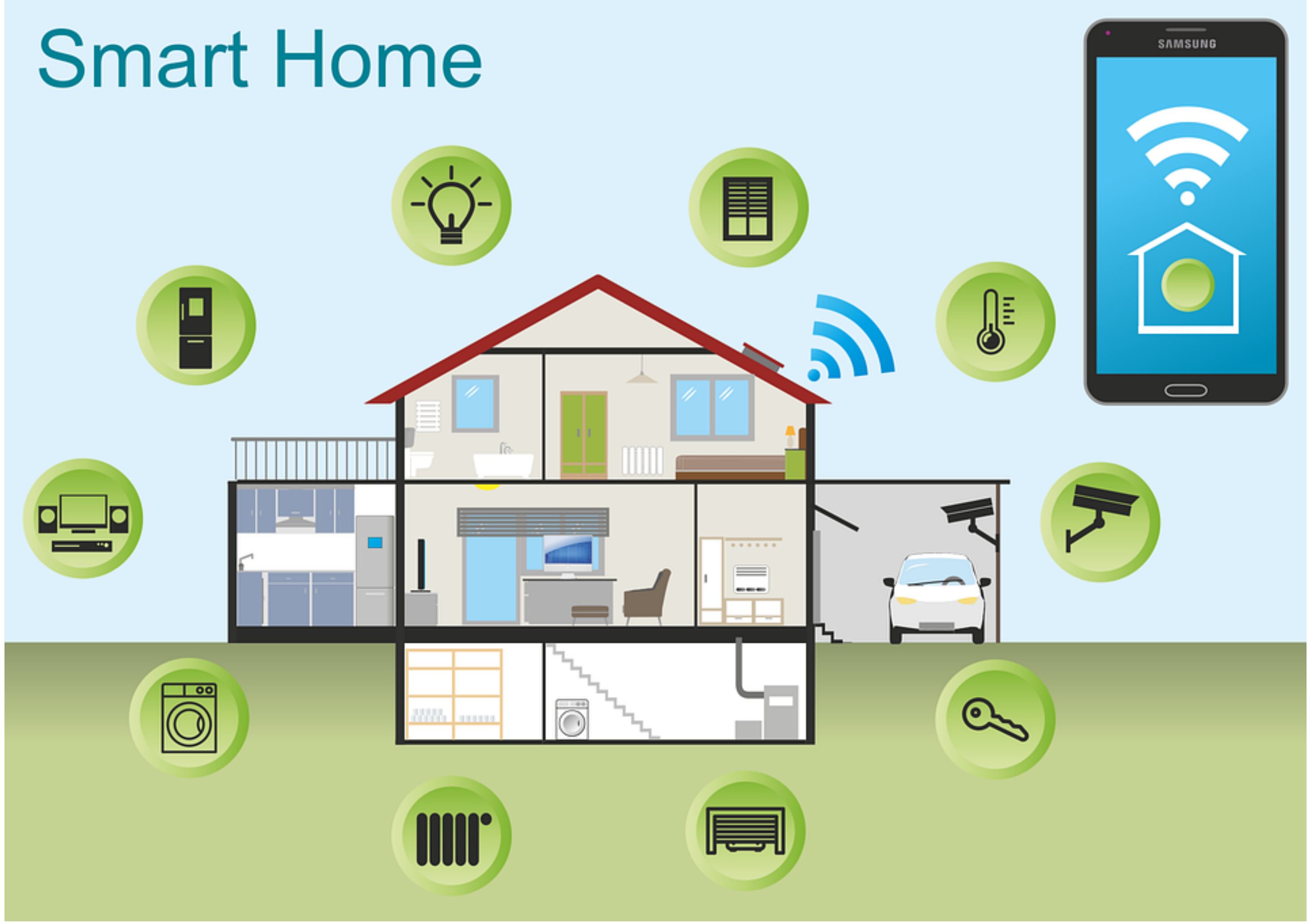
TELECOMMUNICATIONS

- Routers
- Satellite Phones
- Switches

source: wikimedia.org (CC)

source: pixabay.com (CC)

HOME AUTOMATION

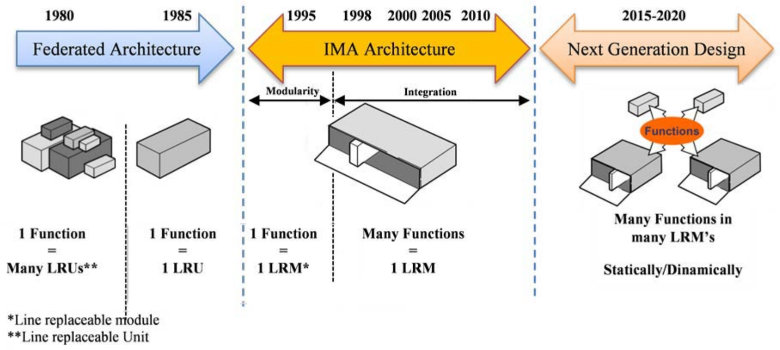


source: pixabay.com (CC)

TRANSPORTATION – AVIONICS

Glass cockpit of the

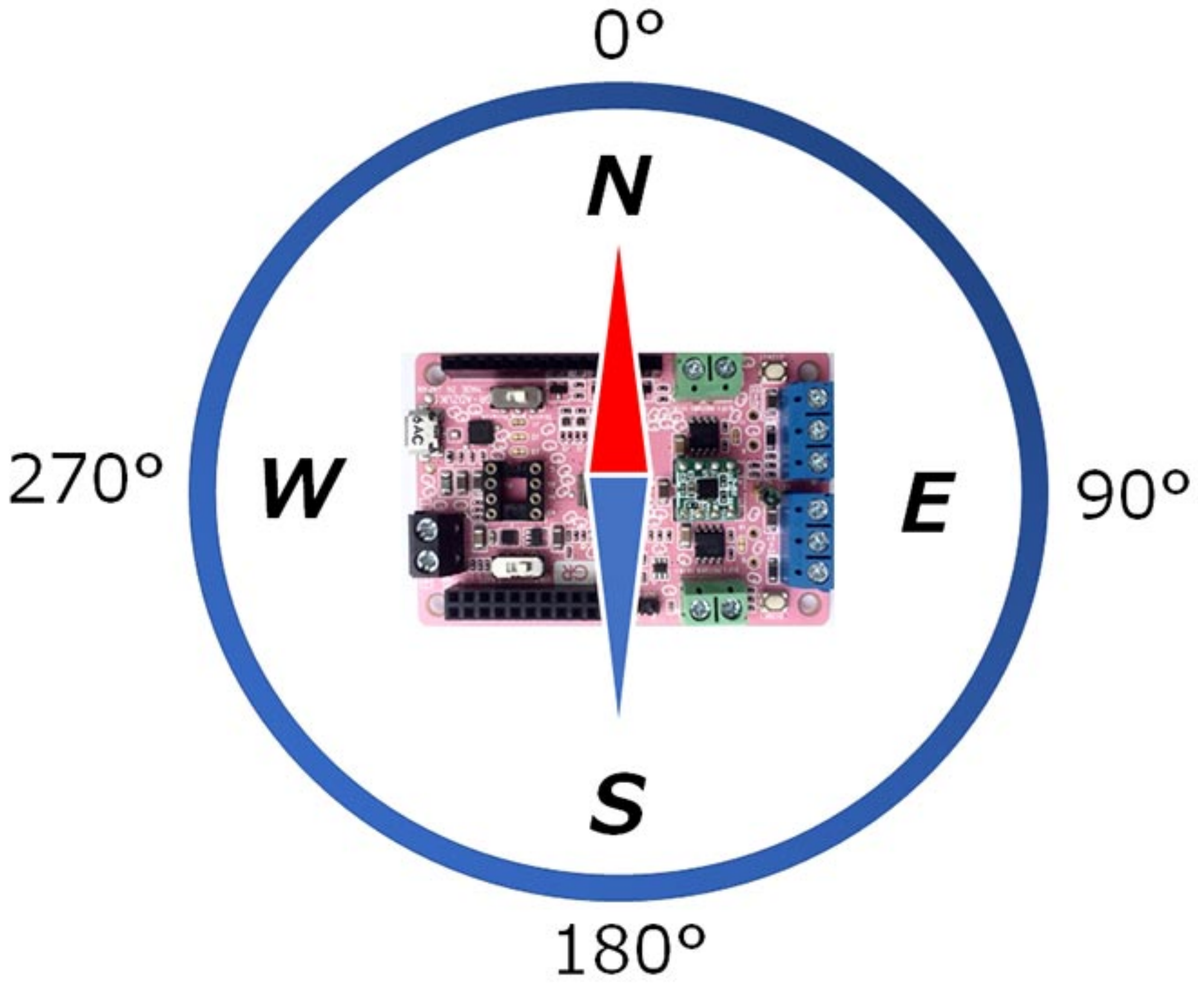
Airbus A350 XWB



source: flickr (CC)

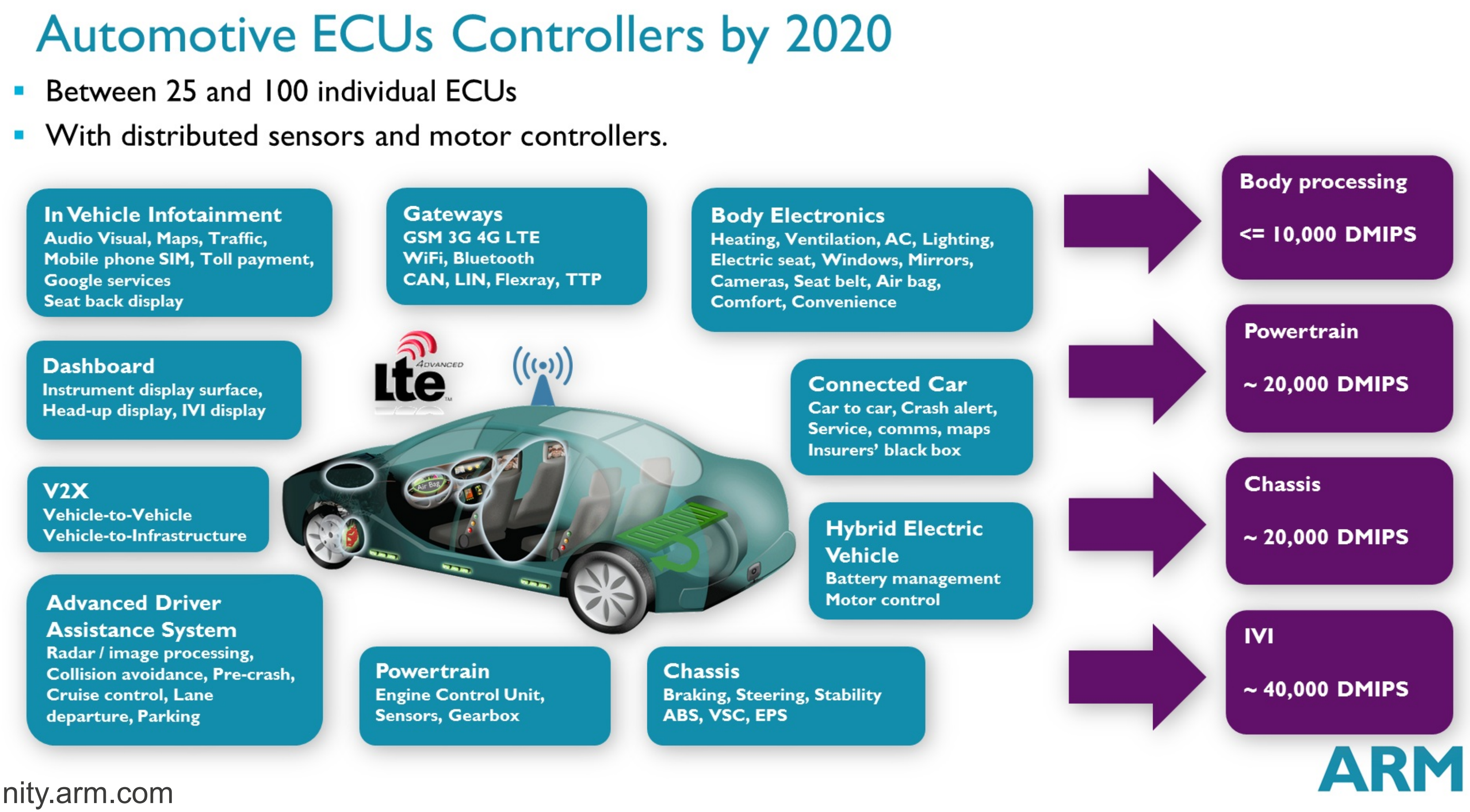
TRANSPORTATION – NAVIGATION

- Automotive GPS
- Electronic Compass



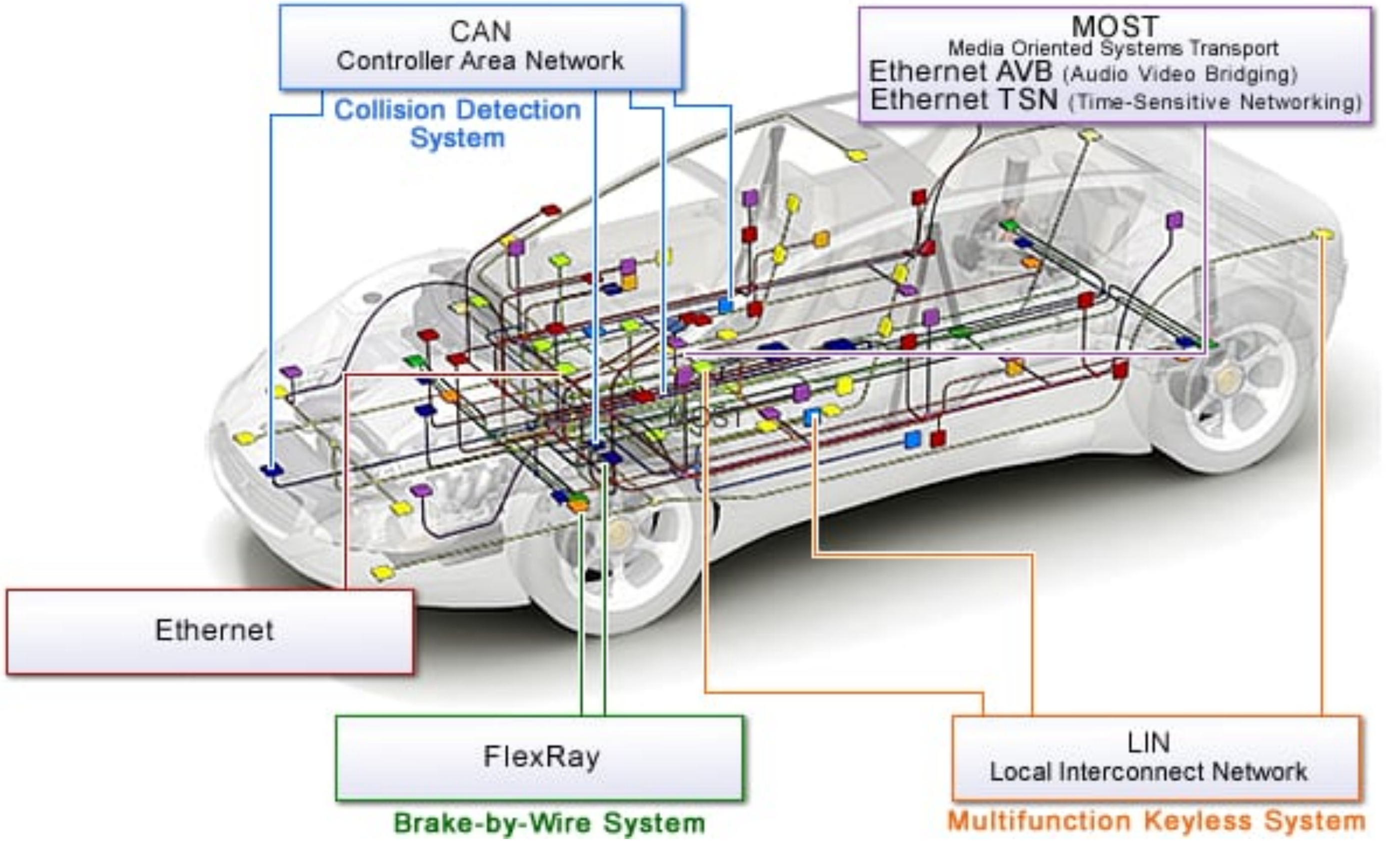
source: Renesas

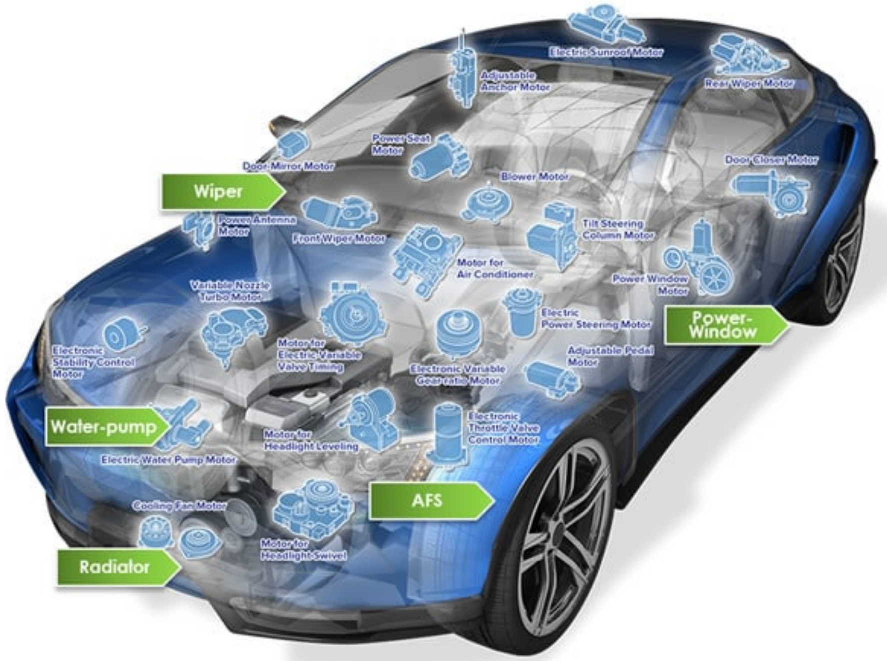
TRANSPORTATION – AUTOMOTIVE



source: community.arm.com

TRANSPORTATION – AUTOMOTIVE





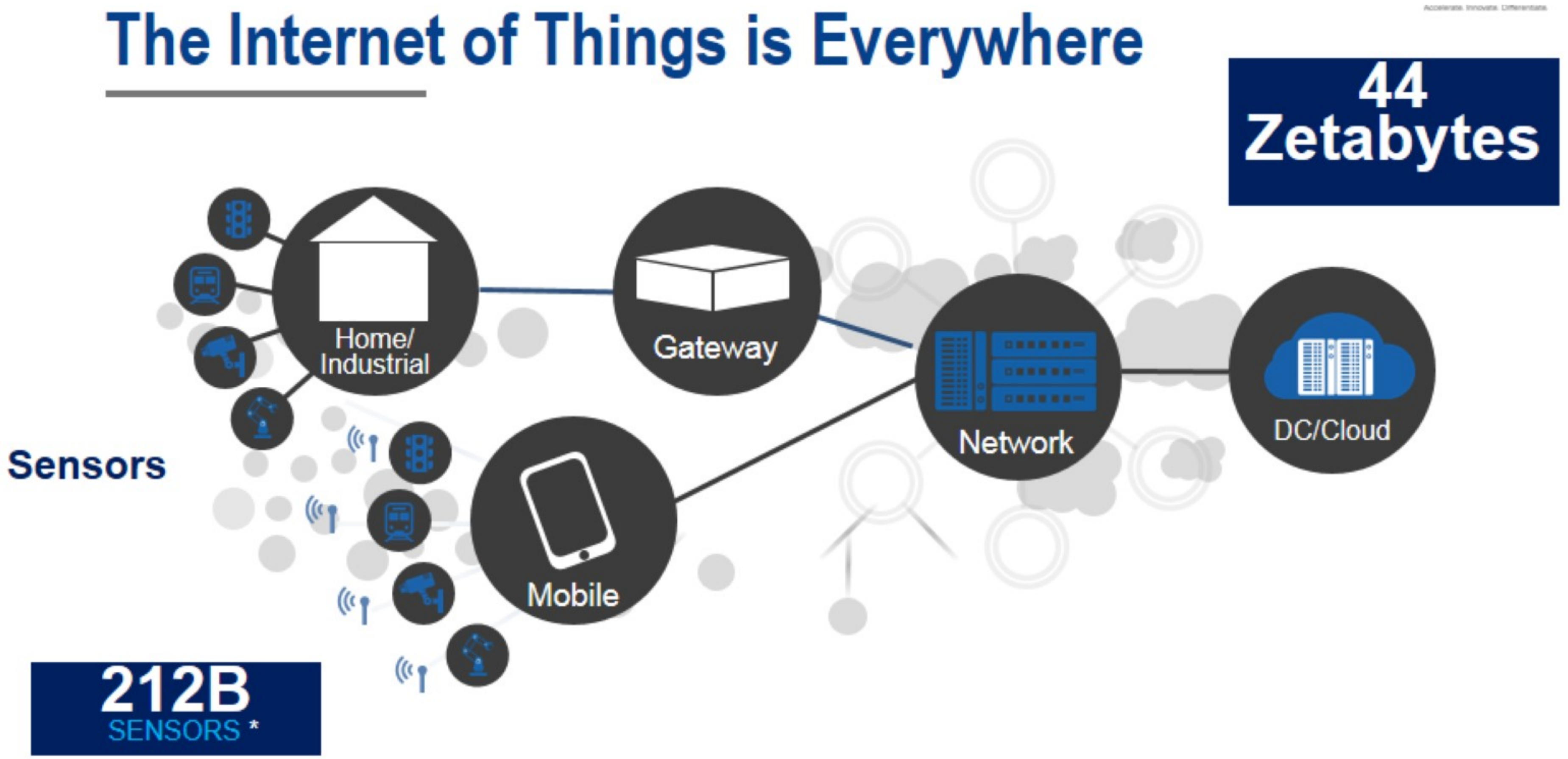
source: Renesas

MEDICAL EQUIPMENT

- CT Scanners
- ECG (Electrocardiogram)
- Blood Glucose Monitor
- Blood Pressure Monitor
- Body Composition Analyzer

source: Renesas

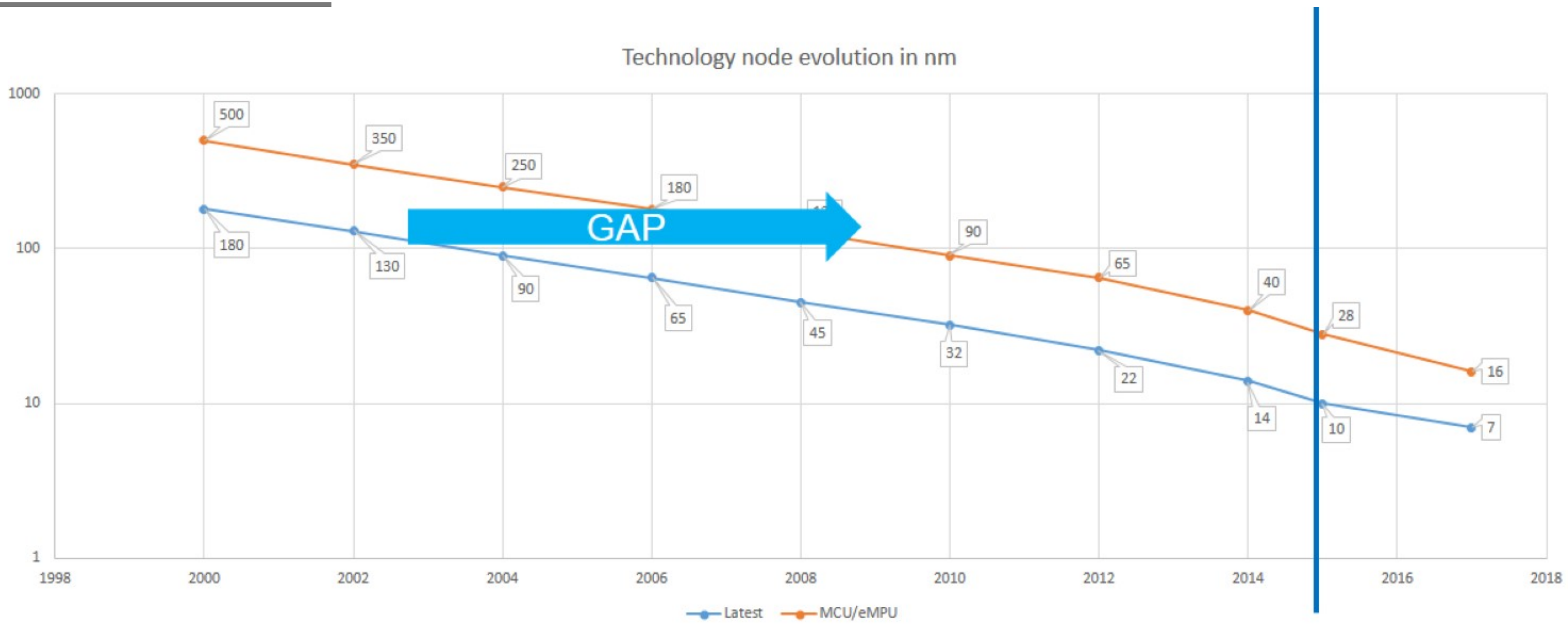
THE IOT ERA



source: Renesas

Five-year gap from state of the art to MCU technology

TRENDS



2015

Technology node = minimum transistor gate length

Source: ITRS, Renesas Electronics – based on product announcement

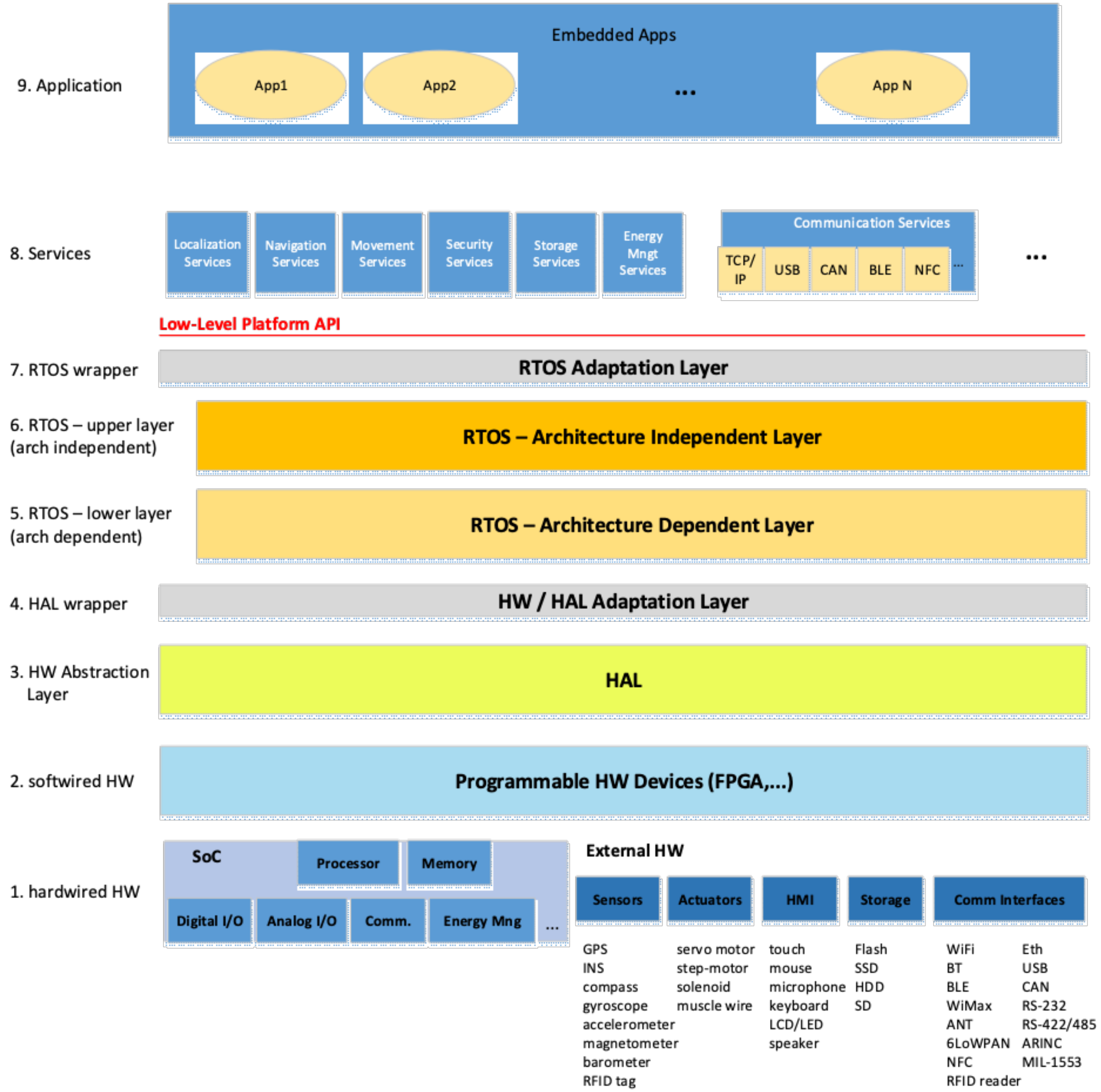
source: Renesas

EMBEDDED SYSTEMS ARCHITECTURE – GENERIC MODEL

- One of the important characteristics of Embedded Systems (ES) is its diversity. Hence, a truly generic model for an ES does not exist.
- The model presented here attempts to represent a large set of the existing ES. Hence, it is adequate to understand Embedded Systems concepts.

suggested reading: (see IEEEXplore.ieee.org)
D. Renaux, F. Pottker, "Applicability of the CMSIS-RTOS Standard to the Internet of Things", *10th Workshop on Software Technologies for Future Embedded and Ubiquitous Systems - SEUS/ISORC 2014*, June 2014.

EMBEDDED SYSTEMS ARCHITECTURE GENERIC MODEL



source: Authors

1. HARDWIRED HW

- This level is composed of hardware devices (MCU, Memory, I/O, connectors) whose connection is determined by the copper traces on a PCB, hence, not changeable after fabrication.
- Currently, a significant portion of the HW functionality of an ES is integrated into a SoC (System on Chip). External HW consists of the remaining HW not in the SoC and comprises, among others, sensors, actuators, Human-Machine Interface devices, Storage devices, and a large variety of communication interfaces.

Skywire Renesas Synergy PMOD Kit

2. SOFT WIRED HW

- In contrast to Level 1, the soft wired HW level consists of components whose connection is programmable, hence, can be modified at any time.
- Currently, the most common programmable devices are FPGAs, however, a variety of programmable logic devices (PLD) are available: FPGA, CPLD, GAL, PAL, PLA, and even ROM.
- The two hardware layers (1 and 2) compose the physical part of an embedded system. The remaining layers (3 to 9) are software layers.

BIG IDEAS FOR EVERY SPACE

SOFTWARE LAYERS

The upper layers (3 to 9) are software layers.

Embedded Systems Software have two distinct characteristics:

1. Typically the development environment (compiler toolchain) generates a single binary file that integrates all software components: device drivers, libraries (RTOS, Services, ...) and the Application. Hence, avoiding the process of reading an executable file and loading it on memory.
2. While on desktops applications are changed and upgraded quite frequently, in embedded systems, typically a single multitasking application is executed along the life of the device. Upgrades may occur but are much less frequent.

3. HARDWARE ABSTRACTION LAYER

- The HAL is comprised of a set of functions that directly access the hardware devices. These functions are also called device drivers.
- A well-designed HAL provides to the upper levels a standardized interface, providing an easy interchange of devices. For instance, if all communication devices have the same API then replacing a comm. interface (such as SPI) for another (such as I2C) is straightforward. The Renesas SSP (Synergy Software Package) is an example of such.
- Developing a device driver for the HAL requires expertise in both hardware and software. Such a development is a complex and time-consuming task typically performed in C and sometimes mixing with assembly language.

4. ADAPTATION LAYER

- An Adaptation Layer (or Wrapper) is a means of providing a common interface for different device drivers.

- If a HAL is not carefully designed, or if device drivers from different vendors are integrated in the same solution, then the software interface to the upper level may not be regular, meaning that different devices have access functions with different signatures. Such a scenario poses severe difficulties for portability and changes.
- The adaptation layer is a simple translation layer aiming at converting the non-standard interface to a standardized one. Such a translation can often be done at compile time, hence, not imposing any runtime penalty.

5. AND 6. RTOS

- Embedded Operating Systems have significant differences to O.S. used in desktop computers, including a very small memory footprint and a reduced amount of functionality. Most embedded O.S. have to provide support for real-time systems, hence, they are termed RTOS (Real-Time Operating System).
- The implementation of a well designed RTOS has at least two layers, so that different software modules implement the architectural dependent code and the architectural independent code. This approach improves modularity and improves the portability to other architectures.

7. RTOS ADAPTATION LAYER

This is a wrapper that translates the API of a given RTOS to a standard API, such as CMSIS-RTOS. Once a standard API is provided to the upper levels, all of the software in the Services and Application layers can be reused in different platforms without rewriting the calls to the RTOS.

8. SERVICES

- To cope with the large amount of functionality implement by software in current Embedded Systems, code reuse is almost mandatory. Thus, off-the-shelf software components are integrated to form the final solution. Typically, these components come in the form of libraries that are linked at compile time.
- A large variety of software components is available providing functionality for: TCP/IP stacks, USB stacks, communication protocols, georeferencing, navigation, security, storage, among many others.

9. APPLICATION LAYER

- The top layer of the model is the Application Layer. This is the software layer that implements the specific functionality of each embedded system.
- In this layer, several concurrent tasks cooperate to provide the required functionality. Concurrent programming is the most common approach to cope with the software complexity of current Embedded Systems.
- The RTOS provides the management of the concurrency, among many other services.

Renesas.com