

Du microcontrôleur au microprocesseur

Quelle architecture pour quel projet ?



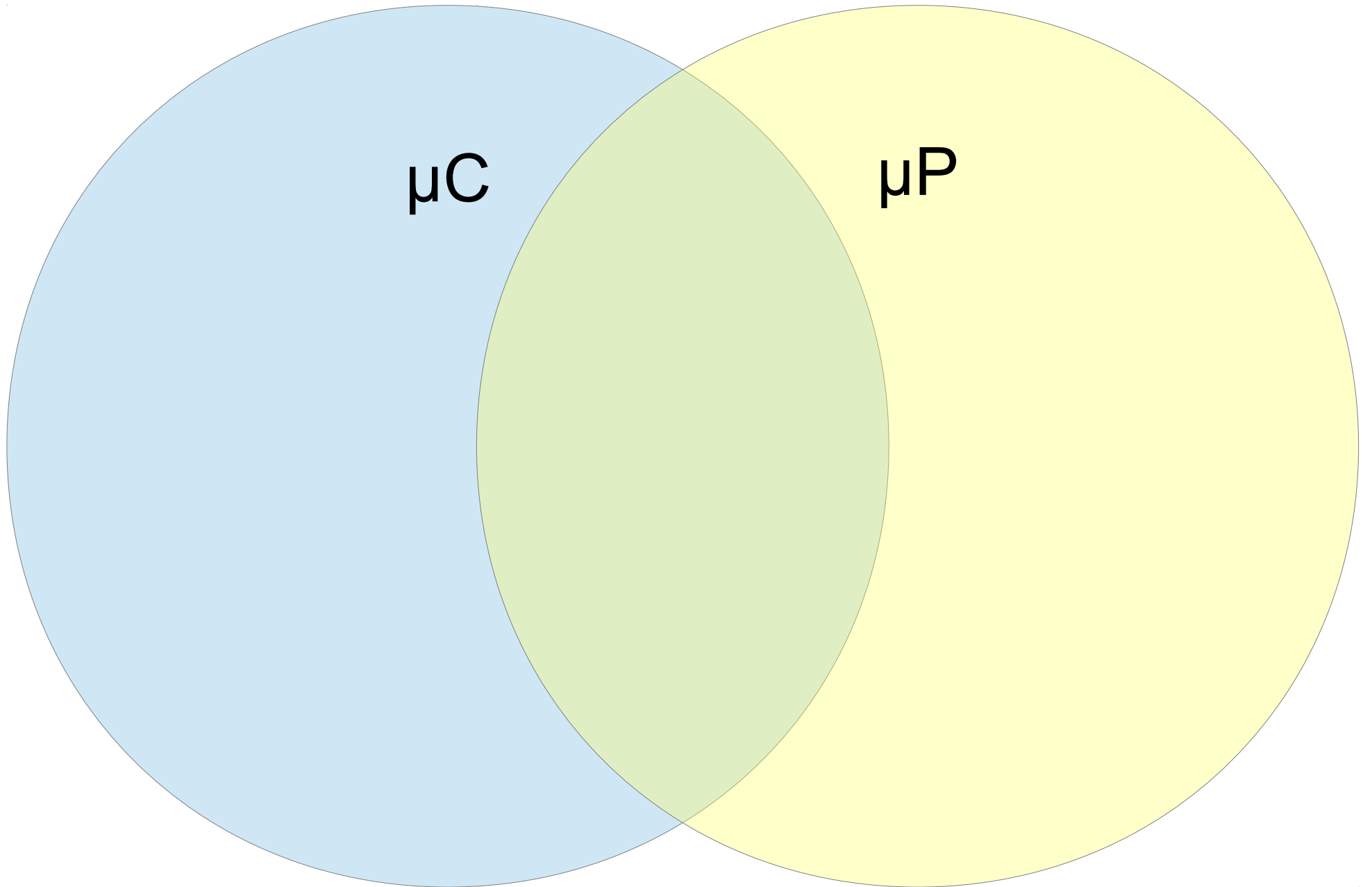
François Beaulier Ingénieur freelance

www.ingelibre.fr

- Conseil architecture hard / soft
- Développement logiciel Linux RT / drivers / microcontrôleurs
- Etude cartes électroniques / systèmes embarqués
- Linux embarqué et logiciels libres
- Expertise CAN / CanOpen, stack canfestival

- 20 ans bureau d'étude en PME dans l'industrie
- Microcontrôleurs 8bits 32 bits logiciel / cartes
- Linux embarqué depuis ~2000 sur PC puis 'computer on module' ARM

μC vs μP



μC vs μP

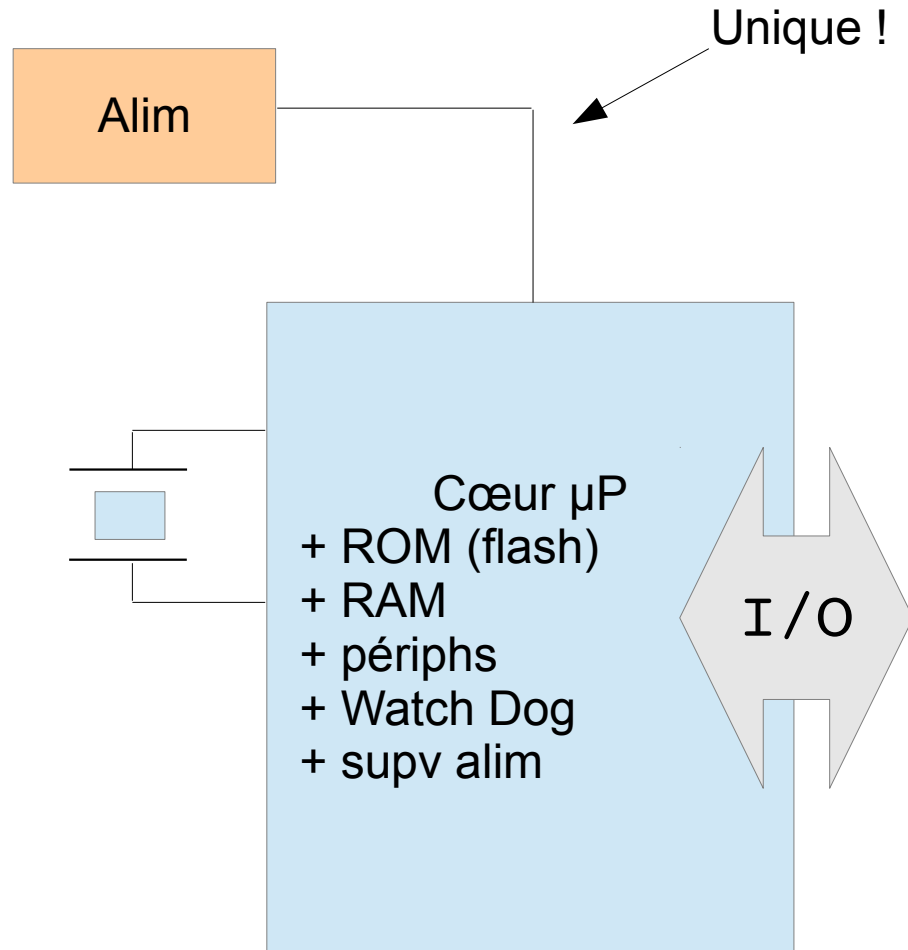
Choisir une architecture

« Culture »

Informaticiens

Electroniciciens

Microcontrôleurs



SRAM : 256ko max
Flash NOR: 2Mo max

Freq : 200Mhz max

OS : aucun ou RTOS

Graphique : ¼ VGA

Microcontrôleurs

- Traitement simple
- Solution faible coût
- Temps réel
- Déterminisme
- Consommation
- Encombrement
- Sécurité / sureté
- Large gamme
- Design carte simple

Evolution des designs

- Avant : Systèmes enfouis, beaucoup de périphs sur bus micro, IHM rudimentaire
- Après : de plus en plus de communication avec le monde extérieur, généralisation des interfaces série, IHM graphique
- Tendence : migration vers le 32 bits, RTOS (portabilité), écosystème logiciel

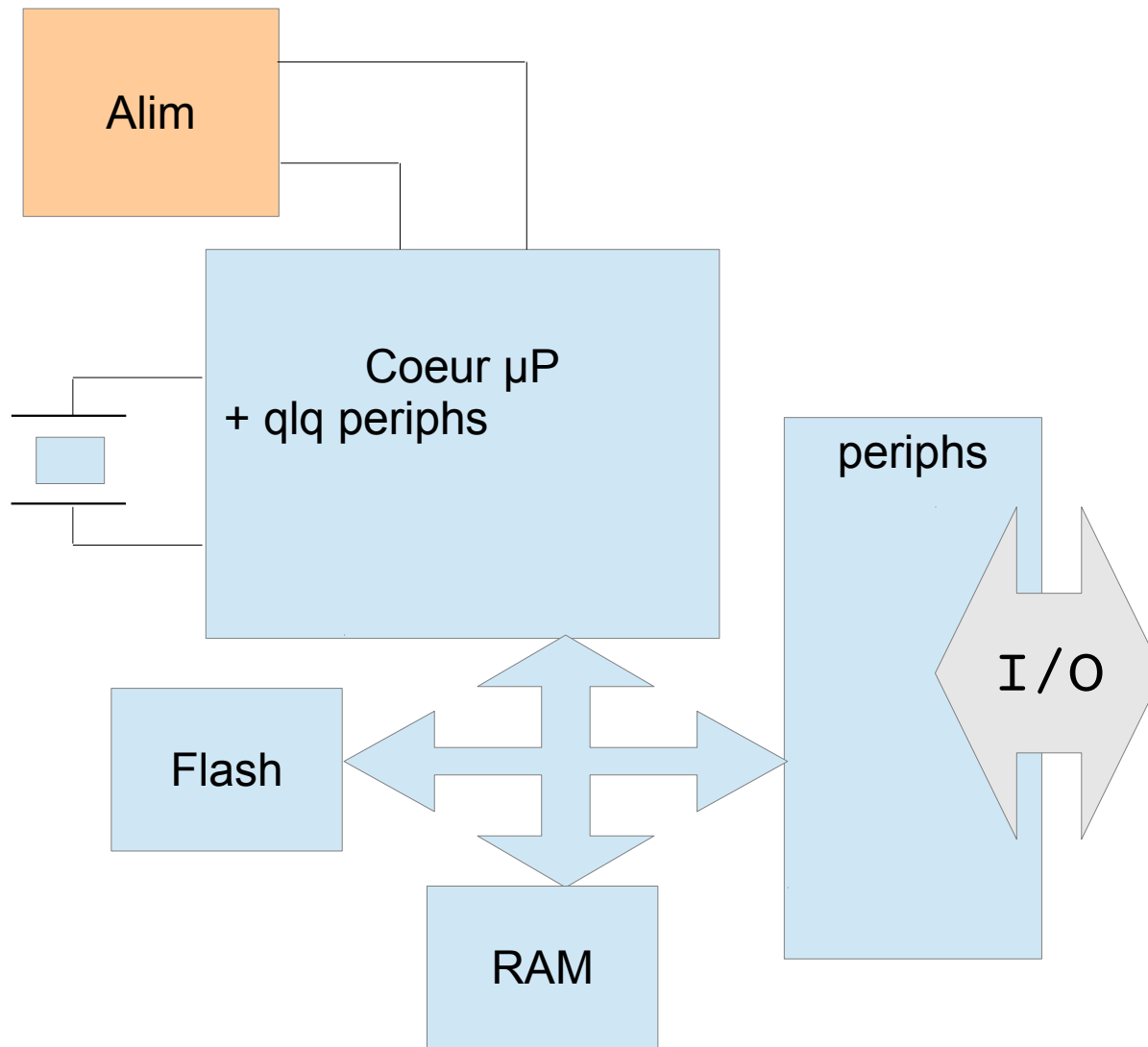
Des microcontrôleurs qui montent

- + de Flash, + de RAM et des fréquences plus hautes
- Des interfaces de communication toujours plus nombreuses
Ethernet, USB, CAN en plus des UARTs, SPI, I2C etc ...
- FPU, MPU, contrôleur mémoire (même SDRAM!)
- Contrôleurs graphiques avec accélération 2D pour des résolutions plus élevées
- Environnement logiciel plus riche

Microprocesseurs

- Traitements complexes
- Volume mémoire
- Communication avec l'univers bureautique
- Graphismes > QVGA
- OS Drivers fichiers
- Ecosystème langages

Microprocesseurs



SDRAM : 64Mo min
Flash NAND: 128Mo min

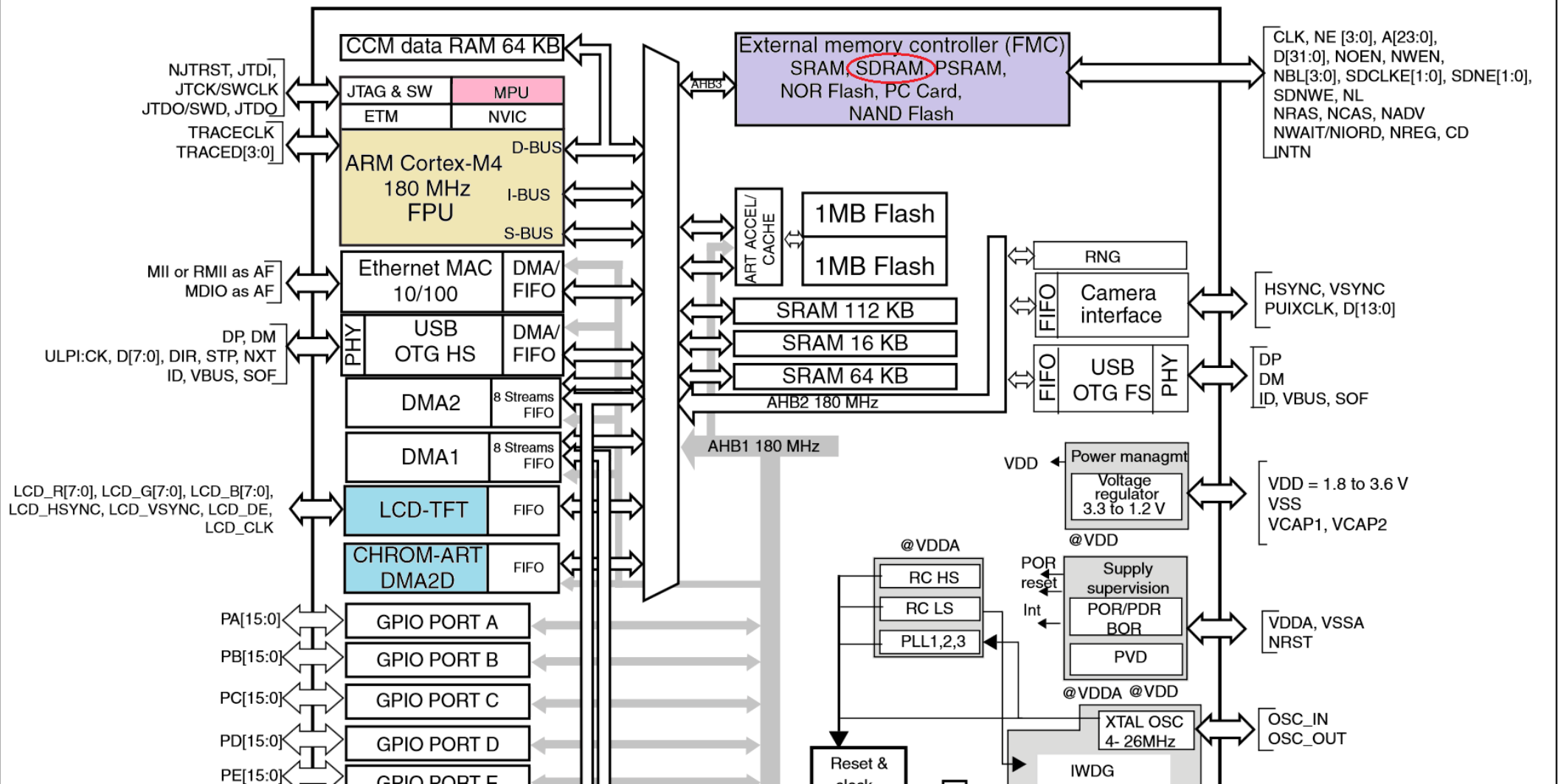
Freq 200 Mhz min

OS : Linux, Windows etc

Graphique : → HD

Des microcontrôleurs qui montent

Figure 4. STM32F427xx and STM32F429xx block diagram



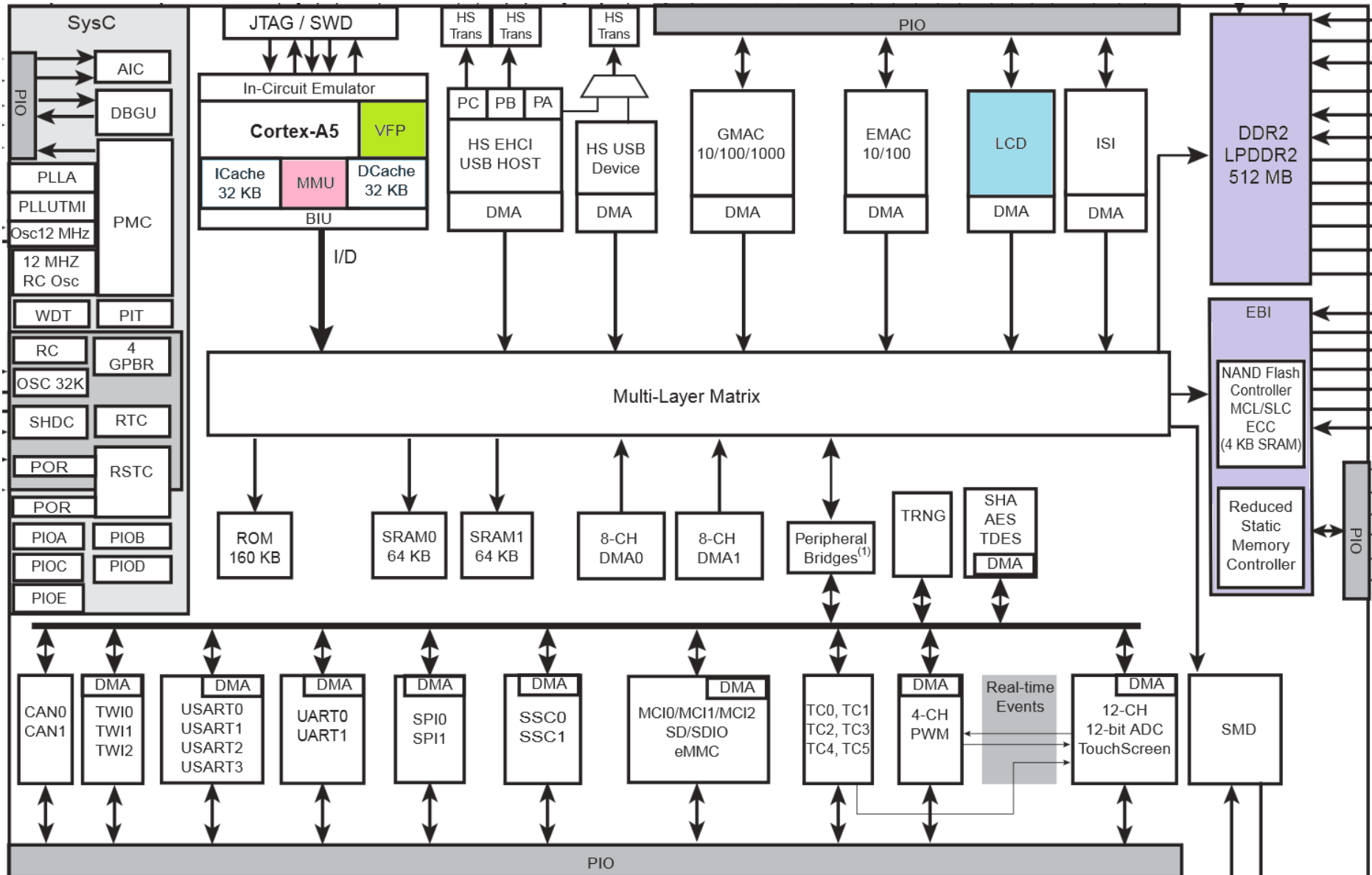
Et des microprocesseurs qui descendent

Les processeurs se déclinent en super microcontrôleurs appelés
'system on chip'

- Coût comparable à un μC
- Intègre des périphériques comme un μC : UART, entrées analog, CAN, timers PWM etc
- Nécessite toujours SDRAM et Flash externe
- Alimentation reste plus complexe (1.2V, 1.8V, 3.3V)
- Cœur processeur évolué pour OS Linux ou autre
- Quelle différence avec un microcontrôleur classique haut de gamme ?

Et des microprocesseurs qui descendent

Exemple ATMEL SAMA5D3



Microcontrôleur vs system on chip

Le système on chip est architecturé pour faire tourner des OS évolués, il intègre notamment une **MMU**

- La MMU est un composant indispensable pour faire tourner Linux, sur les μC on ne trouve qu'une MPU qui est beaucoup plus légère.
- En plus de la MMU un cœur de SOC est optimisé pour les OS.
- Le contrôleur SDRAM ne se rencontre que rarement sur un μC
- Un SOC intègre également une petite quantité de ROM et de RAM pour le bootloader qui doit pouvoir s'exécuter alors que le contrôleur de SDRAM n'est pas encore paramétré.

μC vs SOC : matériel

	μC	SOC
Coût moyen μC haut de gamme SOC entrée de gamme	~10€	~20€
PCB	Simple 2/4 couches	Complexe 6 couches
Alimentation	Simple 3.3V	Double ou triple
Validation	Simple	Complexe

μC vs SOC : logiciel

	μC	SOC
Volume de code système mis en jeu (stabilité)	100Ko	10Mo
Environnement de cross-développement	~3000€ Ou outils GNU	Gratuit et simplifié si hôte et cible Linux
Test et debug de l'application	Difficile, outils spécifiques	Plus simple grâce à l'OS
Déploiement, mise à jour	Difficile	Simple, possibilités multiples
Protection du code	Facile	Difficile

Design à base de SOC : le module CPU

- Payer le design, la qualification d'une carte à base de SOC, avec la génération du BSP et sa maintenance, ne pourra se justifier que pour des quantités importantes
- Pour réduire ces coûts le « computer on module » est un choix intéressant
- A première vue, un COM qui coûtera 30€ - 100€ peut sembler onéreux. Pourtant cela peut s'amortir très rapidement si le module et le BSP fourni sont de qualité. Attention aux petites économies contre-productives.

Design à base de SOC : le module CPU

- Il existe des standards : QSEVEN, ETX, XTX, COM-EXPRESS et d'autres
- Plusieurs fabricants proposent des gammes de modules compatibles : l'évolution du produit s'en trouve facilitée.
- Développer sa carte porteuse avec les interfaces / alim / connecteurs propres au métier : enveloppe 5 -> 20k€

COM : BSP

- Face à l'ampleur de l'offre en COM attention à la qualité du support ! Le BSP doit être maîtrisé par le fournisseur ou par un intervenant tierce.
- Le BSP fourni est adapté à la carte d'évaluation, prévoir un travail de portage sur votre carte métier
- Le temps à passer sur le BSP est souvent sous évalué
- Penser aux outils de déploiement

Les avantages du développement sur OS

- Portabilité, indépendance vis à vis du matériel
- Ecosystème beaucoup plus développé
- Richesse des outils et langages de programmation
- Système de fichiers
- Passerelles avec l'univers bureautique
- Possibilités pour l'architecture de l'application (process / thread / édition de liens dynamique / langages de script etc..)
- Disponibilité de compétences

La solution mixte SOC + μ C

De nombreux designs utilisent cette architecture : un SOC connecté à un microcontrôleur avec une liaison de type SPI + Irq

Permet de concilier les avantages des deux mondes

Le système Linux peut reprogrammer le MCU pour assurer la synchronisation des versions

La communication MCU / SOC peut être une tâche non négligeable !

Permet de retrouver une facilité de réalisation de cartes comparable au microcontrôleur

La solution mixte SOC + μ C

- En écrivant le driver adéquat on peut « remonter » proprement sous Linux les E/S du microcontrôleur et les interfaces type UART et CAN.
- On concentre sur le microcontrôleur la partie temps réel dur ou nécessitant du déterminisme, de la sécurité. Également ce qui est spécifique et demanderait du dev kernel sous Linux.
- Le SOC ou module reste le plus possible dans du standard, déjà dispo avec le BSP
- Freescale intègre ce principe dans sa techno VYBRID VF6xx qui intègre sur un chip un cortex-A5 et un cortex-M4

Conclusion

Chute des prix sur les processeurs et les mémoires Flash et RAM

+

Dynamique de GNU / Linux et du logiciel libre

=

De nouvelles opportunités pour réaliser des systèmes
embarqués plus innovants que jamais !