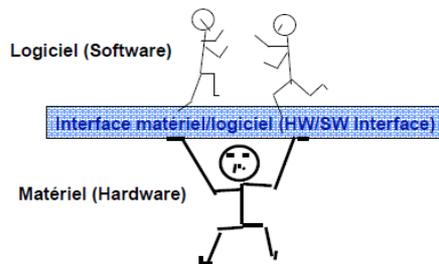


# Master 1 : Systèmes embarqués et Mobilité

## Matière : Processeurs embarqués (PE)

### Cours 0.1 : Introduction aux systèmes embarqués (SE)

Année 2022 / 2023



# Syllabus : Objectifs

---

La matière se décompose en deux grandes parties :

1. La première partie porte sur les processeurs embarqués. Elle se propose d'initier les étudiants à la conception de ces derniers.
2. La deuxième partie aborde les circuits programmables, principalement les FPGA.

# Syllabus : Programme

- ❑ Introduction aux systèmes embarqués
- ❑ Les processeurs
  - CISC/RISC
  - Pipeline et superpipeline
  - Architectures scalaires et superscalaires
  - Architecture VLIW
  - Contrôleurs/DSP
- ❑ Les composants reconfigurables (FPGA)
  - La technologie des circuits FPGA
    - Les ressources de routage
    - Les cellules de base
    - L'organisation générale des circuits FPGA
    - Les ressources spécifiques (PLL, E/S rapides, multiplieurs, mémoires)
  - La reconfiguration dynamique des FPGA
  - Quelques familles de composants FPGA
  - Environnements de simulation

# Syllabus : Bibliographie

- ❑ *Architecture de l'ordinateur*, Andrew Tanenbaum, Pearson, 5eme édition, 2009.
- ❑ *Computer Architecture: A Quantitative Approach*, David A. Patterson, John L. Hennessy, Morgan Kaufmann, 4th edition, 2009.
- ❑ *Architecture et technologie des ordinateurs*, Paolo Zanella, Yves Ligier, Dunod, 5<sup>ème</sup> éd., 2013.
- ❑ *Architecture de l'ordinateur*, Robert Strandh, Irène Durand, Dunod, 2005.
- ❑ *Architecture des machines et des systèmes informatiques*, Alain Cazez, Joëlle Delacroix, 3<sup>ème</sup> édition, Dunod, 2008.
- ❑ *Computer organization and design: the hardware/software interface*, David A. Patterson, John L. Hennessy, Morgan Kaufmann, 4th edition, 2008.
- ❑ *Architecture de l'ordinateur* Nicholas P. Carter, Dunod, 2002 - 360 pages
- ❑ *Introduction to reconfigurable computing*, BOBDA, édition 2007, 350 pages
- ❑ *Fine & coarse grain computing reconfigurable computing*, Sousris Dimitios, Vassiliadis S. , édition 2007, 300 pages

# Emploi du temps

---

- ❑ **Dimanche : 9h45-11h15, TDs, salle PG**
- ❑ **Lundi : 8h00-9h30, Cours, salle PG**
- ❑ **Mercredi: 9h45-11h15, TDs, salle PG**

# Syllabus : Evaluation

Contrôle des Connaissances		Pondération en %
Examen final		50
Travaux dirigés 3 Micro interrogations		25
Travaux pratiques	Coef 3	25
Assiduité	Coef 1	
Total		100%

---

**boite email commune :**

Identifiant : [sem1.ubma@gmail.com](mailto:sem1.ubma@gmail.com)

Mot de passe : Lase\_202223

# QCM 1

1. L'unité arithmétique et logique (UAL) :

- effectue des calculs
- décode les instructions
- modifie le registre d'état
- stocke des données

2. Un registre est une zone :

- en mémoire centrale
- dans le processeur
- dans l'UAL
- dans le séquenceur

3. L'horloge associée au processeur est reliée :

- au décodeur
- au séquenceur
- à l'UAL
- au pointeur d'instruction

4. Le registre d'instruction :

- contient l'adresse de la prochaine instruction
- contient l'instruction en cours de traitement
- contient les opérandes de l'instruction en cours
- contient des micro-commandes

5. Le cycle d'instruction est composé de cinq phases :

- 1 - recherche de l'instruction, 2 - exécution de l'instruction, 3 - stockage du résultat, 4 - recherche des opérandes, 5 - décodage de l'instruction

Quel est l'ordre des phases ?

- 1, 3, 5, 4, 2
- 2, 3, 5, 4, 1
- 4, 2, 3, 1, 5
- 1, 5, 4, 2, 3

6. MOV AX, [adresse] est :

- une instruction en langage machine
- une instruction en langage d'assemblage
- une instruction en langage exécutable
- une instruction en langage binaire

7. Dans MOV AX, [adresse] les éléments en gras sont :

- des opérateurs
- des opérations
- des opérables
- des opérandes

# QCM 1

8. Dans MOV AX, [adresse] AX est :
- une zone en mémoire centrale
  - une variable
  - une constante
  - un registre
9. Le cycle de développement est composé de 5 phases : 1 – analyse, 2 – compilation, 3 –algorithme, 4 – tests, 5 – programmation
- 1, 3, 5, 2, 4
  - 2, 3, 5, 4, 1
  - 4, 2, 3, 1, 5
  - 1, 5, 4, 2, 3
10. Classez les langages suivants du plus simple au plus sophistiqué :
- 1 – Windev, 2 – Langage C, 3 – Langage Machine, 4 – Access, 5 – Assembleur
- 1, 3, 5, 4, 2
  - 2, 3, 5, 4, 1
  - 3, 5, 2, 4, 1
  - 4, 2, 3, 1, 5
11. Un processeur superscalaire peut effectuer plusieurs calculs en même temps :
- vrai
  - faux
12. La mémoire cache sert à réduire les temps d'attentes du processeur :
- vrai
  - faux
13. Pour doubler la rapidité de calcul d'une machine, il suffit de doubler la fréquence d'horloge du processeur :
- vrai
  - faux

# QCM 2

- ❑ Utilise des instructions symboliques qui représentent les codes d'opération et les positions de mémoire.
  - Langage assembleur
  - Langage machine
  - Langage non procédural.
- ❑ Langage se traduit par la présence, l'absence ou le changement de direction d'un courant électrique, de champs magnétiques ou de rayons lumineux.
  - Icône
  - Photostyle
  - Réponse vocale
  - Représentation binaire
- ❑ Traduit les instructions et dirige le traitement.
  - Manette de jeu
  - Poste de travail
  - Souris électronique
  - Unité de commande

# QCM 2

- ❑ Quelle affirmation est fausse ?
  - Un programme écrit dans un langage dit « compilé » va être traduit une fois pour toutes par un programme annexe 'le compilateur' afin de générer un nouveau fichier qui sera autonome
  - Un programme écrit dans un langage interprété a besoin d'un programme 'l'interpréteur' pour traduire au fur et à mesure les instructions du programme
  - Le langage utilisé par le processeur c'est à dire des données telles qu'elles lui arrivent, est appelé langage machine. Il s'agit de données élémentaires de 0 à F en hexadécimal.
  - Un langage de programmation est une notation conventionnelle destinée à formuler des algorithmes qui les appliquent
  
- ❑ L'équivalent en hexadécimal du nombre binaire 1011001100011100 se termine par :
  - A
  - G
  - F
  - C

# QCM 2

- ❑ Le nombre 0,75 est égal à :
  - $0,11_2$
  - $0,3_4$
  - $0,6_8$
  - Les 3 réponses sont vraies
- ❑ On considère une machine ayant la configuration suivante : taille de la mémoire centrale TMC = 1 Mo, taille du bus adresse TBA = 20 bits, Le nombre de mots adressables est :
  - $2^{TBA}$
  - $2^{TBA-1}$
  - $TBA^2$
  - $TMC^{TBA}$
- ❑ La capacité d'un circuit mémoire possédant un bus adresse de 14 bits et un bus de données de 8 bits est :
  - 8ko
  - 16ko
  - 32ko
  - Aucune de ces réponses n'est juste

# QCM 2

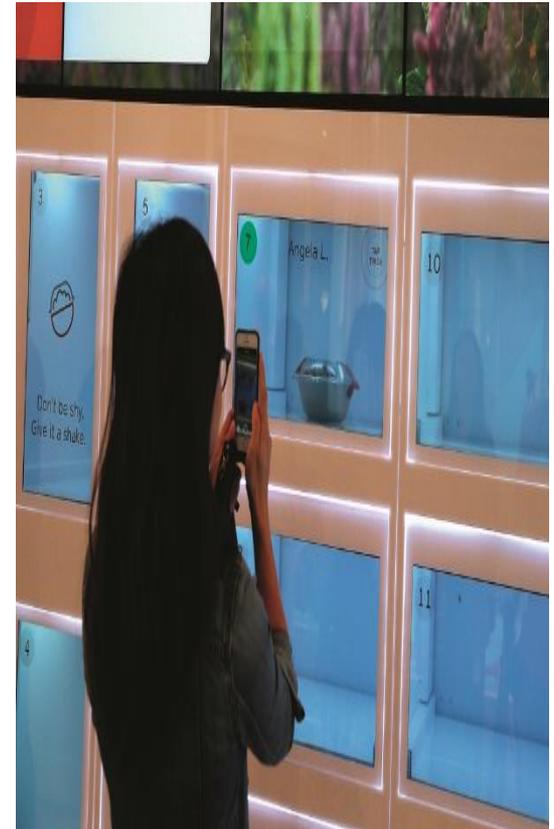
- ❑ La taille mémoire adressée par un processeur est de 1Mo. Les registres et les mots sont de 16 bits. La longueur de l'adresse est :
  - 16 bits
  - 8 bits
  - 24 bits
  - 20 bits
  
- ❑ La taille mémoire principale gérée par un microprocesseur est de 16Mo et celle de la mémoire cache est de 256ko. L'adresse aura une taille de :
  - 24 bits
  - 16 bits
  - 40 bits
  - 20 bits

# Où se trouvent les SE ?



# Où se trouvent les SE ?

- ❑ Un établissement complètement automatisé à San Francisco. Il est ainsi possible de ne plus avoir aucune interaction avec un être humain en se rendant au restaurant.
- ❑ Quand vous arrivez au bout de la (très longue) file d'attente pour commander votre repas au restaurant *Eatsa*, à San Francisco,
- ❑ "vous tombez nez à nez non pas avec un caissier, mais avec un iPad".
- ❑ Puis, quand votre salade bio est prête, "ce n'est pas un serveur qui vous l'apporte : elle apparaît dans un cube numéroté".
- ❑ Même principe pour le paiement : tout se fait par carte et en ligne, le restaurant n'acceptant pas le liquide.



# Où se trouvent les SE ?

## □ La reconnaissance faciale se développe

- Chaque visage est unique. Les logiciels de reconnaissance faciale sont donc capables de parfaitement vous identifier.

Cette technologie est en pleine expansion.

- Par exemple : pour payer chez un commerçant avec sa carte bancaire plutôt que taper le code, le commerçant scanne votre visage et s'assure ainsi que vous êtes bien le propriétaire de la carte Bancaire, qu'elle n'a pas été volée. C'est un moyen d'identification plus sûr qu'un code à quatre chiffres.

# Où se trouvent les SE ?

## □ La reconnaissance faciale se développe

- L'autre moment où l'on a besoin d'un code à quatre chiffres, c'est le digicode pour rentrer chez soi.  
Là encore, le logiciel de reconnaissance faciale vous identifie et ouvre la porte.  
Ce logiciel est également utilisé dans les hôtels, au bureau mais également pour les accès aux parcs d'attraction.
- Certains petits malins passent leur journée à voler du papier toilette dans les WC des sites touristiques.  
Sauf que désormais avec la reconnaissance faciale, on vous identifie et au bout de trois ou quatre passages, l'accès vous est refusé.

# Les SE sont polymorphes



# Les systèmes embarqués

---

- ❑ Les systèmes embarqués (SE) sont donc **partout, discrets, efficaces et dédiés** à ce à quoi ils sont destinés.
- ❑ Omniprésents, ils le sont déjà et le seront de plus en plus.
- ❑ *Ils sont bourrés d'**électronique** plus ou moins complexe et d'**informatique** plus ou moins évoluée.*

# Les systèmes embarqués

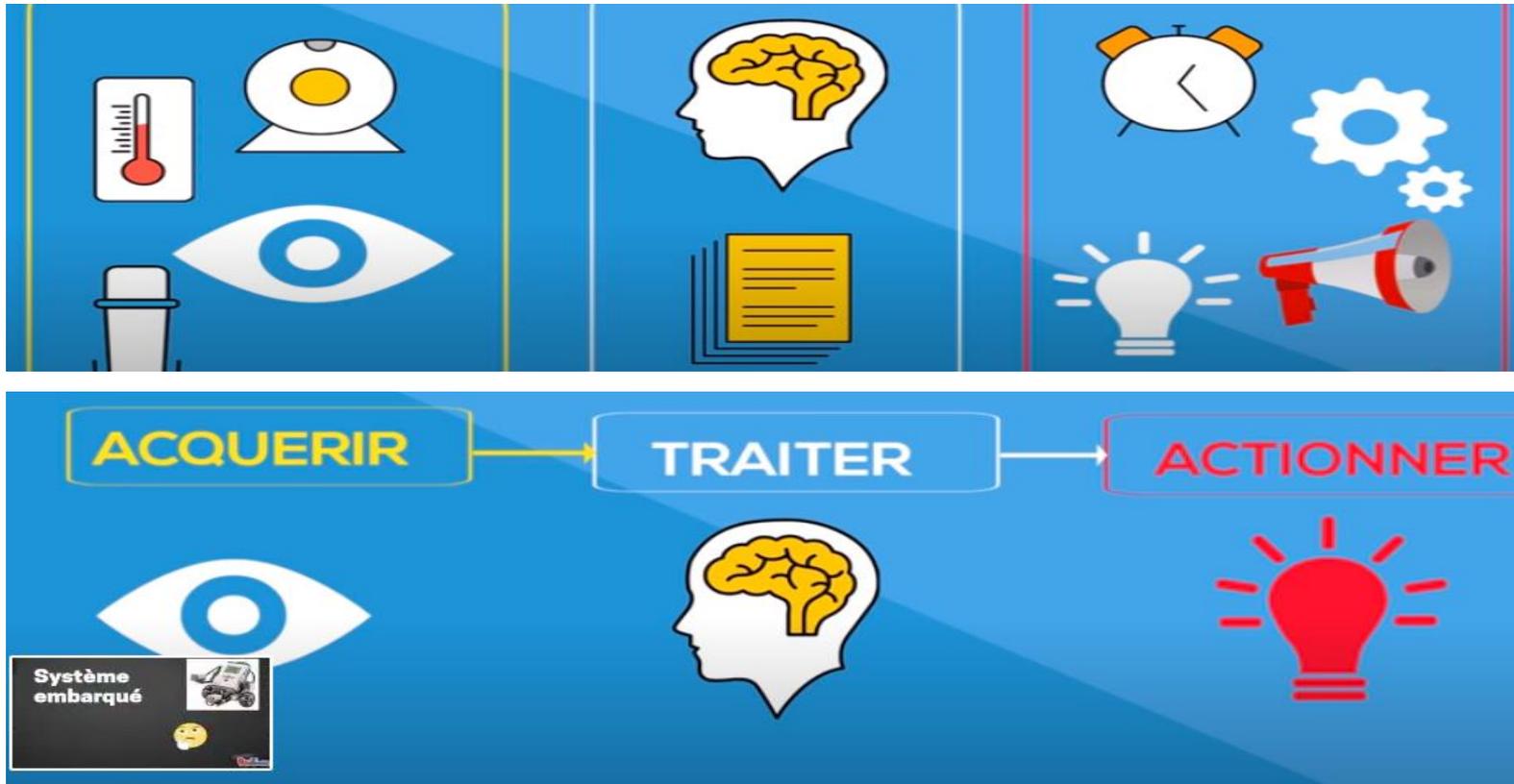
- ❑ Système autonome qui peut fonctionner sans intervention humaine : caisse automatique de supermarché, remontée mécanique automatique, portail automatique pour accès à une résidence, voiture automatique
- ❑ Un système embarqué est un système informatique qu'on va intégrer dans le monde réel. par exemple : Tout le traitement électronique qu'il y a dans une voiture, dans un appareil photo, une cafetière, une machine à laver.
- ❑ Tous les calculateurs qui ne sont pas un ordinateur, qui ne sont pas un PC ou un laptop. Systèmes électroniques qui sont inclus dans des systèmes plus grands

# SE – Tentatives de Définition

**Il existe plusieurs définitions de SE :**

- ❑ Un SE peut être défini comme un système de calcul électronique (Hw) et informatique (Sw) autonome conçu pour réaliser une tâche particulière.  
Souvent, un SE ne possède pas des entrées/sorties standards comme un clavier ou un écran d'ordinateur (PC).
- ❑ Un SE (embedded system) désigne un matériel électronique comprenant au moins un microprocesseur (microcontrôleur) et un logiciel dédié à sa gestion;
- ❑ Le système matériel et l'application sont intimement liés et noyés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type PC.

# Autonomie : Chaine d'information

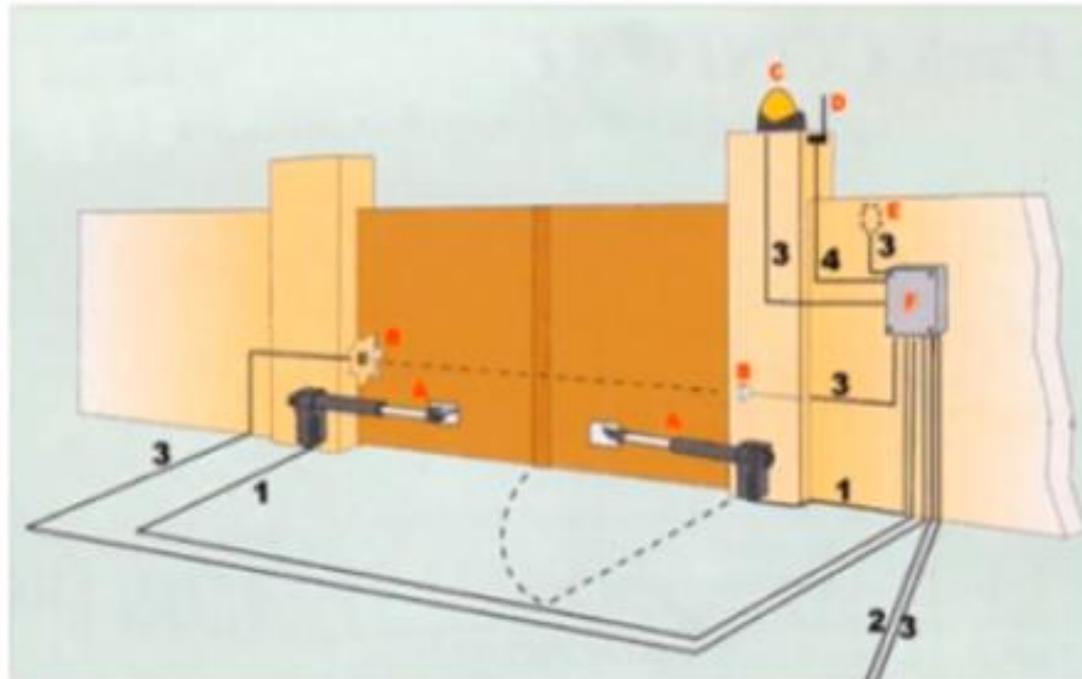


# Systeme embarqué

---



# Exemple d'un système embarqué



- A** vérins
- B** cellules photoélectriques
- C** gyrophare ou flash
- D** antenne
- E** clavier ou contact à clef
- F** armoire de commande

- 1** câble 4 x 1,5 mm<sup>2</sup>
- 2** câble 3 x 1,5 mm<sup>2</sup>
- 3** câble téléphone
- 4** câble coaxial

Vers boutons poussoirs (commande manuelle  
interphone, entrée, garage ...)

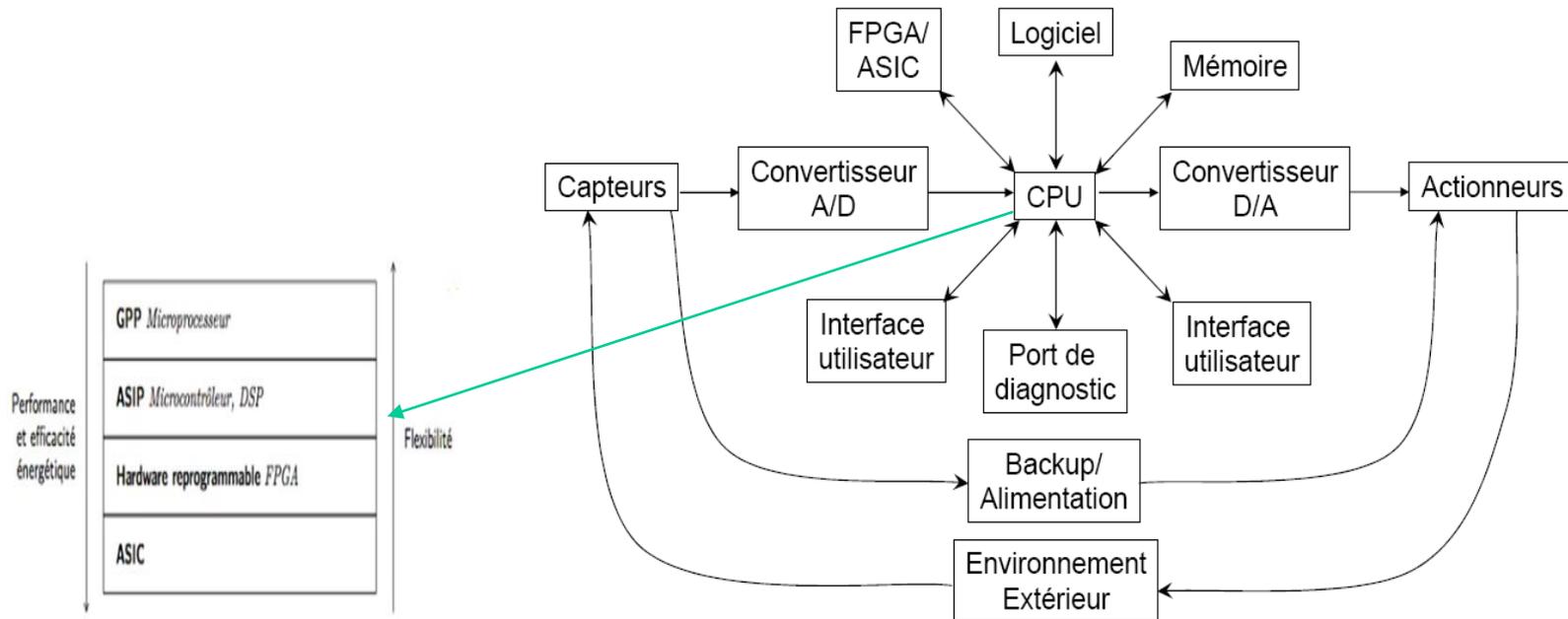
Vers tableau de distribution  
(circuit signalé 10 A)

# SE - PC différences

- ❑ L'interface IHM peut être aussi simple qu'une led qui clignote ou aussi complexe qu'un système de vision de nuit en Temps Réel.
- ❑ Des circuits numériques FPGA, ASIC ou des circuits analogiques sont utilisés en plus pour augmenter les performances du système ou sa fiabilité.
- ❑ Le logiciel a une fonctionnalité fixe à exécuter et est spécifique à une application.
- ❑ Le système d'exploitation réside dans une puce et non sur un disque dur !

# SE - Éléments

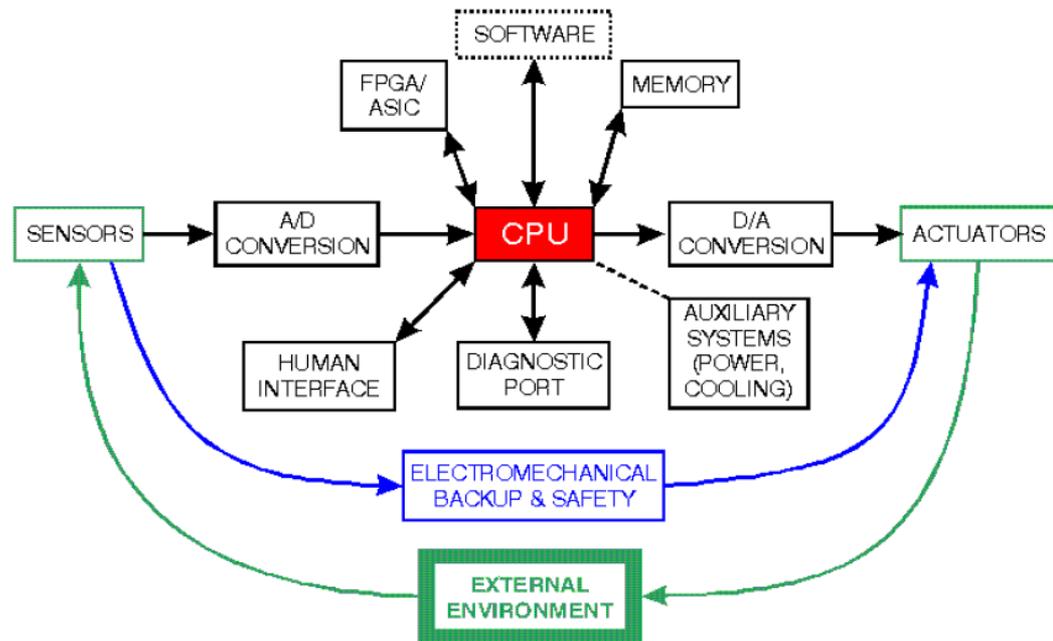
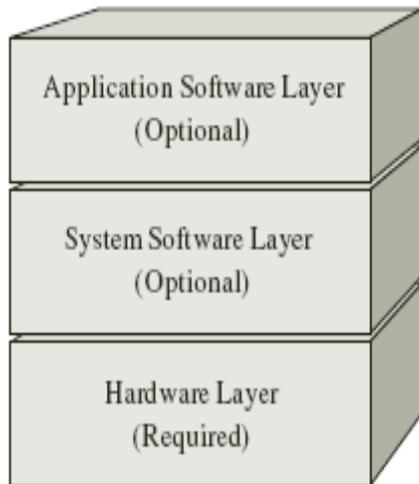
## □ Composants d'un SE



Différents composants d'un système embarqué

# SE - Éléments

## □ SE components



Modèle d'un SE : Architecture en couches

# SE - Éléments

---

- ❑ On retrouve en entrée des capteurs généralement analogiques couplés à des convertisseurs A/N.
- ❑ On retrouve en sortie des actionneurs généralement analogiques couplés à des convertisseurs N/A.
- ❑ Au milieu, on trouve le calculateur mettant en œuvre un processeur embarqué et ses périphériques d'E/S.

*Il est à noter qu'il est complété généralement d'un circuit FPGA jouant le rôle de coprocesseur afin de proposer des accélérations matérielles au processeur.*

# SE - Eléments

Sur ce schéma théorique se greffe un paramètre important : le rôle de **l'environnement extérieur**.

Contrairement au PC ronronnant bien au chaud dans un bureau, un système embarqué doit faire face à des environnements plus hostiles. Il doit faire face à un ensemble de paramètres agressifs :

- Variations de la température.
- Vibrations, chocs.
- Variations des alimentations.
- Interférences RF
- Corrosion.
- Eau, feu, radiations.
- ...

# SE - Éléments

- ❑ L'environnement dans lequel opère le système embarqué n'est pas contrôlé ou contrôlable. Cela suppose donc de prendre en compte ce paramètre lors de sa conception. On doit par exemple prendre en compte les évolutions des caractéristiques électriques des composants en fonction de la température, des radiations, ...
- ❑ Enfin pour terminer cette partie, les systèmes embarqués sont aujourd'hui fortement communicants. Cela est possible grâce aux puissances de calcul offertes par les processeurs pour l'embarqué (32 bits en particulier) et grâce aussi à l'explosion de l'usage de la connectivité Internet ou connectivité IP.
  - ❑ La connectivité IP permet fondamentalement de contrôler à distance un système embarqué par Internet. Ce n'est en fait que l'aboutissement du contrôle à distance d'un système électronique par des liaisons de tout type : liaisons RS.232, RS.485, bus de terrain, ...

# Rôle des systèmes embarqués

- Offrir aux systèmes électroniques l'intelligence nécessaire :
  - À un fonctionnement autonome
    - Capacité de traitement du logiciel
    - Possibilité de prise de décision
  - À l'enregistrement d'informations pendant le fonctionnement du système, pour
    - Tracer en « temps-réel » le fonctionnement
    - Suivre l'historique du fonctionnement
    - Isoler plus facilement les causes de défaillance

# Rôle des systèmes embarqués

- Faciliter le développement des systèmes :
  - Fonctions traditionnellement dévolues au matériel réalisées de façon logicielle
  - Plus grande souplesse lors de la phase de conception
  - Évolutivité directement proportionnelle aux possibilités informatiques du système
    - $\mu$ contrôleur : évolutions possibles mais limitées
    - Système d'exploitation : possibilités infinies

# Quelles sont les caractéristiques d'un système embarqué ?

- espace mémoire ;
- puissance de calcul ;
- autonomie ;
- délais d'exécution ;
- sécurité ;
- fiabilité.

# 4 types de SE

- ❑ Calculs généraux
  - ❑ Application similaire à une application de bureau mais empaquetée dans un système embarqué.
  - ❑ jeu vidéo, set-top box (décodeur tv par ex).
- ❑ Systèmes de contrôle (automatismes)
  - ❑ Contrôle de systèmes en Temps Réel.
  - ❑ Moteur d'automobile, process chimique, process nucléaire, système de navigation aérien.
- ❑ Traitement du signal
  - ❑ Calcul sur de grosses quantités de données.
  - ❑ Radar, Sonar, compression vidéo.
- ❑ Communication & réseaux
  - ❑ Transmission d'information et commutation.
  - ❑ Téléphone, Internet.

# SE - Éléments

- ❑ Cela permet l'emploi des technologies modernes du web pour ce contrôle à distance par l'utilisateur:
  - ❑ Il suffit d'embarquer un **serveur web** dans son équipement électronique pour pouvoir le contrôler ensuite à distance, de n'importe où, à l'aide d'un simple navigateur. Il n'y a plus d'IHM spécifique à concevoir pour cela, ce rôle étant rempli par le navigateur web. Cela est une réalité : les chauffagistes proposent maintenant des chaudières pouvant être pilotées par le web !
- ❑ Il faut aussi noter la **montée en puissance des communications sans fil** dans l'embarqué au détriment des communications filaires pour limiter le câblage et faciliter la mise en place du système embarqué.
  - ❑ Le Wifi et toutes les normes de réseaux sans fil IEEE 802.15 comme Zigbee ont le vent en poupe dans l'embarqué et surtout en domotique (réseaux de capteurs sans fil par exemple).
- ❑ Mais ne nous méprenons pas sur ces facilités et commodités, cela a bien sûr un revers : **la sécurité du système embarqué** puisque connecté à Internet.
  - ❑ La sécurité des systèmes embarqués est donc cruciale aujourd'hui et doit être prise en compte dès leur conception !

# Le marché des SE

---

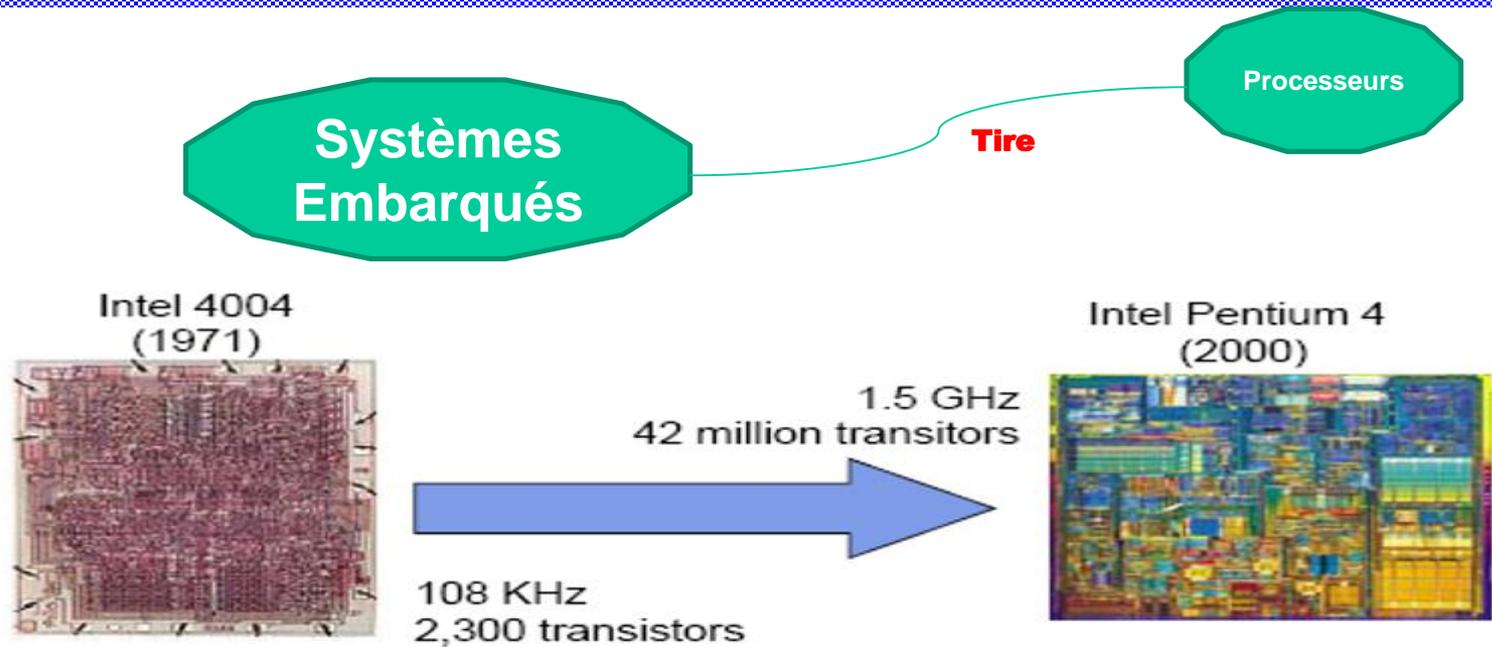
- ❑ L'embarqué est un terme plus général qui regroupe plusieurs notions selon le contexte :
  - ❑ Le marché des systèmes embarqués.
  - ❑ Les systèmes embarqués.

Le mot embarqué signifie aussi que le SE fait partie d'un plus gros système appelé « embedding system ».

# Le marché des SE

- Selon une étude de la société **Transparency Market Research**,
  - Le marché mondial des systèmes embarqués évalué à 153 milliards de dollars en 2016 et devrait atteindre 223 milliards de dollars en 2021.
  - Le secteur est composé de deux catégories d'acteurs : matériel et logiciel.
    - *La catégorie concentrée sur le matériel représente 98% du marché en termes de chiffre d'affaires*
    - *mais la partie logicielle 2% est en croissance.*

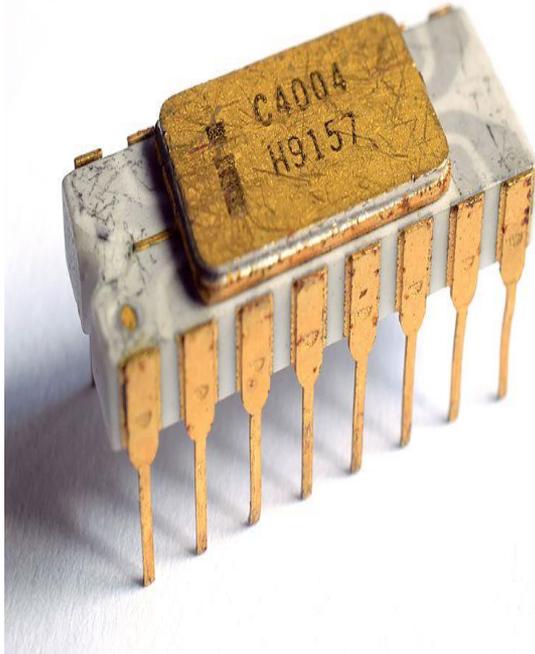
# Développement de l'électronique



*Si la vitesse de l'automobile augmentait de la même façon que les processeurs dans la même période, il serait possible aujourd'hui de parcourir 4000 km en moins de 13 secondes !*

# Développement de l'électronique

## ❑ Intel 4004



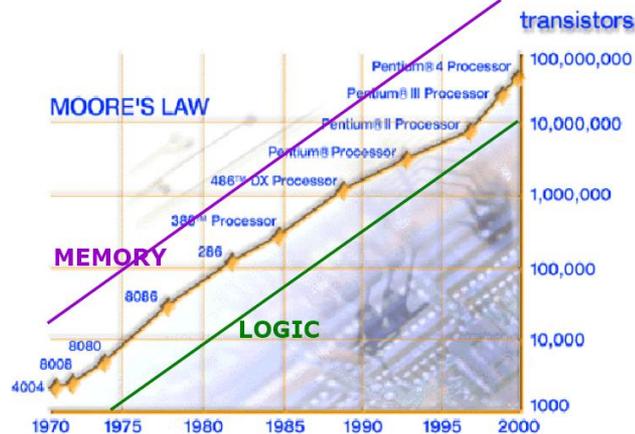
Caractéristiques	
Production	1971 - 1981
Fabricant	Intel
Fréquence	740 kHz
Finesse de gravure	10 $\mu\text{m}$
Architecture	4 bits
Boîtier	DIP à 16 broches

Comparaison avec ENIAC  
(premier ordinateur) :  
surface : 167 m<sup>2</sup>  
poids : 30 tonnes

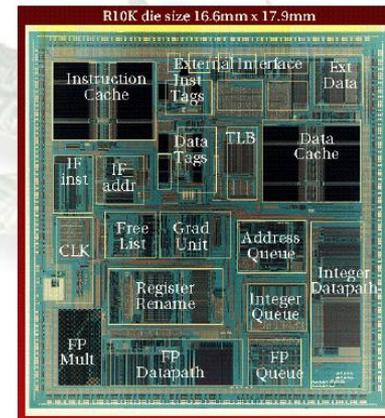
# Développement de l'électronique

L'évolution des microprocesseurs suit la loi de Moore (1970) :  
Pour une surface de silicium donnée, le nombre de transistors intégrés double tous les 18 mois.

Le nombre de transistors est passé de 2300 sur le Intel 4004, à plusieurs dizaines de millions de transistors pour les



## MIPS R10000

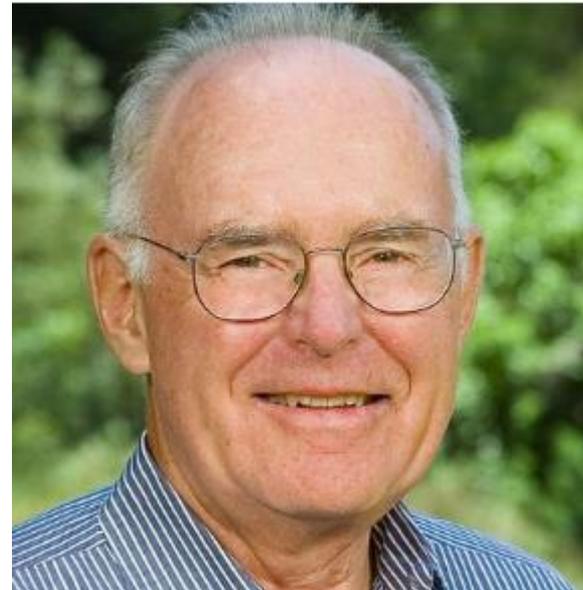


2,600,000 transistors

# Gordon Moore

Gordon Moore (93 ans)  
Cofondateur d'Intel  
Société créée en 1968

La société Intel  
(abréviation de  
**INT**egrated  
**E**lectronics)  
estimée à 254,88  
milliards de USD à la  
bourse de New York  
Nasdaq sous sigle INTC  
en 2018.



Gordon Moore (Intel) :  
12.6Mds en xx/03/2021

# Tableau des microarchitectures x86 Intel

Architecture	Marque commerciale
	Intel 8086, Intel 8088, Intel 80186, Intel 80188
	286
	386
	486
P5	Pentium
	Pentium MMX
P6	Pentium Pro
	Pentium II, Celeron, Pentium II Xeon
	Pentium III, Celeron, Pentium III Xeon
NetBurst	Pentium 4, Pentium D, Celeron, Xeon
P6	Pentium M, Celeron
	Core Solo, Core Duo, Pentium Dual-Core, Celeron, Xeon
Core	Core 2 Solo, Core 2 Duo, Core 2 Quad, Pentium Dual-Core, Celeron, Xeon
Nehalem	
Sandy Bridge	Core i7, Core i5, Core i3, Pentium, Celeron, Xeon
Haswell	
Skylake	Intel Core i3, Intel Core i5, Intel Core i7, Intel Core i9, Pentium

# La loi de Moore ?

---

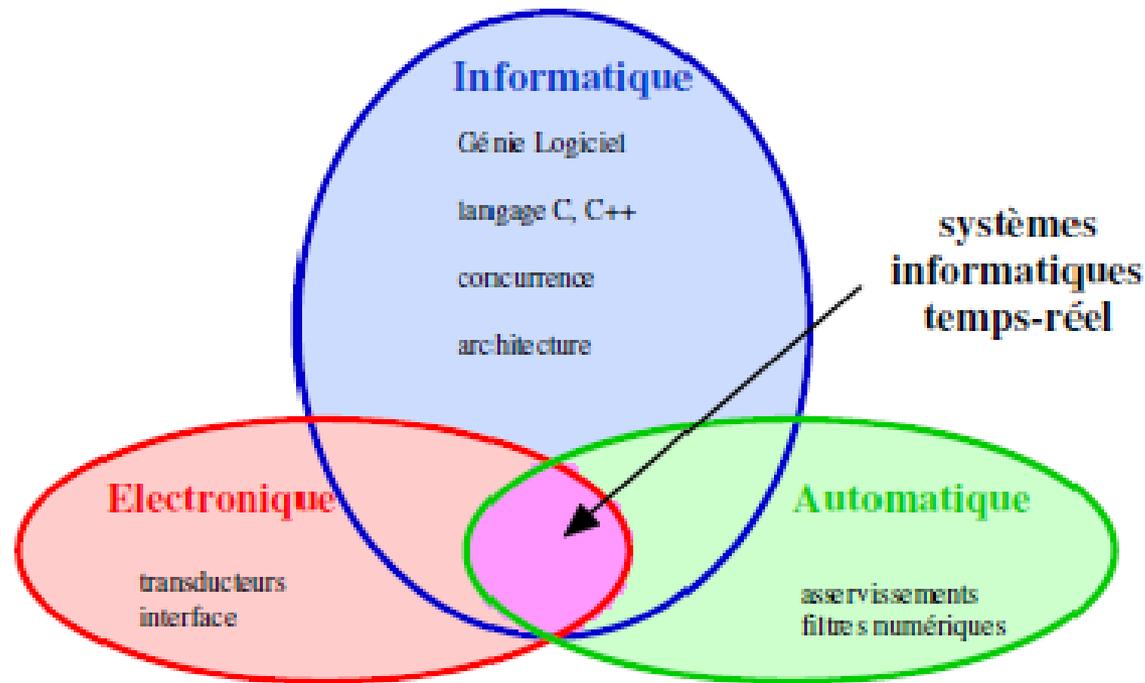
- ❑ L'industrie des **semi-conducteurs** s'attend à un arrêt du rétrécissement de la taille des transistors en 2021. Mais des solutions alternatives émergent pour continuer à faire évoluer les microprocesseurs.
- ❑ **La loi de Moore résistera-t-elle ?**

# Notion de temps-réel

---

- ❑ On entend souvent parler de Temps Réel dès que l'on parle de système embarqué.
- ❑ Le Temps Réel est un concept un peu vague et chacun a sa propre idée sur la question.

# Où se situe le temps réel ?



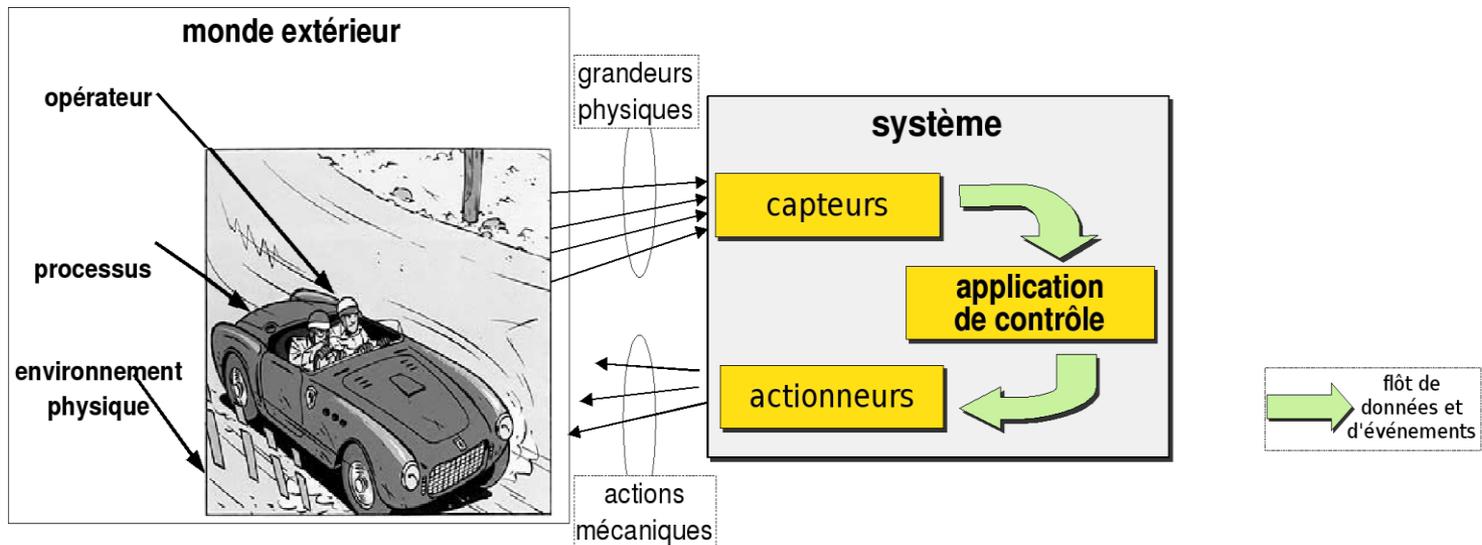
# Informatique temps réel

- ❑ "En informatique temps réel, le comportement correct d'un système dépend, non seulement des résultats logiques des traitements, mais aussi du temps auquel les résultats sont produits.
  
- ❑ Objectifs :
  - ❑ Déterminisme logique : les mêmes entrées appliquées au système produisent les mêmes résultats.
  - ❑ Déterminisme temporel : respect des contraintes temporelles (ex : échéance).
  - ❑ Fiabilité : le système répond à des contraintes de disponibilité (fiabilité du logiciel et du matériel).

# Informatique temps réel

- ❑ On qualifie de **temps réel**, une application mettant en œuvre un système informatique dont le **comportement est conditionné** par l'évolution dynamique de l'état du **procédé** qui lui est **connecté**.
- ❑ Ce système informatique est alors chargé de suivre ou de piloter ce procédé en **respectant des contraintes temporelles** définies dans le cahier des charges de l'application.
- ❑ Un système temps-réel doit réagir aux stimuli de son environnement dans un délai temporel imposé par l'environnement, l'application.
  - ❑ Ex: système de freinage, Contrôle automatisme, procédés industriels, ...

# Informatique temps-réel



- Systemes temps réels :**
  - En charge du contrôle d'un processus
  - Liés à la dynamique du processus à contrôler
  - Soumis à des contraintes temporelles

# Classification des SE temps-réel

- ❑ **Systemes à contraintes strictes ou critiques (hard, critical) :**  
Il est impératif que le système réponde dans les temps (délais).  
Exemples : SE utilisés dans l'automobile - système de freinage, suspension, etc. l'aéronautique, l'aérospatiale, ...
- ❑ **Systemes à contraintes Souples (Soft) :** Une faible probabilité de ne pas respecter des limites temporelles peut être tolérée.  
Exemples : systèmes multimédia ; télécommunications ou vidéo.
  - Pour un système vidéo : *si une image est affichée en retard ou une trame de données transmise sur une ligne téléphonique est perdue, cela ne remet pas en cause le fonctionnement correct de l'ensemble du système.*
  - *Dans l'automobile : le déclenchement automatique des essuie-glaces à la détection de gouttes de pluie est une tâche non critique;*

# Application en temps réel

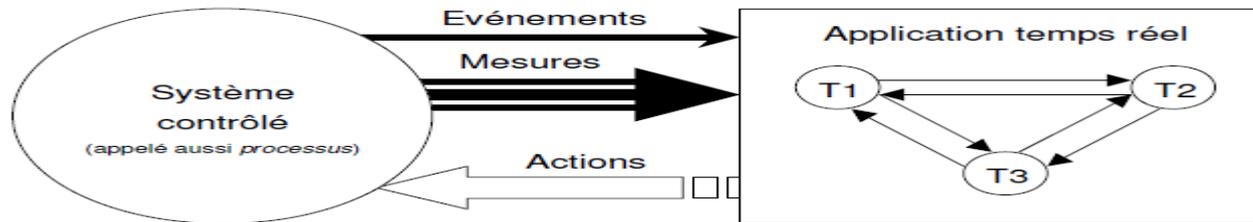
- ❑ Une application en temps réel met en œuvre des systèmes informatiques ou informatisés coopérant avec l'homme et destinés à la perception, l'observation, l'aide à la décision et la conduite de procédés dynamiques.
- ❑ Ainsi, de nos jours, l'informatique « temps réel » est présente dans de nombreux domaines industriels, que ce soient :
  - des systèmes embarqués dans des équipements de haute technologie (aéronautique, nucléaire),
  - ou des systèmes embarqués produits en grande quantité, à coût modéré, pour des équipements plus classiques (automobiles, capteurs intelligents, signalisation).
- ❑ La prise en compte du temps dans ces systèmes informatiques peut s'effectuer sous plusieurs approches, pour autant les interactions entre procédé et système doivent être instantanées, d'où le rôle centralisateur et ordonnanceur joué par l'exécutif.

# Grandeurs contraintes TR

---

- Mesures scientifiques : 10ns - 1ms
- Systèmes radar : 1 $\mu$ s - 1ms
- Systèmes vocaux : 10 $\mu$ s - 10ms
- Robotique : 1ms - 10 ms
- Contrôle de stockage : 1s - 1mn
- Contrôle de fabrication : 1mn - 1h
- Contrôle de réactions chimiques : 1h et plus
- Mission spatiale : mois ou années

# Représentation d'une application temps réel



## □ Différence entre des événements et des mesures :

- les événements sont *fugaces*. Ils peuvent être considérés comme des signaux limités dans le temps de type impulsion ; sitôt émis, ils disparaissent. Exemples : tops d'horloge (événement périodique) ou pression sur une touche (événement sporadique).
- Les mesures sont des *informations permanentes*. Le système doit prélever de manière explicite ces informations pour assurer sa mission. Exemples : une température ou l'état d'un interrupteur.  
Les actions entreprises par l'application temps réel servent à contrôler et commander le système contrôlé.

**Il est impératif de bien prendre en compte le fait que le système contrôlé a sa propre dynamique. Il l'impose à l'application temps réel, et non l'inverse !**

# Exemple d'application temps réel

- ❑ Un système embarqué à bord d'un avion a pour but de fournir la position instantanée de l'avion par rapport à un point de référence terrestre (un aéroport par exemple). Pour ce faire, il tire ses informations d'un certain nombre de capteurs.
- ❑ Le temps de traitement global correspondant au travail à réaliser, est finalement le temps mis pour :
  - acquérir toutes les données d'entrée en provenance des capteurs,
  - traiter effectivement ces données,
  - fournir le résultat correspondant.
- ❑ Dans une première approche, on peut affecter une tâche à chaque étape. Chaque tâche est indépendante des deux autres.
  - Notons T1 le temps d'acquisition, T2 le temps de traitement des données et T3 le temps nécessaire à la présentation des résultats.
- ❑ Il est impératif que l'inéquation suivante soit respectée pour garantir un fonctionnement correct de l'application temps réel :  
$$T1 + T2 + T3 < Tl \quad (Tl : \text{temps limite de production du résultat})$$

# Exemple d'application temps réel

---

- Typiquement, le temps de réponse d'un système ( $T_r$ ) à une sollicitation est le suivant :

$$T_r = T_a + T_c + T_p$$

( $T_a$  : temps d'acquisition,

$T_c$  : temps de calcul,

$T_p$  : temps de présentation du résultat)

# SE & Temps-Réel

- ❑ Une idée reçue est de mélanger Temps Réel et puissance de calcul du système embarqué. On entend souvent : Être temps Réel, **c'est avoir beaucoup de puissance** : des MIPS (Million d'Instructions Per Seconde, des MFLOPS « opérations à virgule flottante par seconde » (en anglais, Mega FLoating point Operations Per Second).
- ❑ Ce n'est pas toujours vrai. En fait, être Temps Réel dans l'exemple donné précédemment, c'est être capable d'acquiescer l'interruption périodique (moyennant un temps de latence de traitement d'interruption imposé par le matériel), traiter l'information et le signaler au niveau utilisateur (réveil d'une tâche, libération d'un sémaphore, ...) dans un temps inférieur au temps entre deux interruptions périodiques consécutives.

# SE & Temps-Réel

- ❑ Dans le pire des cas, le traitement en Temps Réel sera réalisé en logique câblée tout simplement.
- ❑ Il convient donc avant de concevoir le système embarqué de connaître la durée minimale entre 2 interruptions ; ce qui est assez difficile à estimer voire même impossible.
- ❑ C'est pour cela que l'on a tendance à concevoir dans ce cas des systèmes performants et souvent surdimensionnés pour respecter des contraintes Temps Réel mal cernées à priori.

*Ceci induit en cas de surdimensionnement un surcoût non négligeable.*

# Développement des SE

- ❑ Du point de vue technique, la conception d'un système embarqué demande à son concepteur d'être pluridisciplinaire : électronique, informatique, réseaux, sécurité, ...
- ❑ Mais le concepteur se doit aussi d'être un bon gestionnaire car concevoir un système embarqué revient finalement à un exercice d'optimisation : minimiser les coûts de production pour des fonctionnalités optimales.
- ❑ Le système embarqué se doit d'être :
  - ❑ Robuste
  - ❑ Simple
  - ❑ Fiable
  - ❑ Fonctionnel. Le système doit toujours fonctionner correctement.
  - ❑ Sûr, surtout si la sécurité des personnes est en jeu.
  - ❑ Tolérant aux fautes.

# Développement des SE

- ❑ D'autres contraintes sont aussi à prendre en compte :
  - ❑ L'encombrement
  - ❑ Le poids
  - ❑ Le packaging : difficulté de faire cohabiter dans un faible volume, électronique analogique, électronique numérique et RF sans interférences.
  - ❑ L'environnement extérieur
  - ❑ La consommation électrique. Le système embarqué nomade doit être de faible consommation car il est alimenté par des batteries. Une consommation excessive augmente le prix de revient du système embarqué car il faut alors des batteries de plus forte capacité.
  - ❑ Le coût. Beaucoup de systèmes embarqués sont fabriqués en grande série et doivent avoir des prix de revient extrêmement faibles.
  - ❑ Le temps de développement. Dans un marché concurrentiel, il convient d'avoir un système opérationnel le plus rapidement possible pour être le premier sur le marché.

# Développement des SE

- ❑ Devant toutes ces contraintes, le concepteur adopte des règles de bon sens :
  - ❑ Faire simple.
  - ❑ Utiliser ce que l'on a déjà fait ou fait par d'autres.
  - ❑ Ne pas se jeter sur les technologies dernier cri. Quelle est leur pérennité dans le temps ?
  - ❑ Ne pas se jeter sur le dernier composant sorti surtout s'il est grand public. Quelle est sa pérennité dans le temps surtout s'il l'on travaille pour la défense où l'on demande une maintenance sur 30 ans !
  - ❑ Utiliser des technologies éprouvées qui ont fait leur preuve. Ces technologies peuvent d'ailleurs avoir plusieurs générations de retard par rapport à leurs homologues grand public.

*Pour le grand public, le concepteur de systèmes embarqués peut sembler faire de l'inertie face aux nouvelles technologies mais il faut le comprendre : c'est pour le bien du système qu'il conçoit surtout si la sécurité des personnes est en jeu ... Cela en partie explique le décollage difficile des logiciels libres et de Linux pour l'embarqué.*

# A retenir

- ❑ Quel avenir aux systèmes embarqués ?

Un avenir radieux.

- ❑ Quelles sont les évolutions techniques à venir ?

Deux pistes semblent se dessiner :

- ❑ Un couplage fort entre matériel et logiciel via le développement conjoint matériel/logiciel ou codesign et l'approche système sur silicium SoC (System on Chip).
- ❑ L'explosion de l'électronique/informatique ambiante (ubiquitous computing) bon marché couplée à l'Internet ambiant.

Les réseaux de capteurs (sans fil) seront omniprésents notamment en domotique : maison intelligente, ville intelligente, ...



**Thanks!**