

La conception de systèmes embarqués pour l'Internet des Objets

14 novembre 2023

Objectifs

- **La conception d'un système embarqué: cas concrets**
 - Qu'est ce qu'un système embarqué ?
 - Les grandes étapes de conception: de l'idée à la commercialisation.
 - Évoquer des problématiques récurrentes
 - Les microcontrôleurs
 - La radio
 - La gestion de la consommation
 - Les outils de débogages
 - Les BUS
 - L'industrialisation



NOUS
DONNONS VIE
À VOS PRODUITS



CONCEPTEUR DE SOLUTIONS INTELLIGENTES

UNE VOCATION

Donner vie à vos produits



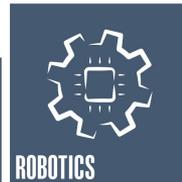
NEW MOBILITY



MEDICAL DEVICES



SMART ENERGY



ROBOTICS



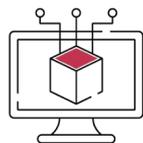
CONSUMER



R&D



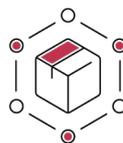
MAKER



ARCHITECT



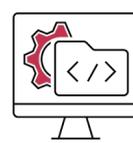
PROJECT MANAGEMENT



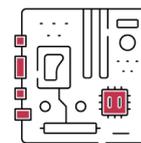
INDUSTRIALIZATION



MECHANICS



SOFTWARE



HARDWARE

Rtone partners



UN PARTENAIRE UNIQUE

avec 8 domaines de compétences

8 domaines de compétences pour vous accompagner dans **TOUTES** les étapes de votre projet.

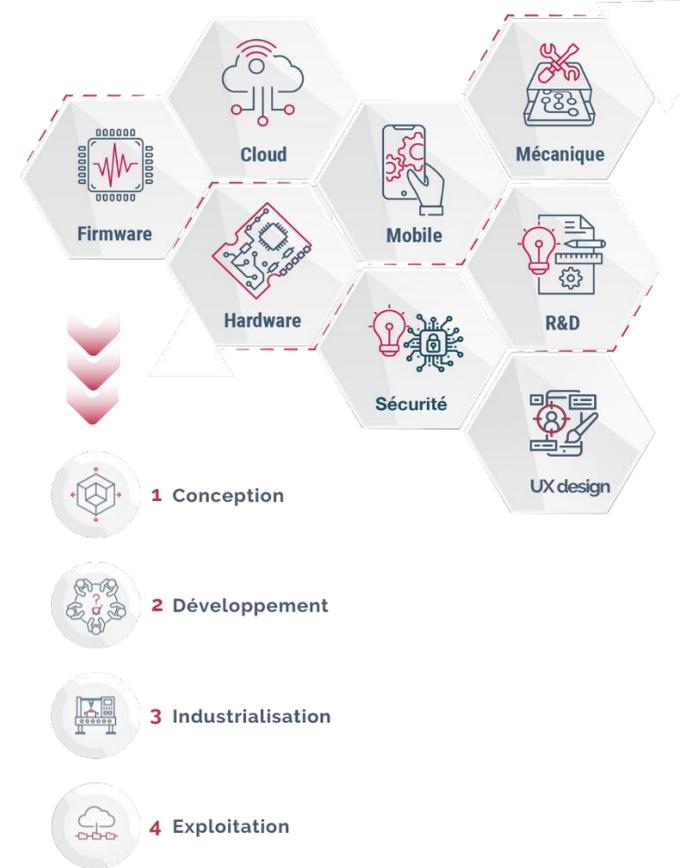
De la définition des usages à l'exploitation des données, notre équipe vous accompagne dans la conception de votre **produit intelligent et connecté**.



Grâce à l'association d'un bureau d'études maîtrisant le hardware et le digital ainsi qu'un écosystème de partenaires de production, Rtone apporte des solutions globales et innovantes sur le marché des systèmes IoT.



Rtone est à même de vous apporter une solution technique pertinente pour réussir à valoriser et sécuriser votre passage au business des objets connectés.



A vos côtés de l'idée jusqu'à la commercialisation et le succès de votre produit.

Rtone - Quelques Références

- Rbee Solar – Web App de monitoring des installations photovoltaïques

Rtone a conçu la Web App Rbee Solar, et développé des compteurs photovoltaïque connectés et intelligents.

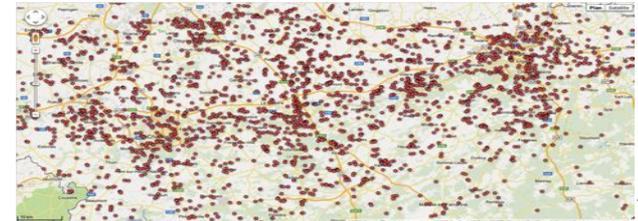
Rbee Solar est une solution complète permettant le contrôle et l'optimisation de l'efficacité énergétique des installations photovoltaïques.

- 20 000+ maisons équipées dans 10 pays, 4 continents, 5 langues.
- Connection GPRS.
- Facturation et paiement en ligne.
- Projet de recherche européen FP7 PVCrops

Rtone - Quelques Références



- Rbee Solar – Web App de monitoring d'installations PV



Rtone - Quelques Références



- SOGEXI Lacroix – Plateforme de gestion Smart City

Gateway Linux 3G et application Web de contrôle et monitoring de l'éclairage publique. Et maintenant panneaux et feux tricolores

- 20 000 lampadaires connectés.
- Panneaux à messages variables
- Feux tricolores
- 400 villes.
- 10 Pays
- 3G et CPL



Rtone - Quelques Références

- Schneider / Energy Pool – Gateway Linux de “Demand Response”

Software embarqué pour une passerelle de contrôle de consommation électrique de sites industriels et de modulation de la demande d’électricité.

- Software CPL.
- VPN – Sécurité.
- France, Turquie, Norvège, Corée, ..
- 3G.




Rtone - Quelques Références



- Nespresso – Conception Hardware de machine à café

Upcoming Nespresso coffee machine (2016) was co-designed by Rtone

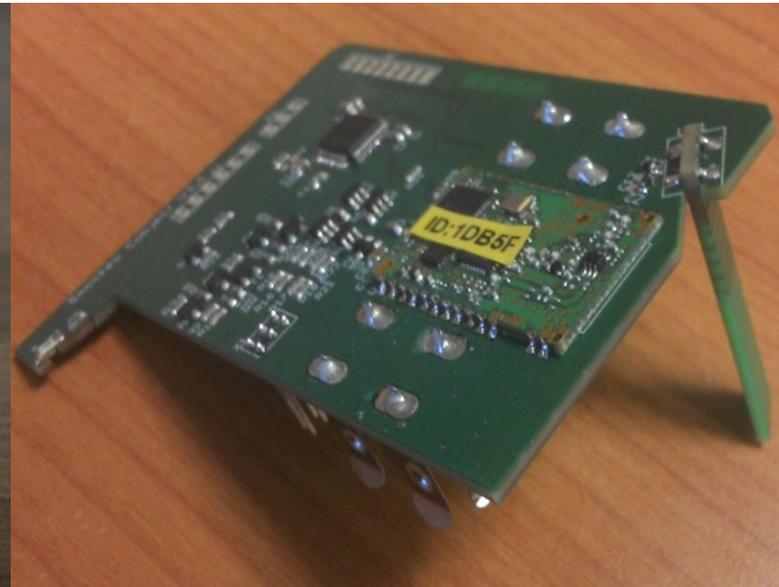
- 3 Millions d'unités par an
- Hardware low cost (forte contrainte)
- Optimisation des coût
- Optimisation mémoire firmware
- Certification au niveau mondial



We are not allowed to unveil the 2016 new design

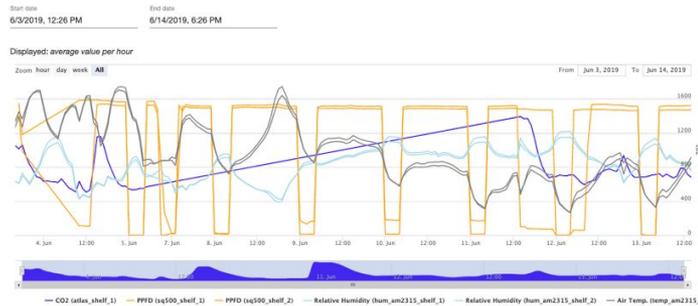
Rtone - Quelques Références

- CObject - Détecteur de Fumée SIGFOX



Rtone - Quelques Références

- La Grangette - Indoor Farming



Unit sensors

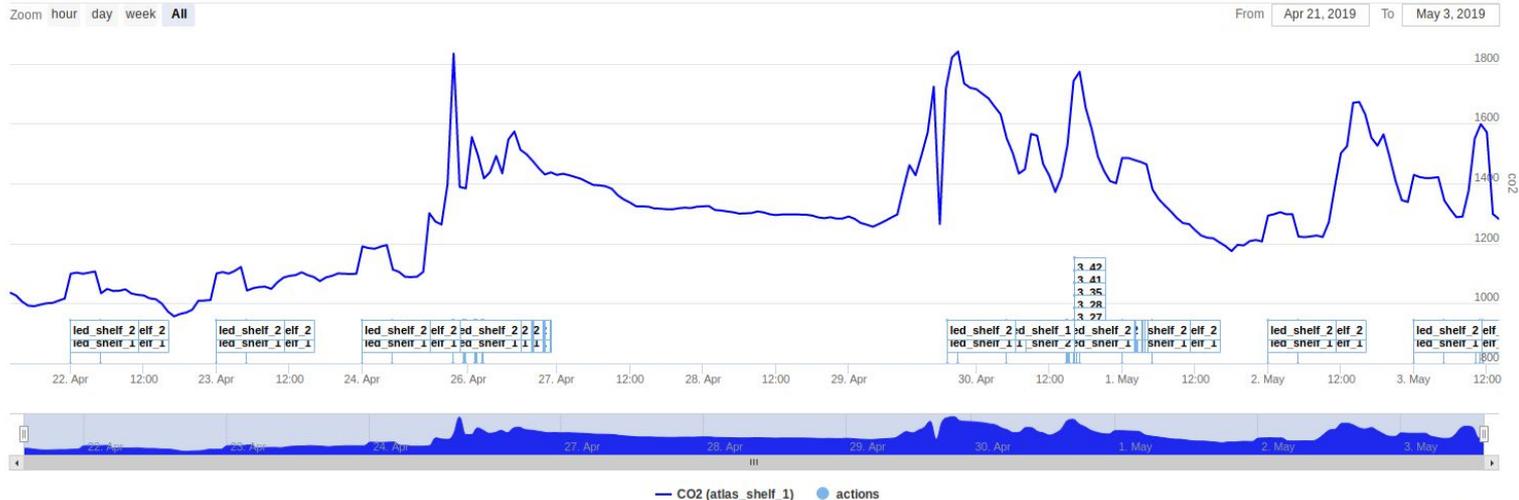
- CO2 atlas_shelf_1
- Dissolved Oxygen do_atlas_pot
- Electrical Conductivity ec_atlaspro_pot
- Oxido-Reduction Potential orp_atlas_pot
- PPF0 sq500_shelf_1
- PPF0 sq500_shelf_2
- pH ph_atlaspro_pot
- Relative Humidity hum_am2315_shelf_1
- Relative Humidity hum_am2315_shelf_2
- Air Temp. temp_am2315_shelf_1
- Air Temp. temp_am2315_shelf_2

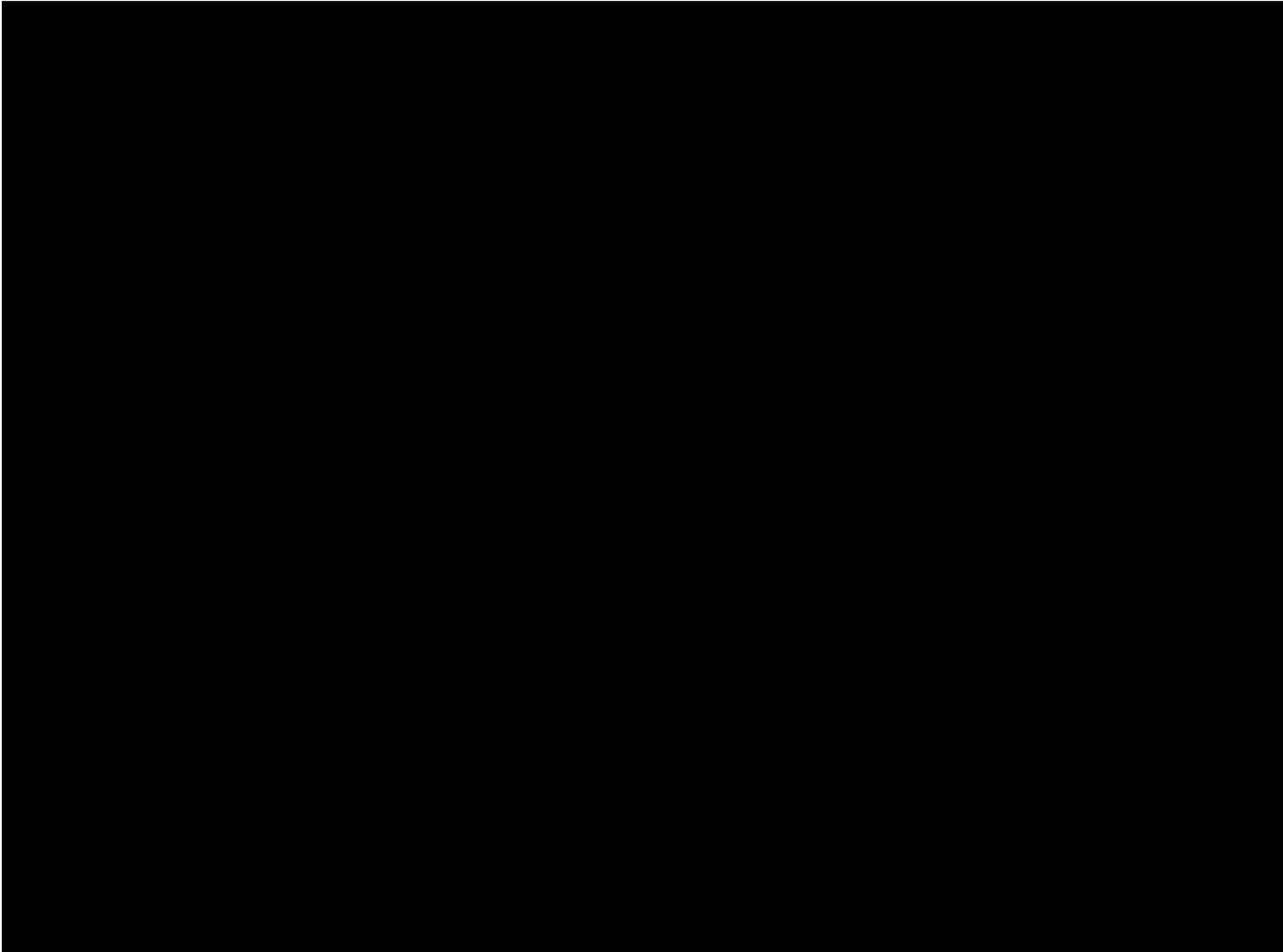
Action types

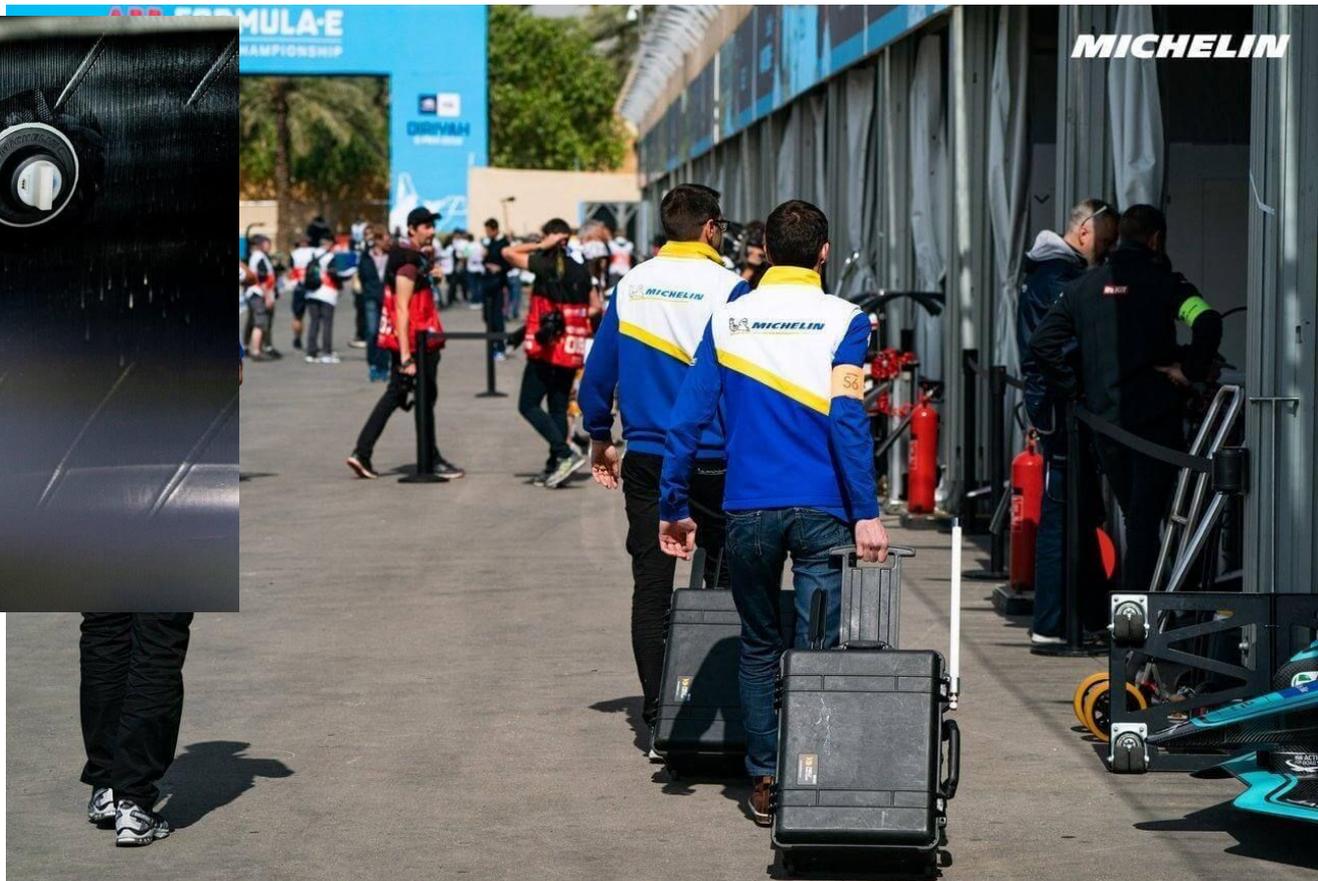
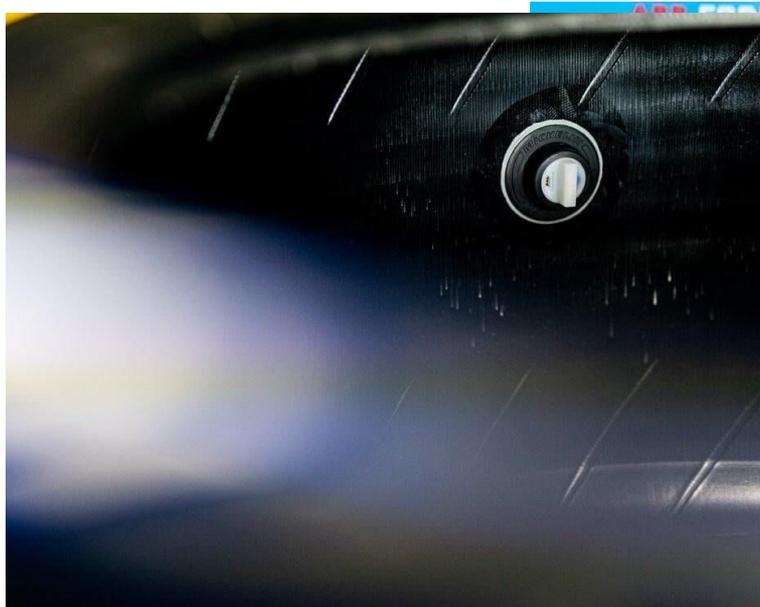
- planting
- irrigation
- light
- nutrition

Start date: 4/21/2019, 4:47 PM End date: 5/3/2019, 4:47 PM

Displayed: average value per hour







Rtone - Quelques Références



- In&Motion – 1er Airbag destiné aux skieurs professionnels dans le monde

Rtone à conçu un airbag destiné aux skieurs professionnels dans le monde.

- Approuvé par la Fédération Internationales de Ski (FIS)
- 7 capteurs
- Connexion sans fils pour détecter la perte de ski.
- Application mobile de configuration et de collecte des données.

Rtone - Quelques Références



sw

deign





THE NEW SMART TORQUE WRENCH

Rtone - Quelques Références

- Tracker intégré dans montre de luxe
 - Communication par 2G
 - GPS, accéléromètre, ...
 - Batterie custom
 - Intégration dans le cuir
 - Application web



Rtone - Quelques Références



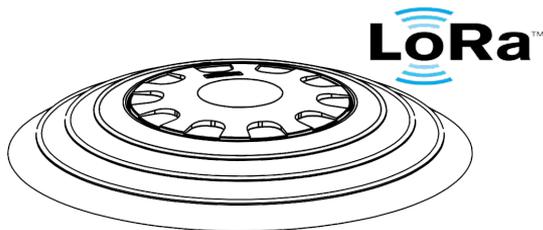
- Localisation Indoor pour Proximity Marketing
 - Transmission d'information par ultrason
 - Beacon BLE



Rtone - Quelques Références

- Mais aussi ...

- Cigarette électronique “smart”
- Plateforme IoT
- Etui de charge par induction pour smartphone QI
- Bracelet d’alerte
- Radiateur connecté
- Borne de parking Lora



Rtone - Quelques Références



FROM OUR R&D *(WITH LOVE)*



kiwink
VISIBLE LIGHT COMMUNICATION

A zero cost smartphone connection

KLOCZ

Leading-edge IoT Geolocation

Rtone - R&D

- Module IoT de localisation indoor/outdoor

- GPS, LPWAN, BLE, WiFi, GSM
- Accéléromètre, Gyroscope, ...

- Projets R&D Collaboratif

- FUI LUG2 : plateforme de visualisation 3D
- FUI PACLIDO
- H2020 FED4Fire+
- BEES-ML



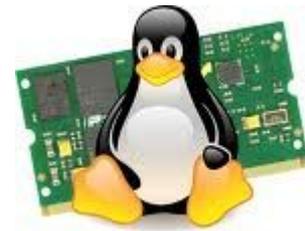
Les système embarqués

- Microcontrôleurs

- SoC, Firmware, Temps réel
- Proche du hardware - le “Soft Hard”



yocto
PROJECT



- Gateway Linux

- Linux embarqués : Project Yocto, ARM Poky, .
- Milieu industriels : pilotage des automates, contrôle à distance ...
- Interface Ethernet, Wifi, ModBus, GPRS, EnOcean

- Autour d'Android

- L'UNIX embarqué “grand public” : écran d'information, UI
- Smartphone, “l'IoT Gateway”
- Innovation : le smartphone modulaire



La conception d'un système embarqué

- Avant Projet
 - Spécification, Faisabilité, Choix techniques
- Hardware
 - Peu flexible, temps de développement long
 - Coûts importants, normes, certifications
 - Délai de production, stock, gestion de l'obsolescence
- Software
 - Firmware, smartphone, serveurs
 - Développements plus flexibles
 - Attention aux standards
 - Exploitation, mise à jour, ...



Les Etapes

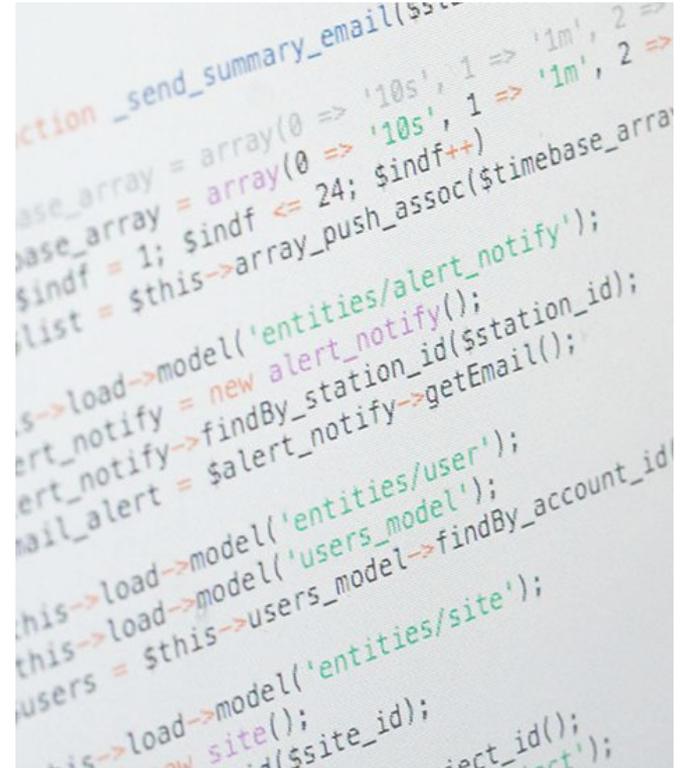
● Hardware

- Pré-étude
- Choix des
- Prototypage rapide
- Conception carte
- Routage
- Run de prototypes (2 en général)
- Pré-série
- Certifications
- Série
- Industrialisation (tests, production, ...)



Les Etapes

- Software
 - Mockup
 - Prototypage **rapide** (sur dev. board)
 - Agilité
 - Portage sur **PCB**
 - Débogage
 - Tests **d'Integration**
 - Production
 - Maintient en conditions opérationnelles



```
function _send_summary_email($st...
base_array = array(0 => '10s', 1 => '1m', 2 =>
base_array = array(0 => '10s', 1 => '1m', 2 =>
$indf = 1; $indf <= 24; $indf++)
$list = $this->array_push_assoc($timebase_array,
$this->load->model('entities/alert_notify');
alert_notify = new alert_notify();
alert_notify->findBy_station_id($station_id);
mail_alert = $alert_notify->getEmail();
$this->load->model('entities/user');
$this->load->model('users_model');
users = $this->users_model->findBy_account_id($
$this->load->model('entities/site');
new site($site_id);
site_id();
site');
```

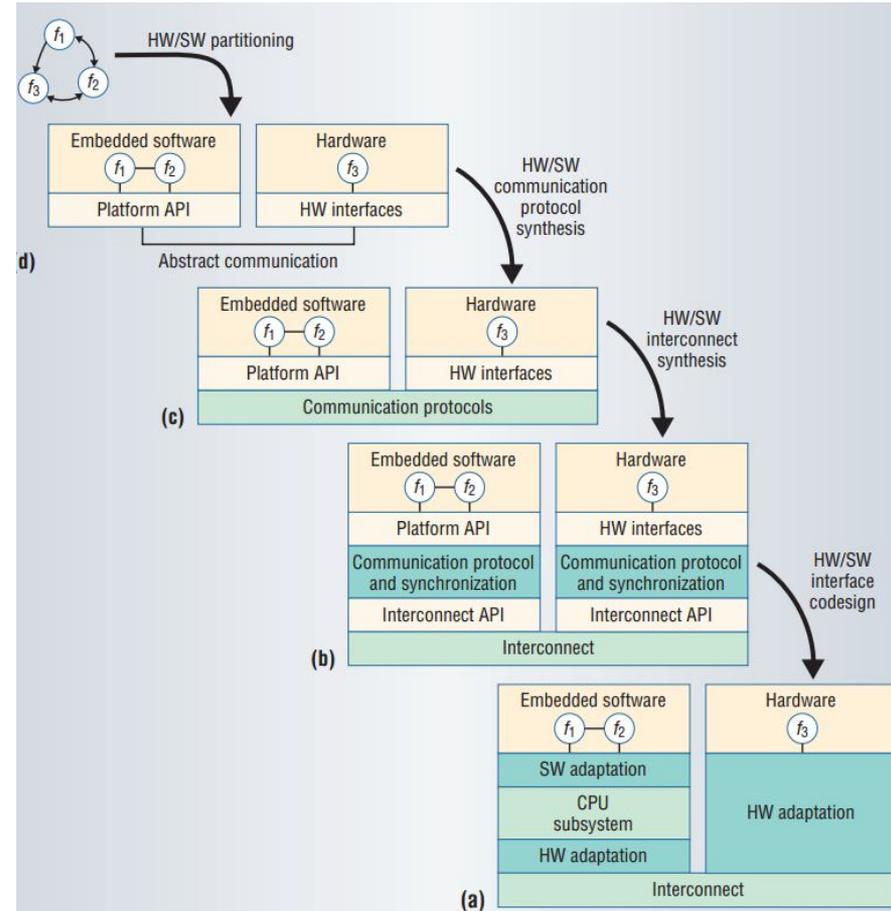
Le Codesign

- Definition
 - Une méthodologie de conception qui supporte le **développement coopératif** et concurrent du **matériel** et du **logiciel** (cospécification, co-développement, et co-vérification) afin d'obtenir des fonctionnalités partagées et d'**atteindre les performances espérées** .

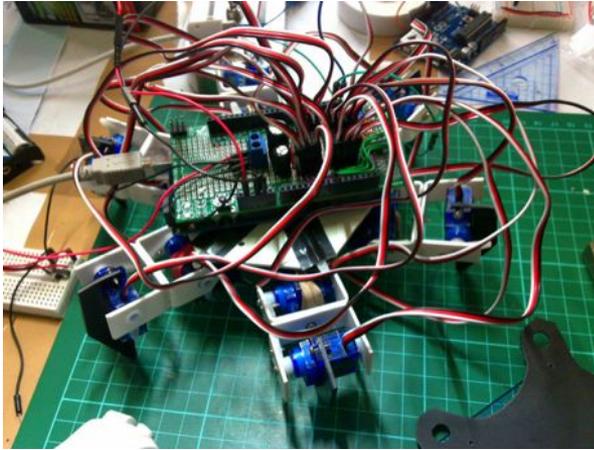
Le Codesign - Les Etapes

- Specification
 - Lister les fonctionnalités
- Modélisation
 - Conceptualisation
 - Affiner les spécifications
 - Matériel et logiciel
- Partitionnement
 - Partage logiciel/materiel
- Synthèse et optimisation
- Validation
- Intégration

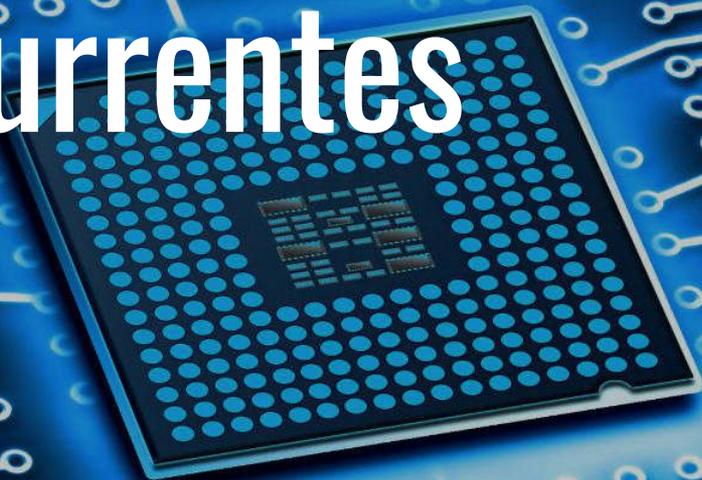
Codesign flow



En Résumé ...



Des problématiques techniques récurrentes

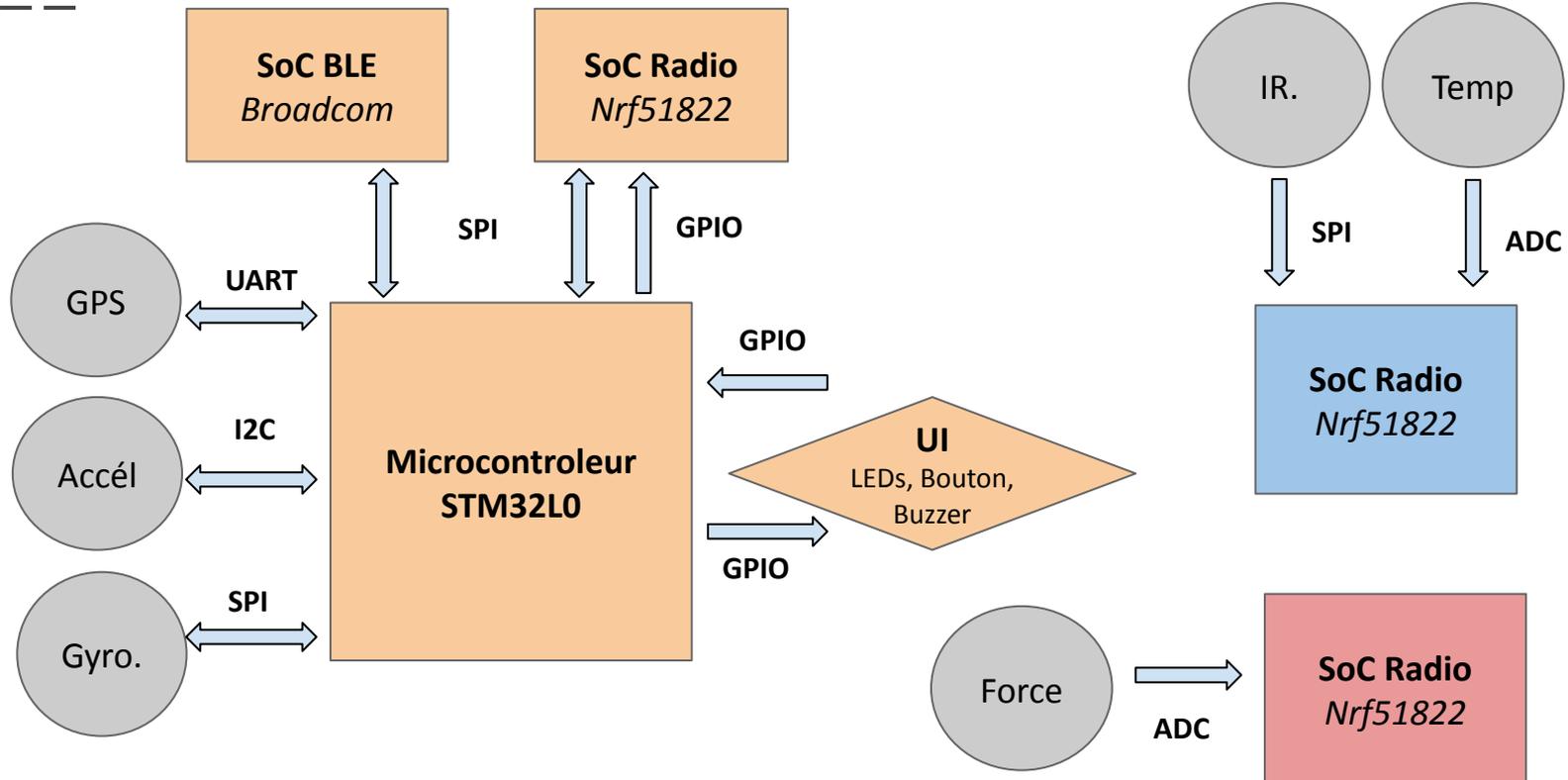


Des contraintes

- Coûts
- Temps Réels
- Tâches préemptives
- Autonomie
- Impacte Mémoire
- Configuration sans fil
- Générer des logs
- Topologie Body Area Network
- Faible latence



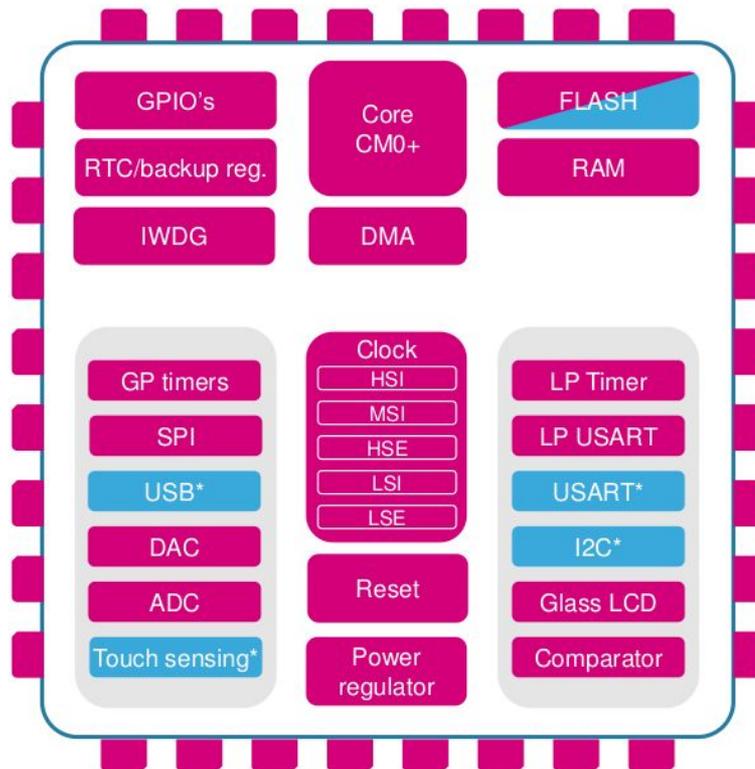
Un exemple de système



Le Microcontrôleur



life.augmented



Différents modes de programmations

- **Scrutation (polling)**
 - Coûteux en temps (multiplié par le nombre de périphériques à interroger)
 - **Bloquant**
 - **Implémentation** : Appel classique à une fonction dans le programme
- **Interruption**
 - Demande à l'initiative du périphérique
 - Prise en compte rapide de l'évènement
 - Non bloquant
 - **Implémentation** : Interruption **asynchrone** d'un programme puis retour au même endroit à la fin du traitement

Interruptions (rappel ?)

- Définition
 - Événement produisant l'interruption du programme en cours d'exécution pour exécuter une routine d'interruption
 - Une interruption provient d'une source d'interruption et peut se produire n'importe quand!
 - Une source d'interruption peut être ignorée (par configuration)
- Une routine d'interruption (ISR) doit
 - Sauvegarder le contexte (valeur des registres)
 - Prendre en charge la demande d'interruption et la traiter (le + rapidement possible)
 - Restaurer le contexte
 - Retour au programme

Interruptions sur STM32

- Le microcontrôleur STM32 a deux types d'interruptions :
 - Les exceptions système (HardFault, ...)
 - Les interruptions externes numéroté de 0 à 59
- Le numéro de l'interruption en cours de service est connue par:
 - le registre special IPSR (Interrupt Program Status Register)
 - le registre ICSR (Interrupt Control State Register) du NVIC (Nested Interrupt Controller) dans le champs VECTACTIVE
- Les interruptions ont des priorités permettant d'avoir un système préemptif

Définir les priorités et partage des ressources

- Quels sont les événements prioritaires ?
 - Temps réel “soft” ou “hard” ?
 - Interruption Timer pour allumer une LED ?
 - Interruption SPI pour recevoir une donnée cruciale ?
 - Radio ?

Définir les priorités

- Une interruption plus prioritaire doit **interrompre** une interruption **moins prioritaire** en cours d'exécution
- Une interruption moins prioritaire **ne doit pas interrompre** une interruption plus prioritaire en cours d'exécution
- Fixer des **priorités non modifiables**
- Elle doit offrir la possibilité de choisir le niveau de priorité de la plupart des sources d'interruption

Modes de programmations

— — —

- Un peu de code ... **Scrutation** (polling)

```
int main(void)
{
    HAL_Init();
    /* Configure the system clock to 24 MHz */
    SystemClock_Config();
    /* -1- Enable each GPIO Clock (to be able to program the configuration registers) */
    LED3_GPIO_CLK_ENABLE();
    LED4_GPIO_CLK_ENABLE();

    /* -2- Configure IOs in output push-pull mode to drive external LEDs */
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_PULLUP;
    GPIO_InitStruct.Speed = GPIO_SPEED_HIGH;
    GPIO_InitStruct.Pin = LED3_PIN;
    HAL_GPIO_Init(LED3_GPIO_PORT, &GPIO_InitStruct);

    GPIO_InitStruct.Pin = LED4_PIN;
    HAL_GPIO_Init(LED4_GPIO_PORT, &GPIO_InitStruct);
}
```

```
/* -3- Toggle IOs in an infinite loop */
while (1)
{
    HAL_GPIO_TogglePin(LED3_GPIO_PORT, LED3_PIN);
    /* Insert delay 100 ms */
    HAL_Delay(100);
    HAL_GPIO_TogglePin(LED4_GPIO_PORT, LED4_PIN);
    /* Insert delay 100 ms */
    HAL_Delay(100);
}
}
```

Modes de programmations

— — —

- Un peu de code ... **Interruption**

```
int main(void)
{
    HAL_Init();

    /* Configure the system clock to 24 MHz */
    SystemClock_Config();

    /* -1- Initialize LEDs */
    BSP_LED_Init(LED3);

    /* -2- Configure EXTI_Line0 (connected to PA.00 pin) in interrupt mode */
    EXTI0_IRQHandler_Config();

    /* Infinite loop */
    while (1)
    {
    }
}
```

```
/**
 * @brief EXTI line detection callbacks
 * @param GPIO_Pin: Specifies the pins connected EXTI line
 * @retval None
 */
void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if (GPIO_Pin == GPIO_PIN_0)
    {
        /* Toggle LED3 */
        BSP_LED_Toggle(LED3);
    }
}
```

Optimiser son code

- Complexité Algorithmique
 - Rappels (... 3TC ...) : $O(n)$, $O(n^2)$
 - Exemple : Comment calculer d'une moyenne glissante ?
 - Optimiser la **mémoire** ?
 - Optimiser l'**efficacité algorithmique** ?
 - Optimiser la **précision** ?
- Optimisation de la mémoire
 - Taille de la flash restreinte
 - Mémoire Volatile (**RAM**) limité
 - Bien choisir la **taille des variables**: uint8, uint16, uint32, char ??

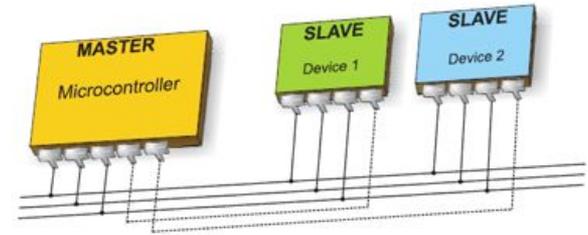
Real Time Operating System

- Avec ou sans RTOS ?
 - IHM, besoin de flexibilité/évolutivité
 - Gestion réseau
- Les composant
 - Ordonnanceur : gestion des tâches (thread)
 - Abstraction Materiel: driver USB, LCD, ...
 - Protocole réseau
- Quelques exemple
 - FreeRTOS : un des + utilisés
 - Zephyr
 - Contiki, RIOT : orientés WSN, réseau
 - CMSIS RTX, ChibiOS, ...

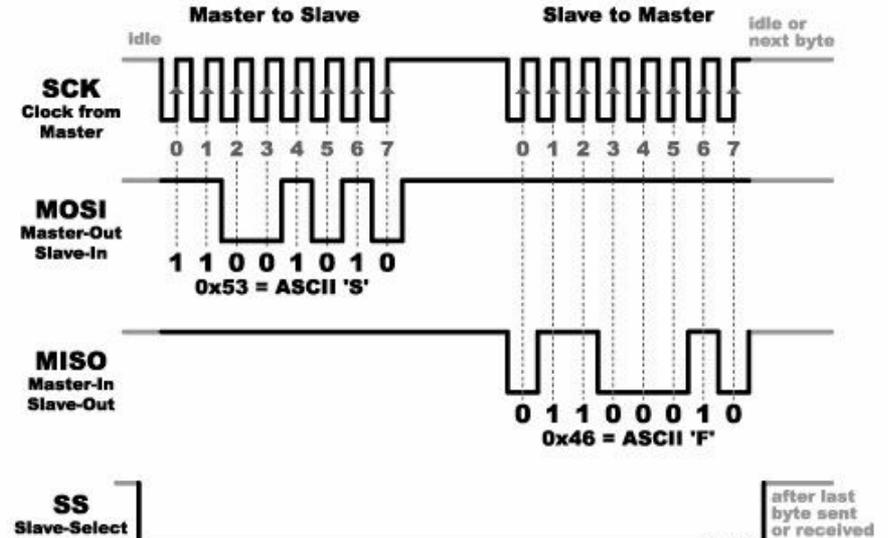
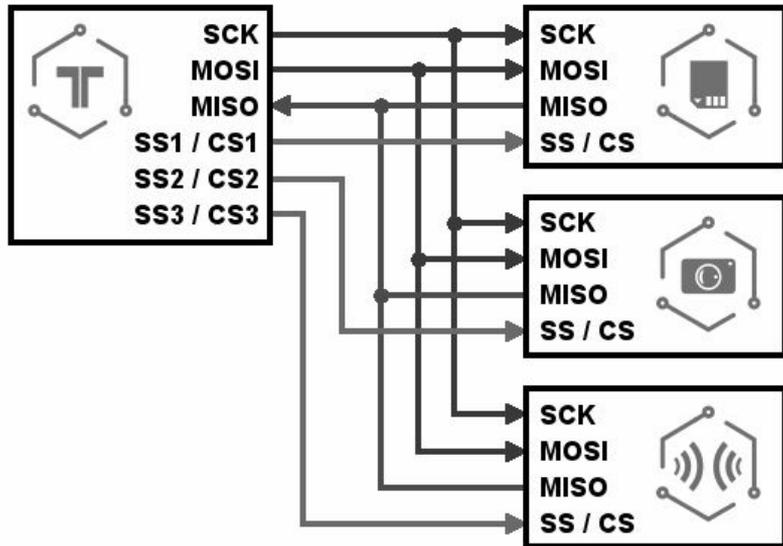


SPI: Serial Peripheral Interface

- 4 Fils
 - **SCK** : horloge
 - **SDO** : sortie des données
 - **SDI** : entrée des données
 - **SS** : choix du périphérique
- Débits pouvant aller jusqu'à qqes MBit/s
- Quelques caractéristiques :
 - **Synchrone**
 - Type maître-esclave (single master, multi slaves)
 - Unidirectionnel (Une ligne par direction)
 - **Full duplex**



SPI: Serial Peripheral Interface

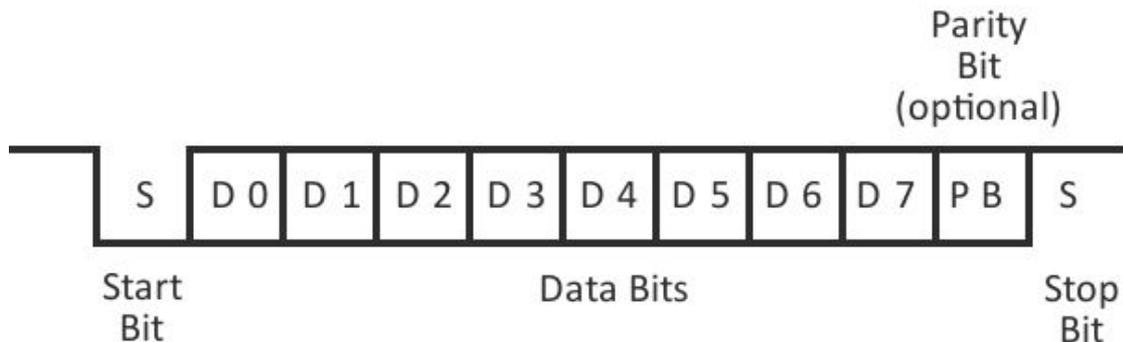


SPI: Serial Peripheral Interface

- Avantages
 - Débit plus important que pour le protocole I2C
 - Aucun arbitre car aucune collision possible
 - Très simple
- Inconvénients
 - Monopolise plus de Pin que l'I2C
 - Aucun adressage possible, il faut une ligne de sélection par esclave
 - Pas d'acquittement
 - S'utilise sur des plus courtes distances que les liaison série ou CAN

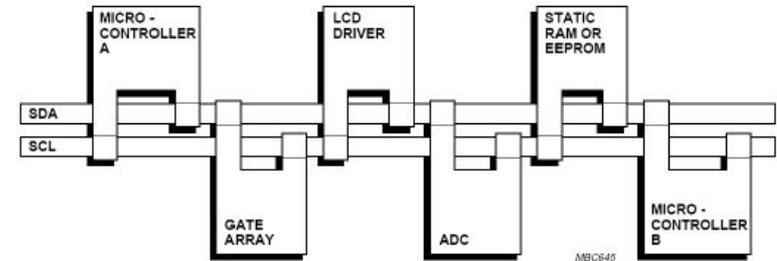
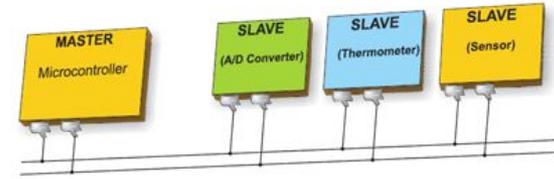
UART : Universal Asynchronous Receiver Transmitter

- Bus asynchrone : pas d'horloge
 - 2 fils pour transmission et réception (TX et RX)
 - Optionnel : 2 fils pour le handshake (RTS/CTS)
- Quelques variations possibles
 - Bits de début et d'arrêt, et format de données fixe



I²C : Inter Integrated Circuit

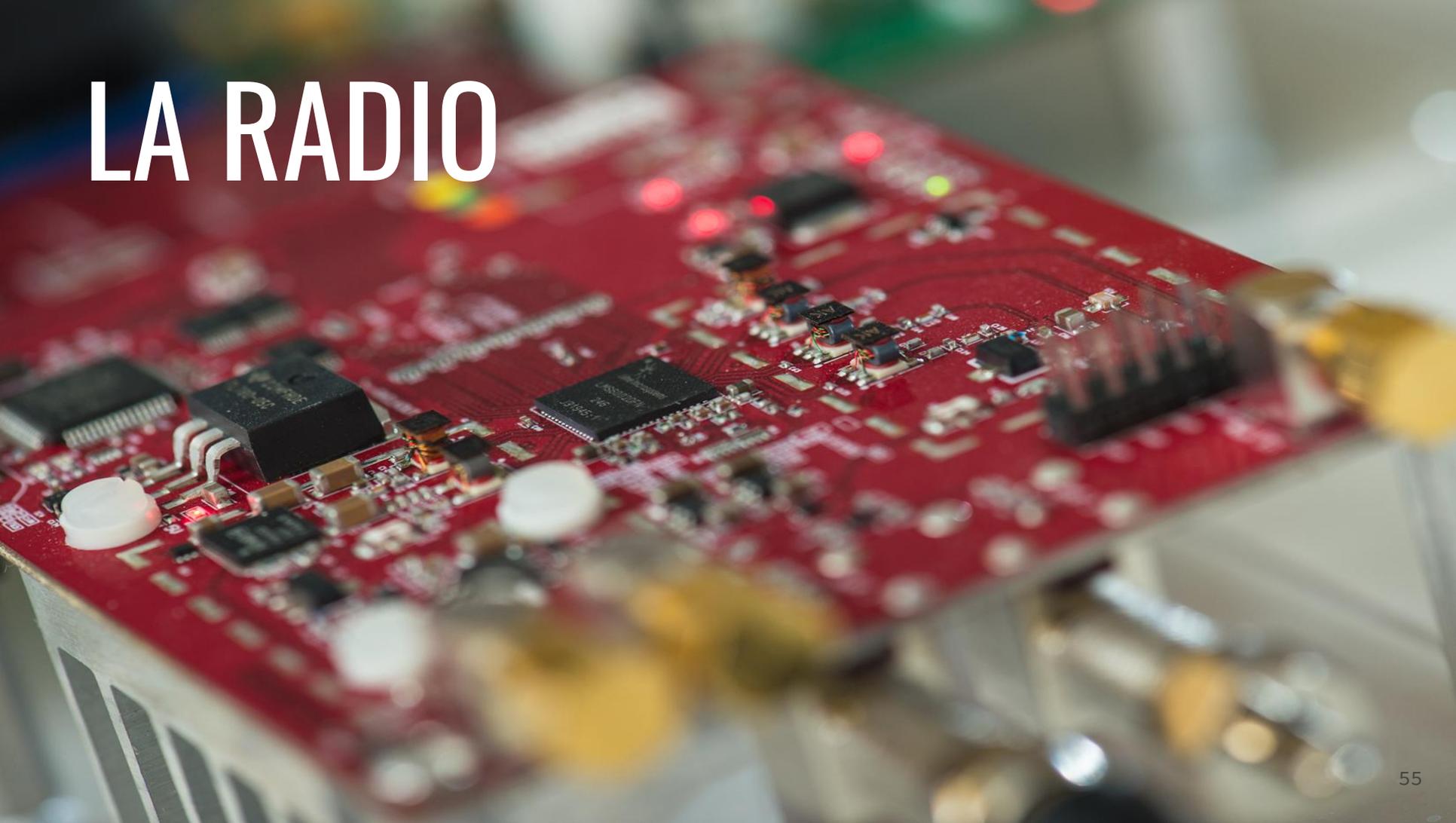
- 2 Fils
 - SDA : signal de donnée
 - SCL : signal d'horloge
 - (Masse : signal de référence)
- Communication synchrone
 - Master/Slave
 - Mode multi-master
- Exemple
 - EEPROM, Afficheur LCD, RTC



Quel BUS / Interface choisir ?

- Interfaces du capteur
- Alternate Functions du microcontrôleur
- SPI
 - Principalement pour les hauts débits, 25MHz environ horloge
 - Maître unique (qui peut être fait via S / W), l'arbitrage de données n'est pas possible
- I2C
 - principalement pour le contrôle et de transfert à basse vitesse
 - environnement bus multi-maîtres, l'arbitrage intégré, ...
- UART:
 - simple communication P2P, 115kbps max,

LA RADIO



Les Interfaces Radio

- Gazell : 2.4 GHz Propriétaire
 - Gazell de Nordic Semiconductor
 - Faible Latence (< 10ms)
- Bluetooth Low Energy
 - Données des capteurs
 - Configuration du système
 - Transfert de log



Bluetooth Low Energy



Low Energy

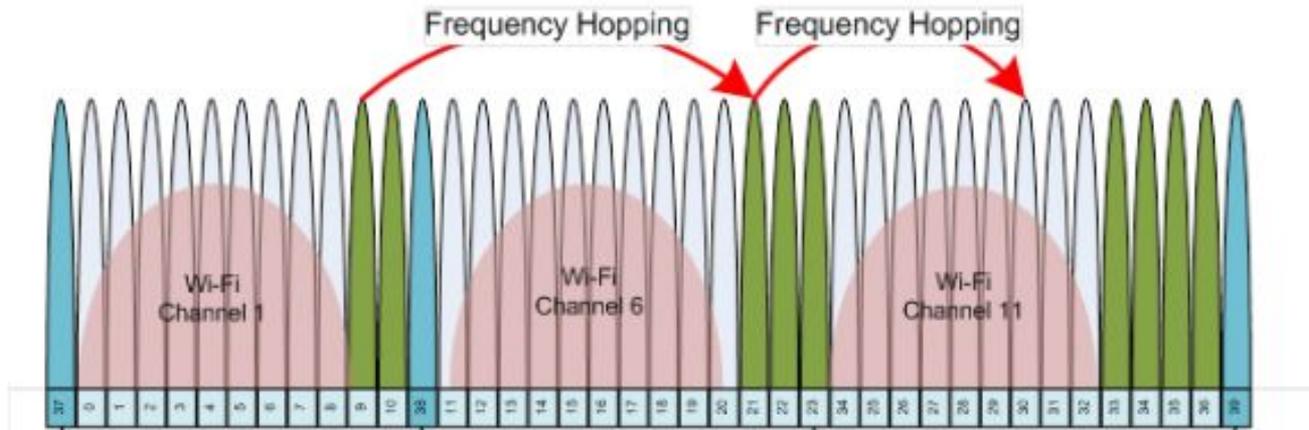


- Intérêt
 - Faible consommation, faible portée, peu de données échangées
 - Connexion avec un smartphone
 - Faible coût
- Low power
 - Intervals de connexion entre 3ms et 4s+
 - Paquets de petite taille : 31 bytes max de payload
 - 1 Mbps on-air
 - tps RF très court // tps en Sleep
- 5.0 : augmentation de la portée, du débit, ...

Bluetooth Low Energy

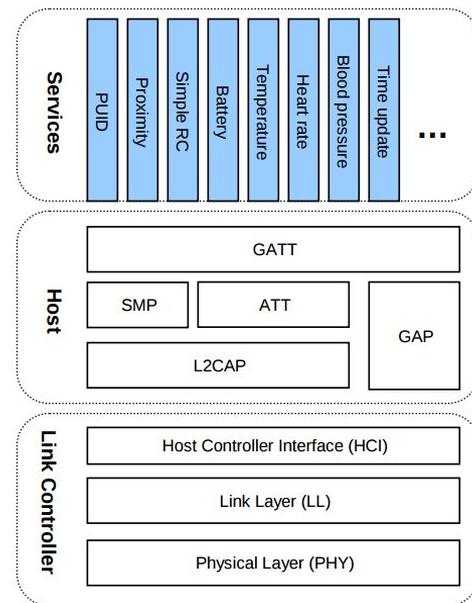
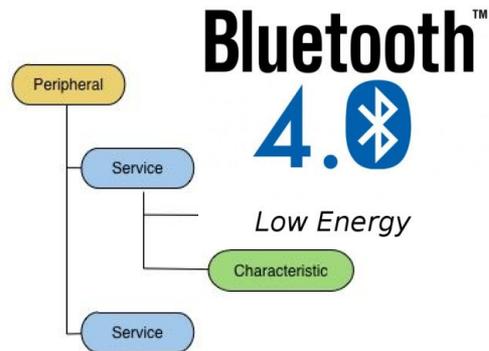


- 2.4 GHz ISM
- GFSK 1 Mbps
- Canaux PHY
 - 3 Canaux Advertising
 - 37 Canaux Data



Bluetooth Low Energy

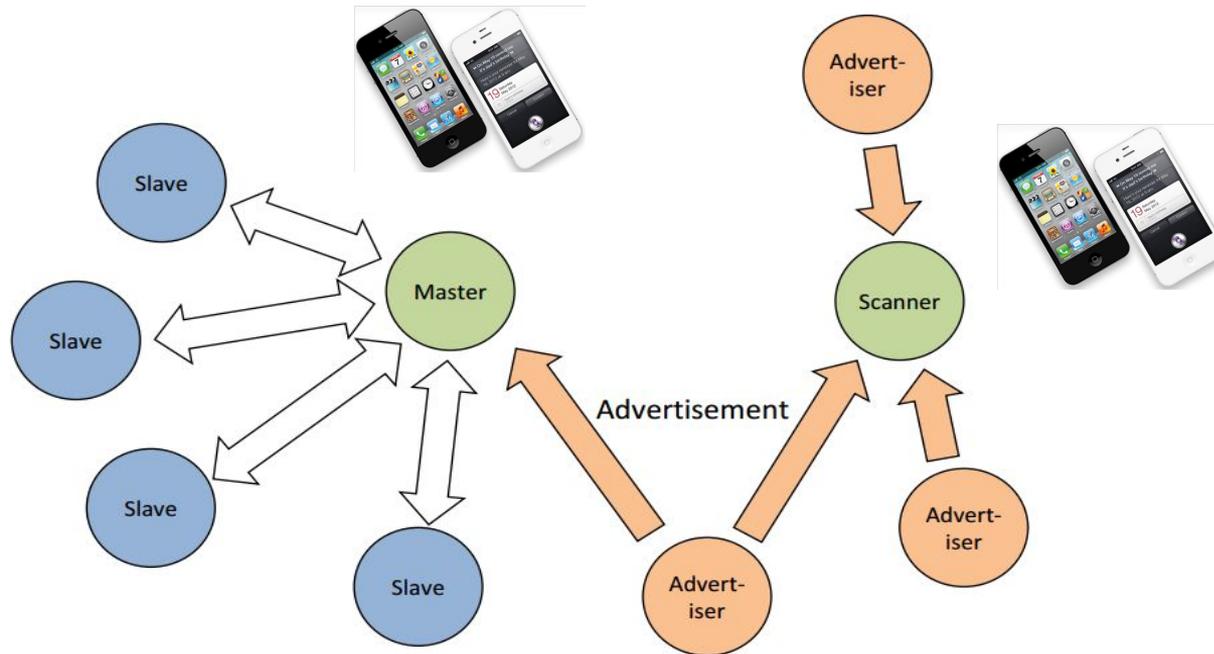
- Le Périphérique
 - Broadcast les advertising
 - Rôle *Slave* lors de la connexion
 - Expose des *Services* et *Caractéristiques* accessibles en écriture/lecture.
 - Possède les données
 - Peut émettre des *Notifications/Indications*
- La Centrale (smartphone)
 - Scan
 - Initie la connexion
 - Rôle *Master*



Bluetooth Low Energy

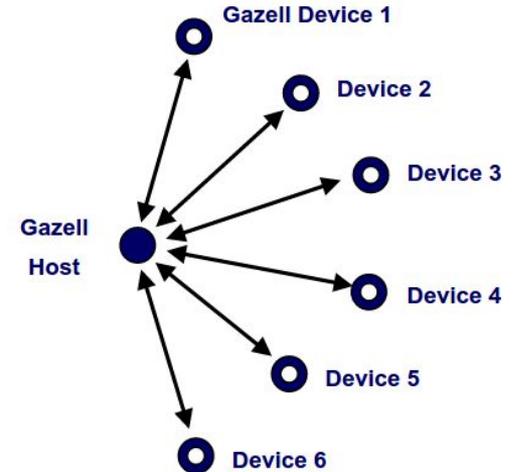


- Topologie



Protocole Radio Gazell 2.4GHz

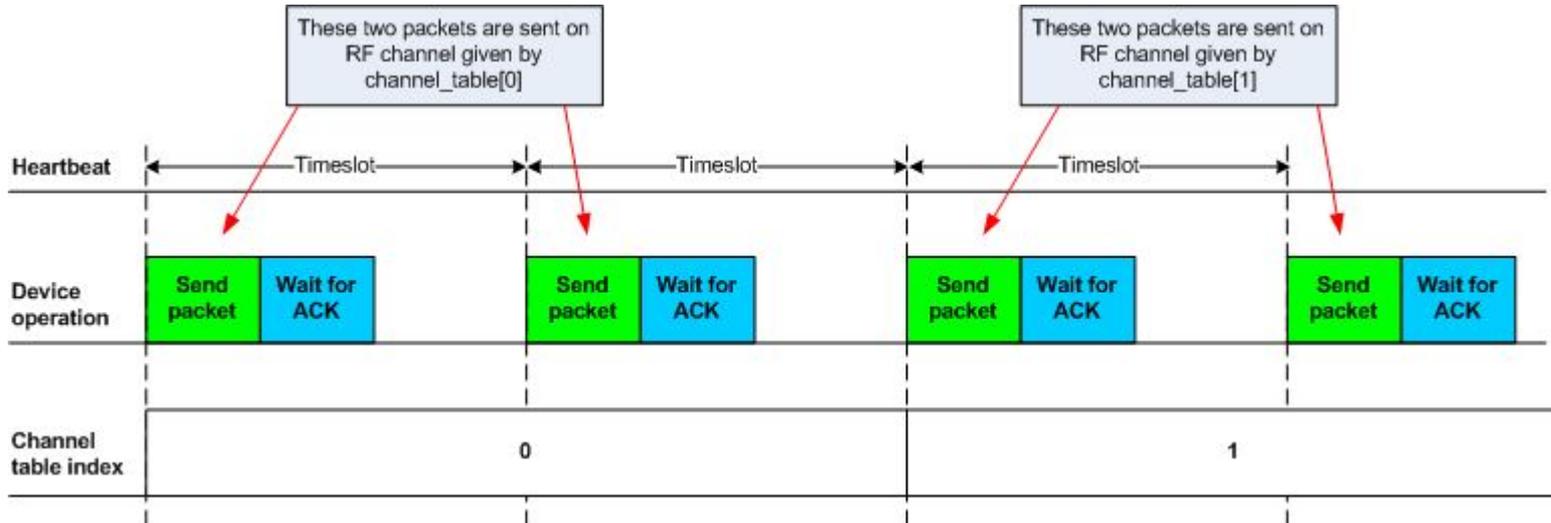
- Protocole propriétaire Nordic Semiconductor
- Bande ISM 2.4GHz
- Topologie en étoile 6-1
- Très faible latence: 2 ms
- Frequency Hopping
- Bi-directionnel
 - Les devices initient toujours la connexion



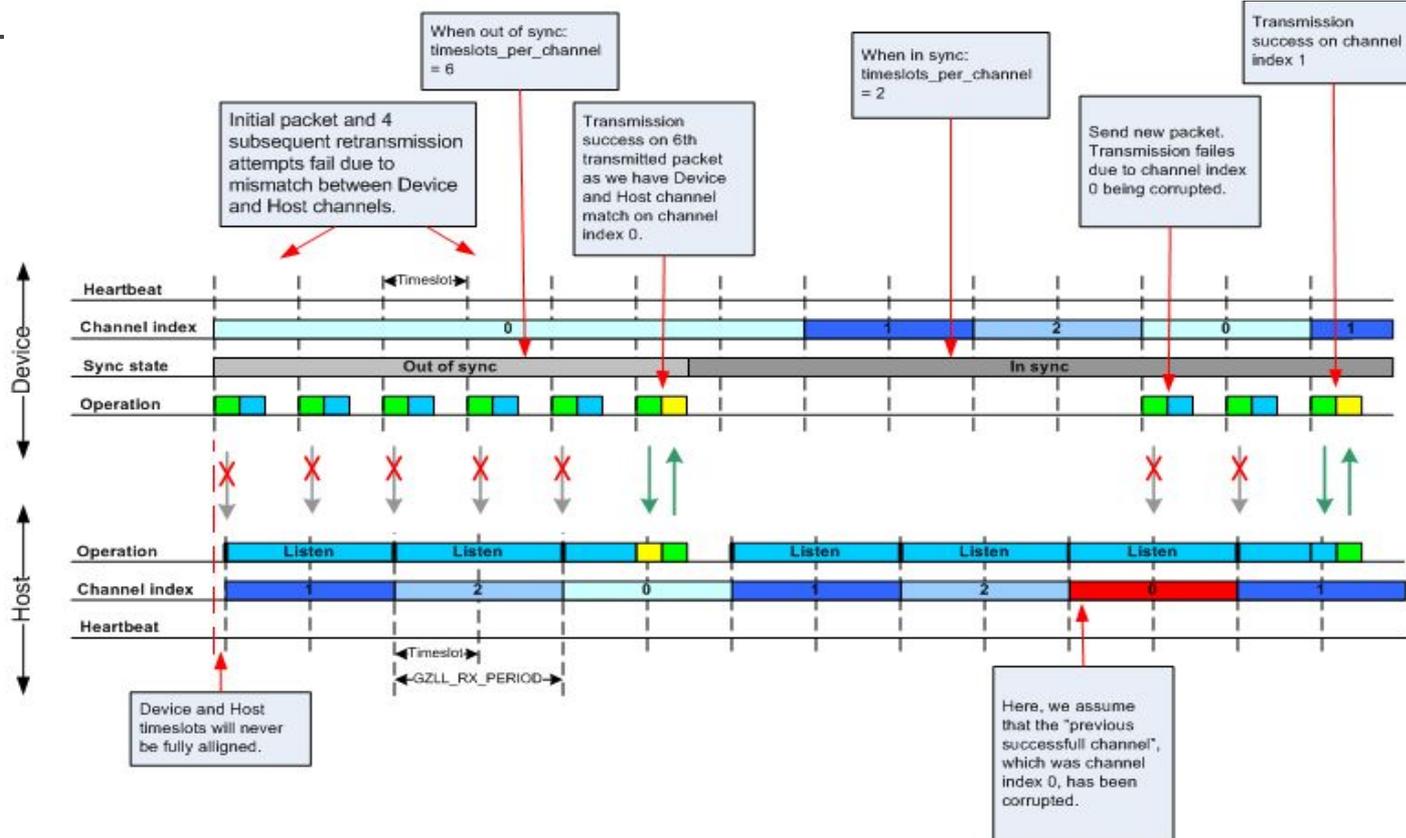
Protocole Radio Gazell 2.4GHz



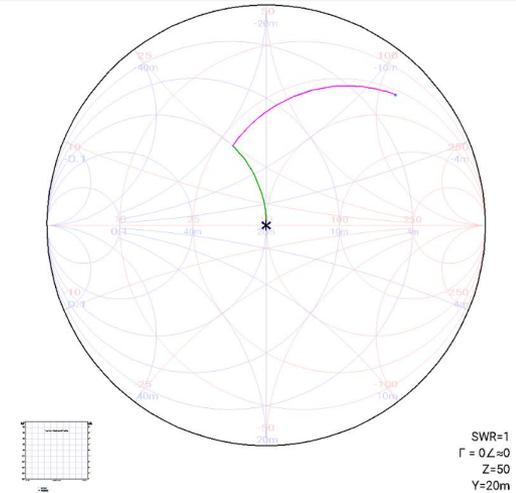
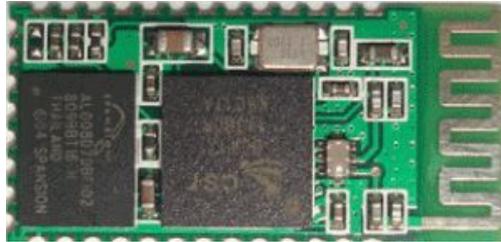
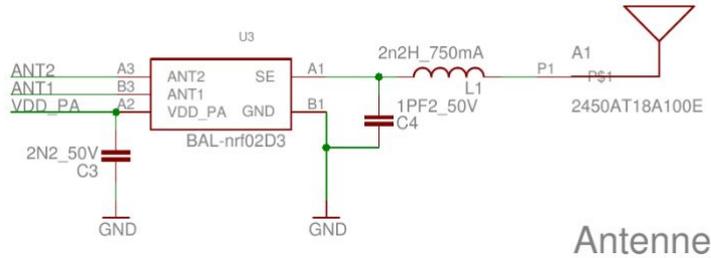
- Principe du Frequency Hopping
 - timeslot de 600us



Protocole Radio Gazell 2.4GHz

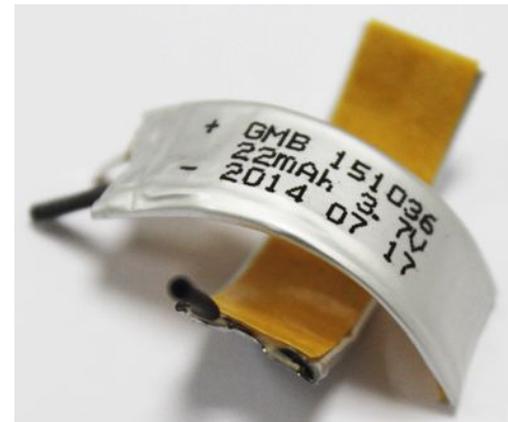


Attention aux antennes !



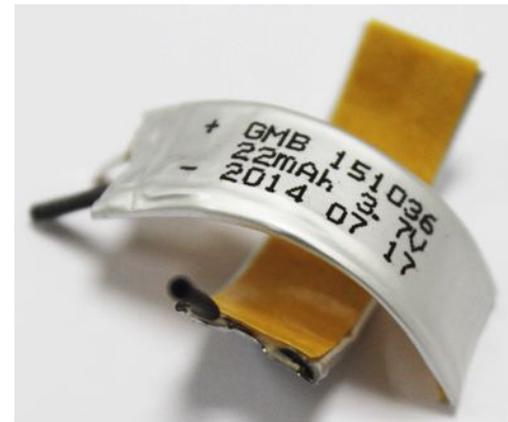
La gestion de l'énergie

- Autonomie, recharge, durée de vie, ...
- Difficulté souvent occultée de l'IoT
- Bien choisir la source d'énergie
 - TX BLE=27 mA, Sigfox=65 mA, GSM/GPRS : 2 A
 - Le RX en attente de trames peut être très « coûteux » !
- Piles / Batteries
 - Qualité, qualité
 - Attention à l'autodécharge des batteries Lithium (2 à 5% par mois)
- Harvesting
 - Encore rare et difficile à mettre au point
 - Technologies d'avenirs



Points de Vigilances ...

- Modes Low Power du MCU
 - Suis-je dans **le bon mode LP** ?
- GPIOs
 - fuites de courant
- Capteurs
 - à mettre en **mode LP** / désalimenté
- Hardware
 - Pull Up, Capacité, Court Circuit, ...
- La radio => mode LP des SoC
 - Souvent gérée “automatiquement”



Le choix des horloges CPU

Clock Source	Frequency	Conso	Precision 25 °C/0-85 °C	Wakeup time
MSI (default)	65kHz-4.2MHz (2.1MHz default)	0.7-15µA	±0.5% / ±3%	3.5 µs (Vcore Range 1/2)
HSI	16MHz	100µA	±1% / ±2.5%	3.7 µs
HSE external crystal	1-24MHz	~500µA	~±0.01% (100ppm)	1ms
PLL	2-32MHz	~350µA	N/A	100µs (2MHz input)
LSI	37kHz	0.4µA	50%	200µs

Le choix des horloges CPU

- Choix de l'Oscillateur

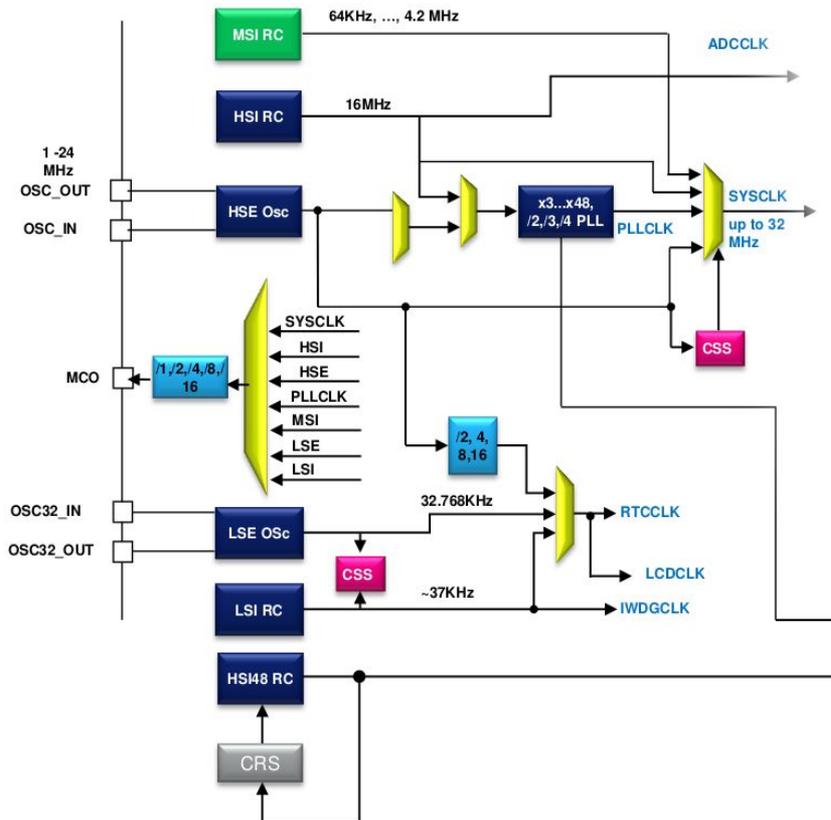
- Interne/Externe
- Vitesse Max
- Précision, **Température**
- **Consommation**

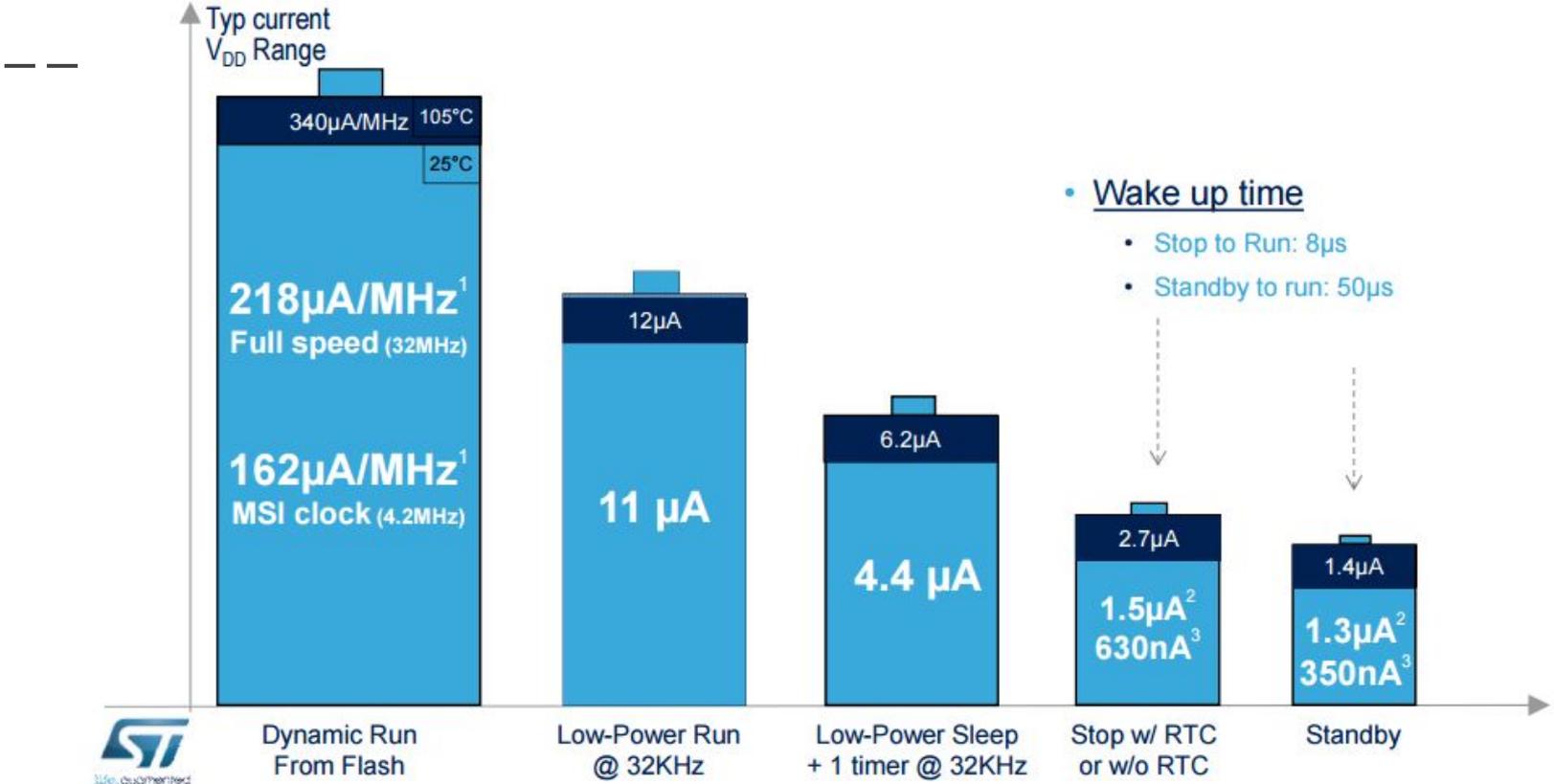
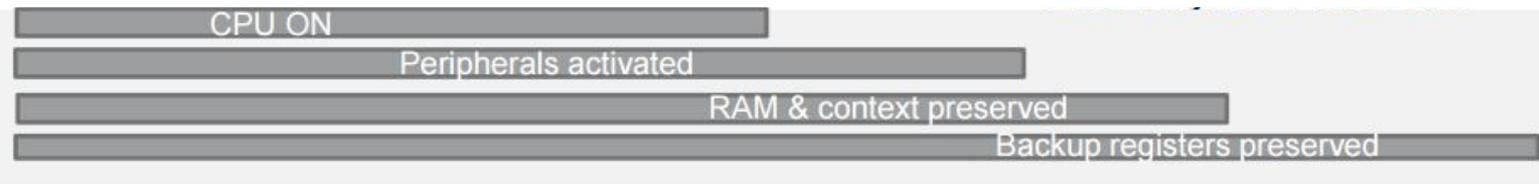
- Choix des pre scalaires

- Vitesse d'**horloge CPU**
- PLL

- Horloge des périphériques

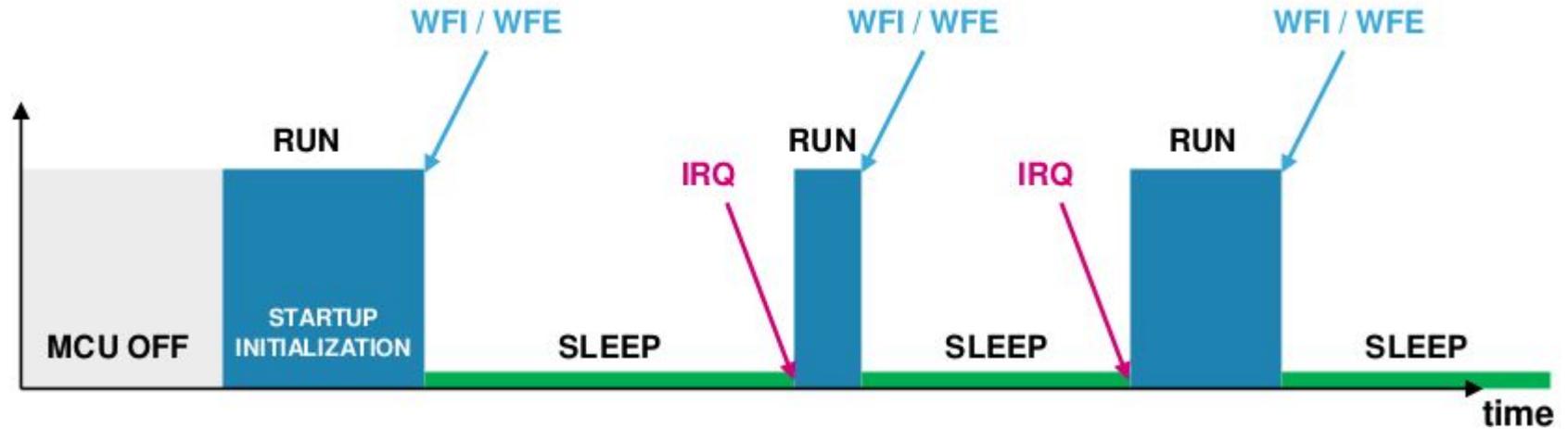
- Vitesse des bus





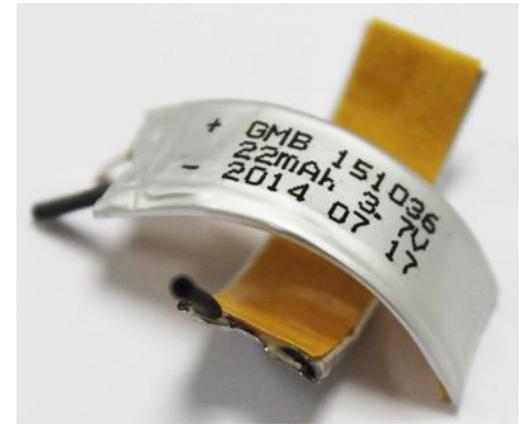
Mode name	Entry	Wakeup	Effect on VCORE domain clocks	Effect on VDD domain clocks	Voltage regulator	IO state	Wakeup latency
Low power run	LPSDSR and LPRUN bits + Clock setting	NA.	None	None	In low power mode	All I/O pins keep the same state as in the Run mode	None
Sleep Sleep now or sleep on- exit)	WFI	Any interrupt	CPU CLK OFF no effect on other clocks or analog clock sources	None	ON		None
	WFE	Wake-up event					
LP Sleep Sleep now or sleep on- exit)	LPSDSR bits+ Clock setting + WFI	Any interrupt		None	In low power mode		Regulator wakeup time from low power mode +FLASH wakeup time
	LPSDSR bits + WFE	Wake-up event					
Stop	Clear PDDS, LPSDSR bits + SLEEPDEEP bit + WFI or WFE	Any EXTI line (configured in the EXTI registers, internal and external lines)	All VCORE domain clocks OFF	HSI and HSE and MSI oscillators OFF	ON, in low power mode (depending on PWR_CR)		MSI RC wakeup time + regulator wakeup time from Low-power mode + FLASH wakeup time
Standby	Set PDDS bit + SLEEPDEEP bit + WFI or WFE	WKUP pin rising edge, RTC alarm (Alarm A or Alarm B), RTC Wakeup event, RTC tamper event, RTC timestamp event, external reset in NRST pin, IWDG reset			OFF	all I/O pins are high impedance (1)	Reset phase

Utiliser les modes Low Power



Etude (théorique) de consommation

- Déterminer la conso. de chaque capteurs
 - En veille
 - En utilisation
- Déterminer la conso. du MCU
 - En fonction du CPU en RUN
 - En STOP
- Déterminer le temps passé dans chaque mode
- Estimation de la conso. moyenne du système
 - $\text{Conso} = \text{Somme} (\text{Conso}_{\text{composant}} \times \%_{\text{temps passé}})$
- Choix de la batterie
- Estimation de l'autonomie



Exemple de Bilan Consommation

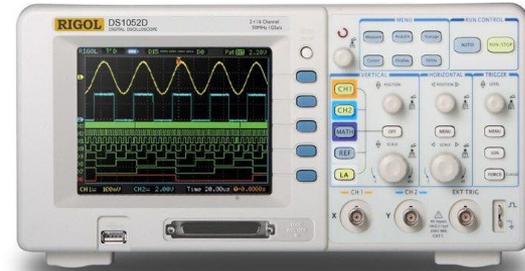
— — —

MESURE DE CONSO AVEC KEYSIGHT			
Mode STOP sans réveil avec LPTIM (uA)	2,41		
Moyenne en STOP avec période de reveil à 1.5s (uA)	5,8	Batterie (mA)	550
Moyenne en mode utilisation (uA) avec 1 msg/minute	505		
Moyenne en mode utilisation entre chaque msg (uA)	107	BILAN / RESULTATS	
Moyenne call de 10sec (mA) avec viber+buzz+screen)	12	DUREE DE VIE (JOURS)	600,0249361
Moyenne reception MSG WAIT durée 4.5s (mA)	5,22	DUREE DE VIE SUR LE SOCLE SANS UTILISATION (J)	9508,990318
Moyenne pendant 1 utilisation de 20min	550		
HYPOTHESES DE DIMENSIONNEMENT			
Nombre d'utilisation par jour	10	1 journée (min)	1440
Nombre de msg de la passerelle par utilisation	10	% passé sur le socle	0,9305555556
Durée d'un MSG (s)	4,5		
Temps d'utilisation par jours	100	% passé en utilisation	0,06944444444
Durée d'une utilisation (min)	20		
AUTONOMIE			
Conso moyenne les jours d'utilisation	43,59166667		
Conso moyenne les jours fermé	5,8		
Conso moyenne journalière	38,19285714		

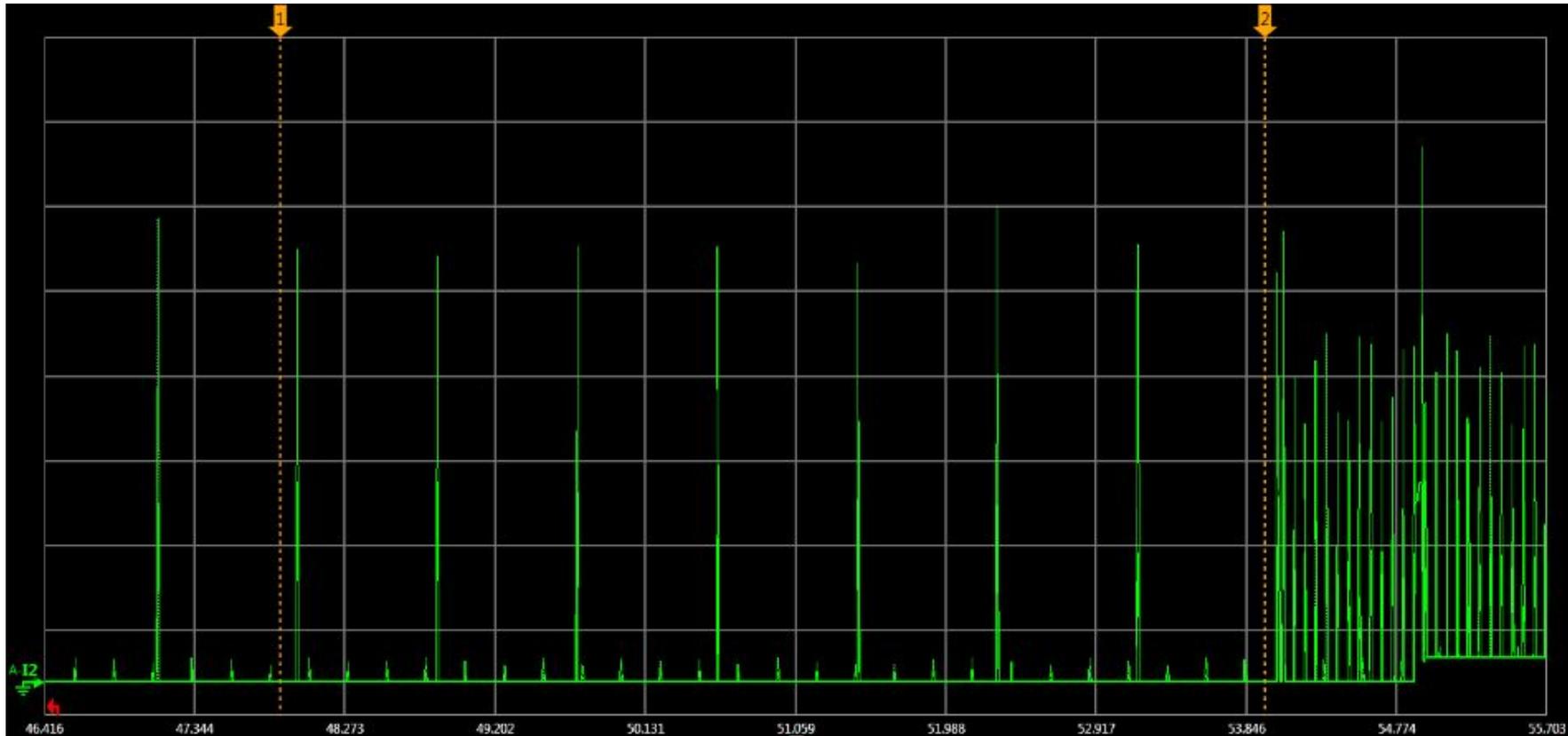
Etude (pratique) de consommation



- Mesure à l'oscilloscope
- Vérification des modes LP
- Duty Cycle
- Tx Radio
- Paramètres BLE
 - Advertising Interval
 - Connection Interval







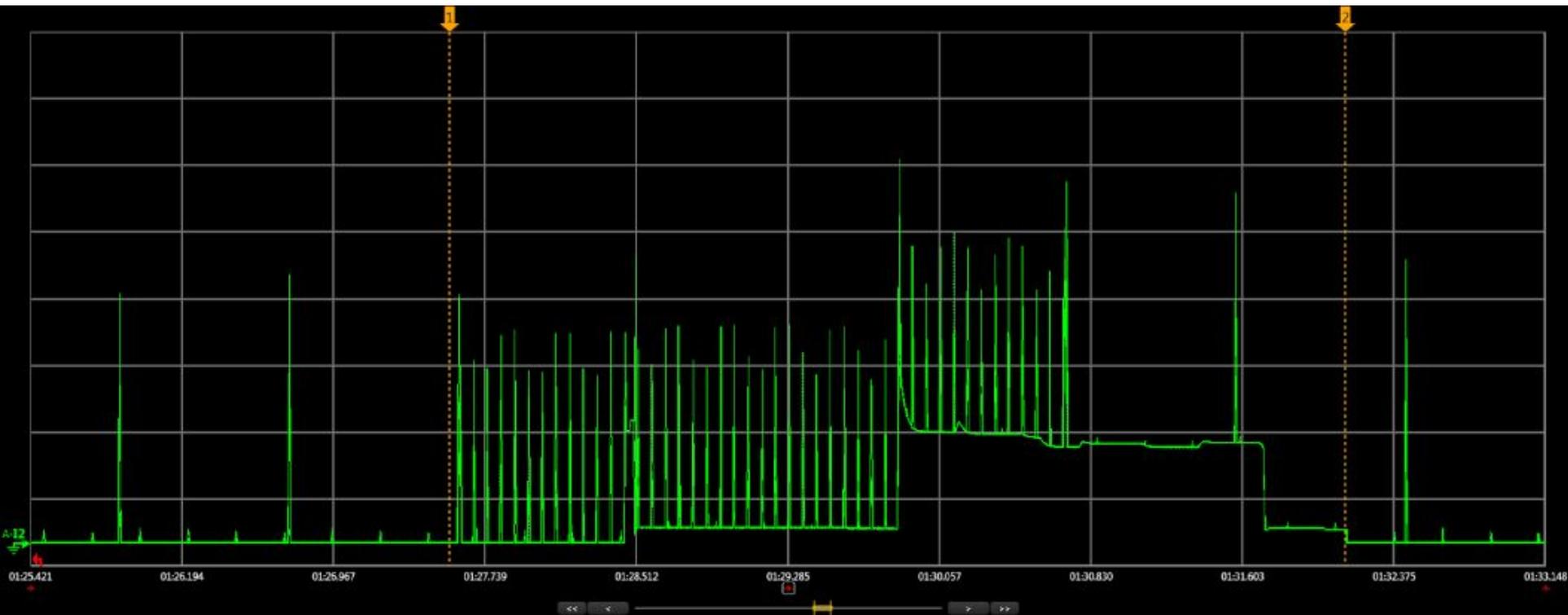
46.416 47.344 48.273 49.202 50.131 51.059 51.988 52.917 53.846 54.774 55.703



Stopped Markers & Measurements + Auto Scroll Ranges... AUTO SCALE 🔍 🔍

Time	Marker 1	Measurements Between Markers					Marker 2
	47.871918 s	$\Delta = 6.085345$ s Freq = 164 mHz					53.957263 s
	Avg	Min	Avg	Max	Peak to Peak	Charge / Energy	
A-12	11.01 μ A	10.256 μ A	107.026 μ A	28.067471 mA	28.057215 mA	181 nA h	
						Avg	
						10.984 μ A	

929 ms / Duration: 000:06:30 Period: 1.00352 ms Min/Max File: datalogdata177.dlg Trigger Data Log Run Button Source: 14585A



Stopped Markers & Measurements

Time	Marker 1	Measurements Between Markers					Marker 2
01:25.421	01:27.557120	$\Delta = 4.568023$ s Freq = 219 mHz					01:32.125143
A-12	Avg 10.367 μ A	Min 5.277 μ A	Avg 5.220296 mA	Max 36.573071 mA	Peak to Peak 36.567794 mA	Charge / Energy 6.624 μ A h	Avg 1.273384 mA

773 ms / Duration: 000:02:30 Period: 1.00952 ms Min/Max File: datalogdata155.dlg Trigger: Data Log Run Button Source: 14585A

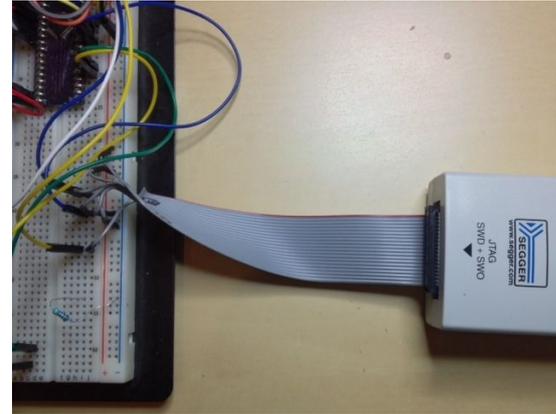
Bootloader

- Rôle
 - Mise à jour des firmware : BLE, STM32, Nordic
- Comment
 - “Over the Air” pour le BLE
 - USB pour STM32 et Nordic
- Fonctionnement
 - 2 firmwares sont présents en flash
 1. Le bootloader
 2. L’application principale
 - Attention au vecteur d’Interruption



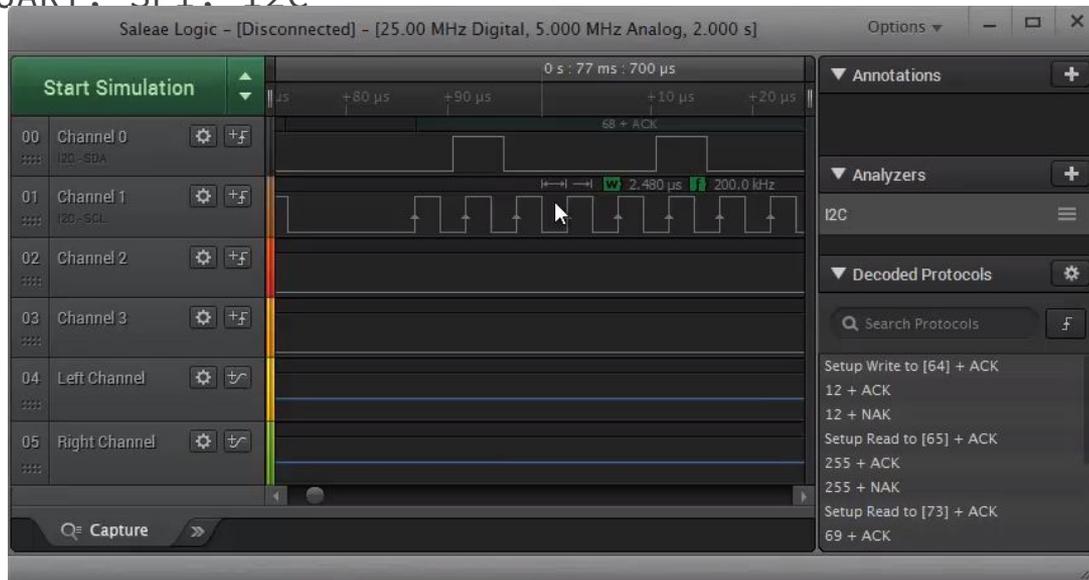
Les Outils de Debugage

- Sonde JTAG
 - Inspecter/flasher un firmware
 - Breakpoint d'exécution
 - Visualiser la mémoire
- Analyseur Logic
 - Espionner un BUS : UART, SPI, I2C
- L'oscilloscope
 - Fréquence du CPU, vitesse de BUS
 - Optimisation du temps de process
 - Signaux logiques



Les Outils de Debugage

- Analyseur Logic
 - “Espionner” un BUS : UART, SPI, I2C



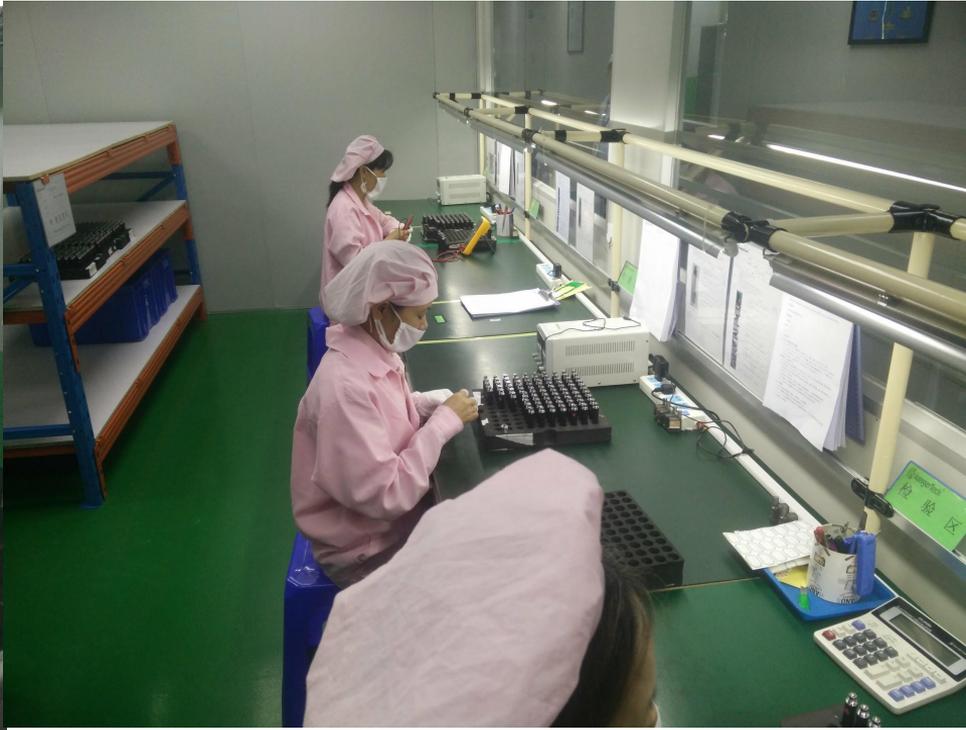
Les Certifications

- CEM
 - Test Emission Radio
 - Immunités
- Certification CE
- Certification Sigfox















Conclusion

- Nous sommes entourés de systèmes embarqués !!

Leur **complexité** se situe au niveau **logiciel** et **matériel**

Ils posent de nombreuses contraintes de conception:

- Temps de conception, échéance, **coût**
- Consommation, **autonomie**
- **Encombrement**

Les méthodologies de conception aident à gérer et améliorer le processus de conception !!

Questions ?

- A vous de jouer !

POURQUOI,
POURQUOI,
POURQUOI ?

