Conception et implémentation des objets connectés

Dr. Ing. Chiheb Ameur ABID

Contact: chiheb.abid@fst.utm.tn

Année universitaire : 2021 - 2022

Plan

Objet connecté

Définition d'un Objet Connecté (OC)

- Un dispositif dont la finalité première n'est pas d'être un système informatique ni une interface d'accès au web
  - ☞ Machine à café, une serrure, machine à laver, etc.
- Intéraction indépendante avec le monde physique de manière indépendante sans intervention humaine.
- Les éléments clés d'un OC
  1. Les données produites ou reçues, stockées ou transmises.
  2. Les algorithmes pour traiter ces données.
  3. L'écosystème dans lequel il va réagir et s'intégrer.

Objet connecté

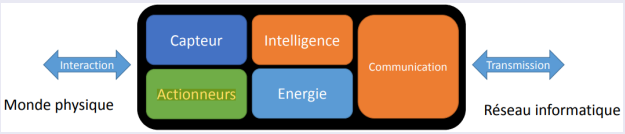
Exemples d'OCs

- Les objets "traditionnels" : ordinateur, tablette, smartphone, etc.

- Nouveaux objets connectés : appareils électroménagers, instruments de mesure, robots, serrures, machines-outils, bennes à ordures, drones, jouets, montres, véhicules, etc.

Solutions technologiques

Caractéristiques générales d'une plateformes pour l'IoT



- Capteurs : recueillir des informations depuis le monde physique.
- Actionneurs : agir sur le monde physique en modifiant son état.
- Intelligence : traitement local des données
- Communication : codage et transmission des données, protocoles standards ou dédiés,communication filaire ou sans fil.
- Énergie : alimentation de la plateforme en énergie électrique.
  - Doit être adaptée à l'application

Solutions technologiques

Solutions technologiques

Deux approches majeures

1. Systèmes construits autour d'un OS embarqué (Raspberry Pi, Beaglebone, etc.)
   - ✔ Ouverts, puissants, langages de programmation multiples
   - ✖ Parfois complexes à mettre en œuvre, prise en main longue, réactivité moyenne, coût relativement élevé, interfaçage plus difficile
2. Systèmes dédiés compacts à logiciel propriétaire (ARDUINO, GENUINO, INTEL GALILEO, ESP8266 etc.)
   - ✔ Très réactifs, très faible coût, fonctionnement plus robuste (pas de couches logicielles), interfaçage aisé, prise en main très rapide.
   - ✖ Moins puissants, langages de programmation plus limités, moins flexibles sur le plan logiciel.

Plan

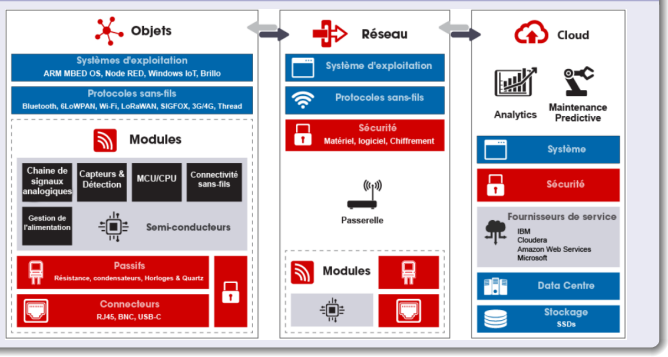
Schéma général d'une solution IOT

Composantes d'une solution IOT

- Généralement, une solution IoT est formée des composants suivants :
  1. Objet (Module-capteur-actionneur)
  2. Passerelle (Gateway)
  3. Cloud (Informatique en nuage)

Architecture fonctionnelle d'une solution IOT

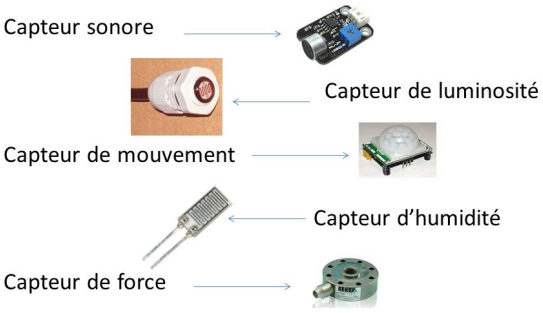
Architecture fonctionnelle d'une solution IOT



Niveau 1 : Capteur/actionneur

Les capteurs

- Un capteur permet de fournir des informations sur le monde extérieur
- Il a comme rôle de saisir une grandeur physique et de la convertir en informations numériques



Niveau 1 : Capteur/actionneur

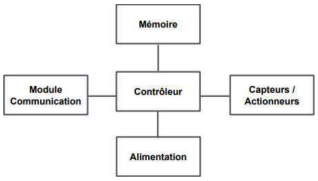
Les actionneurs

- Un actionneur permet d'agir sur le monde extérieur.
- Il a comme rôle de convertir une valeur électrique en une action physique

Niveau 1 : Capteur/actionneur

Les cartes à micro-contrôleur

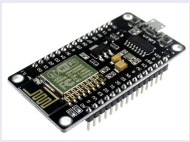
- Avec les actionneurs et les capteurs, on associe des cartes à micro-contrôleurs pour extraire les informations utiles ou commander les actionneurs



Niveau 1 : Capteur/actionneur

Exemples de cartes : NodeMCU

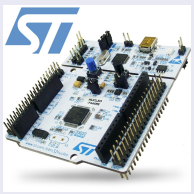
- Un microcontrôleur avec Wifi



  - 80Mhz par un processeur 32bits RISC (ESP8266) + 96K de RAM + une mémoire flash de 512Ko à 4Mo
  - Connectivité Wifi 802.11 b/g/n supportant le WEP, WPA/2/WPS et réseau ouvert
  - SPI, I²C, UART et un port ADC
  - Alimenté en 3,3V et consomme entre 60mA et 200mA avec le Wifi activé
  - Prix entre 2 € et 10 €

Niveau 1 : Capteur/actionneur

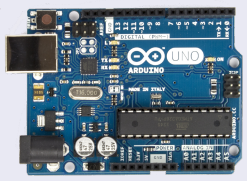
Exemples de cartes : STM32



- Basé sur un micro-contrôleur ARM 32 bits (24 à 200Mhz)
- SPI, I²C, UART, DAC et ADC
- DSP

Niveau 1 : Capteur/actionneur

Exemples de cartes : Arduino



- Basé sur le micro-contrôleur ATMega + 4K à 256 de RAM
- SPI, I²C, UART et un port ADC de 8 à 10 canaux
- De 23 à 53 lignes d'entrée-sortie universelles

Niveau 1 : Capteur/actionneur

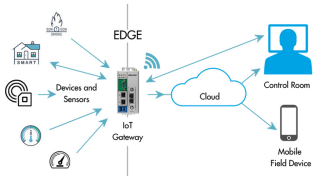
Exemples de cartes : Pycom Lopy4

- Basé sur le micro-contrôleur ESP32 (dual core de 160 à 180 MHz)
- Connectivité Wifi, Bluetooth, LoRa, Sigfox

Niveau 2 : Passerelle

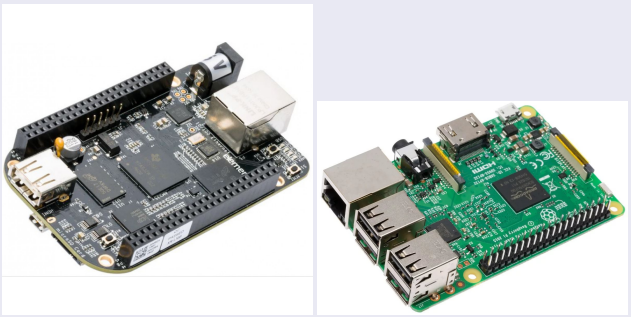
Niveau 2 : Passerelle

- Une passerelle (gateway) est une combinaison de composants matériels et logiciels utilisés pour connecter un réseau à un autre
- Les gateways permettent de relier les capteurs ou les nœuds de capteurs avec le monde extérieur
- Les gateways sont donc utilisées pour la communication de données en collectant les mesures effectuées par les nœuds de capteurs et en les transmettant àl'infrastructure Internet.
- La gateway peut faire des traitements locaux sur les données avant de les relayer au Cloud



Niveau 2 : Passerelle

Exemples de passerelles



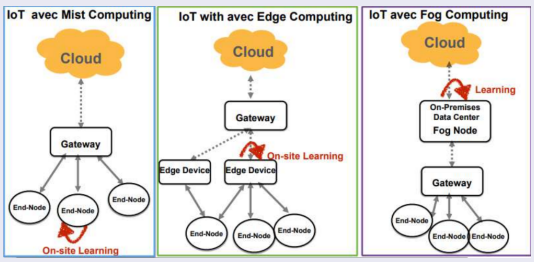
Niveau 3 : Cloud computing

Niveau 3 : Cloud computing

- Le niveau 3 est un choix technologique (optionnel) qui permet d'alléger la charge du travail vers le Cloud et de faire des traitements locaux on the Edge
- Trois solutions techniques sont possibles pour l'implémentation du 3ème niveau
  - Fog computing : un calcul décentralisé en traitant les données IoT au niveau des nœuds locaux (Fog) avant de relayer l'information vers le cloud.
  - Edge computing : le traitement des données IoT se fait à l'extrémité du réseau (Gateways ou des nœuds intermédiaires entre objets et gateways)
  - Mist Computing : le traitement des données se fait localement dans le nœud capteur.

Niveau 3 : Cloud computing

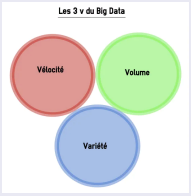
Fog vs Edge vs Mist



L'exploitation et l'indispensable corrélation de données avec les outils Big Data

Big Data : Mega données

- Le Big Data est une solution pour permettre à tout le monde d'accéder en temps réel à des bases de données géantes. Il vise à proposer un choix aux solutions classiques de bases de données et d'analyse (plate-forme de Business Intelligence en serveur SQL...).
- les informations présentant trois caractéristiques principales
  1. Le volume : volume considérable de données à traiter
  2. La variété : une grande Variété d'informations (venant de diverses sources, non-structurées, organisées, Open...)
  3. La vélocité : atteindre certain niveau de fréquence de création, collecte et partage de ces données



L'exploitation et l'indispensable corrélation de données avec les outils Big Data

IoT et Big Data : deux technologies inextricablement liées

- Le nombre d'objets connectés augmente
  - Le volume de données générées par l'internet des objets explose
- Stockage des données pour les IOT doit répondre aux exigences suivantes :
  - Le volume : la quantité de données à stocker peut être massive
  - Variété : différents dispositifs et différents types de capteurs produisent des formes de données très différentes
  - Rapidité : de nombreux cas IoT nécessitent l'analyse des flux de données pour prendre des décisions instantanées
  - Véracité : dans certains cas, les capteurs produisent données ambiguës et inexactes
- Utiliser les bases de données NoSQL, pour "not only SQL", qui ne sont pas fondées sur l'architecture classique des bases de données relationnelles
  - Développées à l'origine pour gérer du big data
  - Principale caractéristique : la représentation des données est notable par l'absence de schéma (schemaless)

Plan

Technologies disponibles

Wifi

- IEEE : 802.11a/b/g/n/ac
- Besoins en mémoire : 1Mo +
- Autonomie avec Pile : Jours
- Vitesse de transfert : 11, 54, 108, 320, 1000 Mb/s
- Portée (environ) : 300m
- Bande : 2,4 à 6 Ghz

Technologies disponibles

Bluetooth

- IEEE : 802.15.1
- Besoins mémoire : 250 ko +
- Autonomie avec Pile : Mois
- Vitesse de transfert : 1 Mb/s
- Portée (environ) : 10m
- Bande : 2,4 à 2,5 Ghz

Technologies disponibles

ZigBee

- IEEE : 802.15.4
- Besoins en mémoire : 4-32 ko
- Autonomie avec Pile : Années
- Vitesse de transfert :20-250 kb/s
- Portée (environ) : 100m
- Bande : 868 Mhz

Technologies disponibles

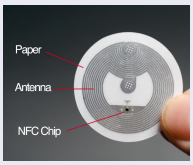
RFID

- RFID (Radio Frequency Identification) : procédé permettant de récupérer des données à distances grâce à un système composé :
  - Une étiquette radio contenant des informations
  - Un lecteur permettant de récupérer les informations d'une étiquette radio à distance
- Fréquences utilisées
  - Basses fréquences (LF) : 125 à 134 Khz
  - Hautes fréquences (HF) : 13,56 Mhz
  - Ultra-Hautes fréquences (UHF) : 850 à 950 Mhz

Technologies disponibles

NFC

- L'interface NFC (Near Field Communication) : une technologie dérivée de RFID permettant une communication à très faible distance entre deux objets
- Bande : 13,56 Mhz
- Distance : 0 et 20 cm
- Débit : 106 Kb/s et 424 Kb/s
- Fréquence utilisée (HF) : 13,56 Mhz



Plan

Le réseau LPWAN

LPWAN en bref

- LPWAN (Low Power Wide Area Network) : un réseau à longue portée et à faible consommation énergétique
  - Réseau dédié à l'IOT et M2M

Pour transmettre un signal sans fil sur une longue distance
- Soit augmenter la puissance du signal
- Soit réduire sa bande passante
- Soit faire les deux

- Les technologies LPWAN exploitent des bandes de fréquences avec et sans licence, et ont les caractéristiques suivantes :
  - Portée : de quelques kilomètres dans les zones les plus denses à des dizaines de kilomètres dans les zones rurales.
  - Autonomie : Plusieurs années.
  - Bande passante : De 0.1 à plusieurs centaines de kbits/sec.

Le réseau LPWAN

Catégories des réseaux LPWAN

- LPWANs cellulaires : `Nb-IoT` et `LTE-M`
- LPWANs non cellulaires : `sigfox` et `LoRaWAN`

Choix d'un réseau LPWAN

- Quatres principaux paramètres pour choisir un réseau LPWAN
  1. Portée
  2. Débit
  3. Consommation d'énergie
  4. Coût de déploiement

Le réseau sigfox

Le réseau sigfox

- Mis en place par la startup française au début des années 2010
- Repose sur une technologie brevetée de bande ultra-étroite UNB (Ultra Narrow Band)
- Utilise des fréquences sans licence ISM (Industriel, Scientifique et Médical)
  - 868 MHz en Europe
  - 915 MHz en Amérique du Nord
  - 433 MHz en Asie

Le réseau sigfox : principe de fonctionnement (1/5)

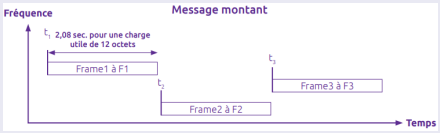
Bande ulta-étroite

- Utilise 192 KHz de la bande non licenciée ISM pour échanger des messages par liaison radio
  - Chaque message occupe 100 Hz (zones ETSI) ou 600 Hz (zones FCC)
  - Transfert avec un débit de 100 ou 600 bits par seconde selon la région

Le réseau sigfox : principe de fonctionnement (2/5)

Accès aléatoire

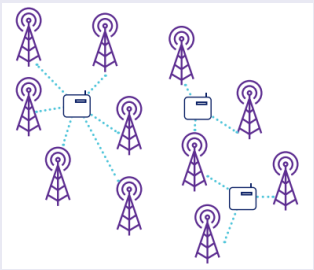
- La transmission n'est pas synchronisée entre l'objet et le réseau
- L'objet émet un message sur une fréquence aléatoire, puis envoie succesivement deux répliques sur des fréquences différentes
  - Ce que l'on appelle "la diversité temporelle et fréquentielle"



Le réseau sigfox : principe de fonctionnement (3/5)

Réception coopérative

- Contrairement aux protocoles cellulaires, les objets ne sont pas attachés à une station de base spécifique.
  - Le message envoyé est reçu par n'importe quelle station de base à proximité
  - Le nombre de stations de base recevant chaque message est de trois en moyenne



Le réseau sigfox : principe de fonctionnement (4/5)

Messages courts

- La taille d'un message envoyé par un objet (messages ascendants) est comprise entre 0 et 12 octets
- La taille de la charge utile des messages descendants est fixe : 8 octets
- Exemple de tailles de charge utile
  - Coordonnées GPS avec une précision de 3 m : 12 octets
  - Température comprise entre -100° et +200°, avec une précision de 0,004° : 2 octets
  - Vitesse jusqu'à 255km/h : 1 octet
  - Statut d'un objet : 1 octet

Réglementation

- La réglementation européenne stipule que nous pouvons occuper la bande de fréquence publique pendant 1 pour cent du temps
- Le cycle de service de la station de base en émission est de 10 pour cent
  - Garantit 4 messages descendants par objet
  - L'objet peut recevoir davantage de messages s'il reste des ressources disponibles au niveau des stations de base

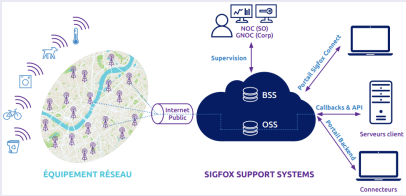
Le réseau sigfox : principe de fonctionnement (5/5)

Communication bidirectionnelle

- Le message descendant est déclenché par l'objet
- Une fois un objet a émis son message, l'objet retourne en veille pendant 20 s puis se réactive pendant 25 s pour recevoir le message descendant émis par la station de base
- La fréquence descendante est la fréquence du premier message montant plus un delta défini.

Le réseau sigfox : architecture

Architecture du réseau



- Le message descendant est déclenché par l'objet
- Une fois un objet a émis son message, l'objet retourne en veille pendant 20 s puis se réactive pendant 25 s pour recevoir le message descendant émis par la station de base
- La fréquence descendante est la fréquence du premier message montant plus un delta défini.

Le réseau LoRaWAN

Le réseau LoRaWAN

- Créé par la startup française Cycleo en 2009
- Depuis 2015, le LoRaWAN est porté par une association à but non lucratif appelé The LoRa Alliance, qui regroupe plus de 500 entreprises
- LoRaWAN se repose sur une modulation à étalement de spectre appelé CSS (Chirp Spread Spectrum)
  - Le message transmis est répété plusieurs fois à des fréquences différentes
- LoRa, comme le SigFox, opère sur une plage de fréquences ISM sans licence
  - 868 MHz en Europe
  - 915 MHz en Amérique du Nord.

Plan

Historique

L'histoire d'ARM

- Tandis qu'Intel produit des microprocesseurs 8086 et 80286, des employés d'une compagnie anglaise se sont inspirés par les travaux de l'heure travaillent à la conception d'un ordinateur
- En 1985, la compagnie ACORN produit un premier processeur qui résulte de ces travaux : le Acorn RISC Machine (ARM)
- La complexité du ARM est réduite par choix et par faute de moyens, ainsi le jeu d'instructions est simplifié : naissance des processeurs RISC
  - Acorn Archimedes (1987) : ordinateur conçu à base d'ARM
- Alors que le 8086 gagne en popularité, ACORN et Apple collaborent pour fonder la compagnie ARM ltd en 1990
- ARM et Intel feront évoluer leurs produits en parallèle dans des marchés différents

La philosophie d'ARM

La philosophie d'ARM

- ARM Holdings
  - Développe des architectures de micro-processeurs et des jeux d'instructions
  - Ne construit aucun micro-processeur comme tel !
  - La compagnie licencie la technologie à d'autres qui les fabriquent à leur façon clients : Apple, Nvidia, Samsung, Texas Instruments, etc.
- Micro-processeurs ARM
  - Supportent 32 et 64 bits
  - L'architecture la plus utilisée au monde (100 milliards de processeurs produits en 2017)

ARM est partout

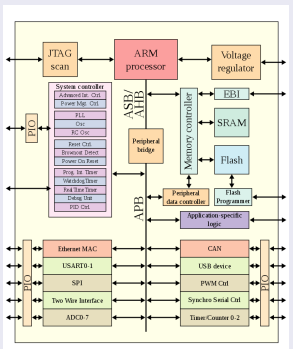
ARM est partout

- Plusieurs versions de processeurs, utilisées partout !
  - ARM7TDMI(-S) : Nintendo DS, Lego NXT
  - ARM946E-S : Canon 5D Mark ii (caméra)
  - ARM1176JZ(F)-S : Raspberry Pi
  - Cortex-A9 : Apple iPhone 4S, iPad2
  - Cortex-A15 : Nexus 10

ARM est partout

ARM est partout

- Aujourd'hui, ARM est surtout connu pour ses systèmes sur puce (SoC), intégrant sur une seule puce : microprocesseur, processeur graphique (GPU), DSP, FPU, SIMD, et contrôleur de périphériques. Ceux-ci sont présents dans la majorité des smartphones et tablettes.
- ARM propose des architectures, qui sont vendues sous licence de propriété intellectuelle aux concepteurs.
- Différentes options sont possibles ainsi les constructeurs peuvent prendre ce qui les intéresse pour les compléter avec leurs options propres ou de concepteurs tiers



Architecture des ARMs 32 bits

Architecture des ARMs 32 bits

- Architecture RISC pour laquelle tout passe par des registres
- ARM possède 37 registres de 32 bits (seulement 17 sont visibles)
  - 16 registres généraux
  - le compteur ordinal `R15` (`PC`) de 32 bits
  - le registre d'état CPSR de 32 bits
- Chaque instruction est codé sur 32 bits
- Accès à 2^{32} octets = 4GB de mémoire
- Organisation des données dans la mémoire "Little ou Big endian" au choix
- Opère sur 8, 16 et 32 bits

Plan

Versions et implémentation de v1 à v6

Versions et implémentation

- Version 1 : ARM1
  - Pas vraiment commercialisée (quelques centaines d'exemplaires)
- Version 2 : ARM2
  - 27 registres (16 accessibles simultanément)
  - 4 modes de fonctionnement (mode utilisateur avec certaines ressources non disponibles, mode interruption pour gérer les interruptions externes, mode interruption rapide avec plus de ressources dédiées, mode superviseur pour l'exécution du système d'exploitation)
  - Pipeline d'exécution à trois étage (lecture, décodage, exécution)
  - 8MHz, 4 à 5 MIPS
- Version 2aS : ARM250 et ARM3
  - Ajout d'un cache unifié (données et instructions) de 4Ko
  - Ajout d'une instruction d'échange de données monolithique et atomique entre un registre et la mémoire (environnement multiproc)
  - Version mise en œuvre dans l'ARM3 (26 à 33MHz) et ARM250 (12MHz)

Versions et implémentation de v1 à v6

Versions et implémentation de v1 à v6

- Version 3 : ARM6, et ARM7
  - Véritable adressage 32 bits
  - Ensemble de registres pour le maintient de l'état du processeur
  - ARM6 : plusieurs variantes (coprocesseur, gestionnaire de la mémoire, cache modifié, etc.)
  - fréquence : 26-33 MHz
  - ARM7 fonctionnellement identique à l'ARM6 mais avec des fréquences plus élevées, cache énergétiquement plus performant, meilleure gestion de la mémoire,
  - fréquence : 40MHz et plus
- Version 4 : ARM8, ARM9 et StrongARM
  - Certaines versions optionnelles dans la v3 sont intégrées dans la v4 : multiplication étendue, etc.
  - Pipeline 5 étages : Lecture instruction décodage, exécution, mémoire et écriture registre.
  - ARM8 : ARM7 + unité d'exécution spéculative, politique d'écriture retardée pour le cache, multiplication 64 bits, 80MIPS à 80MHz
  - StrongARM : cache Harvard (données et instructions séparés), 100-200MHz
  - ARM9 : ARM8 + cache Harvard

Versions et implémentation de v1 à v6

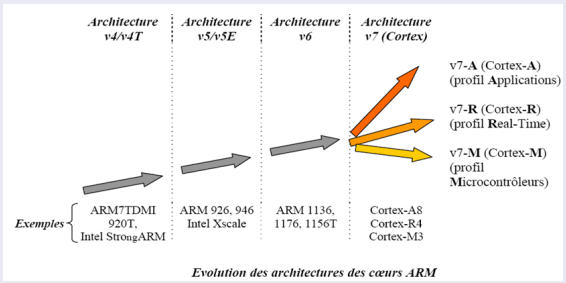
Versions et implémentation de v1 à v6

- Version 5 : ARM10, Xscale
  - V4 + instructions supplémentaires, unités d'exécution multiples, pipeline à 6 étages
  - ARM10 : 2*32Ko de cache, gestionnaire de mémoire compatible avec les OS embarqués, 4000MIPS, 350 MHz
  - Introduction de Jazelle DBX : technique permettant d'exécuter le bytecode Java
  - Intel Xscale : gestion de plusieurs périphériques, jusqu'à 624MHz
- Version 6 : ARM11
  - Amélioration multimédia
  - Extension SIMD (Single Instruction Multiple Data stream)
  - Support multiprocesseurs
  - Pipeline 8 étages, unité de prédiction de branchement.

Versions et implémentation

Versions et implémentation

- L'architecture et le jeu d'instructions du coeurs ARM ont évolué depuis la première version ARM-v1 jusqu'à la version ARM-v7 depuis laquelle on observe l'apparition de l'appellation Cortex



Evolution des architectures des cœurs ARM

Versions et implémentation

Versions et implémentation

	Application Cortex Processors			Cortex-A15		
	Embedded Cortex Processors			Cortex-A9		
	Classic ARM Processors			Cortex-A8		
CORE			ARM11MP	Cortex-A5	Cortex-M7	
		ARM926	ARM176JZ	Cortex-R7	SC300	SC00
	ASC100	ARM968	ARM1136J	Cortex-R5	Cortex-M4	Cortex-M1
	ARM7TDMI	ARM946	ARM1156T2	Cortex-R4	Cortex-M3	Cortex-M0
Family	ARM7TDMI	ARM9E	ARM11	Cortex-A/R	Cortex-M	Cortex-M
Architecture Version	ARMv4T	ARMv5TJ	ARMv6	ARMv7A/R	ARMv7M/ME	ARMv8M

Versions et implémentation v7

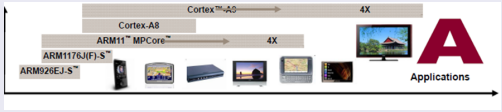
Versions et implémentation v7

- Version 7
  - Données manipulées : Word (32 bits), Halfword (16 bits) ou Byte (8 bits)
  - Pas de contrainte sur le fait d'être Big ou little endian
  - Communication avec les périphériques via un mécanisme de mapping mémoire
  - Utilisation : le groupe ST Microelectronics utilise l'ARM7 pour les DSP (Digital Signal Processor) qui vont constituer l'étage de décodage numérique des prochains récepteurs radio DRM (Digital Radio Mondiale)

Les séries Cortex

La série A

- Le profil A (séries Cortex-A) : destiné pour faire tourner des applications complexes telles que les systèmes d'exploitation embarqués (linux, Windows) nécessitant une puissance de traitement élevée et un système de gestion de mémoire virtuelle (MMU : Memory Mangement Unit)



Les séries Cortex

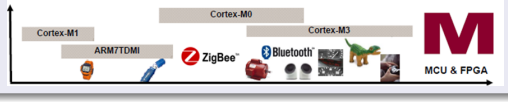
La série R

- Le profil R (séries Cortex-R) : destiné principalement aux applications temps réel à contraintes très sévères et exigeant une haute fiabilité et un temps de réponse faible

Les séries Cortex

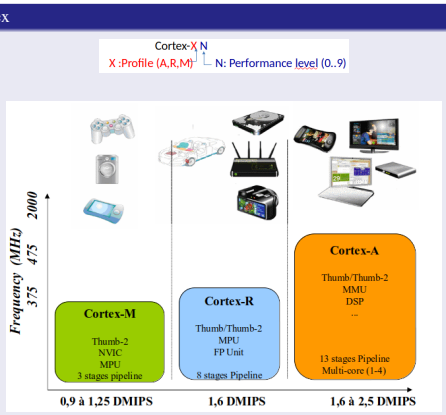
La série M

- Le profil M (séries Cortex-M) : optimisé pour des applications nécessitant une basse consommation en énergie, un coût réduit et un comportement déterministe pour le traitement des interruptions. C'est le profil qui est destiné à être intégré dans les microcontrôleurs (utilisés surtout pour la commande des machines)



Les séries Cortex

Les séries Cortex



Plan

La raspberry Pi

Présentation de la raspberry Pi

- Nano-ordinateur monocarte à processeur ARM conçu par David Braben (UK) 2011-2012
- Destiné à encourager l'apprentissage de la programmation informatique
- Plus de 12 millions d'exemplaires vendus
- Disponible à moins de 40 € (entre 10 et 35 €) selon la version

Il est généralement vendu "nu" (carte mère seule, sans boîtier, alimentation, clavier, souris ni écran) dans l'objectif de diminuer les coûts et de permettre l'utilisation de matériel de récupération.

Les systems-on-chip BCM2835/2836/2837

Les systems-on-chip BCM2835/2836/2837

Plusieurs générations de Raspberry Pis ont été produites. d'architecture ARM.

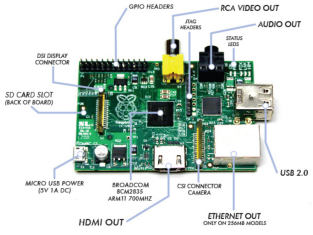
- Tous les modèles disposent d'un système SoC (System on a Chip) Broadcom avec une unité centrale de traitement (CPU) d'architecture ARM et une unité de traitement graphique (GPU)
- Les architectures ARM sont des architectures de type RISC 32 bits (ARMv1 à ARMv7) et 64 bits (ARMv8) développées par ARM Ltd depuis 1990.
- Le SoC Broadcom BCM2835 utilisé dans la première génération de Raspberry Pi comprend un CPU ARM11 (ARMv6) de 700 MHz, un GPU VideoCore IV et de la RAM. Il possède un cache de niveau 1 (L1) de 16 Ko et un cache de niveau 2 (L2) de 128 Ko principalement utilisé par le GPU (Graphics Processing Unit).
- Le Raspberry Pi 3+ utilise un SoC Broadcom BCM2837B avec un processeur ARM Cortex-A53 (ARMv8-A) quatre coeurs 1,4 GHz à 64 bits, avec une mémoire cache L2 partagée de 512 Ko.

Les versions de la raspberry Pi

Depuis 2012, la Raspberry Pi s'est déclinée en plusieurs versions

Raspberry Pi 1

- Modèle A : ARMv6 à 700 MHz Broadcom 2835 - RAM 256 Mo
- Modèle A+ : Lecteur de carte microSD - GPIO 40 broches
- Modèle B Rev1 : RAM 512 Mo - 2 ports USB 2.0 - 1 port réseau Fast Ethernet (10/100 Mbit/s) ; Rev2 : JTAG - I2C
- Modèle B+ : 4 ports USB 2.0 - micro SD ;



Raspberry Pi modèle B

Les versions de la Raspberry Pi

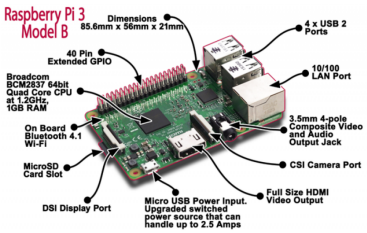
Raspberry Pi 2

- Modèle B : quatre cœurs ARMv7 à 900 MHz Broadcom BCM2836 - RAM 1Go

Les versions de la Raspberry Pi

Raspberry Pi 3

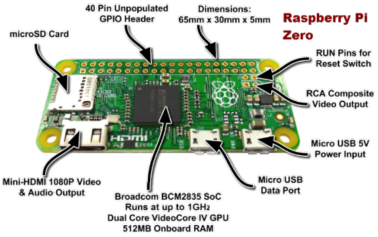
- Modèle 3B : quatre cœurs ARM Cortex-A53 à 1,2 GHz Broadcom BCM2837 64 bits - Wifi 802.11n et Bluetooth 4.1 ;
- Modèle 3B+ : quatre cœurs ARM Cortex-A53 cadencé à 1,4 GHz Broadcom BCM2837B 64 bits - WiFi Dual-band 802.11ac et Bluetooth 4.2
- Modèle A : ARMv6 à 700 MHz Broadcom 2835 - RAM 256 Mo



Les versions de la raspberry Pi

Raspberry Pi Zero

- Modèle Zero W : ARMv6 à 1 GHz Broadcom BCM2835 - Mini-HDMI - Wifi 802.11n et Bluetooth 4.1



Les versions de la raspberry Pi

Raspberry Pi 4

- Modèle B : quatre cœurs ARMv8 Cortex-A72@1.5Ghz, 2xUSB 2.0+2xUSB 3.0, 2 mini-HDMI, Mémoire 1, 2 ou 4 Go, etc.

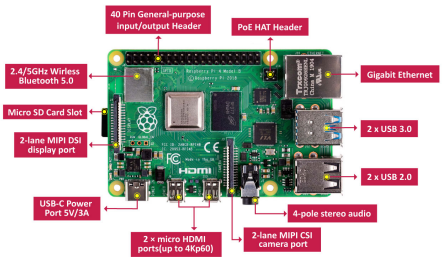


Tableau comparatif

Generation	Model A		Model B						Zero		
	1	1 +	1	1 +	2	2 ver 1.2	3	3+	PCB ver 1.2	PCB ver 1.3	W (wireless)
Release date	February 2013	November 2014	April–June 2012	July 2014	February 2015	October 2016	February 2016	14 March 2018	November 2015	May 2016	28 February 2017
Target price (USD)	$25	$20	$35	$25	$35	$35	$35	$35	$5	$5	$10
Instruction set	ARMv6Z (32-bit)				ARMv7-A (32-bit)	ARMv8-A (64/32-bit)			ARMv6Z (32-bit)		
SoC	Broadcom BCM2835				Broadcom BCM2836	Broadcom BCM2837		Broadcom BCM2837B0	Broadcom BCM2835		
FPU	VFPv2; NEON not supported				VFPv3 + NEON	VFPv4 + NEON			VFPv2; NEON not supported		
CPU	1× ARM1176JZF-S 700 MHz				4× Cortex-A7 900 MHz	4× Cortex-A53 900 MHz	4× Cortex-A53 1.2 GHz	4× Cortex-A53 1.4 GHz	1 GHz single-core ARM1176JZF-S		
GPU	Broadcom VideoCore IV @ 250 MHz (BCM2837: 3D part of GPU @ 300 MHz, video part of GPU @ 400 MHz) OpenGL ES 2.0 (BCM2835, BCM2836: 24 GFLOPS / BCM2837: 28.8 GFLOPS) MPEG-2 and VC-1 (with license), 1080p30 H.264/MPEG-4 AVC high-profile decoder and encoder (BCM2837: 1080p60)										
Memory (SDRAM)	256 MB (shared with GPU)	512 MB (shared with GPU) as of 4 May 2016. Older boards had 256 MB (shared with GPU)	1 GB (shared with GPU)						512 MB (shared with GPU)		

Systèmes d'exploitation & RPi

Distributions

- La Raspberry Pi n'ayant pas de mémoire de masse interne ni de système d'exploitation intégré, on aura besoin d'une carte SD préchargée avec un système d'exploitation.
- Depuis sa commercialisation, la Raspberry Pi exécute des variantes du système d'exploitation libre GNU Linux-Debian. Suivant les modèles, on pourra aussi exécuter :
  - GNU/Linux : Debian (**Raspbian**), Fedora, Arch Linux, Gentoo, Slackware, Ubuntu, Suse, etc.
  - Windows 10 IoT Core
  - Android Pi
  - Autres : Firefox OS, RISC OS, NetBSD, FreeBSD, etc.
  - Distributions spécialisées : médias center (LibreELEC/OpenElec, OSMC/Rasbmc, etc), audio, retrogaming, réseaux (Kali), serveurs (OwnCloud), etc.

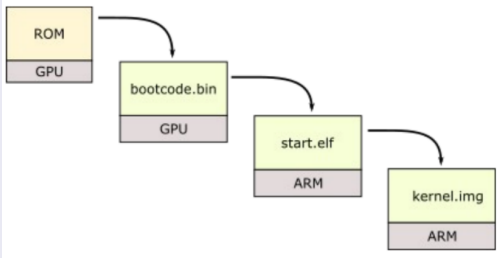
Distribution officielle : Raspberry OS (Raspbian)

Raspberry OS (Raspbian)

- C'est la distribution officielle supportée par la fondation Raspberry
- Basée sur Debian et compilée pour la Rapsberry Pi (ARM)
- Distribution classique et non pas embarquée ni temps réel
- livré pré-installé avec de nombreux logiciels pour l'éducation, la programmation et l'utilisation générale
- Elle existe en trois versions (05/2021)
  - Lite (444 Mo)
  - Desktop (1180 Mo)
  - Desktop avec des applications recommandées (2867 Mo)
- Dernière version basée sur Debian Buster avec le noyau linux version 5.10

Séquence de Boot

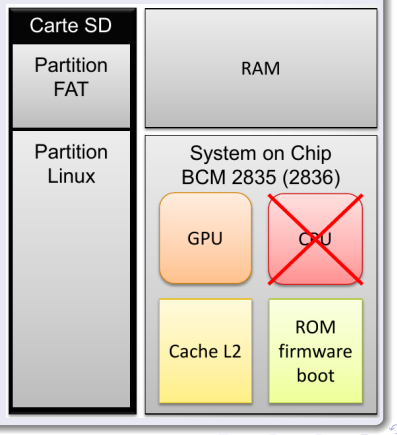
Démarrage du système d'exploitation



Séquence de Boot

Démarrage du système d'exploitation : Mise sous tension

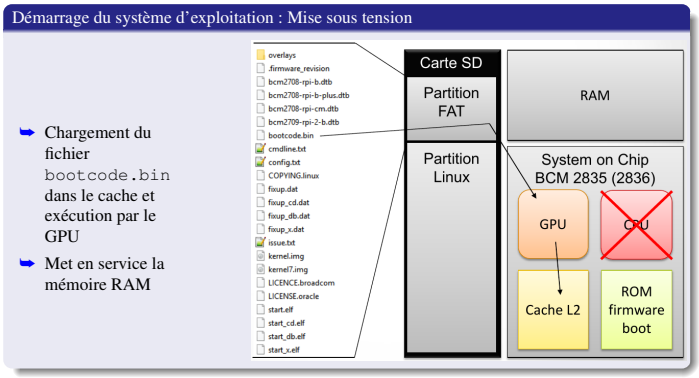
- GPU activé, CPU en veille, Carte SD désactivée
- Le GPU commence par exécuter le programme (bootloader premier étage) en ROM



Séquence de Boot

Démarrage du système d'exploitation : Mise sous tension

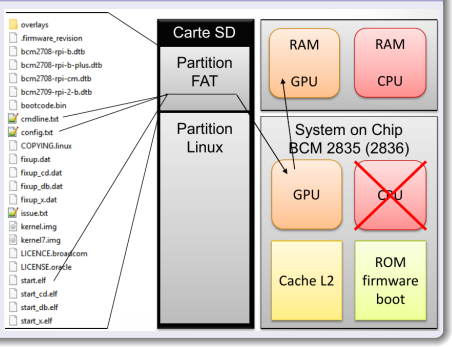
- Chargement du fichier `bootcode.bin` dans le cache et exécution par le GPU
- Met en service la mémoire RAM



Séquence de Boot

Démarrage du système d'exploitation : Mise sous tension

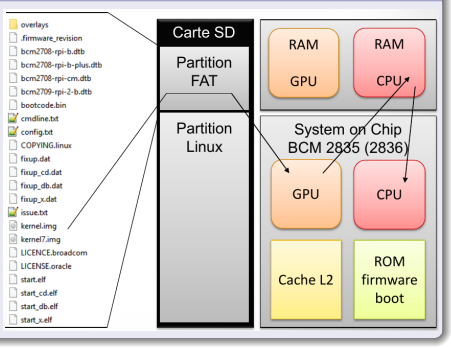
- Le GPU exécute `start.elf`
- Chargement de `config.txt` et `cmdline.txt`
- Séparation de la mémoire en deux zones



Séquence de Boot

Démarrage du système d'exploitation : Mise sous tension

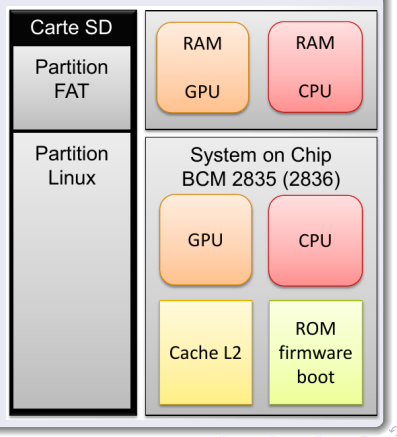
- Chargement du noyau (kernel.img)
- Le GPU cède le contrôle au CPU afin d'exécuter le noyau



Séquence de Boot

Démarrage du système d'exploitation : Mise sous tension

- Le CPU exécute le noyau
- Montage de la partition racine
- Fin de chargement du noyau
- Lancement du premier processus



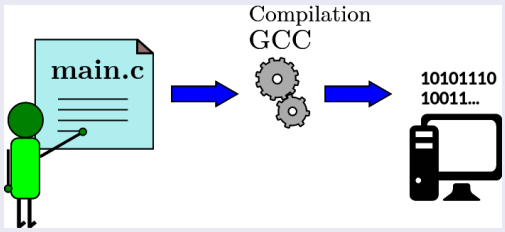
Plan

Les langages C/C++

Les langages C et C++

- Performance (Langages compilés)
- Empreinte mémoire faible
- Portabilité
- Bibliothèques existantes (C depuis 1971, C++ depuis 1983)

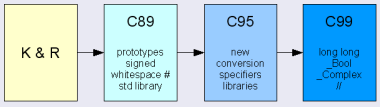
Compilation



Les langages C/C++

Historique

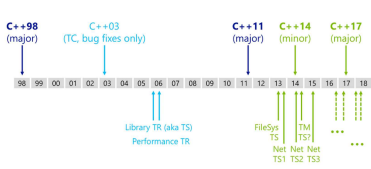
- En 1970, Ken Thompson, créa un nouveau langage : Le B
  - Simple, mais trop dépendant de l'architecture utilisée
- En 1971, Dennis Ritchie commence à mettre au point le successeur du B, le C.
  - Portable,
  - Langage bas niveau : performant (peut créer du code aussi rapide que de l'assembleur)
  - Permet de traiter des problèmes de haut niveau
- En 1989, l'ANSI (American National Standards Institute) normalisa le C sous les dénominations ANSI C ou C89



Les langages C/C++

Historique

- En 1983, Bjarne Stroustrup des laboratoires Bell crée le C++
  - Basé sur le C, il garde une forte compatibilité avec le C
- En 1999, l'ISO (International Organization for Standardization) proposa une nouvelle version de la norme : le C99
  - Reprise de quelques bonnes idées du langage C++
- La première normalisation de C++ date de 1998 (C++98)



Compilation

Principaux compilateurs C/C++

- GCC (GNU Compiler Collection) : il s'agit d'un ensemble d'outils Open Source analysant plusieurs langages (C, C++, Objective C, Ada, Go, Fortran...)
  - Supporte une multitude d'architectures cibles
  - GCC autorise la compilation croisée
  - GCC est disponible sous la plupart des systèmes de type Unix (comme Linux) et sous Windows en utilisant l'environnement Cygwin
- Clang : c'est la principale alternative au compilateur C++ de GCC
  - Il s'intègre au sein du projet plus global LLVM (framework de développement de compilateur) pour générer une représentation compilée intermédiaire traduisible ensuite sur différentes architectures
  - Clang, de conception plus récente, possède une architecture plus modulaire que GCC et permet d'obtenir des programmes offrant des performances plus ou moins similaires à GCC
- Intel C++ Compiler : compilateur propriétaire (et gratuit) est développé par Intel pour la compilation à destination de ses propres processeurs (de type x86) en utilisant certaines optimisations maison.
- Visual C++ Compiler : compilateur propriétaire de Microsoft pour Windows accompagne l'environnement de développement Visual Studio de Microsoft

Compilation

Compilateur GCC

- GCC est un logiciel libre capable de compiler divers langages de programmation, dont C, C++, Objective-C, Java, Ada et Fortran
- Pour faire référence précisément aux compilateurs de chaque langage

  gcc : le compilateur C de GNU
  g++ : le compilateur C++ de GNU
  gcj : le compilateur Java de GNU
  gobjc : le compilateur Objective-C de GNU
  gobjc++ : le compilateur Objective-C++ de GNU
  gnat : le compilateur Ada de GNU
  gfortran : le compilateur Fortran de GNU

Compilation

Compilateur GCC

- Installer GCC

```
$ sudo apt-get install build-essential
```

- Compiler un programme
  - Programme écrit C

```
$ gcc main.c fonctions.c -o Programme
```

  - Programme écrit en C++

```
$ g++ main.cpp fonctions.cpp -o Programme
```

- Pour le lancer le programme

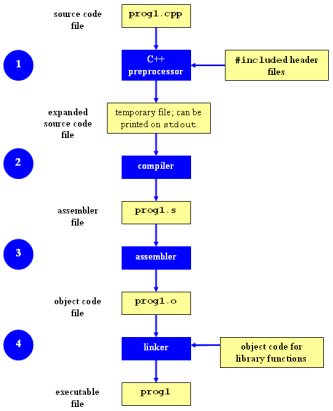
```
$ ./Programme
```

Compilation

Toolchain

Un toolchain est constitué par un ensemble d'outils permettant de produire le fichier exécutable

- C++ Preprocessor : Inclure le contenu des fichiers entêtes (headers), générer les macros et remplacer les constantes définies par *#define*
  g++ $-E$ fichier.cpp
- Compiler : Traduire en assembleur
  g++ $-S$ fichier.cpp
- Assembler : produire fichier objet binaire en traduisant les instruction assembleur en langage machine (binaire)
  g++ $-c$ fichier.cpp
- Linker : édition des liens du fichier objet avec les autres fichiers objets de n'importe quelle bibliothèque utilisée afin de produire le fichier exécutable

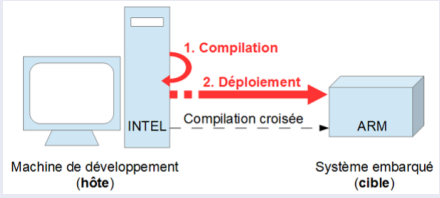


Compilation native et croisée

Compilation native sur un système embarqué

- ✔ Simple
- ✖ Charge de la compilation (temps)
- ✖ Installer un IDE, des outils, des bibliothèques, etc.

Compilation croisée

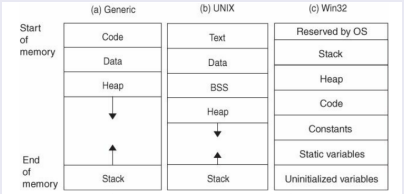


Modèle mémoire

Un processus

Un processus est une instance d'un programme en cours d'exécution chargé en mémoire vive

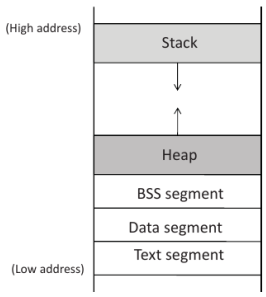
Organisation de la mémoire d'un processus



Modèle mémoire

Modèle mémoire

- Code/Text : contient les instructions du programme (lecture seule)
- Data : variables globales et statiques initialisées par le programmeur
  static int v=0 ;
- BSS (Block Started by Symbol) : variables globales et statiques non initialisées
  Exemple : static int v ;
- Heap : allocation dynamique
  Exemple : int *p=new int ;
- Stack : fonctionne en mode LIFO ; stockage de variables locales (définies dans une fonction) et les informations relatives aux appels des fonctions



MERCI POUR VOTRE ATTENTION

Questions ?