Les systèmes embarqués
Introduction

Richard Grisel – Professeur des Universités – Université de Rouen
Nacer Abouchi – Professeur – ESCPE Lyon

Introduction aux systèmes embarqués

- Définition.
- Caractéristiques d’un système embarqué.
- Notions de codesign.
- Les contraintes dans la conception des systèmes embarqués.
- Méthodologie de conception.

Définition (1)

- “Embedded system”: tout système conçu pour résoudre un problème ou une tâche spécifique mais n’est pas un ordinateur d’usage général.
- Utilisent généralement un microprocesseur combiné avec d’autres matériel et logiciel pour résoudre un problème de calcul spécifique.
- Système électronique et informatique autonome ne possédant pas des entrées-sorties standards.
- Le système matériel et l’application sont intimement liés et noyés dans le matériel et ne sont pas discernables comme dans un environnement de travail classique de type PC.

Définition (2)

- N'est pas visible en tant que tel, mais est intégré dans un équipement doté d'une autre fonction; on dit aussi que le système est enfoui, ce qui traduit plus fidèlement le terme anglais «embedded».
- Une faible barrière existe entre les systèmes embarqués et les systèmes temps réels (un logiciel embarqué n'a pas forcément de contraintes temps réel).
- La conception de ces systèmes est fiable (avions, système de freinage ABS) à cause de leur utilisations dans des domaines à fortes contraintes mais également parce que l'accès au logiciel est souvent difficile une fois le système fabriqué.

Définition (3)

- Les microprocesseurs s'étendent depuis de simples microcontrôleurs 8 bits aux 64-bit le plus rapidement et les plus sophistiqués.
- Le logiciel système inclus s'étend d'un petit directeur à un grand logiciel d'exploitation en temps réel (RTOS) avec une interface utilisateur graphique (GUI). Typiquement, le logiciel système inclus doit répondre aux événements d'une manière déterministe et devrait toujours être opérationnel.
- Les systèmes embarqués couvrent aussi bien les commandes de navigation et de commande de trafic aérien qu'un simple agenda électronique de poche.

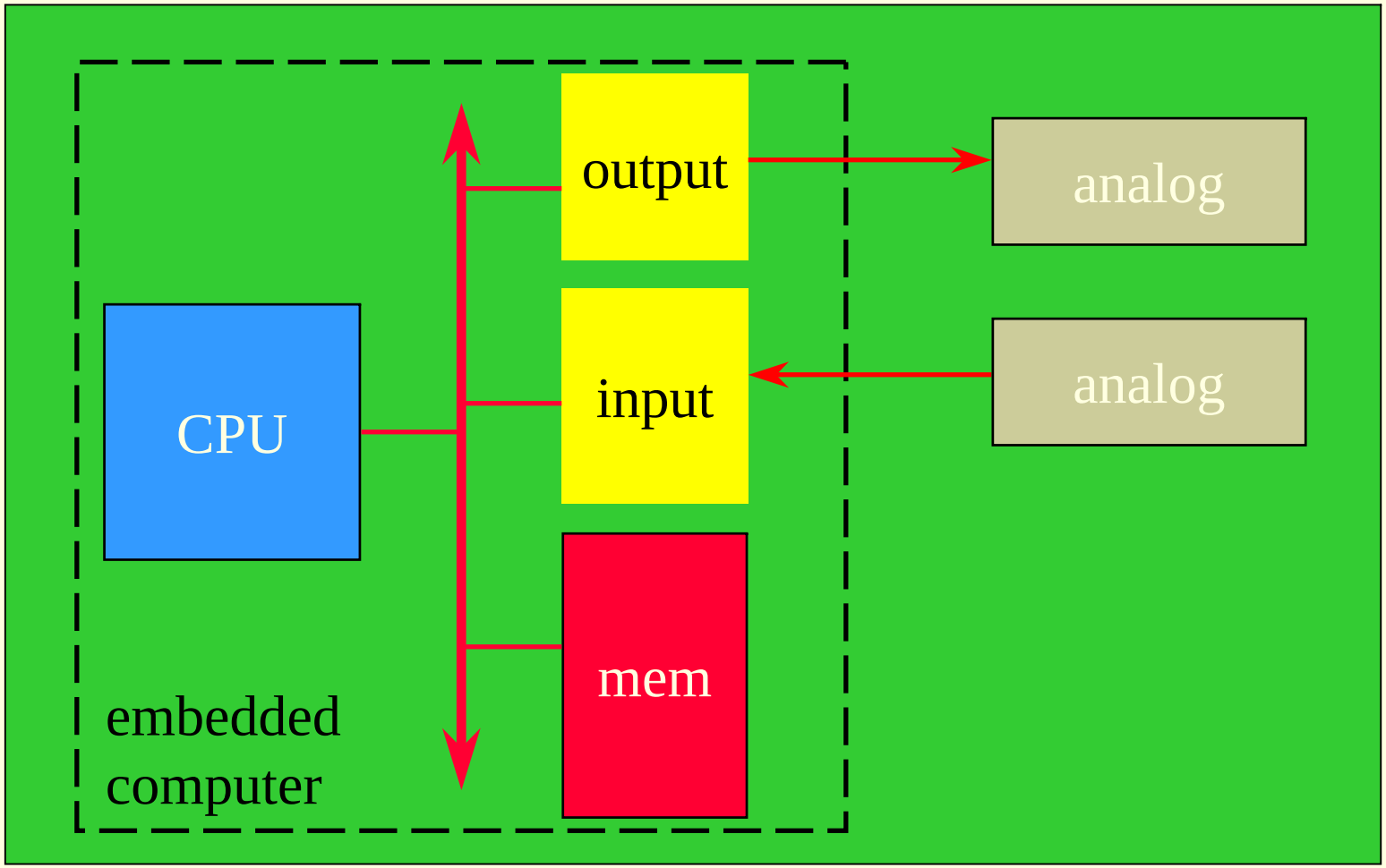
Les types de systèmes embarqués

- **Calcul général :**
  - Jeu vidéo.
- **Contrôle de systèmes en Temps Réel :**
  - Système de navigation aérien.
- **Traitement du signal :**
  - Radar, Sonar,
- **Transmission d'information et commutation :**
  - Téléphone, internent.

Quelques exemples

- **Équipement mobile et bureautiques :**
  - Répondeurs,
  - Copieurs,
  - Téléphone portable,
  - Imprimante.
- **Équipement dans le bâtiment :**
  - Ascenseurs, escalators,
  - Système de surveillance,
  - Contrôle d’accès,
  - Systèmes d'éclairage.

“Embarquement” d’un ordinateur



Quelques exemples

- **Équipement de production :**
  - Productions automatisées,
  - Systèmes de commande d'énergie,
  - équipements de stockage,
- **Transport :**
  - Avionique,
  - Trains, Automobiles (+ de 100 processeurs),
  - Contrôle de navigation,
- **Communications :**
  - Satellites,
  - GPS,
  - Téléphonie mobile,

Historique (récent)

- Fin des années 1940: Le processeur Whirlwind du MIT est conçu pour des applications temps réel
- A l'origine pour contrôler un simulateur de vol.
- Le premier microprocesseur est l'Intel 4004 au début des années 1970.
- La calculatrice HP-35 utilise plusieurs circuits pour implémenter un microprocesseur en 1972.

Historique (suite).

- Les automobiles utilisent des systèmes de contrôle du moteur avec microprocesseurs depuis les années 1970.
  - Contrôle du mélange fuel/air, gestion du moteur, etc.
  - Modes de fonctionnement multiples: démarrage, croisière, montées, etc.
  - Baisse des émissions polluantes, consommation optimisée.

Caractéristiques d'un système embarqué (1)

- **Fonctionnement en Temps Réel :**
  - Réactivité : des opérations de calcul doivent être faites en réponse à un événement extérieur (**interruption matérielle**).
  - La validité d'un résultat dépend du moment où il est délivré, (deadlines).
  - Rater une échéance peut causer une erreur de fonctionnement.
  - La plus part des systèmes sont «multirate » : traitement d'informations à différents rythmes.

Caractéristiques d'un système embarqué (2)

- **Faible encombrement, poids et consommation :**
  - Consommation électrique minimisée,
  - Difficulté de packaging (analogique, numérique et RF),
  - Batterie de 8 heures et plus (PC portable : 2 heures).
  - Environnement sévère (Température, vibrations, variations d'alimentation, interférences RF, corrosion, eau, feu, radiations, etc.),

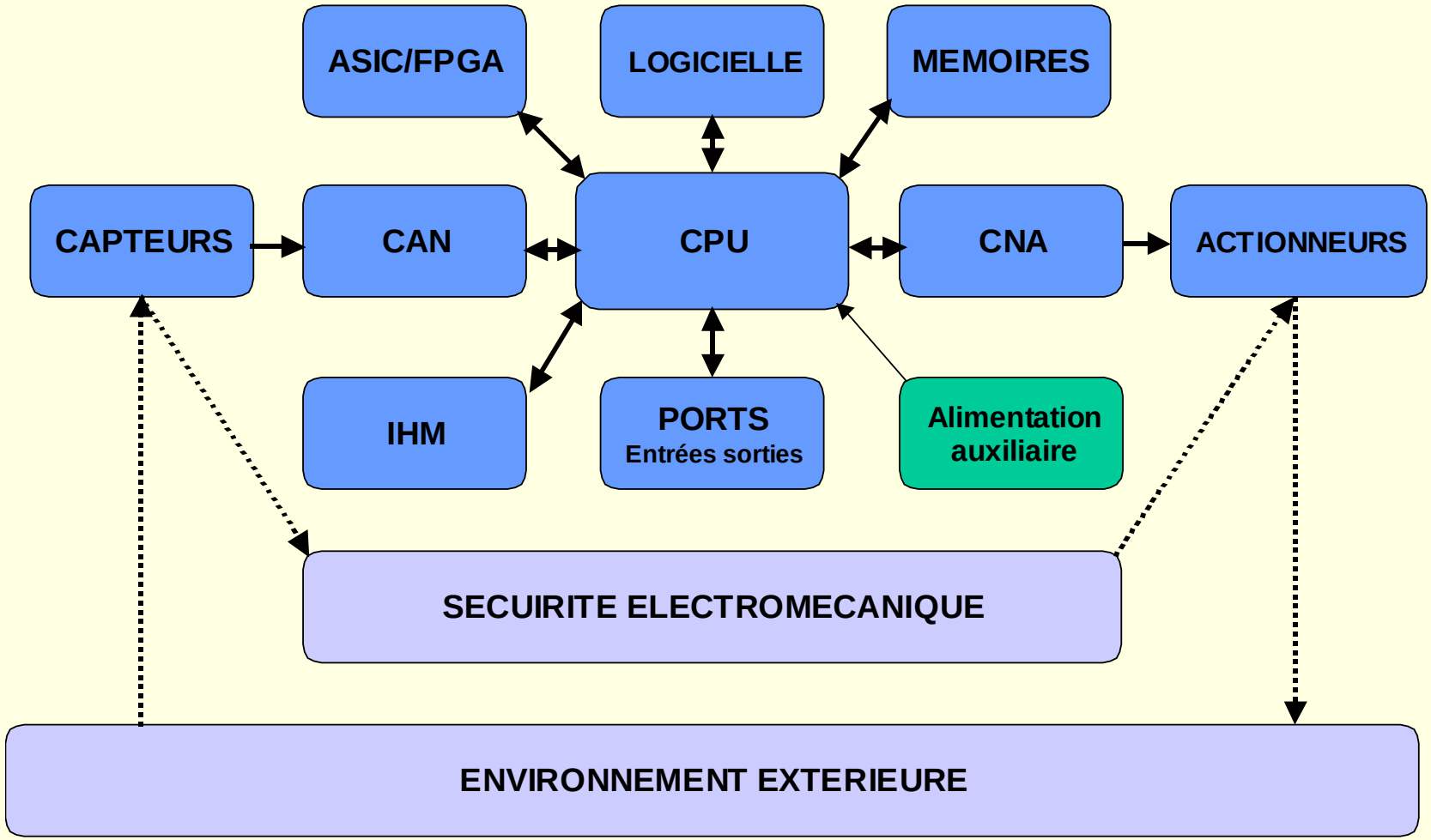
Le système n'évolue pas dans un environnement contrôlé (évolutions des caractéristiques des composants en fonction de l'environnement).

Caractéristiques d'un système embarqué (3)

- **Coût, sûreté et sécurité :**
  - Le système doit toujours fonctionner correctement (faible coût et une redondance minimale),
  - Sûreté de fonctionnement du logiciel (système opérationnel même quand un composant électronique « lâche ».

Beaucoup de systèmes embarqués sont fabriqués en grande série et doivent avoir des prix de revient extrêmement faibles

Système embarqué typique



Les systèmes embarqués et le temps réel

- Un système embarqué doit généralement respecter des contraintes temporelles fortes (Hard Real Time).
- On y trouve enfoui un système d'exploitation ou un noyau Temps Réel (Real Time Operating System, RTOS).
- "Un système est dit Temps Réel lorsque l'information après acquisition et traitement reste encore pertinente". Cela veut dire que dans le cas d'une information arrivant de façon périodique (interruption), les temps d'acquisition et de traitement doivent rester inférieurs à la période de rafraîchissement de cette information.

Ne pas mélanger Temps Réel et rapidité de calcul du système donc puissance du processeur

Le fonctionnement temps réel

- Les opérations doivent être faites avec des échéances (deadlines) précises.
  - “Hard real time”: le manquement des échéances provoque une faute
  - “Soft real time”: le manquement des échéances cause des dégradations de performances.
- La plupart des systèmes sont “multi-rate” : Les opérations doivent être gérées à des vitesses (très) différentes.

Les systèmes embarqués et les RTOS

- La question d'utiliser un système d'exploitation Temps Réel ou non ne se pose pratiquement plus pour les raisons suivantes :
  - Simplifications de l'écriture de l'application embarquée.
  - Portabilité.
  - Évolutivité.
  - Maîtrise des coûts.
  - Le système d'exploitation peut être même « maison » : encore dans 50 % des cas !
- **Quelques exemples :**
  - Linux, RTLinux, Windows CE, µCOS, PALM OS

Spécifications non fonctionnelles

- Beaucoup de systèmes sont des systèmes à production de masse qui doivent avoir des coût de fabrication bas
  - Mémoire limitée, puissance du microprocesseur, etc.
- La consommation est un facteur critique pour les systèmes fonctionnant sur piles (ou batteries).
  - Une consommation excessive augmente le coût même en cas d'alimentation par le secteur.

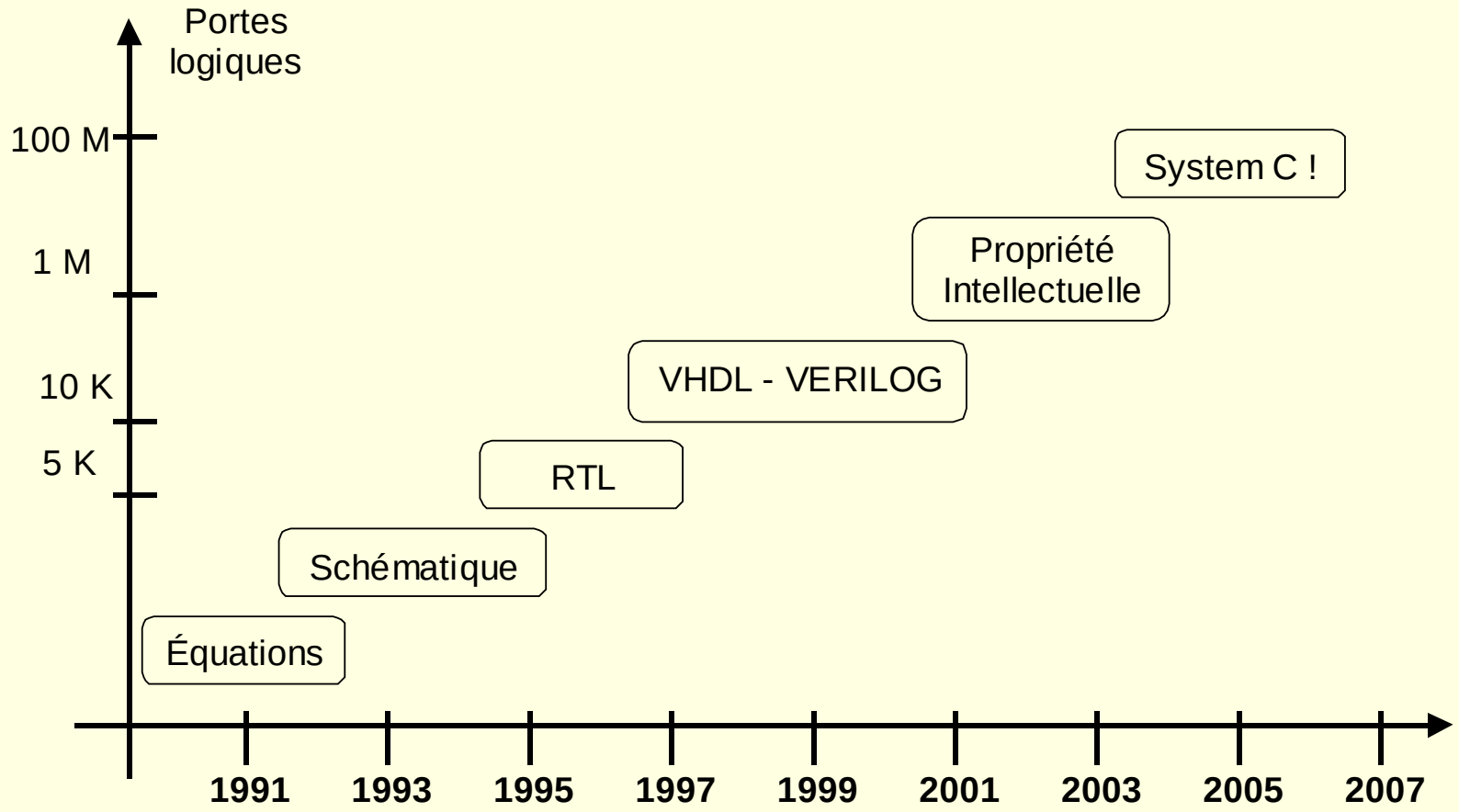
Equipes de conception

- Nombre d’ingénieurs faible.
- Les échéances sont contraintes.
  - 6 mois pour la mise sur le marché est un délai classique.
  - Des échéances sont contraintes (rentrée des classes, concurrence, etc..)

Lien entre le matériel et le logiciel « le codesign »

- Objectif : intégrer un système dans un même composant (*single chi*p). On parle aussi de système sur silicium SoC *(System on Chi*p) ou SoPC *(System on Programmable Chi*p), (loi empirique de Moore).
- Langages de description du matériel pour synthétiser et tester les circuits numériques. On a ainsi une approche logicielle pour concevoir du matériel.
- Les systèmes numériques se sont complexifiés et la mise sur le marché est plus rapide ! (notion de *Design Reuse* et d'*Intellectual Property*).

Evolution de la conception

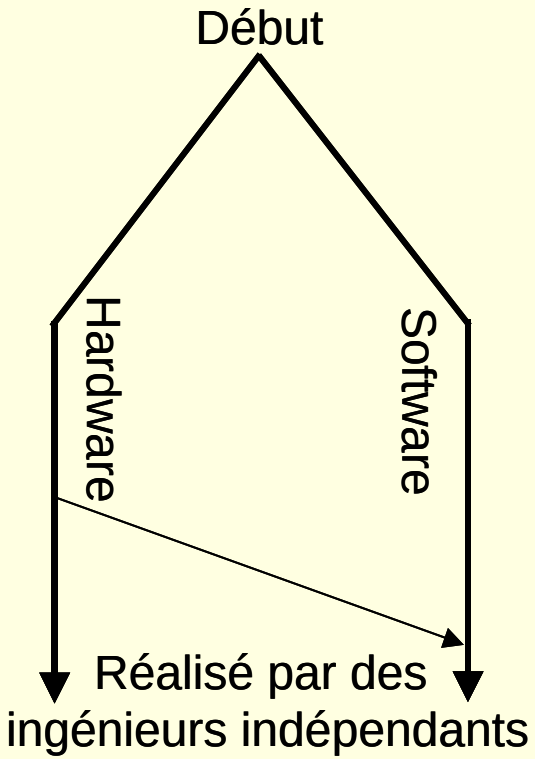


Codesign hardware / software

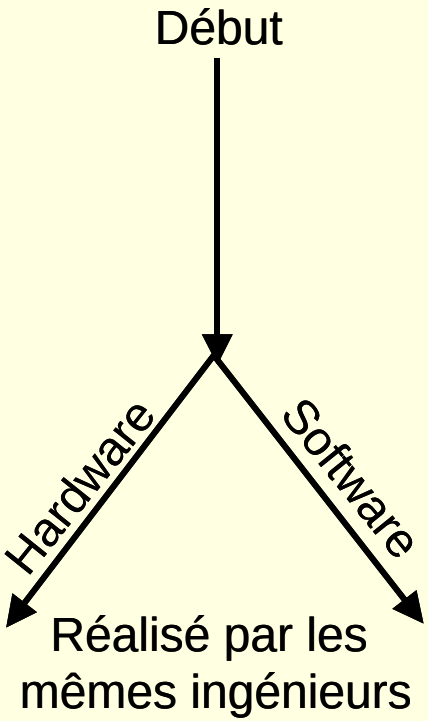
- Le codesign dans la méthodologie de conception d'un système embarqué est de plus en plus utilisé.
- Le codesign permet de concevoir en même temps à la fois le matériel et le logiciel pour une fonctionnalité à implémenter. Cela est maintenant possible avec les niveaux d'intégration offerts dans les circuits logiques programmables.
- Le codesign permet de repousser le plus loin possible dans la conception du système les choix matériels à faire contrairement à l'approche classique où les choix matériels sont faits en premier lieu !

Conception et codesign

Conception traditionnelle



Codesign (flot concurrent)



Les étapes dans le Codesign

- **Spécifications** : liste des fonctionnalités du système de façon abstraite.
- **Modélisation** : conceptualisation et affinement des spécifications produisant un modèle du matériel et du logiciel.
- **Partitionnement** : partage logiciel matériel.
- **Synthèse et optimisation** : synthèse matérielle et compilation logicielle.
- **Validation** : co-simulation.
- **Intégration** : rassemblement des différents modules.
- **Tests d'intégration** : vérification du fonctionnement.

Avantages du codesign

- Amélioration des performances : parallélisme, algorithmes distribués, architecture spécialisée, etc.
- **Reconfiguration** statique ou dynamique en cours de fonctionnement.
- Indépendance vis à vis des évolutions technologiques des circuits logiques programmables.
- Mise à profit des améliorations des outils de conception fournis par les fabricants de circuits logiques.
- programmables : synthèse plus efficace, performance accrue.

Systèmes embarqués et microprocesseurs (1)

- Microprocesseur.
- Microcontrôleur : contient les I/O, la mémoire.
- Digital signal processor (DSP) : microprocesseur optimisé pour le traitement du signal.
- Taille des données dans les systèmes embarqués : 8-bit, 16-bit, 32-bit.

Alternatives: “Field-programmable gate arrays” (FPGAs), logique spécialisée, etc.

Les microprocesseurs sont très efficaces : la même logique permet de faire une multitude de fonctions ce qui simplifient la conception de familles de produits.

Microprocesseur et logique « sur-mesure

- Les microprocesseurs utilisent plus de logique pour implémenter une fonction que l'équivalent en logique "sur-mesure". Mais ils sont souvent au moins aussi rapides :
  - Pipelines à plusieurs étages.
  - Conception optimisée.
  - Technologie VLSI récente.
- La logique "sur-mesure" est adaptée aux systèmes basse consommation.
- Les microprocesseurs ont des possibilités de contrôle de la consommation (mise en sommeil de certaines parties).
- Les techniques de conception logicielle peuvent aider à réduire la consommation.

Microcontrôleur et DSP

Type de processeur	Besoins	Ressources nécessaires	avantages
Microcontrôleur	Contrôle d'entrées sorties	Ports d'entrées sorties « bits »	Pilotage direct d'actionneurs, etc.
	Communications séries avec l'extérieur.	Port série : SPI, I²C, UART, CAN, Microware, etc.	facilite l'interfaçage avec l'extérieur via le réseau, etc.
	Contrôle précis de moteur et actionneurs.	Périphériques spéciaux : compteurs temporisateurs, générateurs de PWM, etc.	Programmation facile.
	Rapidité d'exécution de fonction complexe de contrôle.	Sauts conditionnels, instructions de test au niveau bit, interruption, etc.	Facilité de mise en œuvre au niveau de la programmation des taches.
	Rapidité de prise en compte d'évènements externes.	Interruption externes avec plusieurs niveaux de priorités.	Facilité de mise en œuvre logicielle.
	Entrées analogiques	Convertisseur analogique numérique	Prise en compte matérielle de capteurs externes.
DSP	Filtres numériques	Multiplieur, etc.	Temps de calcul réduit.
	Interface pour Codeur-Decodeur	Port série rapide	Prise en compte de signaux analogiques.
	Quantité importante de données en entrée.	Contrôleur de DMA	Temps de transfert minimisé
	Accès rapide aux données.	Architecture spécifique.	Accélération de l'exécution.

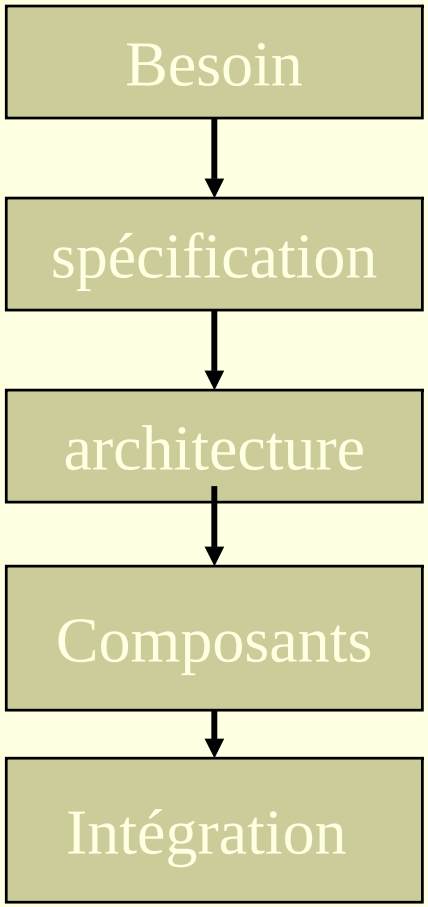
But de la conception (systèmes embarqués)

- Performances : vitesse, échéances fonctionnelles.
- Fonctionnalités et interface utilisateur.
- Coût de fabrication.
- Consommation.
- Autres caractéristiques du cahier des charges (taille du boîtier, etc.)

Methodologies de conception

- Une procédure est nécessaire pour la conception.
- La compréhension de la méthodologie assure que l'on ne va pas passer "à coté" de paramètres importants.
- Les compilateurs, les outils d'aide à la conception logicielle, les outils de CAO, …, peuvent être utiliser pour :
  - automatiser les étapes méthodologiques,
  - surveiller et "tracer" la méthodologie elle-même.

Niveaux d’abstraction



A chaque niveau d’abstraction, on doit analyser le système pour déterminer ses caractéristiques actuelles et l’améliorer pour prendre en compte les détails manquants.

Top-down ou bottom-up

- “Top-down” :
  - on part du plus haut niveau d’abstraction;
  - on “descend” vers le plus détaillé.
- “Bottom-up” :
  - on part des composants de base et on “remonte” vers le système.

Une conception réaliste utilise les deux techniques

Challenges en conception de systèmes embarqués

- De quoi avons nous besoin en terme de HW ?
  - Taille du CPU, taille de la mémoire.
- Comment respecter les délais ?
  - Hardware rapide ou Software intelligent.
- Comment minimiser la consommation ?
  - Mise en sommeil de la logique non utilisée, réduction des cycles d'accès à la mémoire.
- Est ce que cela fonctionne ?
  - Les spécifications sont elles correctes ?
  - Est ce que la réalisation couvre les spécifications ?
  - Comment teste-t-on les caractéristiques temps réel ?
  - Comment teste-t-on en vrai grandeur (données réelles) ?
- Comment travaille ton sur le système ?
  - Observabilité, contrôlabilité, quelle plateforme de développement utiliser ?

Expressions des besoins

- Une description précise de ce que veut l'utilisateur (client) et de ce qu'il espère obtenir

- Besoins fonctionnels :
  - Sorties en fonction des entrées et des paramètres.

- Besoins non fonctionnels :
  - temps nécessaire pour calculer la sortie,
  - taille, poids, etc.,
  - consommation,
  - fiabilité,
  - etc.

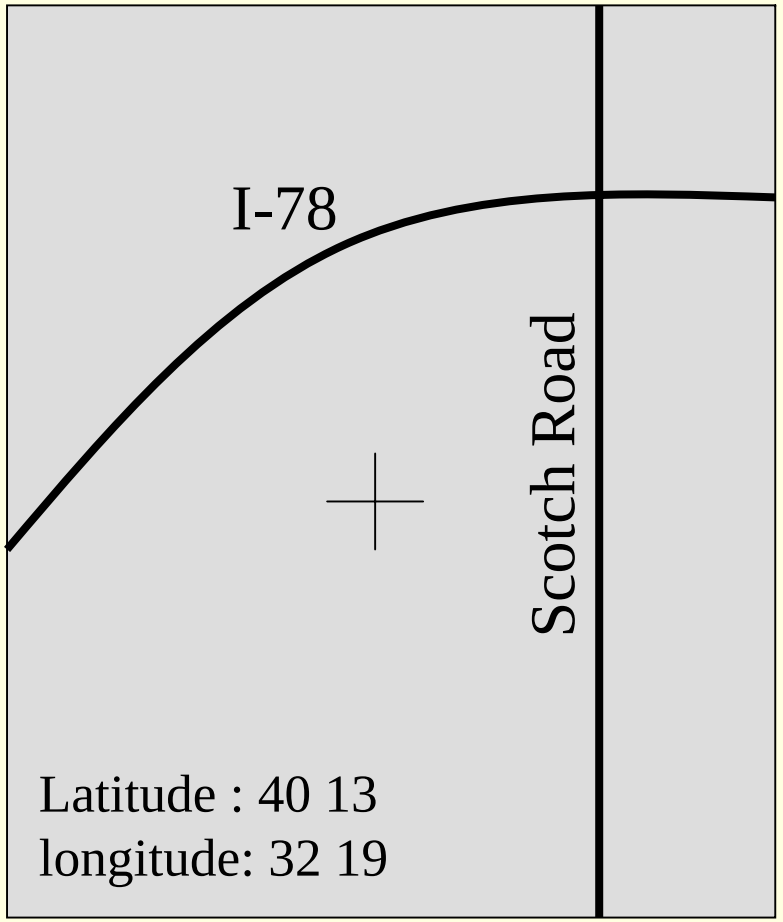
Comprendre le besoin du client et savoir aussi l'identifier !

Modèle de fonctionnalités

- Nom :
- Objectifs :
- Entrées :
- Sorties :
- Fonctions :
- Performances :
- Coût de fabrication :
- Consommation :
- Taille :
- Poids :

Exemple : Système GPS

- La carte obtient la position du GPS, la base de données est locale pour la carte.



Besoins pour le système GPS

- Fonctionnalité : Pour l'automobile, il faut montrer les axes principaux et les repères.
- Interface utilisateur :
  - Au moins 400 x 600 pixels pour l'écran.
  - 3 boutons maximum.
  - Menus déroulants.
- Performances : La carte doit être balayée "doucement", pas plus de 1 seconde à la mise sous tension, calage sur le GPS en moins de 15 secondes.
- Coût : prix de vente de 500 € (approximativement). 100 € de coût pour les fournitures.
- Taille/poids : Doit tenir dans la main.
- Consommation : Doit fonctionner 8 heures avec 4 piles type AA.

Modèle pour le système GPS

nom	Carte GPS
Objectifs	Carte routière GPS pour conducteur
Entrées	1 bouton on/off, 2 de contrôle
Sorties	LCD 400 X 600 rétro éclairé
Fonctions	récepteur GPS; 3 résolutions; affichage lat/lon
Performances	Mise à jour de l'écran en 0.25 sec
Coût de fabrication	€100 (fournitures)
Consommatoin	100 mW
Taille	5cm x 12 cm
Poids	100 g

Spécifications

- Une description plus précise du système :
  - ne doit pas identifier une architecture particulière,
  - donne les entrées au dispositif de conception de l'architecture.
- Peut comprendre des éléments fonctionnels et non fonctionnels.
- Peut être exécutable ou sous une forme mathématique pour preuve formelle.

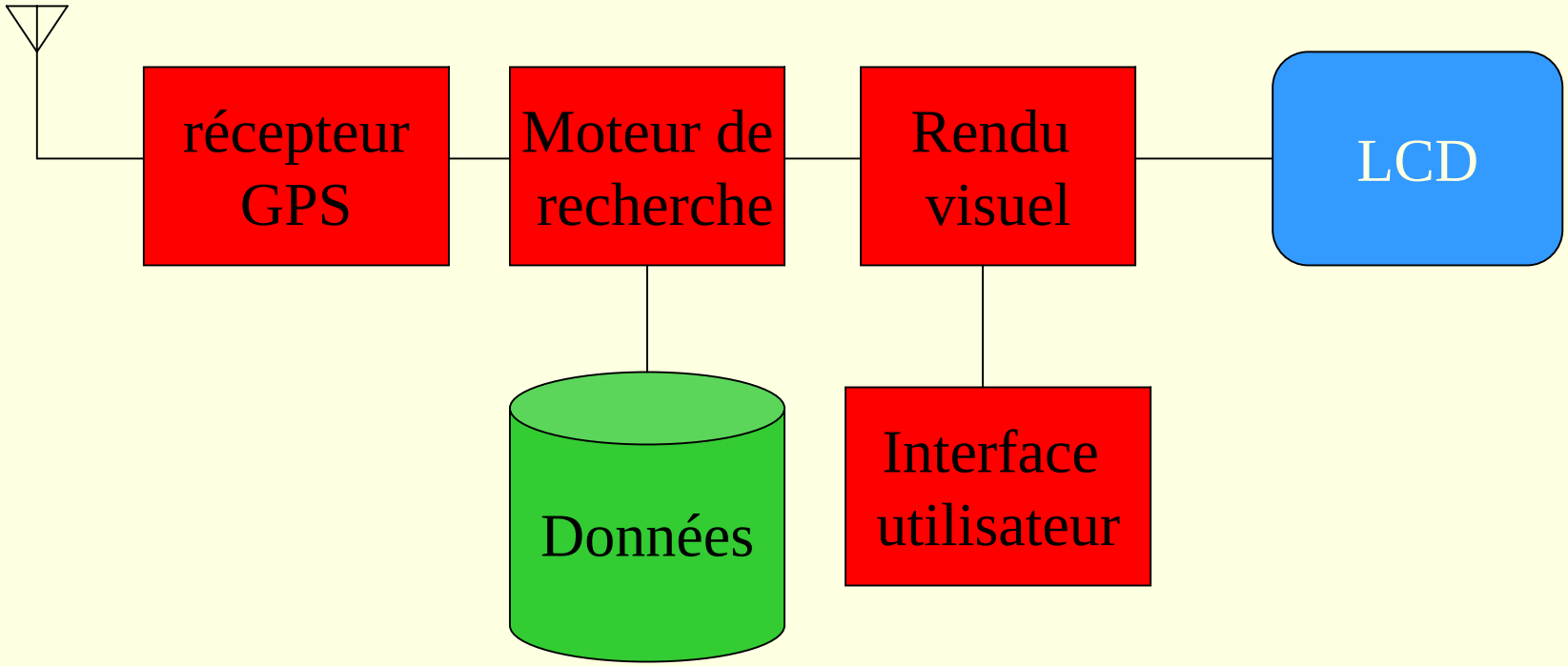
Spécifications pour le GPS

- Doit comprendre :
  - Ce qui est reçu du GPS.
  - Les données de la carte.
  - L’interface utilisateur.
  - Les opérations nécessaires pour satisfaire à la demande du client.
  - Les opérations d’arrière plan permettant au système de continuer fonctionner.

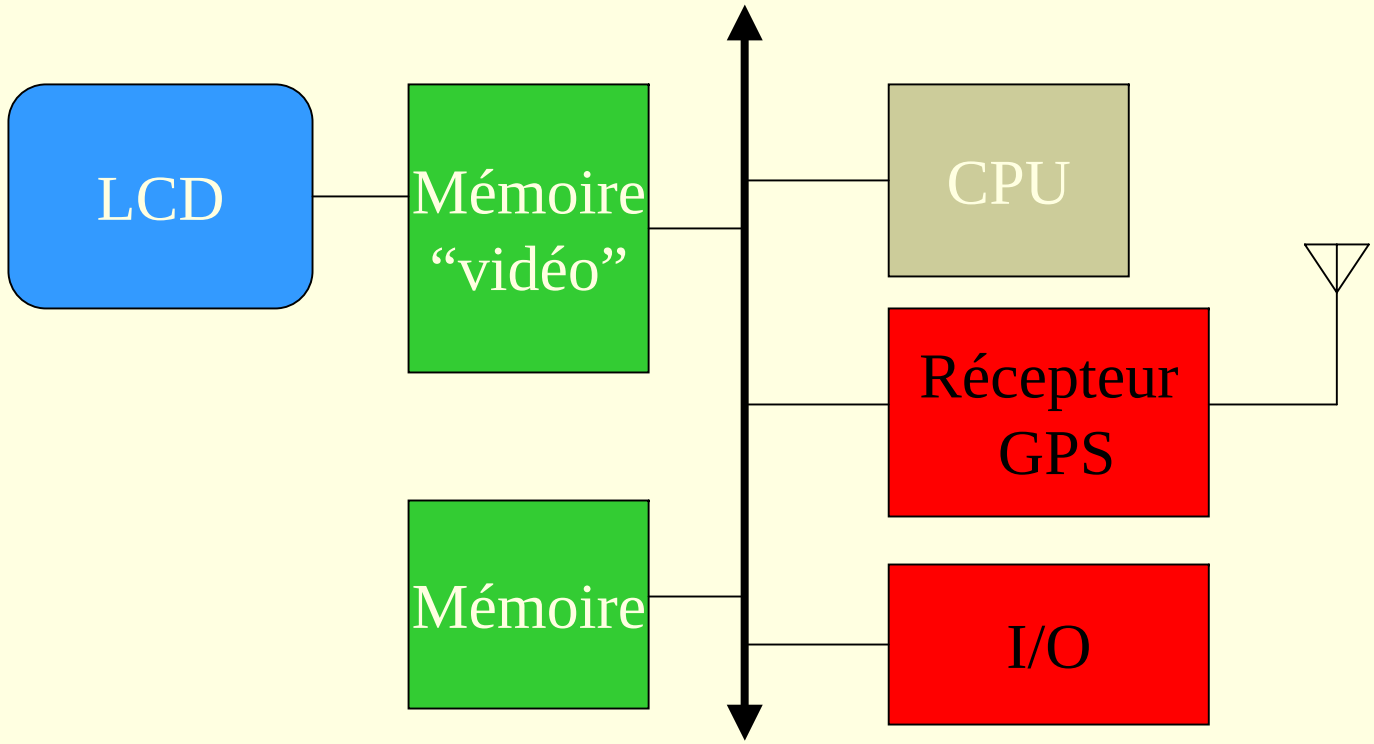
Conception de l’architecture

- Quels sont les composants qui satisfont aux spécifications majeures ?
- Composants matériels :
  - CPUs, périphériques, etc.
- Composants logiciels :
  - Programmes principaux et leurs opérations.
- Doit prendre en compte les spécifications fonctionnelles et non fonctionnelles.

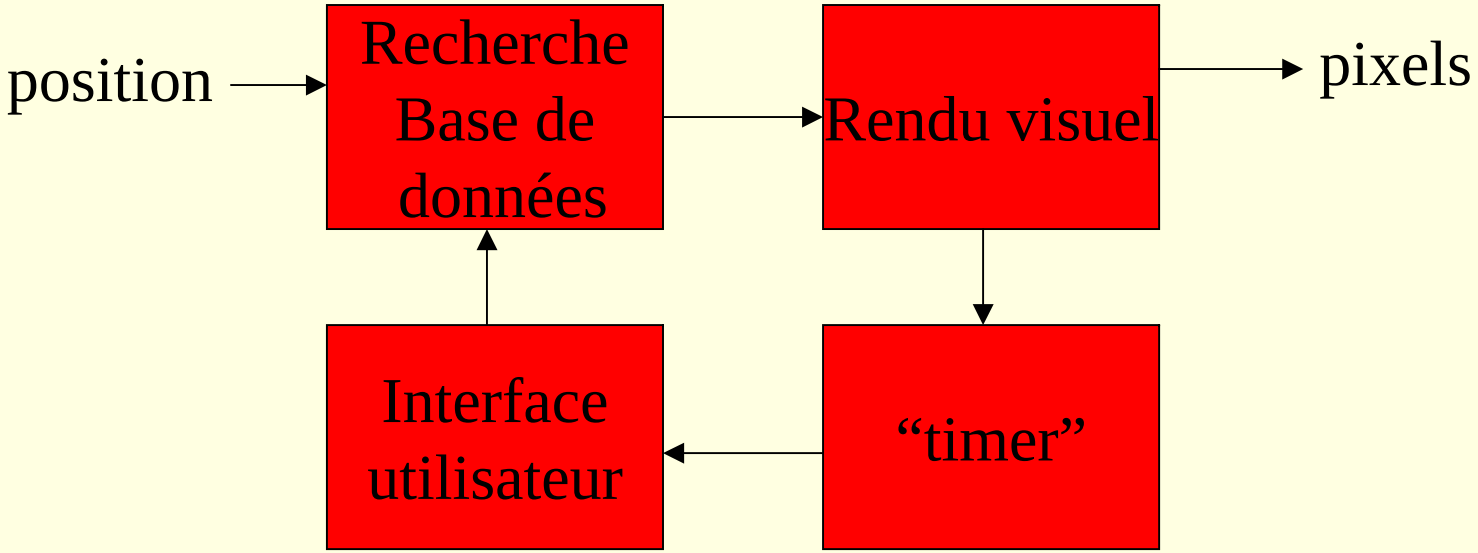
Schéma bloc du système GPS



Architecture matérielle du GPS



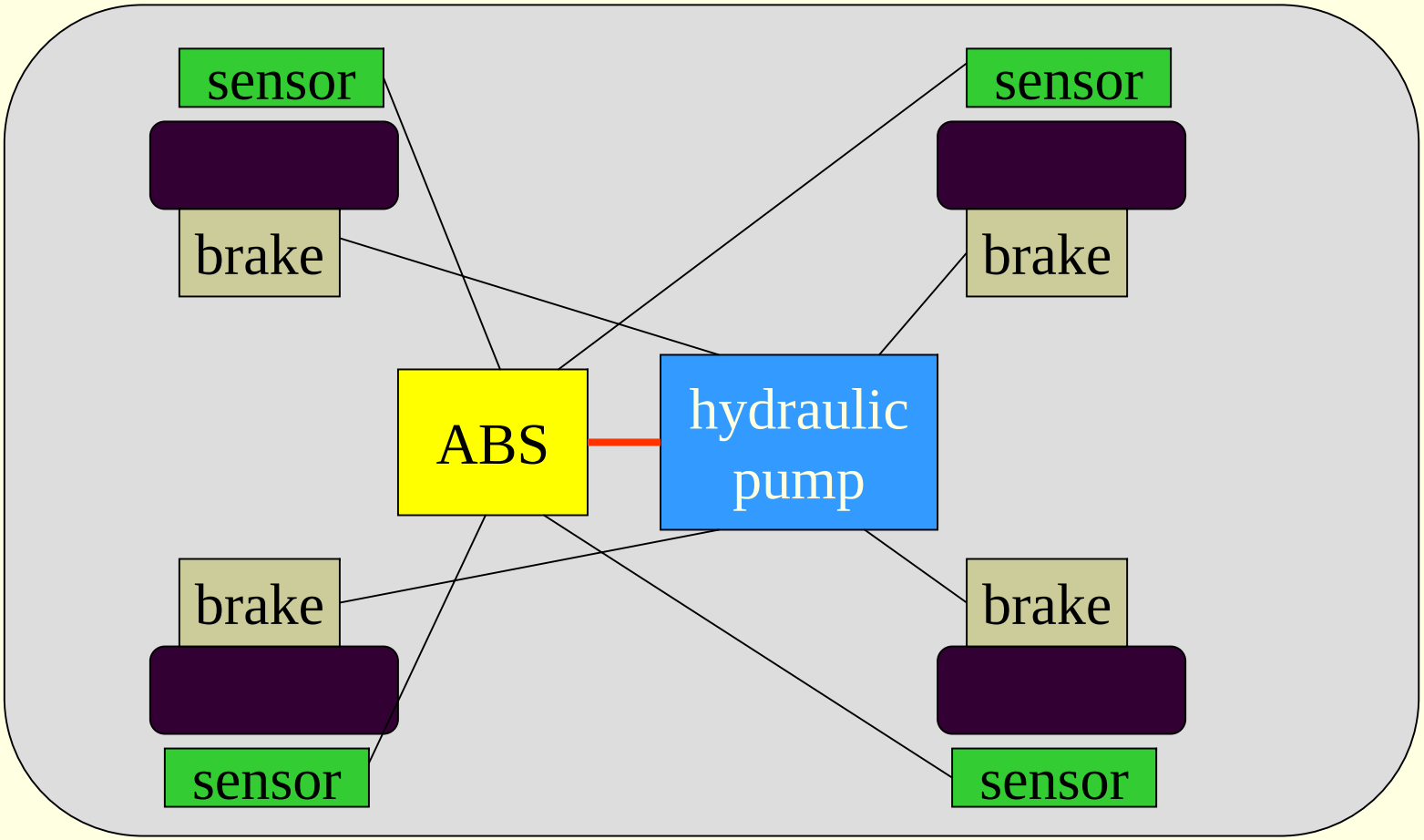
Architecture logicielle du GPS



Exemple du système de contrôle de freinage et de stabilité de la BMW 850i

- Anti-lock Brake System (ABS)
- Automatic stability control (ASC+T): contrôle de la stabilité
- L'ABS et l'ASC+T communiquent.
  - L'ABS a été introduit en premier ce qui a nécessité de s'interfacer avec le module ABS existant.

BMW 850i, (suite)



Conception des composants matériel et logiciel

- Il faut passer beaucoup de temps de réflexion avant de commencer à coder.
- Quelques composants existent, certains peuvent être modifiés à partir d'autres systèmes, d'autres devront être créés.

Intégration du système

- Rassembler les composants.

Beaucoup de “bugs” à cette étape !

- Avoir un plan d’intégration des composants pour couvrir les “bugs” rapidement, tester le plus de fonctionnalités le plus tôt possible.

Résumé

- Nous sommes entourés de systèmes embarqués.

La complexité embarquée se situe au niveau matériel et au niveau logiciel.

- Les systèmes embarqués posent de nombreuses contraintes en terme de conception :

temps de conception, échéances, consommation, encombrement, autonomie, etc.

- **Les méthodologies de conception aident à gérer le processus de conception.**

Les compétences à avoirs

- microprocesseur, microcontrôleur, DSP, mémoires, IO,
- FPGA, VDHL,
- Codesign,
- Programmation en assembleur,
- Programmation C, C++, java,
- Systèmes d'exploitations, linux, RTOS,
- réseau,
- bus.

Connaissance des systèmes numériques.

Travailler en équipe avec des ingénieurs d'autres disciplines.