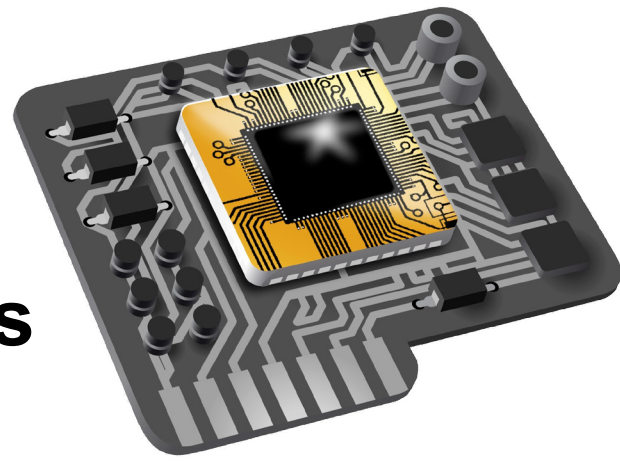


Tour d'horizon des systèmes embarqués

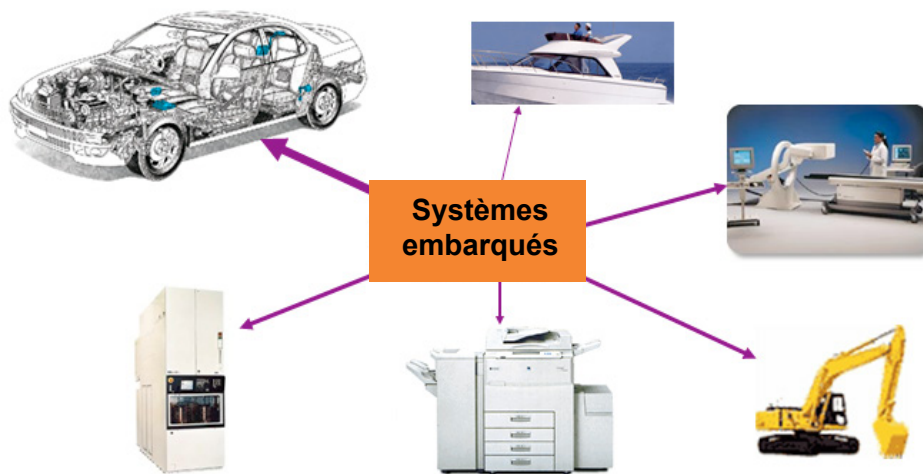


Par Christophe Blaess

Notion de système embarqué

J'aurais tendance à décrire un système embarqué plutôt comme un ensemble informatique qui se trouve intégré dans une entité plus complexe dont il devient une pièce parmi d'autres. Ainsi une carte à microprocesseur (système informatique complet) devient un simple composant d'un tableau de bord, d'un panneau de contrôle industriel, d'un objet grand-public, etc.

Une caractéristique commune à la plupart des systèmes embarqués est le fait qu'ils sont conçus pour répondre à un ensemble de contraintes imposées par leur environnement de fonctionnement ou leur destination. En voici quelques exemples courants :



- **contraintes matérielles** : l'encombrement maximal du système (taille et poids) impose des limitations à l'autonomie (volume des batteries), au refroidissement (radiateurs et ventilateur) et par conséquent à la puissance processeur utilisable ;

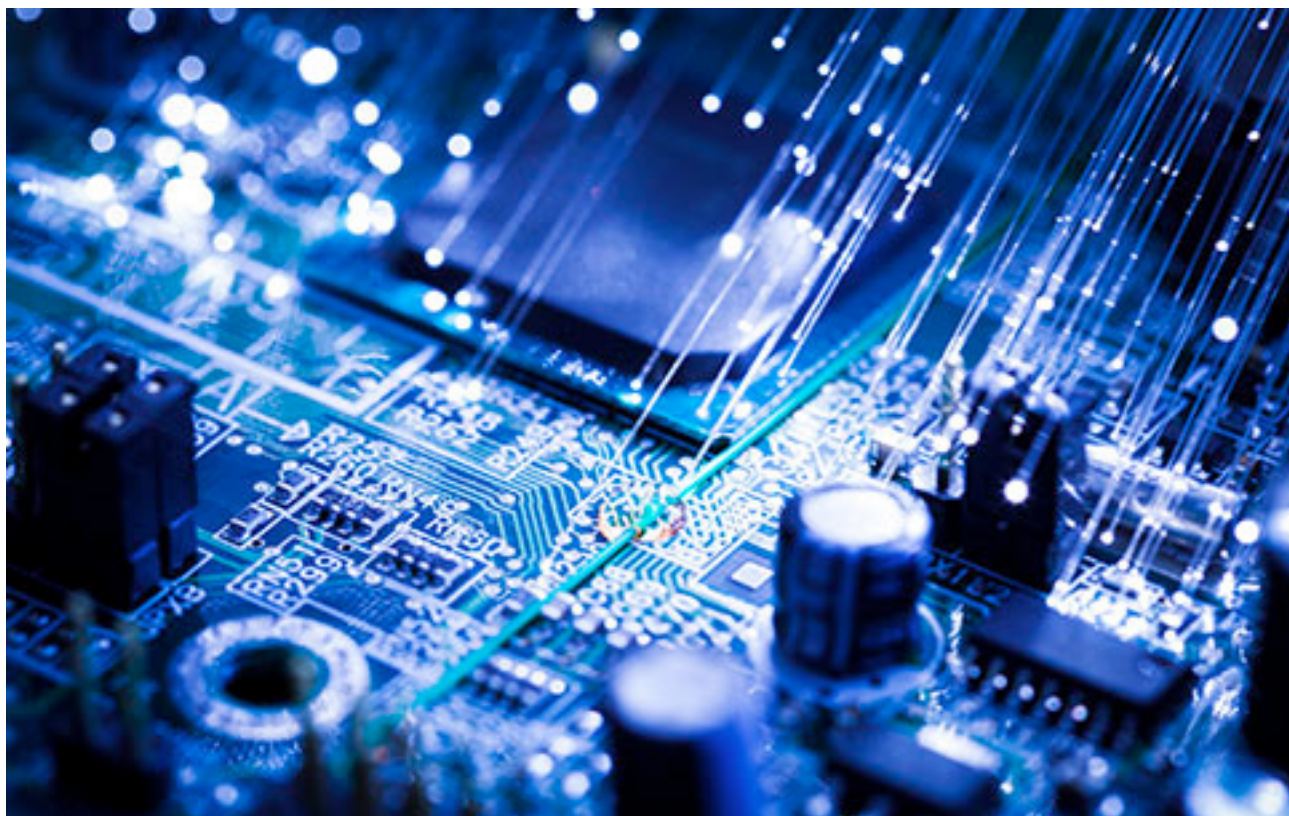
- **contraintes logicielles** : le fonctionnement du système dans des environnements hostiles, soumis à des utilisateurs inexpérimentés ou malintentionnés, sa résilience face à des coupures d'alimentation intempestives et la possibilité de réaliser des mises à jour distantes et automatiques sont des problèmes complexes, et doivent impérativement être pris en considération lors de la conception d'un système embarqué ;

- **contraintes fonctionnelles** : pour qu'un système soit en avance face à la concurrence, il est nécessaire de minimiser les coûts – donc la puissance matérielle disponible – tout en optimisant les fonctionnalités proposées à l'utilisateur. Le logiciel doit tirer parti de toutes les ressources disponibles via des techniques de programmation spécifiques (temps réel, multi-threading, IHM avancée, etc.).

Domaines d'application

Bien que les systèmes embarqués prennent des formes très différentes et représentent une activité transverse de l'informatique, on peut les ranger dans quelques catégories couramment rencontrées :

- **Systèmes embarqués industriels** : ces systèmes ont essentiellement un rôle de pilotage et de supervision de processus de production industrielle. Le cœur du métier est souvent réalisé par des composants spécialisés (systèmes d'acquisition, FPGA, microcontrôleur, ASIC, etc.) dialoguant par l'intermédiaire de protocoles de communication adaptés (CAN, SPI, PCIe...) avec un processeur intégré sur une carte embarquée qui assure les fonctions secondaires (réseau, interface utilisateur, stockage...).
- **Électronique grand public** : les systèmes embarqués sont omniprésents dans les équipements technologiques actuels (smartphones, tablettes, liseuses...) et gagnent sans cesse du terrain dans de nombreux appareils domestiques (électroménager, domotique, etc.). L'interface utilisateur joue un rôle crucial dans le choix du public, les systèmes d'exploitation et environnements logiciels doivent assurer une cohérence d'utilisation entre les différentes plates-formes. C'est le domaine de prédilection des environnements comme Android, iOS, Tizen, etc.
- **Objets connectés, IoT (Internet of Things)** : cette nouvelle branche des systèmes embarqués connaît un essor très important depuis quelques années. Le modèle repose généralement sur la connexion d'objets du quotidien vers des plates-formes de Cloud Computing (en passant par la box Internet locale). L'exploitant engrange d'énormes quantités de données en provenance de ses utilisateurs, qu'il peut traiter avec des algorithmes spécifiques et innovants (issu des recherches dans le domaine du Machine Learning) pour en extraire des informations pertinentes pour l'utilisateur (statistiques, prévisions, optimisations, etc.)



Plates-formes matérielles

Les supports matériels pour l'implémentation des systèmes embarqués se déclinent essentiellement en trois types :

- **Les micro-contrôleurs** : ces petits processeurs bon marché, peu complexes à mettre en œuvre, mais relativement limités dans les traitements qu'ils permettent ont un rôle important dans le domaine des objets connectés de par leur faible encombrement et leur consommation électrique réduite.

On les rencontre également en versions plus puissantes dans les dispositifs industriels pour des tâches simples nécessitant une bonne prédictibilité du temps de traitement des événements. Il existe des cartes à microcontrôleurs permettant une mise en œuvre très rapide pour une expérimentation ou un prototypage, comme les Launchpad de Texas Instrument ou le célèbre Arduino et ses dérivés.

- **Les SOC (System-on-Chip)** : dans un unique circuit intégré sont regroupées les fonctionnalités du microprocesseur, de divers périphériques (contrôleur graphique, réseau, communications série, SPI, i²C, RS232, etc.), ainsi qu'une quantité plus ou moins importante de mémoire flash, voire de mémoire Ram. L'intégration d'un SOC sur une carte personnalisée est un peu plus complexe que celle d'un microcontrôleur, mais reste largement à la portée d'un bureau d'études électroniques.

- **Les SBC (Single Board Computers) ou ordinateurs monocartes** : très employés pour le prototypage ou la réalisation de petites séries, les ordinateurs monocartes proposent l'intégration d'un SOC ou d'un processeur classique sur une carte intégrant toute l'électronique nécessaire pour l'alimentation, la génération d'horloges, les entrées-sorties, etc.

On trouve des SBC industriels durcis, résistants en environnement hostile, aussi bien que des cartes plutôt dédiées à une utilisation hobbyiste ou expérimentale comme le fameux Raspberry Pi.

Il existe des SBC très réduits, tenant sur quelques cm² de circuit électronique et prêts à être intégrés sur une carte de plus grande dimension. On parle dans ce cas de CoM (Computer-on-Module).

Environnements logiciels

Pour les systèmes très réduits, comme les microcontrôleurs, il est fréquent d'employer des OS (Operating Systems – systèmes d'exploitation) très légers – voire aucun OS et de n'intégrer que le code développé spécifiquement pour le projet. Ces OS simplifiés, qui se résument souvent à un ordonnanceur et des mécanismes de synchronisation entre tâches, sont généralement propriétaires et vendus par le fabricant du microcontrôleur accompagnés d'un environnement de développement (compilateur, etc.) spécifique.

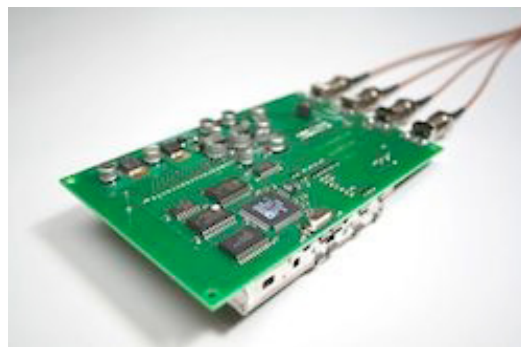
Il est néanmoins important de noter que, même pour des microcontrôleurs, il existe des systèmes d'exploitation libres comme Lepton, FreeRTOS ou RTEMS.

Dans le domaine de l'embarqué avec système d'exploitation (SOC, SBC), la prédominance actuelle de Linux est incontestable. Il s'agit d'un OS libre (dont le code source est intégralement consultable, adaptable, redistribuable) disponible gratuitement, sans aucun coût d'installation initiale ni frais ultérieurs de licence d'exploitation.

L'un des principaux avantages de Linux est la disponibilité immédiate d'une énorme quantité de drivers pour les périphériques, de protocoles réseau, de systèmes de fichiers, etc. En outre il existe un immense catalogue d'applications libres et gratuites couvrant la plupart des domaines de l'informatique actuelle.

Il est important de noter que Linux est porté sur un nombre très important d'architectures et de plates-formes actuelles, du petit processeur embarqué proche d'un microprocesseur au super-calculateur comportant des centaines ou milliers de processeurs travaillant en parallèle. Ceci garantit de disposer d'un environnement homogène sur une large gamme de processeurs et de cartes.

L'interface de programmation pour le développement applicatif respecte le standard Posix et offre des possibilités de développement dans de nombreux langages (C, C++, Python, NodeJS, PHP, etc.).



Formations Orsys sur ce thème :

- BSP Linux embarqué, construire son propre noyau
- Linux temps-réel et embarqué
- Écriture de drivers et programmation noyau Linux
- Systèmes embarqués, développement sur la plate-forme Arduino
- Raspberry Pi 2, créez votre propre système embarqué sous Linux
- Android, construire son propre système embarqué
- Internet des objets, développer des applications connectées en Java



»» Christophe BLAESS »

Expert dans le domaine de Linux Industriel, il conduit de nombreux projets (applications de contrôle aérien, système d'interfaçage d'automates industriels...) et réalise des prestations d'ingénierie et de conseil dans des domaines variés liés à Linux : télévision numérique, informatique médicale, imagerie aérienne, traitement radar embarqué... Il dispense des formations professionnelles (Linux temps-réel et embarqué, Écriture de drivers, Programmation noyau...) en collaboration avec Orsys.
