Embedded Systems

Master 's Degree in Informatics Engineering

Contents

Contents

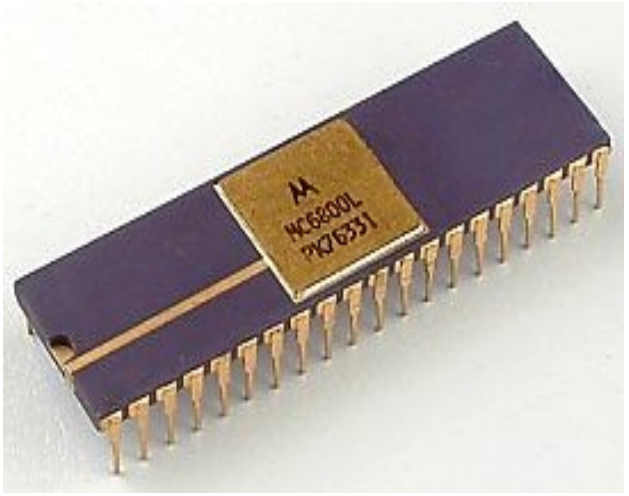
Embedded Systems

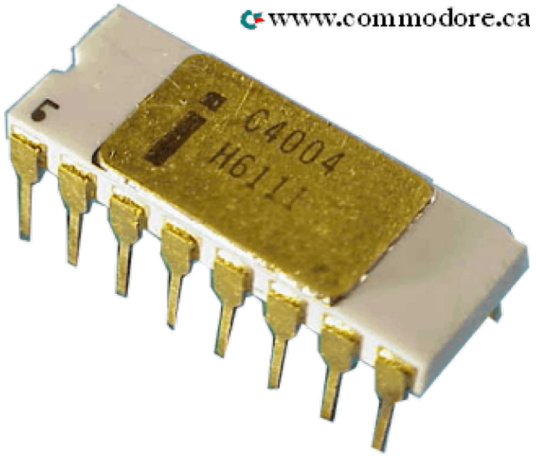
- An embedded system is a computer system designed for specific **control functions** within a larger system, often with real-time computing constraints.
- It is **embedded** as part of a complete device often including hardware and mechanical parts.
- Embedded systems contain processing cores that are typically either **microcontrollers** or **digital signal processors** (DSP)
- The embedded systems are dedicated to specific tasks.
  - Consumer electronics (PDA, videogame consoles, mpX players, ..,)
  - Transportation systems (avionics, inertial guidance systems, GPS, ….)
  - Medical equipment (electronic stethoscopes, heart monitoring, …)

- In the earliest years of computers in 1930 – 40s, computers were sometimes dedicated to a single purpose task.
- One of the first recognizably modern embedded system was the Apollo Guidance Computer, developed by Charles Stark Draper at the MIT Instrumentation Laboratory.
- Mass-produced embedded system - Autonetics D-17 missile guidance computer

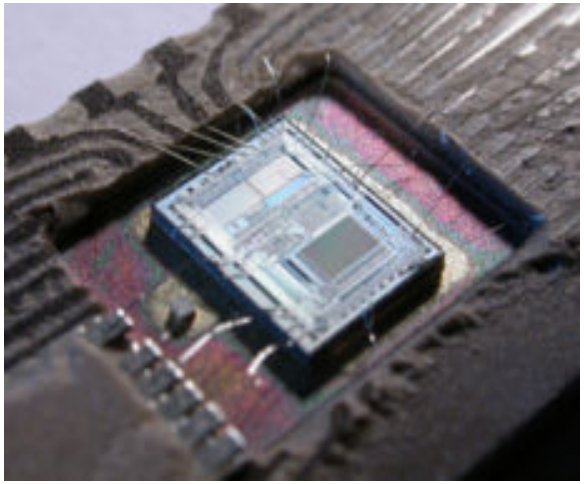
- Since these early applications in the 1960s, embedded systems have **come down in price** and there has been a dramatic rise in processing power and functionality.

  The first microprocessor for example, the Intel 4004 was designed for calculators and other small systems but still required many external memory and support chips.





- By the *mid-1980s*, most of the common previously external system components had been *integrated into the same chip as the processor* and this modern form of the microcontroller allowed an even more widespread use, which by the end of the decade were the norm rather than the exception for almost all electronics devices.



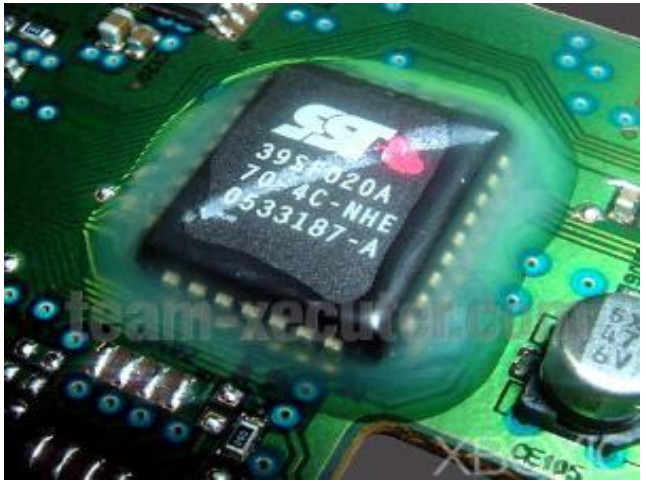
Contents

- Special-purpose
  - Typically, is designed to execute a single program, repeatedly
  - It used to be single-purpose
  - Now, multi-functioned, but single-purpose
  - Tactic and Strategy

- Tightly constrained
  - Low cost
  - Simple systems
  - Fewer components based
  - Performs functions fast enough
  - Minimum power

- Reactive and real-time
  - Reactive: Continually reacts to external events
  - Real-time: Must compute certain results in real-time

- Hardware and software coexist
  - The software written for embedded systems is often called firmware
  - Is stored in read-only memory or Flash memory chips rather than a disk drive



Contents

Differences from usual computer programs

- Several components of vastly different functionalities are found in embedded system software
- Response time constraint and strict deadlines
- All components must use the memory optimally

Differences from usual computer programs

- Each software component execution speed must be optimum
- Software must have controlled complexity and must be thoroughly tested and debugged for errors

Real-time programming: programming the processes or instruction set with constraints of time for its response, process with latencies, and process with deadlines.

- Procedure-oriented C and object-oriented programming C++ and Java languages are used in most embedded systems programming.
- Embedded programming is such that methods to optimize the system memory requirements are also used.

Real Time Operative System

- An **RTOS** is an OS for response time-controlled and event-controlled processes. It is very essential for large scale embedded systems.

Real Time Operative System

- Function
  1. Basic OS function
  2. RTOS main functions
  3. Time Management
  4. Predictability
  5. Priorities Management
  6. IPC Synchronization
  7. Time slicing
  8. Hard and soft real-time operability

is RTOS necessary?

- Software for a large number of small-scale embedded system use no RTOS and these functions are incorporated into the application software.
- For small-scaled systems, RTOS's function can be replaced by C functions.

  For example, instead of the memory allocation and de-allocation functions of RTOS, the C function , malloc and free can be used.
- Software can directly handle inter-process communication

When is RTOS necessary?

However, RTOS is essential when…

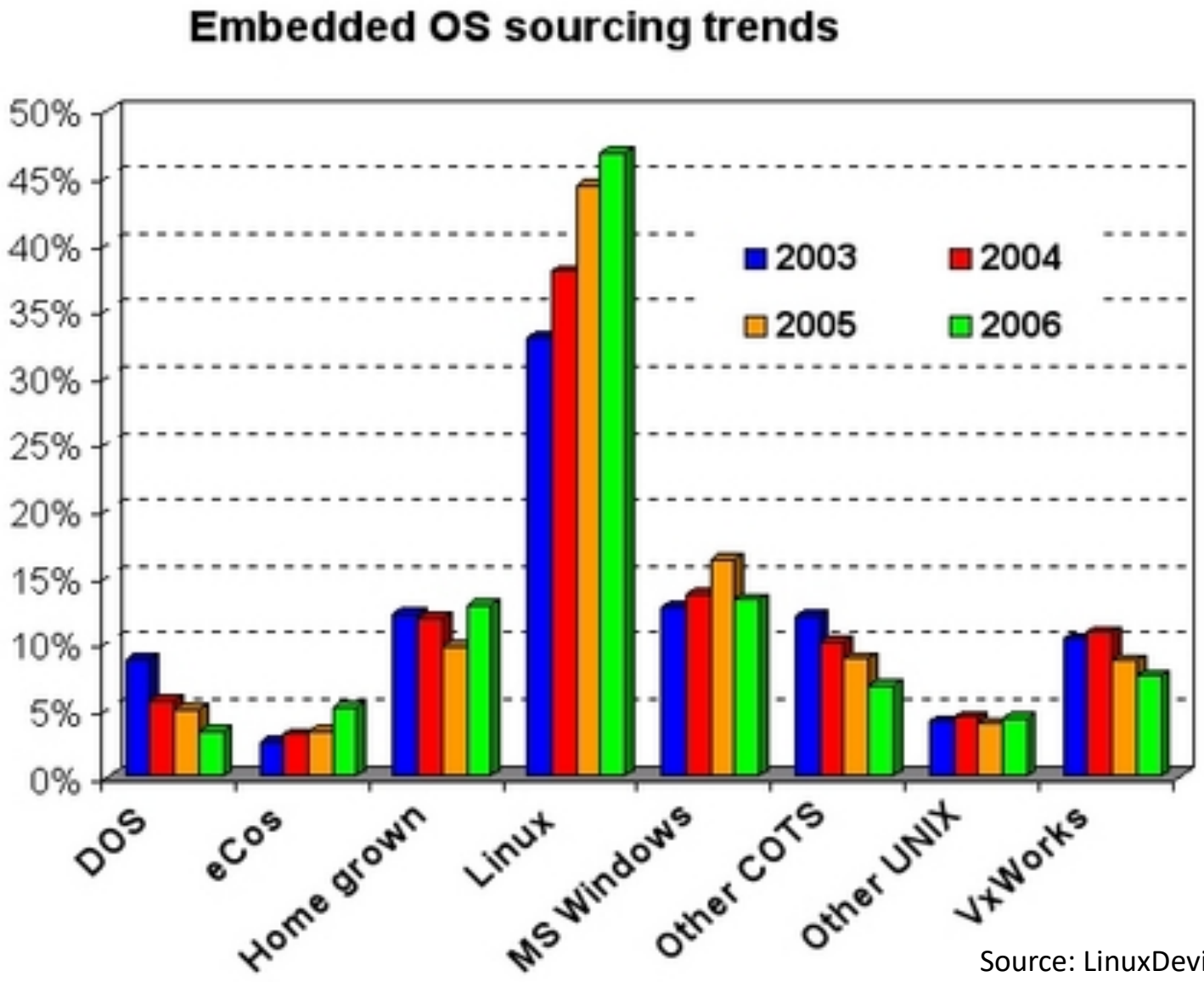
- A common and effective way of handling of the hardware source calls from the interrupts
- I/O management with devices, files, mailboxes becomes simple using an RTOS
- Effectively scheduling and running and blocking of the tasks in cases of many tasks
- …..

In conclusion, an RTOS may not be necessary in a small-scaled embedded system. An RTOS is necessary **when scheduling of multiple processes and devices is important**.

CPU Platforms for embedded systems

- Different from desktop computer

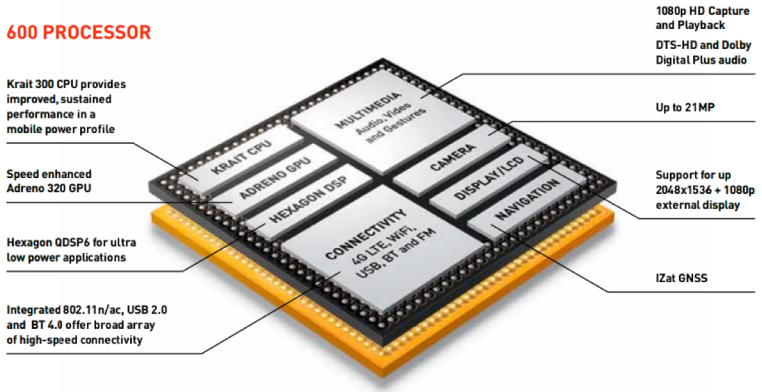
- CPU Architectures: 65816, 65C02, 68HC08, 68HC11, 68k, 8051, ARM, AVR, Blackfin, C167, Coldfire, COP8, eZ8, eZ80, FR-V, H8, HT48, M16C, M32C, MIPS, MSP430, PIC, PowerPC, R8C, SHARC, ST6, SuperH, TLCS-47, TLCS-870, TLCS-900, Tricore, V850, x86, XE8000, Z80, etc.



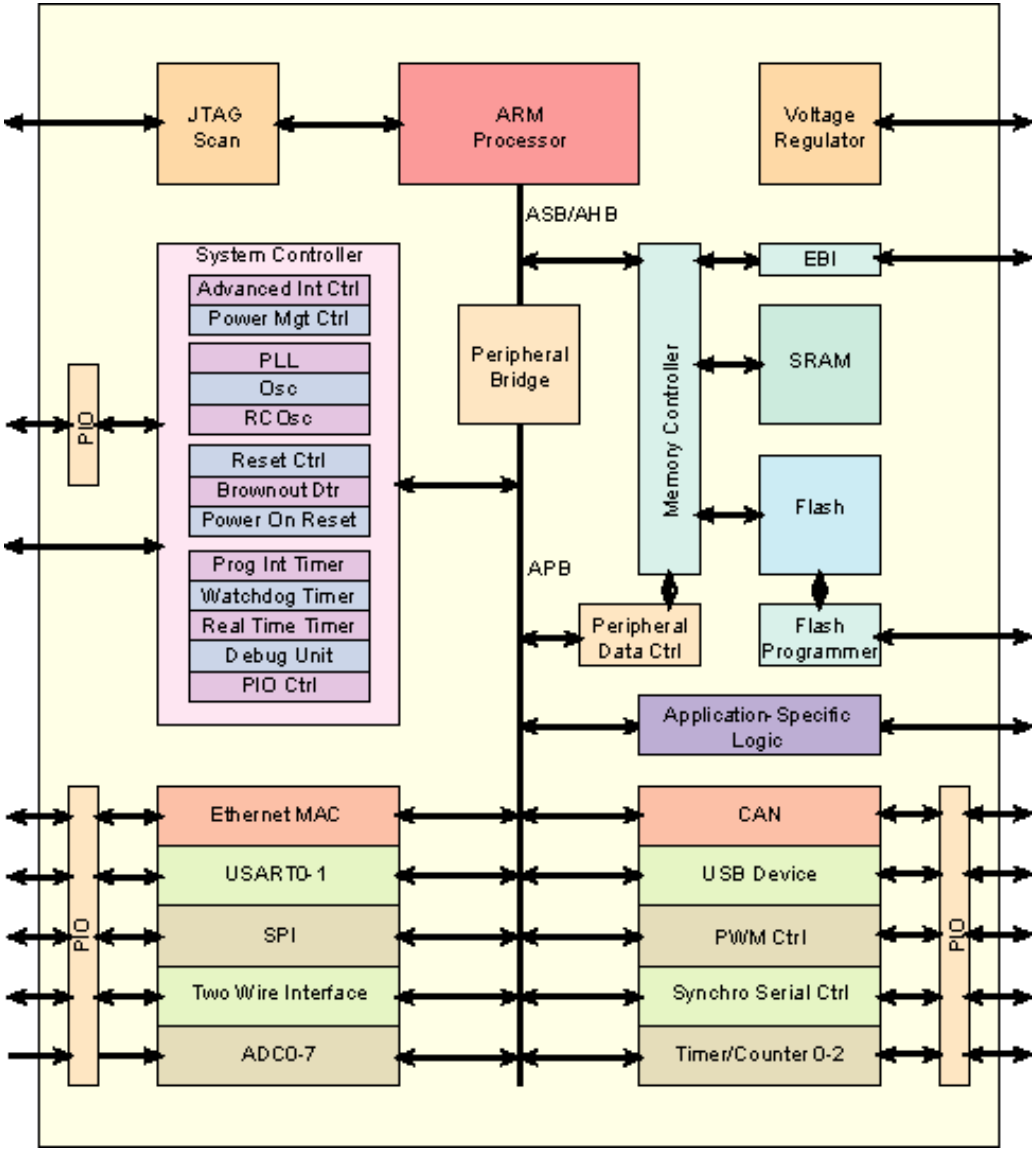
Source: LinuxDevices.com

System on Chip (SoC)

- Integrating all components of a computer or other electronic system into a single integrated circuit (chip).
- It may contain digital, analog, mixed-signal, and often radio-frequency functions – all on one chip.
- A typical application is in the area of embedded systems.
- SiP (System in Package)

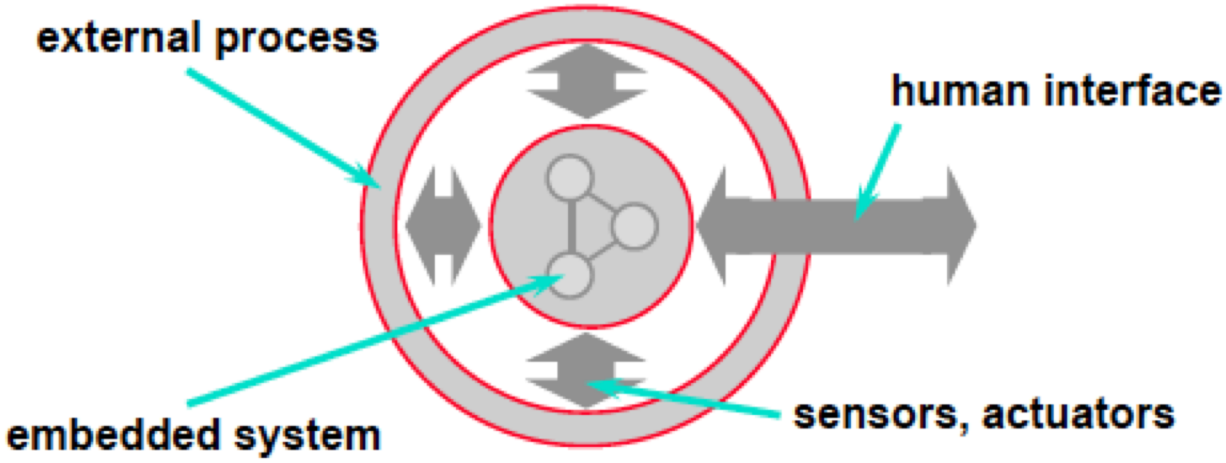


Qualcom SnapDragon 6000

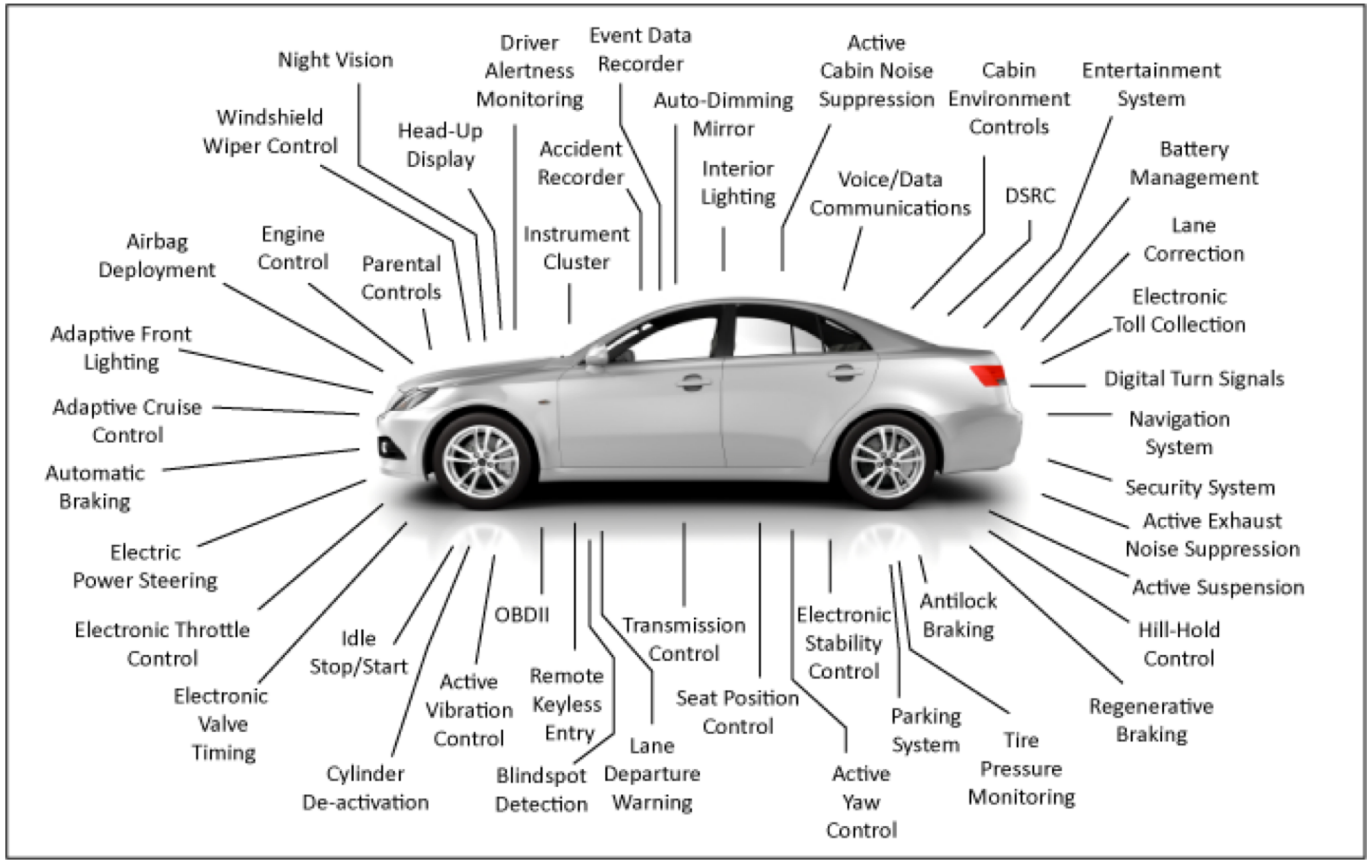

JTAG Scan
ARM Processor
Voltage Regulator
ASB/AHB
System Controller
Advanced Int Ctrl
Power Mgt Ctrl
PLL
Osc
RC Osc
Reset Ctrl
Brownout Dtr
Power On Reset
Prog Int Timer
Watchdog Timer
Real Time Timer
Debug Unit
PIO Ctrl
PIO
Peripheral Bridge
Memory Controller
EBI
SRAM
Flash
APB
Peripheral Data Ctrl
Flash Programmer
Application-Specific Logic
Ethernet MAC
USART0-1
SPI
Two Wire Interface
ADC0-7
CAN
USB Device
PWM Ctrl
Synchro Serial Ctrl
Timer/Counter 0-2
PIO
PIO

Contents

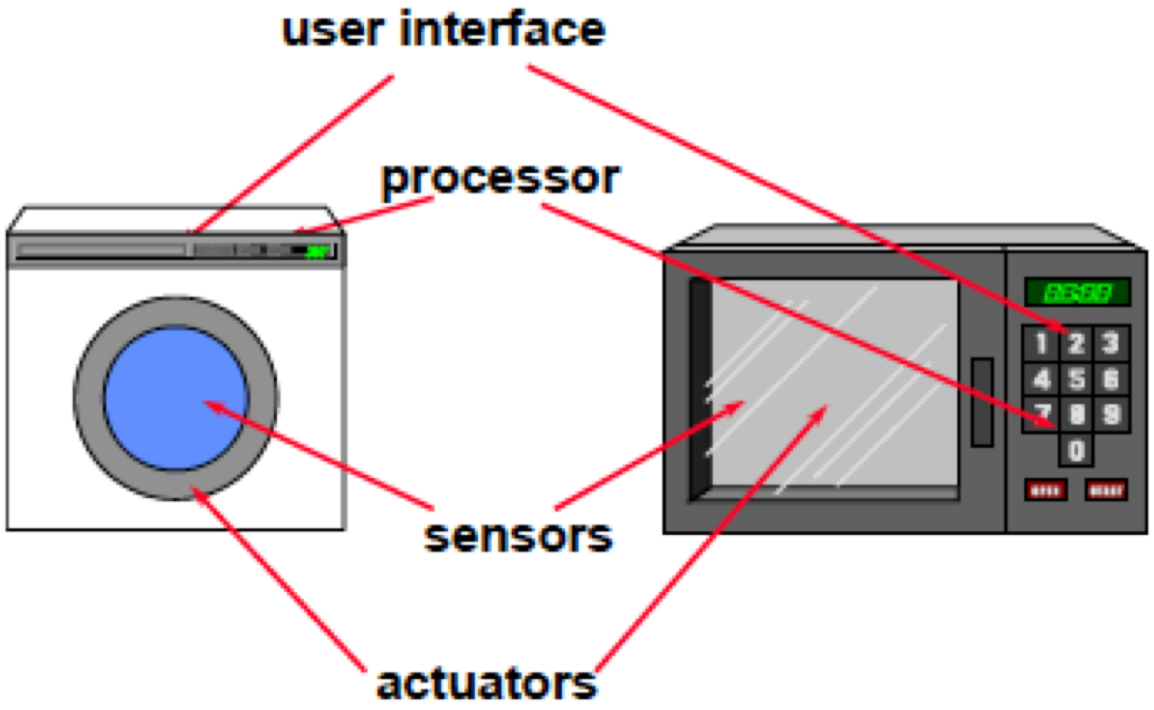
Input & Output is one of the basic principles in computer science – data is given to a processing unit, which processes the data end gives out the results. Concerning embedded systems it is basically the same. Data is measured by sensors, passed to the processing unit and then given out to the actuators.



Example: Car as an integrated control; communication and information system.



Consumer electronics, for example MP3
Audio, digital camera, home electronics, … .



Transducer: A transducer is a device that **converts one form of energy to another**. Energy types include (but are not limited to) electrical, mechanical, electromagnetic (including light), chemical, acoustic or thermal energy. While the term transducer commonly implies the use of a sensor/detector, any device which converts energy can be considered a transducer.

Sensors: is a converter that **measures a physical quantity and converts it into a signal** which can be read by an observer or by an (today mostly electronic) instrument

Actuator: An actuator is a type of motor for **moving or controlling a mechanism** or system. It is operated by a source of energy, usually in the form of an electric current, hydraulic fluid pressure or pneumatic pressure, and converts that energy into some kind of motion

Sensors

- A sensor is a device which **receives and responds to a signal**. A sensor's sensitivity indicates how much the sensor's output changes when the measured quantity changes

- Sensors need to be designed to have a **small effect on what is measured**

- A good sensor obeys the following **rules**:
  - Is sensitive to the measured property only
  - Is insensitive to any other property likely to be encountered in its application
  - Does not influence the measured property

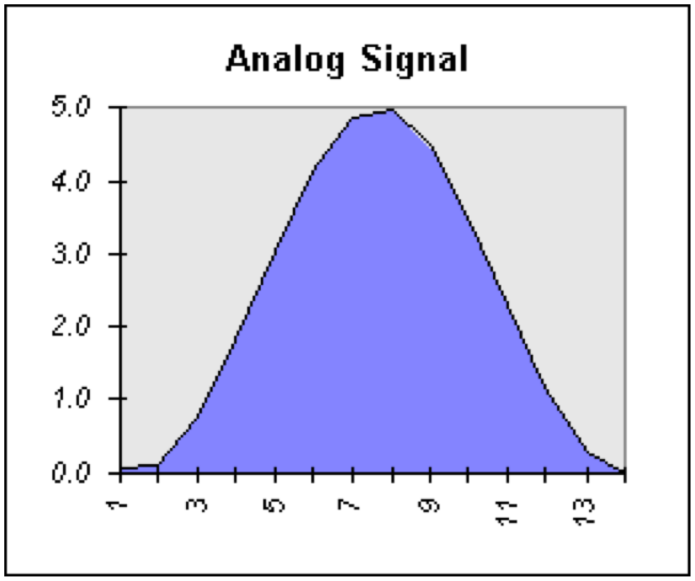
Types of Sensors

1. Acoustic, sound, vibration
2. Automotive, transportation
3. Chemical
4. Electric current, electric potential, magnetic, radio
5. Environment, weather, moisture, humidity
6. Flow, fluid velocity
7. Ionizing radiation, subatomic particles
8. Navigation instruments
9. Position, angle, displacement, distance, speed, acceleration
10. Optical, light, imaging, photon
11. Pressure
12. Force, density, level
13. Thermal, heat, temperature
14. Proximity, presence

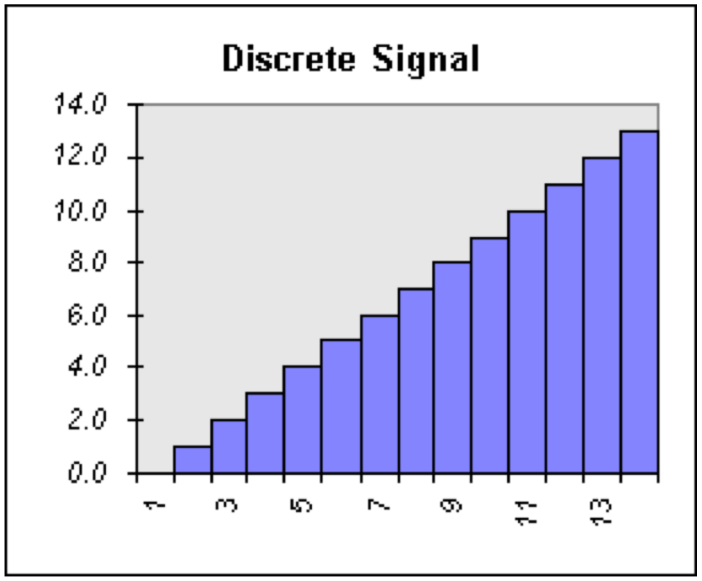
Wikipedia Source

Analog vs Discrete signals

- Environmental signals use to be analogic – light, temperature, …
- Embedded systems process discrete signals
- It is necessary a conversion between them – DSP Processors

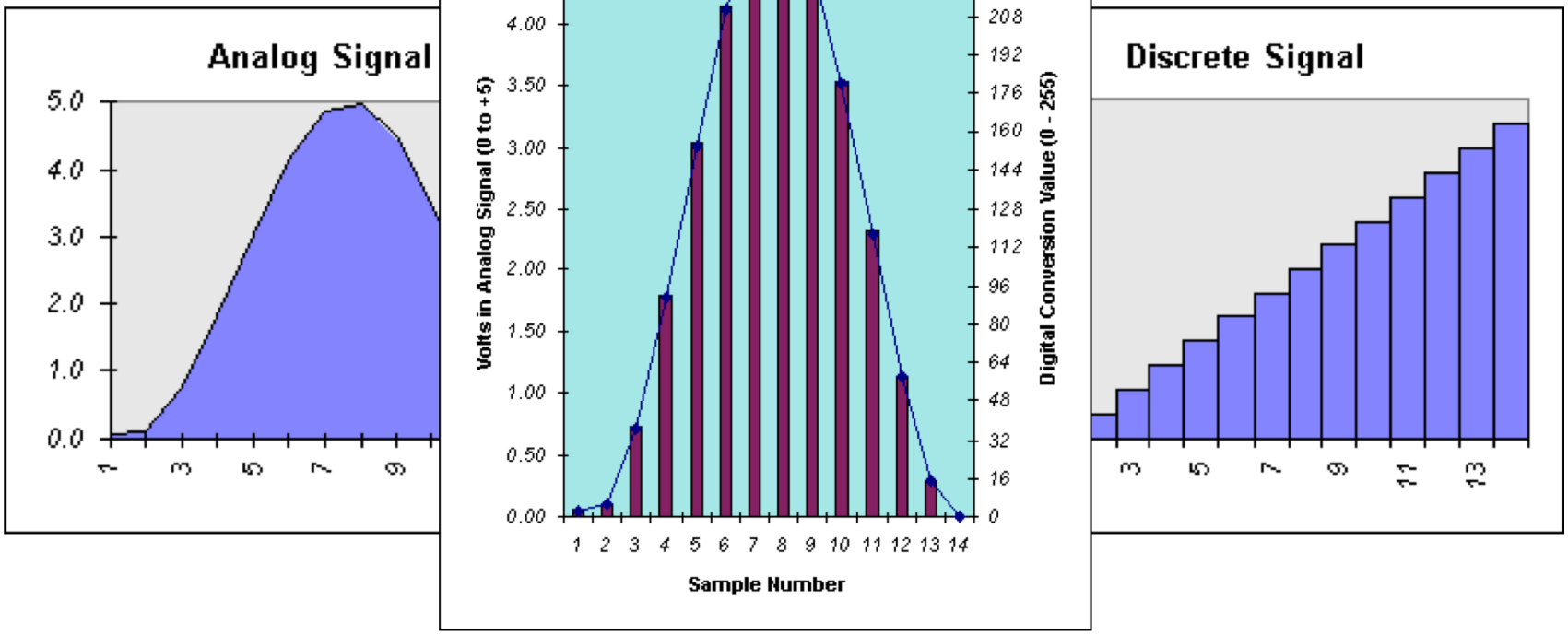


Conversion

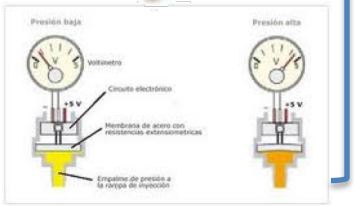


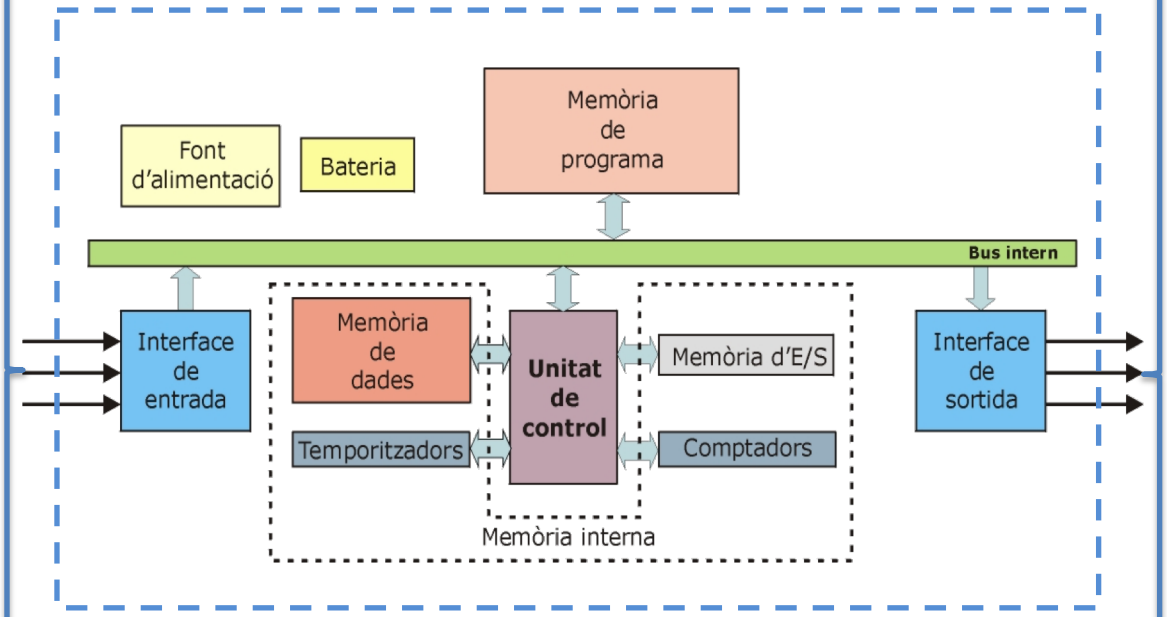
Analog vs Discrete signals

- Environmental signals use to be analogic – light, temperature, …
- Embedded syste
- It is necessary a Processors



Inputs





System Controller

Outputs

Contents

An **input/output system** usually consists of following components:

- Transmission medium: used for exchanging data, e.g. a cable
- Communication port: connects transmission media to the embedded board
- Communication interface: manages data exchange between the processing unit and the device (or device controller)
- I/O controller: manages the I/O device
- I/O Buses: connect components on the embedded board
- Master processor (processing unit) integrated I/O

Input & Output Performance is one of the most important issues of an embedded system design, because I/O *can slow down* the overall performance of the whole system. There are some aspects that can be dealt with concerning I/O ***performance***:

- *The data rates of the I/O devices*: the actual amount of data that is delivered from the I/O device. Commonly it is measured in data per timeslice, Mbits per second for example.

- *The speed of the master processor*: If the processing unit is fast and the I/O device slow it might run idle must of the time, whereas the other way it might occur that the master processor is not able to process all the data delivered by the I/O device

- *How to synchronize the speed of the master processor with the speed of the I/O*: The speed of the processing unit should be designed that it can handle all data delivered by the I/O devices (or vice versa)

Interrupts are the common way of I/O devices communicating with the processing unit. There are other ways like polling or memory mapping.

- An interrupt is an event which stops the master processor executing its current instruction and handling the interrupt with a predefined handling mechanism.

- This can happen asynchronously, therefore it has to be defined how the master processor communicates with the I/O devices. (e.g. If an I/O interrupt handler can be interrupted by another I/O device)

Managing Data is also one of the most important field dealing with input/output. Basically, there are two main types of managing data: managing data serially and managing data in parallel.

- *Serial data transfer*: In serial communication, there are different ways in which communication can occur.
  - Serial Simplex Communication - Only one communication direction exists
  - Serial Half-Duplex Communication - Both directions but only alternately
  - Serial Full-Duplex Communication
    - Asynchronous - Data is packed into 4 to 8 bit, which are sent asynchronously and stored into buffers.
    - Synchronous - Data is transferred related to clock cycles.
- *Parallel data transfer*: Devices involved are capable of handling more than one bit at a cycle simultaneously.

Buses components which have to exchange data are connected via buses that corresponds to a bundle of wires which carry all various kinds of data.

The SPI Bus:

is a simple 4-wire serial communications interface used by many microprocessor/microcontroller peripheral chips that enables the controllers and peripheral devices to communicate each other in a full-duplex mode.

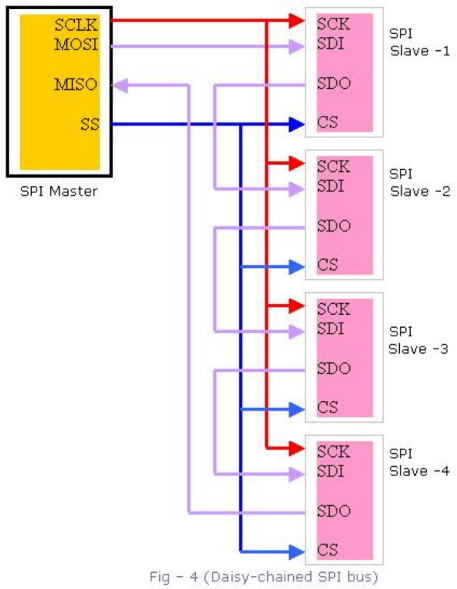


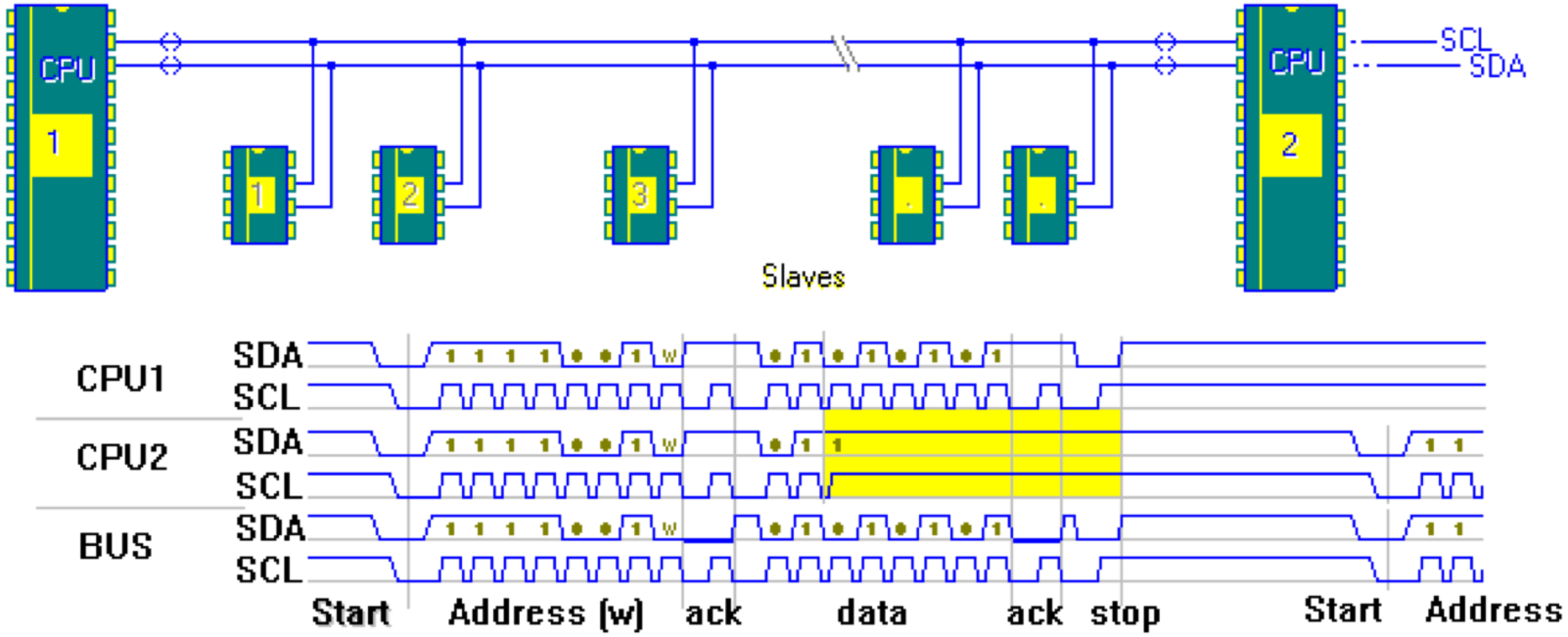
Fig – 4 (Daisy-chained SPI bus)

An SPI protocol specifies 4 signal wires.

1. Master Out Slave In (MOSI) - MOSI signal is generated by Master, recipient is the Slave.
2. Master In Slave Out (MISO) - Slaves generate MISO signals and recipient is the Master.
3. Serial Clock (SCLK or SCK) - SCLK signal is generated by the Master to synchronize data transfers between the master and the slave.
4. Slave Select (SS) from master to Chip Select (CS) of slave - SS signal is generated by Master to select individual slave/peripheral devices.

The I2C Bus:

Consists of 2 active wires and a ground connection. The active wires, called SDA and SCL, are both bi-directional. SDA is the Serial DAta line, and SCL is the Serial CLock line.

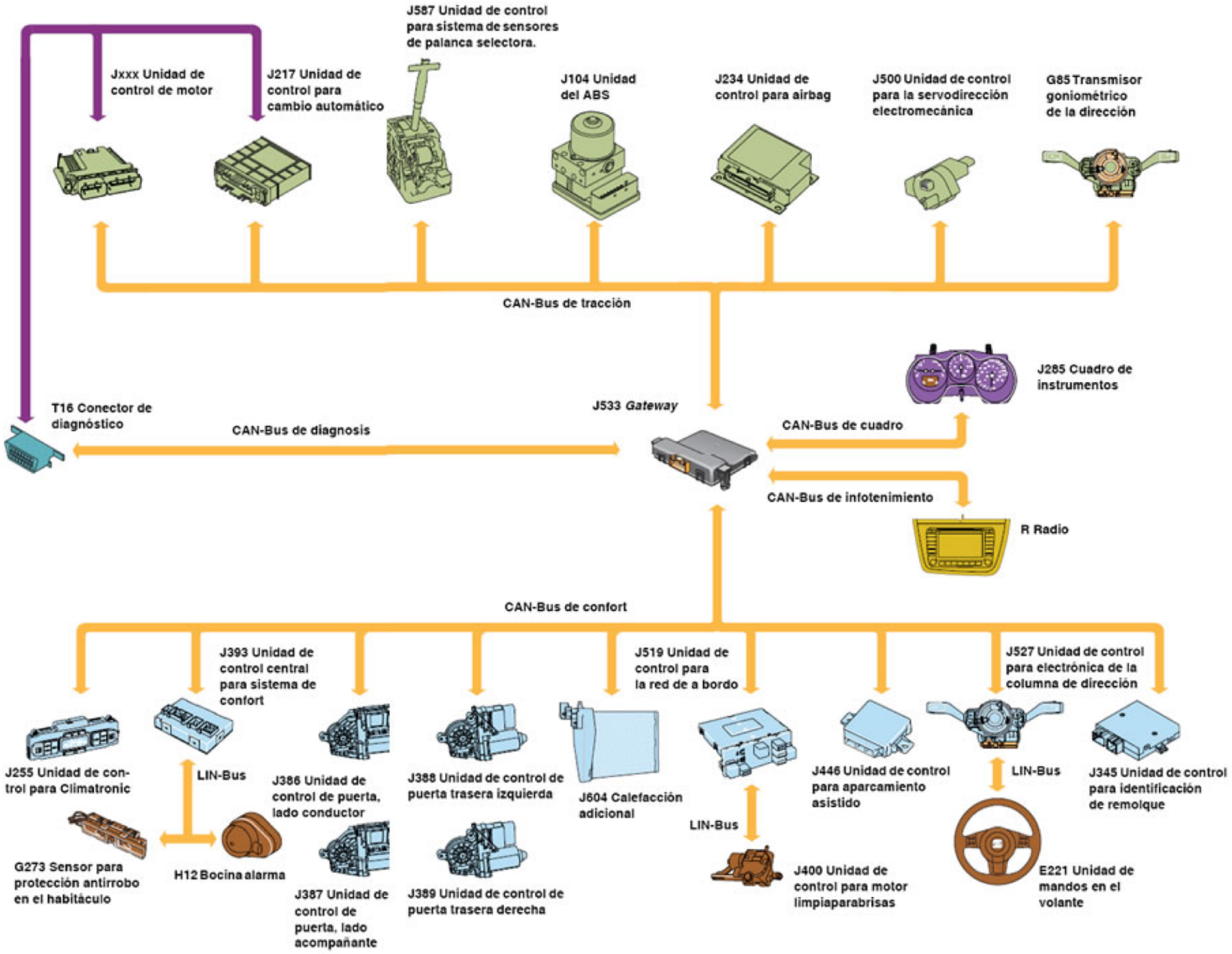


The CAN (Controller Area Network) Bus: Is an asynchronous serial bus to connect control units in cars, send data in real-time at the highest possible level of transmission security and reducing the amount of cables used (up to 2Km per car). The reduction of cable was granted due to the peer to peer structure of the CAN.

Nowadays the CAN - Bus is commonly used in all kinds of vehicles like trains, buses, cars and aircrafts.

The Buses speed varies from 33 Kbit/s to 1Mbit/s, depending on the length of the cable used to connect the control units and the type of CAN-bus implemented.

There are three different implementations: *High-speed CAN, Fault-Tolerant CAN, Single wire CAN*


J587 Unidad de control para sistema de sensores de palanca selectora.
Jxxx Unidad de control de motor
J217 Unidad de control para cambio automático
J104 Unidad del ABS
J234 Unidad de control para airbag
J500 Unidad de control para la servodirección electromecánica
G85 Transmisor goniométrico de la dirección
CAN-Bus de tracción
J285 Cuadro de instrumentos
T16 Conector de diagnóstico
CAN-Bus de diagnosis
J533 Gateway
CAN-Bus de cuadro
CAN-Bus de infotenimiento
R Radio
CAN-Bus de confort
J393 Unidad de control central para sistema de confort
J519 Unidad de control para la red de a bordo
J527 Unidad de control para electrónica de la columna de dirección
J255 Unidad de control para Climatronic
LIN-Bus
J386 Unidad de control de puerta, lado conductor
J388 Unidad de control de puerta trasera izquierda
J604 Calefacción adicional
J446 Unidad de control para aparcamiento asistido
LIN-Bus
J345 Unidad de control para identificación de remolque
G273 Sensor para protección antirrobo en el habitáculo
H12 Bocina alarma
J387 Unidad de control de puerta, lado acompañante
J389 Unidad de control de puerta trasera derecha
LIN-Bus
J400 Unidad de control para motor limpiaparabrisas
E221 Unidad de mandos en el volante