

Université Mohamed Khider – Biskra
Faculté des Sciences et de la technologie
Département : Génie électrique
Réf :



جامعة محمد خيضر بسكرة
كلية العلوم و التكنولوجيا
قسم: الهندسة الكهربائية
المرجع:

Mémoire présenté en vue de l'obtention
Du diplôme de
Magister en : Electronique

Option : Signaux et communications

**Conception d'un système embarqué
pour le suivi préventif d'un château d'eau**

Présenté par :

Radouane LAKEHAL

Soutenu publiquement le **06 Octobre 2019**

Devant le jury composé de :

Dr. Nouredine SENGOUGA
Dr. Abderezzak DEBILOU
Dr. Abdelhamid BENAKCHA
Dr. Abdelkrim OUAFI

Professeur
Maitre de Conférences 'A'
Maitre de Conférences 'A'
Maitre de Conférences 'A'

Président
Rapporteur
Examineur
Examineur

Université de Biskra
Université de Biskra
Université de Biskra
Université de Biskra

Résumé

L'avancée des technologies de l'information fait introduire dans les années 2007 la notion de systèmes embarqués communicants. Ces derniers sont en fait des ordinateurs miniaturisés, spécialisés communicants sans fil, intégrés dans des dispositifs.

Les châteaux d'eau cumulent un volume très important d'eau qui peut provoquer de grave incident s'ils n'ont pas contrôlés néant moins l'accès est difficile car ils sont généralement situés dans des endroits à hauteur élevée et éloigné des habitations. Les possibilités technologiques actuelles permettent un suivi à distance des paramètres des châteaux d'eau.

Dans ce contexte notre projet de magister a pour objectif l'étude et la conception d'un système électronique embarqué implanté dans un château d'eau. Ce système doit acquérir via des capteurs et des instruments correctement étalonnés et configurés les paramètres physiques et chimiques de fonctionnement du château tels que : Le débit, la température, le niveau, le PH et le chlorure etc....

Ces paramètres sont d'une importance majeure dans le fonctionnement du château d'eau. La connaissance de ces paramètres en temps réel permet de protéger le château et son environnement. Le système à concevoir doit être capable de transmettre ces grandeurs via une ligne de transmission sans fil à une chambre pilote de contrôle.

Les outils et les possibilités électroniques offertes : Carte à microprocesseur, module GSM industriel, capteurs et logiciel de développement aide énormément la caractérisation d'un tel système embarqué.

Mots clés : Système embarqué communicant, transmission sans fil, contrôle de paramètres.

Abstract

The progress of information technology is introduced in the year 2007 the concept of communicating embedded systems. These are actually miniaturized computers, wireless communicating specialized integrated in devices.

Water towers cumulated a very large volume of water that can cause serious incident if they have not checked nothing less access is difficult because they are usually located in places with high elevation and remote homes. Current technological possibilities allow followed at a distance parameter water towers.

In this context, our project aims magister study and design of an electronic system installed in a water tower. This system must acquire via sensors and instruments properly calibrated and configured the physical parameters of the castle such as flow rate, temperature, level etc.

These parameters are of major importance in the operation of the water tower. Knowledge of these parameters in real time allows protected the castle and its environment. The system design must be able to transmit these values via a wireless transmission line to a pilot control chamber.

Tools and electronic possibilities: Microprocessor Card, Industrial GSM module, sensors and software development really helps the characterization of such embedded system.

Keywords: Communicating embedded systems, Wireless transmission, control parameters.

ملخص

سمح تنامي أهمية إرسال واستقبال المعلومات خاصة في الأونة الأخيرة أين أصبح العالم بأكمله يتجه نحو التقنيات اللاسلكية والتطور الحاصل في تكنولوجيا المعلومات بظهور الأجهزة المضمنة التوافقية، وهي أنظمة لمعالجة المعلومات وتكون مضمنة في منتج أكبر ومصممة لأغراض محددة، وهي عادة ما تكون غير مرئية مباشرة للمستخدم ومتصلة لاسلكيا.

خزانات (أبراج) المياه تحوي كمية كبيرة جدا من المياه التي يمكن أن تتسبب في حوادث خطير إن لم تفحص ولم تراقب، كما أنه في غالب الأحيان يصعب الوصول إليها لأنها تقع عادة في مناطق مرتفعة ونائية. تمكننا الإمكانيات التكنولوجية الحالية من مراقبة معايير هذه الخزانات عن بعد.

في هذا السياق يهدف مشروعنا الى دراسة وتصميم نظام إلكتروني مثبت في برج مياه، هذا النظام يحصل عن طريق أجهزة الاستشعار على البارامترات الفيزيائية والكيميائية مثل معدل التدفق، ودرجة الحرارة، ومستوى المياه، والـ PH الخ...

هذه البارامترات ذات أهمية كبيرة في تسيير خزانات المياه. متابعة هذه البارامترات في الوقت الحقيقي يتيح حماية الخزان ومحيطه. يجب أن يكون النظام المصمم قادرا على نقل هذه القيم لاسلكيا إلى غرفة التحكم.

الأدوات والإمكانيات الإلكترونية المتاحة: بطاقة المعالجات الدقيقة وحدة GSM، أجهزة الاستشعار والبرمجيات التي تساعد حقا على تصميم هذا النظام المضمن.

الكلمات المفتاحية: الأنظمة المضمنة التوافقية، الإرسال اللاسلكي، مراقبة البارامترات.

Table des matières

Introduction Générale	4
------------------------------------	---

Partie théorique

I- Systèmes embarqués : Introduction

1. Introduction	6
2. Notion de convergence	7
3. Qu'est-ce qu'un système embarqué ?	8
4. Structure d'un système embarqué	9
4.1.1. Composants matériels	10
4.1.2. Composants logiciels	12
4.1.3. Système embarqué typique	13
5. Exemple de systèmes embarqués	15
6. Domaines d'applications des systèmes embarqués	17
7. Classification des systèmes embarqués	18
7.1.1. Système embarqué petit	18
7.1.2. Système embarqué moyen	18
7.1.3. Système embarqué haute performance	19
8. Cycle de vie des systèmes embarqués	19

II- Conception des systèmes embarqués

1. Introduction	23
2. Caractéristiques des systèmes embarqués	23
2.1. Algorithmes complexes	23
2.2. Interface utilisateur	24
2.3. Fonctionnement en Temps Réel.....	24
2.4. Multi-débit.....	24
2.5. Coût de fabrication	24
2.6. Puissance et énergie.....	25
3. Contrainte temporelle des systèmes embarqués	25
4. Conception des systèmes embarqués	26
4.1. Défis de la conception des systèmes embarqués	26
5. Processus de conception du système embarqué	28

5.1. Exigences	29
5.2. Spécification	30
5.3. Conception d'architecture	31
5.4. Conception des composants matériel et logiciel	33
5.5. L'intégration du système	34
6. Méthodologies de conception	34
6.1. Flux de conception	36

III- Communication dans les systèmes embarqués

1. Introduction	40
2. Systèmes embarqués communicants	40
3. Les protocoles de communication série	40
3.1. Inter-Integrated Circuit (I2C)	40
3.2. Serial Peripheral Interface (SPI)	44
4. Technologie de communication sans fil	48
4.1. Généralités sur les réseaux sans fils	48

Partie Pratique

IV- Conception et réalisation

A- Conception du système	60
1. Exigences et spécification	60
2. Architecture	61
3. Composants matériels et logiciel	64
3.1. Matériel utilisé	64
4. Intégration du système	73
4.1. Transmetteur de niveau	73
4.2. Transmetteur de température	75
4.3. Transmetteur de débit	77
4.4. Transmetteur - Récepteur RF 433MHz	79
B- Mise en œuvre et résultats	81

Conclusion et mise en perspective

Bibliographie

Liste des figures

N°	Titre	Page
1.1	La convergence des disciplines	7
1.2	Vue générale d'un Système Embarqué	9
1.3	Les éléments matériels dans un système embarqué	11
1.4	Structure du logiciel dans un système embarqué	12
1.5	Système embarqué typique	14
1.6	Un système embarqué simple	15
1.7	Circuit intégré d'un appareil photo	16
1.8	Domaines d'applications des systèmes embarqués	17
1.9	Classification des systèmes embarqués	18
1.10	Cycle de vie d'un système embarqué	20
2.1	Besoin des applications en puissance de calcul	24
2.2	Principaux niveaux d'abstraction dans le processus de conception des systèmes embarqués	27
2.3	Schéma block de la carte de déplacement GPS	28
2.4	Architectures matérielles et logicielles pour carte de déplacement GPS	32
2.5	Influence d'un retard à l'arrivée sur le marché sur les recettes d'un produit.	33
2.6	Méthodologie de conception simple de système embarqué.	34
2.7	Flux de conception hiérarchique pour système embarqué	34
2.8	Modèle de conception raffinement successif.	35
3.1	Vue Générale du Bus I2C	63
3.2	Connexion des dispositifs au bus I2C	67
3.3	Les conditions START and STOP	69
3.4	Transfert des données sur le bus I2C	70
3.6	Liaison SPI Maître - Esclaves	73
3.7	Liaison SPI Maître - Multi Esclaves	74
3.8	Chronogramme général d'un échange sur une liaison SPI	76
3.9	Les modes de prise en compte des données de la liaison SPI	76
3.10	Les différentes catégories des réseaux sans fils	78
3.11	Représentation de la topologie en étoile.	78
3.12	Représentation de la topologie point à point	79
3.13	Diagramme topologique d'un réseau de type Maître/Esclave	81
3.14	Diagramme topologique d'un réseau chaîné	82
3.15	Réseau 802.11 en mode infrastructure.	83
3.16	Réseau 802.11 en mode ad-hoc.	86
4.1	Schéma bloc du Système de surveillance d'un château d'eau	90
4.2	Architecture matérielle du Système de surveillance d'un château d'eau	91
4.3	Architecture logiciel du Système de surveillance d'un château d'eau	91
4.4	Carte Arduino "Mega 2560"	94
4.5	Brochage de la Carte Arduino "Mega 2560"	94

4.6	Carte Arduino "Nano"	95
4.7	Les formats du capteur de température DS18B20	96
4.8	Le capteur ultrasons HC-SR04	97
4.9	Illustration du signal TRIGGER et ECHO	98
4.10	Le capteur de débit YFS 201	98
4.11	Module GSM SIM900A mini	98
4.12	Le module RF433MHz	99
4.13	Schéma du transmetteur de niveau	
4.14	Configuration du transmetteur de niveau d'eau	
4.15	Schéma du transmetteur de température	
4.16	Configuration du transmetteur de température	
4.17	Schéma du transmetteur de Débit	
4.18	Configuration de transmetteur de débit d'eau	
4.19	Schéma du transmetteur RF 433MHz	
4.20	Schéma du récepteur RF 433MHz	
4.21	Configuration générale du Système de surveillance d'un château d'eau	
4.22	Maquette du système réalisé	
4.23	Affichage local de niveau Château d'eau vide	
4.24	Affichage mobile de niveau Château d'eau vide	
4.25	SMS Château d'eau vide	
4.26	Affichage local de niveau Château d'eau demi plein	
4.27	Affichage mobile de niveau Château d'eau demi plein	
4.28	Affichage local de niveau Château d'eau plein	
4.29	Affichage mobile de niveau Château d'eau plein	
4.30	Affichage local de niveau Château d'eau trop plein	
4.31	Affichage mobile de niveau Château d'eau trop plein	
4.32	SMS Château d'eau trop plein	
4.33	Affichage local du débit Pas de remplissage	
4.34	Affichage mobile du débit Pas de remplissage	
4.35	Affichage local du débit remplissage	
4.36	Affichage mobile du débit remplissage	
4.37	Affichage local température	
4.38	Affichage mobile température	

Liste des tableaux

N°	Titre	Page
2.1	Evolution de la consommation des familles de processeurs INTEL et AMD	30
3.1	Débit des normes USB	64
3.2	Les classes d'émetteurs du standard Bluetooth	75
4.1	Caractéristiques techniques d'Arduino "Mega 2560 "	92

Les châteaux d'eau sont des réservoirs de stockage de grande capacité surélevés qui permettent d'assurer 24 heures sur 24 et 7 jours sur 7 un approvisionnement en eau potable, ils sont bien souvent excentrés des centres-villes, situés sur les points hauts. Les châteaux d'eau sont connus également, par les risques et périls qui y sont associés, tel que :

- Le débordement qui provoque l'inondation et les pertes d'eau qui est de plus en plus rare, et pertes humaines, animales, agriculture. Tout cela engendre des pertes économiques considérables aux différentes régions ou pays.
- La contamination intentionnelle ou non.

En vue d'éviter toutes ces conséquences négatives, une suivie préventif s'impose. Ce suivi qui est basé essentiellement sur la surveillance manuelle, présente. Ce mode de surveillance a tout de même ces insuffisances, à type de :

- Châteaux d'eau construits souvent en zones lointaines, éloignées et isolés
- Difficulté d'accéder surtout en période d'intempéries.
- Difficulté de suivi en temps réel.
- Difficulté de suivre plusieurs paramètres en même temps, ce qui nécessite souvent le recours à des équipes variés avec des compétences prouvées et différentes.

Afin de pallier à ces insuffisances, les systèmes embarqués peuvent en constitue une solution de remplacement au systèmes manuels. Qu'est à la fois efficace et peu coûteuse.

D'ailleurs, pour la société nationale *NAFTAL* ce système est très utilisé pour le suivie des grandes citernes de stockage de carburants qu'elle possède depuis déjà un certain temps.

D'où l'intérêt de notre travail qui vise à concevoir et à réaliser un système embarqué communicant qui peut être utilisé pour le suivi préventif des châteaux d'eau.

1. Introduction :

Ce chapitre est une introduction aux systèmes embarqués (enfouis), c'est-à-dire aux ordinateurs invisibles qui sont intégrés dans un grand nombre d'objets utilisés pour communiquer (téléphone portable, PDA, télévision numérique, etc.), dans les systèmes de transports (automobile, train, avion) et dans les infrastructures grandes ou petites (contrôle des centrales nucléaires, automatisations industriels, etc.).

2. Notion de convergence :

Les premiers ordinateurs ont servi pour des applications militaires. Au cours des années 70, l'arrivée des premiers gros ordinateurs d'IBM ouvre leur champ d'application aux services et au commerce. En parallèle, l'apparition des circuits intégrés permet une augmentation exponentielle de la puissance de calcul. Dans les années 80, la convergence entre les technologies de l'information et les télécommunications ouvre la voie vers internet et le web. Les années 90 ont connu une révolution aussi importante que celle de la société de l'information avec la dissémination massive des systèmes embarqués dans les produits évolués. Plus de 95% des puces produites aujourd'hui sont affectées à des applications embarquées. Elles sont cachées dans des équipements, produits de consommation, et des infrastructures pour y fournir des services divers et variés. Les systèmes embarqués existent depuis dizaines d'années, rien de neuf, mais ce qui est nouveau, c'est qu'ils se situent maintenant à l'intersection de différentes disciplines : *Electronique, Informatique et Télécoms & réseaux*. [1]

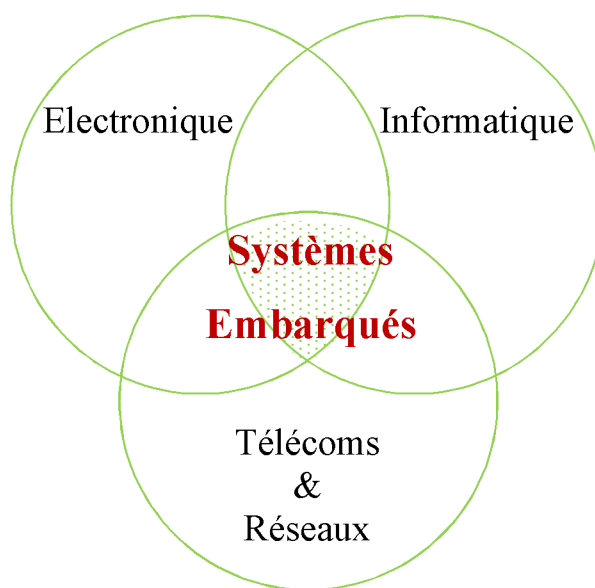


Figure 1.1- La convergence des disciplines

3. Qu'est-ce qu'un système embarqué ?

Nous sommes tous familiers avec l'idée d'un ordinateur de bureau ou ordinateur portable, et le traitement étonnant qu'ils puissent faire. Ces ordinateurs sont à usage général, nous pouvons les amener à faire des choses différentes à des moments différents, en fonction de l'application ou du programme que nous exécutons sur eux. Au cœur de ces ordinateurs se trouve un microprocesseur : un circuit électronique minuscule et fantastique complexe qui contient les fonctions de base d'un ordinateur. Ce qui est moins familier à beaucoup de gens, c'est l'idée que, au lieu de mettre un microprocesseur dans un ordinateur à usage général, il peut également être placé à l'intérieur d'un produit qui n'a rien à voir avec l'informatique, comme un lave-linge, une micro-onde ou un appareil photo. Le microprocesseur est alors adapté à contrôler ce produit. Il est à l'intérieur du produit, mais il ne peut pas être vu, et l'utilisateur ne sait probablement même pas qu'il existe.

En outre, les add-ons qui sont normalement associées à un ordinateur, comme le clavier, la souris ou l'écran, ne peuvent être vu. Ces produits sont appelés systèmes embarqués, parce que l'ordinateur qui les contrôle est intégré à l'intérieur. Car ils ont tendance à se concentrer sur le contrôle, dans de nombreux cas, les microprocesseurs utilisés dans les systèmes embarqués développent des caractéristiques différentes de celles utilisées dans plus de machines à usage général. Ils sont appelés les microcontrôleurs. Bien que beaucoup moins visibles que leurs cousins les microprocesseurs, les microcontrôleurs se vendent beaucoup plus en volume et leur impact est considérable. Pour les concepteurs du système, ils offrent d'énormes possibilités.

Les systèmes embarqués nous entourent et nous sommes littéralement envahis par eux, fidèles au poste et prêts à nous rendre service. On en croise des dizaines par jour sans le savoir. Ils sont donc partout, discrets, efficaces et dédiés à ce à quoi ils sont destinés. Omniprésents, ils le sont déjà et le seront de plus en plus. On parle en fait d'informatique et d'électronique diffuse ou ubiquitaire. Ils sont bourrés d'électronique plus ou moins complexe et d'informatique plus ou moins évoluée. [5]

Un **système embarqué** (SE) peut être défini comme :

Un système électronique et informatique autonome, dédié à une tâche bien précise. [1]

Il ne possède généralement pas des entrées / sorties standards et classiques comme un clavier ou un écran d'ordinateur. Le matériel et le logiciel ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur PC. Ce système électronique est conçu pour réaliser une ou un nombre limité de fonctions particulières, souvent avec des

contraintes temps réel. Il est intégré à l'intérieur d'un système complet, souvent avec des parties mécaniques et des interfaces avec le monde extérieur (capteurs et actionneurs).

4. Structure d'un système embarqué :

Quelle que soit la fonction exécutée par un système embarqué, la vue la plus large de sa structure révèle deux grands ensembles de composants étroitement couplés : un ensemble de composants matériel qui comprend une unité centrale de traitement, typiquement sous la forme d'un microcontrôleur ; et une série de programmes, typiquement comprise comme logiciel, qui donne la fonctionnalité du matériel. La **Figure 1.2** illustre cette vue générale en montrant ces deux grandes composantes et leur interrelation. [2], [8]

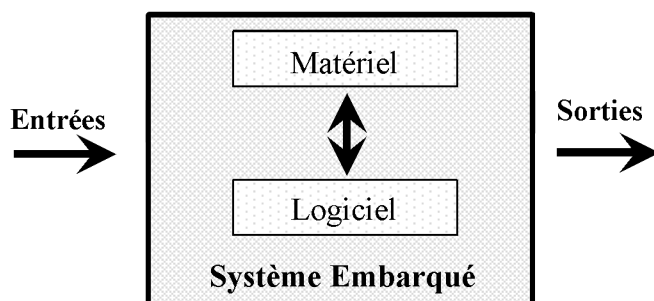


Figure 1.2- Vue générale d'un Système Embarqué

Les entrées dans un système embarquée sont des variables de processus et des paramètres qui arrivent via des capteurs et des ports d'entrée / sortie (E / S). Les sorties sont sous la forme d'actions de contrôle influençant sur les actionneurs du système ou des informations traitées pour les utilisateurs ou d'autres sous-systèmes au sein de l'application. Dans certain cas, on assure l'interfaçage avec l'utilisateur par des touches, des boutons, des diodes électroluminescentes (LED), un afficheur à cristaux liquides (LCD), et d'autres types de dispositifs d'affichage en fonction de l'application.

Considérons par exemple une application de système embarqué sous la forme d'un four micro-ondes. Les éléments matériels de cette application comprennent : le magnétron (Générateur des micro-ondes), le module d'électronique de puissance qui commande le niveau de puissance du magnétron, le moteur qui tourne le plateau, le clavier des chiffres et des paramètres repas, le système d'affichage indiquant le timing et les informations d'état, une sorte de buzzers pour les signaux sonores, et au cœur du four, le contrôleur électronique embarqué qui coordonne le fonctionnement de l'ensemble du système. Les entrées du système comprenant les sélections repas, le temps de cuisson, et le niveau de puissance sont obtenus de l'opérateur humain au moyen d'un

clavier et les informations d'état du magnétron, la température des repas, et les signaux internes de l'état du système à partir de plusieurs capteurs. Les sorties prennent la forme du temps restant de cuisson et le statut du four par un dispositif d'affichage, les niveaux de puissance du système de commande de magnétron, les commandes pour activer ou désactiver le plateau tournant, et des signaux sonores.

Le logiciel est la partie la plus abstraite du système et aussi essentiel que le matériel lui-même. Il comprend les programmes qui déterminent l'ordre dans lequel les composants matériel fonctionnent. Quand quelqu'un décide de préparer un repas préprogrammé dans un four micro-ondes, le logiciel reprend les touches du panneau de commande du four, identifie la sélection de l'utilisateur, détermine le temps et le niveau de puissance de la cuisson, et initialise les radiations micro-ondes dans la chambre, met fin à la rotation de la plaque, et alerte l'utilisateur par un signal audible quand le repas est prêt. Alors que le repas est en cuisson, le logiciel contrôle la température de repas et ajuste la puissance et le temps de cuisson, mais également vérifie le bon fonctionnement des composants internes du four. Dans le cas de la détection d'un dysfonctionnement du système, le programme interrompt le fonctionnement du four pour éviter des conséquences catastrophique. Dans les sections ci-dessous nous détaillons les composants matériels et le logiciel qui intégrés dans un système embarqué.

4.1. Composants matériels :

D'un point de vue général, les composants matériels d'un système embarqué comprend toute l'électronique nécessaire pour que le système exécuter la fonction pour laquelle il a été conçu. Par conséquent, la structure spécifique d'un système particulier pourrait sensiblement différer d'un autre système en fonction de l'application elle-même. Malgré ces dissemblances, trois composants matériels de base sont essentiels dans un système embarqué :

- L'unité centrale de traitement (CPU).
- La mémoire système.
- Un ensemble de ports d'entrées-sorties.

Le processeur exécute des instructions traitant les entrées du système pour prendre les décisions qui guident le fonctionnement du système. La mémoire stocke le programme et les données nécessaires pour le fonctionnement du système. La plupart des systèmes différencient entre la mémoire de programme qui stocke les instructions exécutées par le CPU et la mémoire de données qui stocke les données traitées par le système. Les ports d'E / S assurent la communication

entre l'unité centrale (CPU) et le monde extérieur. Au-delà de ce point, un certain nombre de périphériques d'E / S est nécessaire pour le fonctionnement du système selon l'application. Ceux-ci comprennent :

- Les ports de communication série et / ou parallèle avec d'autres dispositifs ou systèmes. Ports USB, ports d'imprimante, sans fil et ports infrarouges, sont Certains exemples représentatifs de dispositifs d'E / S de communication.
- Les interfaces utilisateur interagissant avec l'humain. Claviers, commutateurs, buzzers, lumières et afficheurs sont des exemples des interfaces E/S.
- Les capteurs et actionneurs électromécaniques interagissant avec l'environnement externe du système. Les capteurs fournissent des entrées relatives à des paramètres physiques tels que température, pression, déplacement, accélération, rotation, etc. les contrôleurs de vitesse du moteur, les contrôleurs de moteurs pas à pas et les relais sont quelques exemples d'actionneurs.
- Les convertisseurs de données (analogique-numérique (CAN) et / ou numérique-analogique (CNA)) pour permettre une interaction avec les capteurs et actionneurs analogiques.
- Les composants de diagnostic pour vérifier et prévoir la robustesse et la fiabilité du fonctionnement du système.
- Les composants de support du système pour fournir des services essentiels comprennent : l'alimentation, générateurs de fréquence d'horloge, timers et gestionnaire d'interruptions etc.

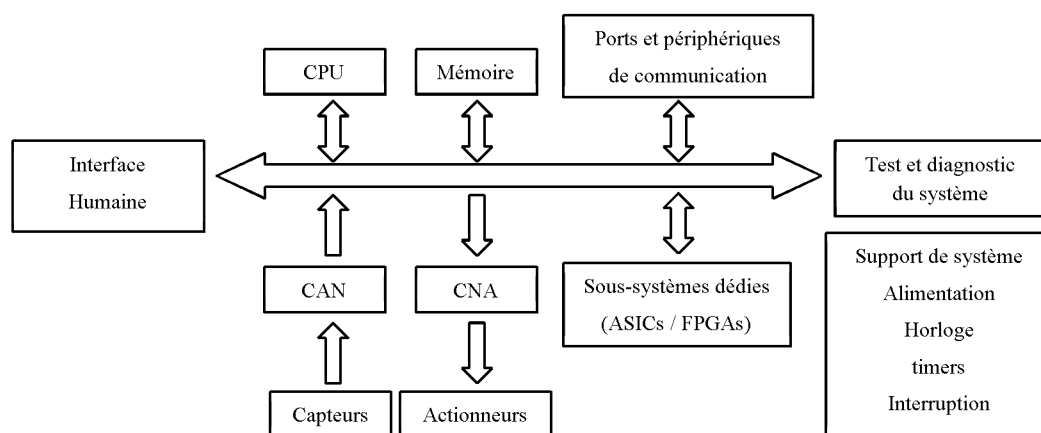


Figure 1.3- Les éléments matériels dans un système embarqué

- D'autres sous-systèmes assurant le bon fonctionnement du système, qui pourraient inclure ASIC, FPGA et d'autres unités selon la complexité de l'application.

La **Figure 1.3** illustre la façon dont ces composants matériels sont intégrés pour fournir la fonctionnalité du système souhaitée [1], [3], [4].

4.2. Composants logiciels :

Les composants logiciels d'un système embarqué comprennent tous les programmes (*micrologiciel*) nécessaires à donner la fonctionnalité au matériel du système. Le micrologiciel n'est pas modifiable par les utilisateurs, bien que certains systèmes puissent fournir des moyens d'exécution des mises à jour. Les programmes sont organisés sous la forme de système d'exploitation (SE, OS). Dans des petites applications les systèmes d'exploitation peuvent être simples et informels, mais dans des applications complexes le système d'exploitation doit être structuré et formel. Dans certains cas, les conceptions sont développées comme système d'exploitation temps réel (RTOS). [2] La **figure 1.4** illustre la structure d'un logiciel de système embarqué. Les principaux éléments dans un logiciel de système comprennent :

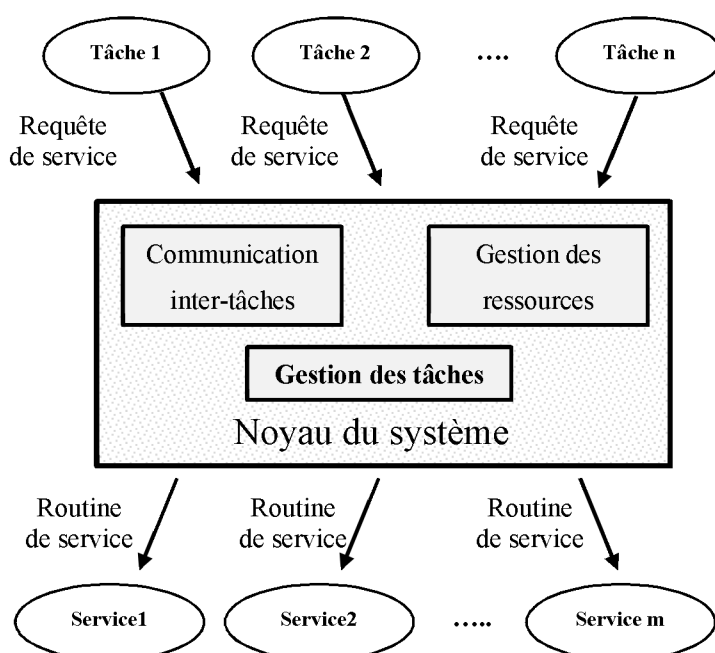


Figure 1.4- Structure du logiciel dans un système embarqué

Tâche système : Le logiciel dans les systèmes embarqués est divisé en un ensemble de petits programmes appelés *Tâches*. Chaque tâche gère une action distincte dans le système et nécessite l'utilisation des ressources spécifiques du système. Les tâches soumettent des demandes (Requête)

de services au noyau pour effectuer leurs actions désignées. Dans notre exemple *four micro-ondes* le fonctionnement du système peut être décomposé en un ensemble de tâches qui comprend, la lecture du clavier pour déterminer les sélections de l'utilisateur, la présentation d'informations sur l'afficheur du four, la mise sous tension du magnétron à un certain niveau de puissance pendant certain laps de temps.

Noyau système : Le composant logiciel qui gère les ressources du système dans une application embarquée est appelée le *noyau*. Les ressources du système sont tous les composants nécessaires pour servir les tâches, ceux-ci comprennent : la mémoire, les dispositifs d'entrées / sorties (E/S), le CPU lui-même, et d'autres composants matériels. Le noyau reçoit des demandes de services de tâches, et les organise selon les priorités imposées par le gestionnaire de tâches. Lorsque plusieurs tâches disputent sur une ressource commune, le gestionnaire des ressources définit la politique de gestion des ressources du système. C'est possible que des tâches système échangent des informations entre eux. Le noyau fournit un cadre qui permet une communication inter-tâches fiable pour échanger des informations.

Services : Les tâches sont servies par des routines de service. Un sous-programme de service est une partie de code qui donne la fonctionnalité à une ressource du système. Dans certains systèmes, ils sont considérés comme des pilotes de périphériques. Les services peuvent être activés par des routines d'interruptions des services (ISR).

4.3. Système embarqué typique :

La définition d'un système embarqué donne un aperçu en ce qui concerne les principales propriétés de ce dernier. Ils sont les suivants :

- Un système embarqué est un système numérique
- Il utilise généralement un processeur
- Il exécute un seul logiciel dédié à une fonctionnalité précise
- Il a des entrées/sorties dédiés plus ou moins limitées (boutons poussoirs, clavier matriciel, afficheur LCD...)

La **Figure 1.5** présente les principales propriétés d'un système embarqué typique. [4]

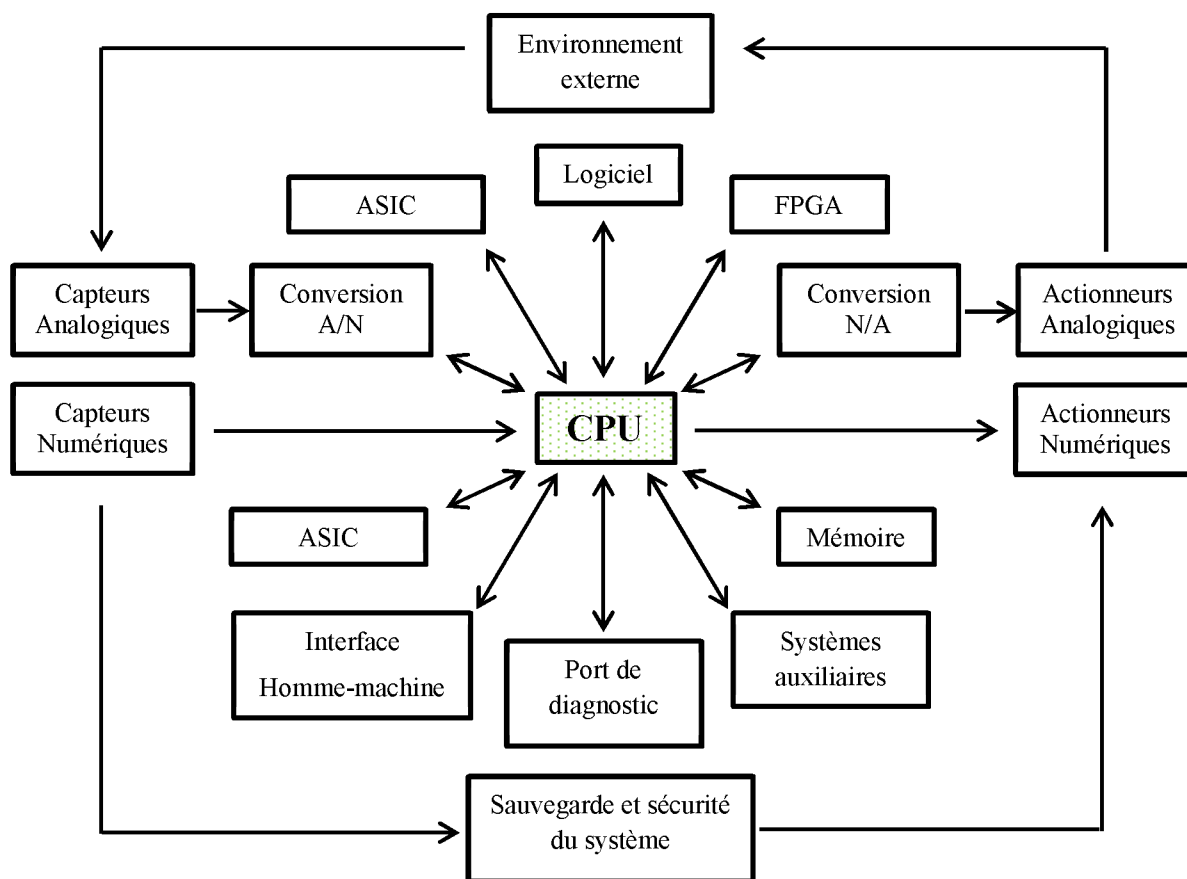


Figure 1.5- Système embarqué typique

- A l'étage d'entrée, nous trouvons généralement des capteurs analogiques couplés à des convertisseurs analogique / numérique. Les capteurs numériques sont interfacés directement au processeur.
- A l'étage de sortie, nous trouvons généralement actionneurs analogiques couplés à des convertisseurs numériques / analogiques. Les actionneurs numériques sont interfacés directement au processeur.
- Au cœur du système se trouve le calculateur, qui utilise un processeur embarqué et ses différents périphériques d'entrée / sortie associés. Il s'appuie généralement sur un ou plusieurs FPGA, circuits logiques programmables, ou même un ou plusieurs circuits intégrés spécifiques (ASIC) jouant le rôle de coprocesseur afin d'accélérer le fonctionnement du processeur embarqué.
- Par conséquent, nous avons un système de contrôle de la rétroaction entre les entrées et les sorties. Nous contrôlons les actionneurs selon les informations reçues des capteurs et les lois de contrôle implantés dans le processeur.

- Un facteur important affecte ce système, le rôle de l'environnement externe. Contrairement à un PC, un système embarqué doit fonctionner dans des environnements plus hostiles. Il doit faire face à plusieurs facteurs agressifs :
 - Variations de température
 - Vibrations, et chocs
 - Variations de la puissance d'entrée
 - L'interférence électromagnétique
 - Corrosion
 - L'eau, et le feu

Il est impossible de contrôler l'environnement dans lequel fonctionne le système embarqué. Par conséquent, il est nécessaire de tenir compte de cet effet au cours de la phase de conception. Par exemple il faut prendre en compte les variations des propriétés électriques des composants en fonction de la température.

5. Exemple de systèmes embarqués :

Comme tout système informatique, le système embarqué comprend un processeur, sa mémoire et les dispositifs d'entrée sortie. C'est l'ordinateur caché de la partie colorée de **la Figure 1.6**, intégré dans le dispositif global avec les interfaces avec le monde extérieur (capteur et actionneur).

Un système embarqué réel va comprendre ces éléments, avec davantage de composants différents pour réaliser la ou les fonctions nécessaires.

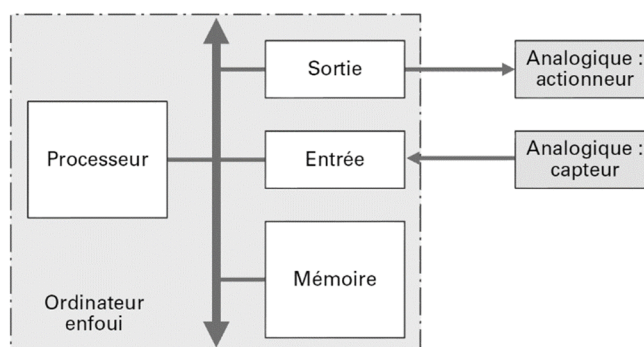


Figure 1.6- Un système embarqué simple

La **Figure 1.7** montre l'exemple classique d'un appareil photo numérique [6], [11]. Le processeur utilisé est rudimentaire (microcontrôleur), il pilote deux coprocesseurs : le préprocesseur *CCD* reconstitue l'image à partir des matrices monochromes fournies par le *CCD*

et le coprocesseur de traitement des pixels effectue les traitements élémentaires nécessaires sur l'image. L'interface avec le monde extérieur inclut la conversion analogique numérique en entrée (à partir des capteurs *CCD*) et la conversion numérique analogique en sortie pour restituer la photographie résultat.

Cet exemple permet de mettre en évidence quelques caractéristiques des systèmes embarqués. Tout d'abord, le circuit intégré, qui intègre un véritable système en un seule puce (SoC pour *system on chip*) ne réalise qu'une seule fonction : prendre des photos, même si cette fonction peut prendre en compte un grand nombre de paramètres d'entrée dans le cas des appareils numériques haut de gamme. S'ajoutent différentes contraintes :

- la taille et le poids. L'appareil photo doit être facilement transportable ;
- la consommation. La durée de vie des piles ou batteries doit être suffisamment longue pour que l'appareil soit utilisable. La source d'énergie doit permettre de prendre typiquement de 400 à 800 photos ;
- le coût. La conception et la réalisation ne doivent pas conduire à un coût qui rendrait l'appareil invendable par rapport aux prix des appareils concurrents.

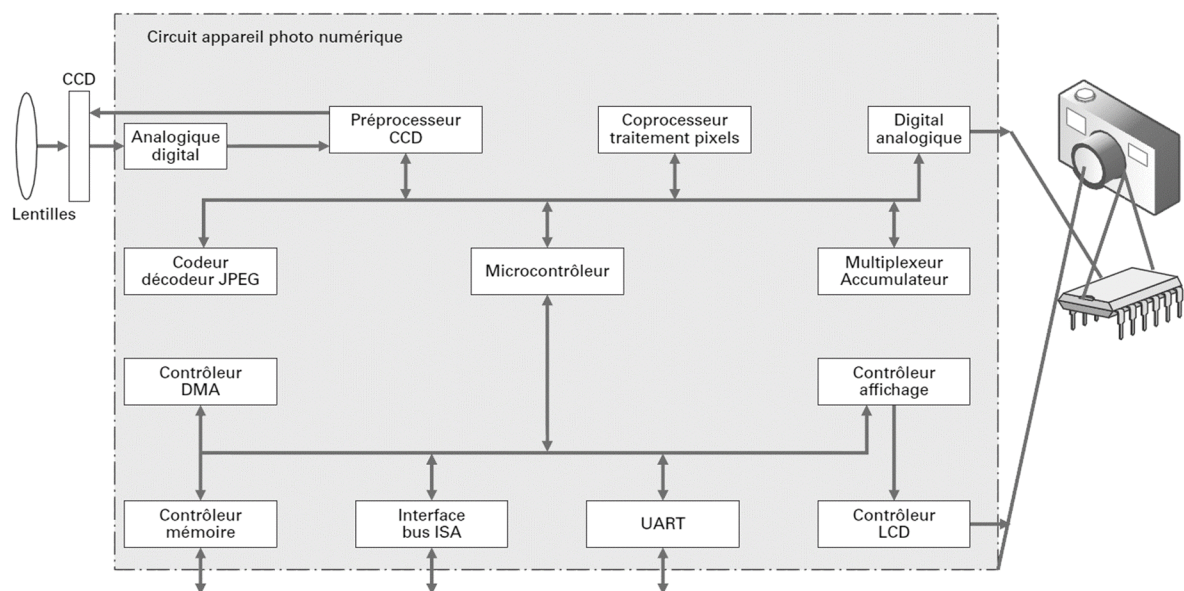


Figure 1.7- Circuit intégré d'un appareil photo

6. Domaines d'applications des systèmes embarqués :

Comme le domaine d'applications des systèmes embarqués est très large, on peut cependant distinguer deux grandes catégories :

- Les systèmes embarqués **orientés contrôle** avec les systèmes utilisés dans les transports (automobile, ferroviaire, avionique) et les systèmes de contrôle des gros équipements industriels, comme par exemple le contrôle des centrales nucléaires. Ces systèmes seront appelés « systèmes critiques » ou « systèmes temps réel dur ».
- Les systèmes embarqués **orientés calcul** ou traitement du signal, avec notamment les systèmes utilisés dans les télécommunications, le multimédia, la télévision numérique, la radio logicielle, etc. [3]

C'est deux catégories comprennent quatre domaines d'applications des systèmes embarqués en fonction des contraintes auxquels ils doivent satisfaire **Figure 1.8** :

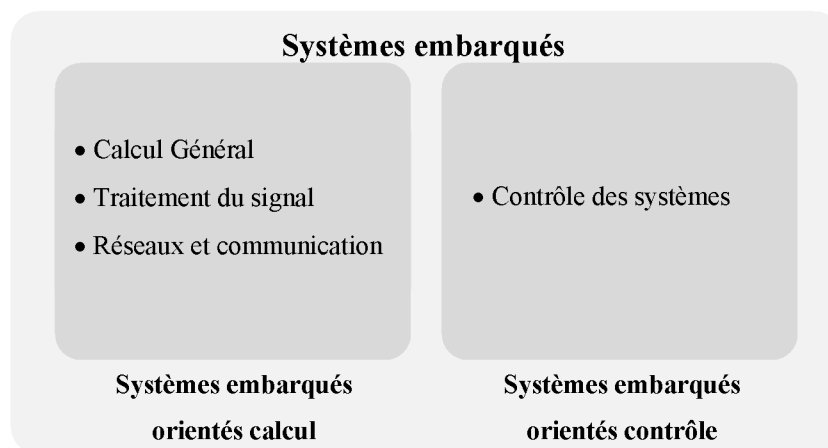


Figure 1.8- Domaines d'applications des systèmes embarqués

- **Contrôle de systèmes** (*Contrôle Systems*).
 - Contrôle de systèmes en Temps Réel.
 - Moteur d'automobile, traitement chimique, traitement nucléaire, navigation aérien.
- **Calcul général** (*Général Computing*).
 - Application similaire à une application de bureau mais empaquetée dans un SE
 - Les jeux de vidéo, set-top box, et TV Box.
- **Traitement du signal** (*Signal Processing*).
 - Calcul sur de grosses quantités de données.
 - Le radar et sonar, le dispositif de compression vidéo.

- **Réseaux et communication** (*Communications and Networking*).
 - Transmission d'information et commutation.
 - Téléphone portable, Dispositifs de l'Internet.

7. Classification des systèmes embarqués

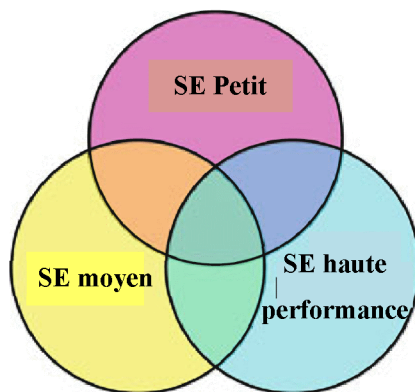


Figure 1.9 Classification des systèmes embarqués

7.1. Petit système embarqué :

Cette classe est généralement basée sur un seul microcontrôleur qui commande toute l'application. [2] Ces systèmes sont hautement intégrés, nécessitent quelques composants analogiques, capteurs, actionneurs, et interface utilisateur, selon les besoins. Ces systèmes fonctionnent avec un minimum ou sans entretien, sont à très faibles coûts, et fabriqués en grandes quantités. Le logiciel de ces systèmes est généralement mono-tâche, et ils utilisent rarement un système opérationnel temps réel (RTOS). On peut prendre le système de surveillance de pression des pneus comme un exemple de cette classe.

7.2. Système embarqué moyen :

Ce type de système est principalement utilisé pour le traitement du signal numérique. La plupart du temps, une architecture de processeur de 16 ± 32 bits est utilisée dans de tels systèmes. Ce système prend en charge la conception complexe de logiciels et de matériel et nécessite des environnements de développement intégrés, des débogueurs, des simulateurs et des outils d'ingénierie de code pour installer et analyser le logiciel. Dans ce cas, l'architecture **RISC** est la plus préférable.

Les dispositifs de prototypage rapide, les processeurs vidéo, les contrôleurs de jeux vidéo les processeurs de réseau et les systèmes d'automatisation sont des exemples de tels systèmes embarqués.

1.1. Système embarqué haute performance :

Cette classe de systèmes embarqués hautement spécialisés nécessitant des calculs rapides, robustesse, tolérance aux erreurs, et de haute maintenabilité. Ces systèmes nécessitent généralement des **ASICs** dédiés, ils sont généralement distribués, et pourraient inclure des **DSPs** et des **FPGA** comme matériel de base. Dans certains cas, la complexité de leur logiciel rend l'utilisation de RTOS pour gérer la multiplicité des tâches. Ils sont fabriqués en petites quantités avec des coûts très élevés. Ce sont le type de systèmes embarqués utilisés dans les applications militaires et aérospatiales, tels que les contrôleurs de vol, les systèmes de guidage de missiles, et les systèmes de navigation des vaisseaux spatiaux. [2]

Comme la **Figure 1.9** montre, les catégories de cette classification ne sont pas mutuellement incompatibles. Entre eux, nous pouvons trouver des zones d'intersections où les caractéristiques de deux ou de trois classes se chevauchent et les applications deviennent difficiles à associer à une seule classe spécifique.

8. Cycle de vie des systèmes embarqués :

Les systèmes embarqués ont une durée de vie limitée pendant laquelle ils subissent différentes étapes, allant de la conception ou de la naissance du système jusqu'à son déclin et dans de nombreux cas, la renaissance. Nous appelons cela le cycle de vie d'un système embarqué. Cinq étapes peuvent être identifiées dans le cycle : Conception ou naissance, Design, croissance, maturité et déclin. La **Figure 1.10** illustre la séquence de ces étapes, ainsi que les phases critiques qui composent chaque phase. [2], [4]

Une conception embarquée est d'abord conçue par l'identification du besoin de résoudre un problème particulier, ou la possibilité d'appliquer la technologie embarquée à un vieux problème. Beaucoup de systèmes embarqués sont conçus par la nécessité de changer la façon dont les vieux problèmes sont résolus ou la nécessité de fournir des solutions rentables pour eux. Autres systèmes sont conçus avec les problèmes eux-mêmes. L'occasion se présente avec la réalisation de solutions efficaces possibles par l'utilisation de systèmes embarqués. Dans tous les cas, l'application où la conception nécessite l'embarquement détermine les caractéristiques qui conduisent à la phase de conception.

Dans la phase de conception, le système passe d'abord par une phase de conception fonctionnelle, où les preuves de concept de prototypes sont développées. Cette phase permet de régler la fonctionnalité du système pour s'adapter à l'application pour laquelle il était créé. Une

analyse de faisabilité qui prend en compte des variables telles que le coût du produit, la fenêtre de marché, et l'obsolescence des composants détermine si la conception et la fabrication du produit se poursuivre ou non. Cette deuxième phase de conception établit la façon dont un prototype devient un produit et la logistique pour sa fabrication. Le design est donc l'étape la plus coûteuse dans le cycle de vie d'un système embarqué, puisque la plupart des coûts d'ingénierie non récurrents s'élèvent ici. Le volume du marché attendu guide l'entrée dans la phase de croissance.

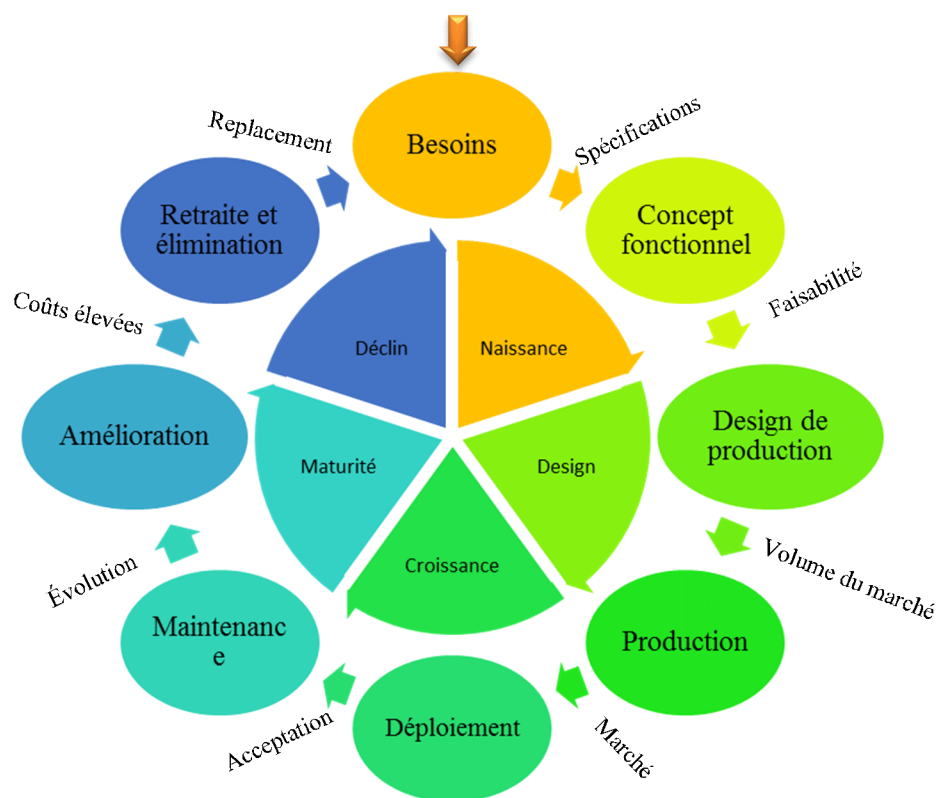


Figure 1.10 Cycle de vie d'un système embarqué

Le stade de croissance se déclenche avec la production du système pour fournir une demande estimée du marché. La production est réaménagée pour déployer le système au début de sa fenêtre de marché. La phase de déploiement implique la distribution, l'installation et la mise en place du système. L'aptitude du système à ses besoins originaux définit son acceptation, conduisant le système embarqué dans sa phase de maturité.

Dans le stade de maturité, le produit est maintenu fonctionnel et à jour. Il s'agit de fournir un appui technique, l'entretien périodique et la révision. Comme l'application évolue, le système peut nécessiter l'habilité à changer ses besoins de l'application par des mises à niveau logiciels et / ou matérielles. Le stade de maturité devrait permettre l'exécution du produit

avec des coûts minimes .Comme le système commence à vieillir, les coûts de maintenance commencent à augmenter, marquant son passage à la phase de déclin.

Dans la phase de déclin, les composants du système deviennent obsolètes et difficiles à obtenir, conduisant le coût de la maintenance, le service et les mises jours à des niveaux qui se rapprochent ou dépassent ceux de remplacer l'ensemble du système à la fois. C'est le point où le système doit être retiré du service. Les conceptions de produits doivent prévoir ce stade par élaborer des façons dont le recyclage et la réutilisation peut devenir possible, en réduisant les coûts et l'impact de la cession. Dans de nombreux cas, le remplacement du système crée de nouveaux besoins et des possibilités d'être satisfait par un nouveau design, ré-initial le cycle de vie d'un nouveau système.

1. Introduction :

Dans ce chapitre, nous intéresserons à l'étude de la conception des systèmes embarqués. Afin de comprendre le processus de conception, d'abord nous avons besoin de connaître les principales caractéristiques et contraintes de ces systèmes. Ensuite, nous explorons un exemple de conception d'un système pour comprendre les principales étapes de la conception d'un système embarqué.

2. Caractéristiques des systèmes embarqués :

Les systèmes embarqués sont beaucoup plus exigeants que le genre d'ordinateur à usage général (PC ou stations de travail). La fonctionnalité est importante à la fois pour les ordinateurs à usage générale et les systèmes embarquée, mais les applications embarquées doivent également répondre à de nombreuses autres contraintes. [6], [17], [10]

- D'une part, les systèmes embarqués doivent fournir des fonctionnalités sophistiquées :

2.1. Algorithmes complexes :

Certaines applications, comme les montres digitales ou le contrôle d'appareils ménagers, ne nécessitent que très peu de moyens de calcul. À l'autre extrémité du spectre, les applications de traitement du signal ou d'images vont utiliser des algorithmes qui peuvent être très sophistiqués et nécessiter des moyens de calcul considérables. **La Figure 2.1** montre la puissance de calcul nécessaire pour une large gamme d'applications de traitement du signal ou celui d'images. [11]

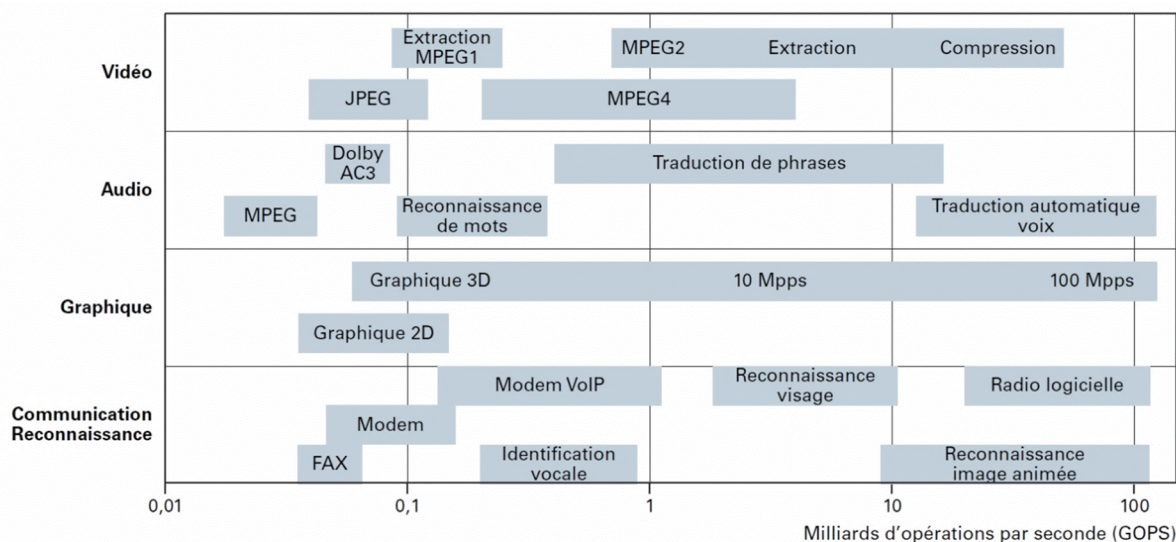


Figure 2.1- Besoin des applications en puissance de calcul

2.2. Interface utilisateur :

Les microprocesseurs sont fréquemment utilisés pour contrôler les interfaces utilisateur complexes pouvant inclure plusieurs menus et de nombreuses options. Les dispositifs de navigation GPS (Global Positioning System) sont de bons exemples d'interfaces utilisateur sophistiquées.

- Pour rendre les choses plus difficiles, les systèmes embarqués doivent souvent être réalisés pour respecter des délais :

2.3. Fonctionnement en Temps Réel :

De nombreux systèmes embarqués doivent fonctionner en temps réel. Si les données ne sont pas prêtes avant un certain délai limite, le système échoue. Dans certains cas, le non-respect des délais est dangereux et peut même mettre des vies en danger. Dans d'autres cas, manquer un délai ne crée pas de problèmes de sécurité mais crée des utilisateurs mécontents.

2.4. Multi-débit :

De nombreux systèmes embarqués ont plusieurs activités en temps réel en même temps. Ils peuvent contrôler simultanément certaines opérations qui s'exécutent lentement et d'autres qui fonctionnent à des vitesses élevées. [10], [14]

Les applications multimédia sont des excellents exemples de comportement multi-débit, les composantes audio et vidéo d'un flux multimédia fonctionnent à des vitesses différentes mais ils doivent rester synchronisés. Le non-respect d'un délai soit sur la composante audio ou vidéo endommage la présentation.

- Les coûts de toutes sortes sont également très importants :

2.5. Coût de fabrication :

Un assez grand nombre de produits d'embarqué sont sur le marché où l'utilisateur ne veut pas payer le supplémentaire pour la performance ou fonctionnalité de plus, Les concepteurs ont donc dû concevoir avec des rapports optimaux entre le prix et la performance. Le problème du coût ne se pose pas dans les mêmes termes pour tous les types de systèmes embarqués. Dans le cas de l'avionique, du ferroviaire, de l'automobile ou d'une centrale nucléaire, le coût du système informatique intégré n'est qu'une partie du coût global, même s'il n'est pas négligeable dans le cas des automobiles bas de gamme. Nous verrons que l'utilisation de composants logiques

programmables, de composants disponibles dans le commerce ou de plates-formes pour le matériel, d'outils logiciels disponibles est un facteur clé pour l'obtention de faibles coûts. Pour les systèmes enfouis dans les composants grand public, fabriqués et vendus en très grande série, les coûts jouent un rôle décisif.

2.6. Puissance et énergie :

La consommation d'énergie affecte directement le coût du matériel, car une alimentation plus grande peut être nécessaire. La consommation d'énergie affecte la durée de vie de la batterie, ce qui est important dans nombreuses applications.

3. Contrainte temporelle des systèmes embarqués :

On entend souvent parler de Temps Réel dès que l'on parle de système embarqué. En fait, un système embarqué doit généralement respecter des contraintes temporelles. Le temps réel est un concept un peu vague. On pourrait le définir comme [3] :

« Un système est dit Temps Réel lorsque l'information après acquisition et traitement reste encore pertinente. »

La littérature fournit pour la notion de « système temps réel » la définition suivante :
« Un système temps réel doit toujours livrer des réponses correctes dans des délais prédéfinis. Le dépassement de ces délais se traduit par un dysfonctionnement du système. Par conséquent, dans un système temps-réel, non seulement le résultat mais aussi l'instant auquel ce dernier est livré sont déterminant. » [17].

Pour les systèmes embarqués il existe deux types de contraintes de temps :

- Une contrainte douce (**Temps réel doux**).
- Une contrainte dure (**Temps réel dur**).

• Temps réel dur

Dans les systèmes embarqués orientés contrôle, le fonctionnement « temps réel dur » implique qu'une tâche définie soit terminée avant la fin d'une période T_{\max} prédéfinie sous peine d'une défaillance grave du système contrôle pouvant être catastrophique. On conçoit assez facilement ce qui peut se passer si le système embarqué d'une Avion n'est pas capable de traiter la tâche voulue lors de l'activation du pilote automatique lorsque l'avion est en vol. Si $T_{\text{t\^a}che}$ est la durée d'exécution de la tâche et T_{\max} la durée maximale, la relation $T_{\text{t\^a}che} < T_{\max}$ doit être

strictement respectée. Cela signifie que le fonctionnement du système embarqué doit être prévisible et déterministe. Un système pour lequel la relation serait vérifiée dans 99,9 % des cas et non vérifiée dans 0,1 % des cas ne peut être utilisé.

- **Temps réel doux**

Dans les systèmes embarqués orientés calcul, traitement du signal ou de l'image, la non-satisfaction de la contrainte $T_{\text{tâche}} < T_{\text{max}}$ ne se traduit pas par des conséquences potentiellement catastrophiques, mais par une dégradation des performances.

4. Conception des systèmes embarqués :

Les applications à bases des systèmes embarqués représentent la grande majorité de production d'applications électroniques. Les appareils de maison tels que le four à micro-ondes, et les machines à laver, les systèmes automobiles tels que les freins ABS et les mécanismes de déploiement de l'airbag, et nos appareils personnels comme les téléphones cellulaires et les lecteurs multimédias ne sont que quelques exemples représentatifs. La sensibilité de coûts vers les ressources de conception de ces systèmes est élevée, à cause des volumes élevés dans lesquels ils sont produits. De plus, la conception doit être complétée, fabriqué et lancé en temps pour maximiser les revenus de produits. Ces contraintes accompagnent la conception des applications embarquées du début à la fin.

4.2. Défis de la conception des systèmes embarqués

Les contraintes externes sont une source importante de difficultés pour la conception de systèmes embarqués. Considérons les problèmes les plus importants qui doivent être pris en compte dans la conception de systèmes embarqués :

- *De quoi avons-nous besoin ?*

Nous avons beaucoup de contrôle sur la quantité de puissance de calcul que nous appliquons à notre problème. Nous pouvons non seulement sélectionner le type de microprocesseur utilisé, mais également sélectionner la quantité de mémoire, les périphériques et plus encore. Comme nous devons souvent respecter à la fois les délais de réalisation et les contraintes de coûts de fabrication, le choix du matériel est important ; trop peu de matériel et le système ne parvient pas à respecter ses délais, trop de matériel et il devient trop cher.

- *Comment respectons-nous les délais ?*

La façon brute de respecter une date limite est d'accélérer le matériel afin que le programme s'exécute plus rapidement. Bien sûr, cela rend le système plus coûteux. Il est également tout à fait possible que l'augmentation de la fréquence d'horloge du processeur ne fasse pas assez de différence sur le temps d'exécution, car la vitesse du programme peut être limitée par le système de mémoire.

- *Comment pouvons-nous minimiser la consommation d'énergie ?*

Dans les applications alimentées par batterie, la consommation d'énergie est extrêmement importante. Même dans les applications sans batterie, une consommation électrique excessive peut augmenter la dissipation thermique. Une façon pour rendre un système numérique consomme moins d'énergie est de le faire fonctionner plus lentement, mais ralentir naïvement le système peut évidemment conduire à des délais non respectés. Une conception soignée est nécessaire pour ralentir les parties non critiques du système pour diminuer la consommation d'énergie tout en atteignant les objectifs de performance nécessaires.

- *Comment pouvons-nous concevoir pour l'évolutivité ?*

La plate-forme matérielle peut être utilisée sur plusieurs générations de produits, ou pour plusieurs versions différentes d'un produit de la même génération, avec peu ou pas de changements. Cependant, nous voulons être en mesure d'ajouter des fonctionnalités en changeant de logiciel. Comment pouvons-nous concevoir une machine qui fournira les performances requises pour un logiciel que nous n'avons pas encore écrit ?

- *Est-ce que ça marche vraiment ?*

La fiabilité est toujours importante lors de la vente de produits aux clients ; car ils s'attendent que les produits qu'ils achètent fonctionnent. La fiabilité est particulièrement importante dans certaines applications, telles que les systèmes de sécurité critiques. Si nous attendons la mise en marche d'un système et que nous essayons d'éliminer les problèmes, nous serons en retard, nous ne trouverons pas assez de problèmes, il sera trop coûteux de les régler, et cela prendra trop de temps. Un autre ensemble de défis provient des caractéristiques des composants et des systèmes eux-mêmes.

Si la programmation d'un ordinateur ressemble à l'assemblage d'une machine sur un banc, la conception d'un système embarqué est souvent comme travailler sur une voiture, délicate et difficile.

5. Processus de conception du système embarqué :

Cette section donne un aperçu du processus de conception de système intégré visant deux objectifs. Tout d'abord, il nous donnera une introduction aux différentes étapes de la conception de systèmes embarqués avant de les approfondir. Deuxièmement, cela nous permettra d'examiner la méthodologie de conception elle-même. [6], [15], [16]

La **Figure 2.2** résume les principales étapes du processus de conception de systèmes embarqués.

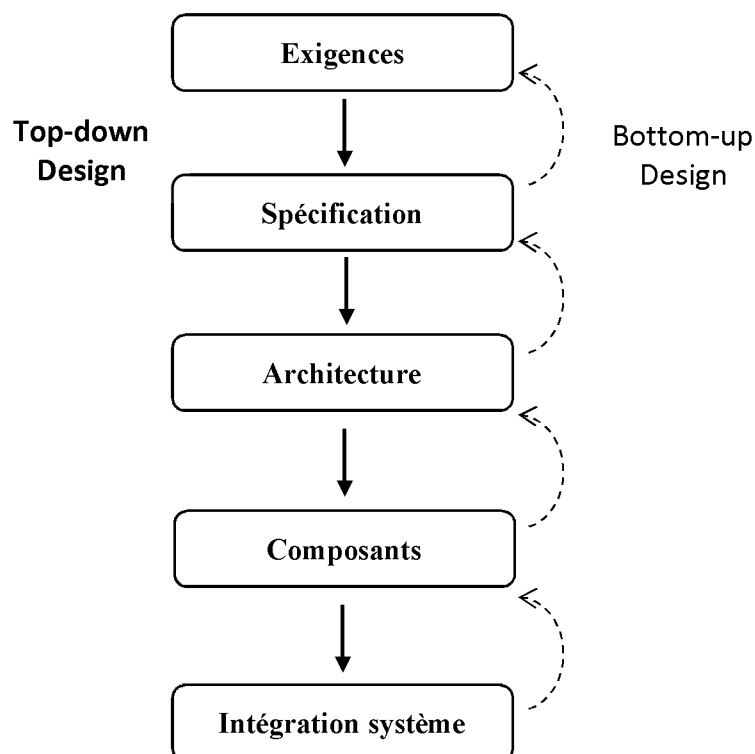


Figure 2.2- Principaux niveaux d'abstraction dans le processus de conception des systèmes embarqués

Nous commençons avec les exigences du système. À l'étape suivante, spécification, nous créons une description plus détaillée de ce que nous voulons. Mais la spécification indique seulement comment le système se comporte, pas comment il est construit. Les détails du système commencent à prendre forme lorsque nous développons l'architecture, ce qui donne au système une structure en termes de grands composants. Une fois que nous connaissons les composants dont nous avons besoin, nous pouvons les concevoir, y compris les modules logiciels et tout le matériel spécialisé dont nous avons besoin. Basés sur ces composants, nous pouvons enfin construire un système complet.

5.1. Exigences :

Avant de concevoir un système, nous devons savoir ce que nous concevons. Les étapes initiales du processus de conception capturent cette information pour l'utiliser dans la création de l'architecture et des composants. Nous procédons généralement en deux phases : d'abord, nous recueillons une description informelle des clients appelés exigences, et nous affinons les exigences dans une spécification qui contient assez d'informations pour commencer à concevoir l'architecture du système.

Les exigences peuvent être fonctionnelles ou non fonctionnelles. Nous devons bien sûr capturer les fonctions de base du système embarqué, mais la description fonctionnelle est souvent insuffisante. Les exigences non fonctionnelles incluent :

- *Performance* : La rapidité du système est souvent une considération majeure à la fois pour l'utilisabilité du système et pour son coût final. Comme nous l'avons noté, la performance peut être une combinaison de mesure de performance telles que le temps approximatif nécessaire pour exécuter une fonction au niveau de l'utilisateur et les échéances auxquelles une opération particulière doit être effectuée.
- *Coût* : Le coût du système est presque toujours pris en compte. Le coût a généralement deux composantes principales : le coût de fabrication comprend le coût des composants et de l'assemblage ; Les coûts d'ingénierie non récurrents (NRE) comprennent le personnel et les autres coûts de conception du système.

- *Taille et poids* : Les aspects physiques du système peuvent varier considérablement selon l'application. Un système de contrôle industriel pour une chaîne de montage peut être conçu pour s'intégrer dans un bâti de taille standard sans limitation stricte de poids. Un périphérique portable a généralement des exigences strictes en termes de taille et de poids.

- *Consommation d'énergie* : La puissance est importante dans les systèmes alimentés par batterie et souvent elle est importante dans d'autres applications. La puissance peut être spécifiée à l'étape des exigences en termes de durée de vie de la batterie et il est un peu probable que le client puisse décrire la puissance autorisée.

5.2. Spécification :

La spécification est plus précise, elle sert comme contrat entre le client et les concepteurs. A ce titre, le cahier des charges doit être soigneusement écrit de manière à ce qu'il reflète fidèlement les exigences du client et qu'il le fasse d'une manière qui puisse être clairement suivie pendant la conception. La spécification est probablement la phase la moins connue de cette méthodologie pour les concepteurs, mais elle est essentielle pour créer des systèmes de travail avec un minimum d'effort de concepteur. Les concepteurs qui n'ont pas une idée précise de ce qu'ils veulent construire au début font généralement des suppositions erronées au début du processus qui ne sont pas évidentes tant qu'ils n'ont pas un système fonctionnel. À ce moment-là, la seule solution est de démonter la machine, de jeter une partie et de recommencer, cela prend beaucoup de temps et de frais supplémentaires.

La spécification doit être suffisamment compréhensible pour que quelqu'un puisse vérifier qu'elle répond aux exigences du système et aux attentes du client. Elle devrait également être suffisamment clair pour que les concepteurs sachent ce qu'ils ont besoin de construire. Les concepteurs peuvent rencontrer différents types de problèmes à cause des spécifications imprécises. Si le comportement de certaines entités dans une situation particulière n'est pas clair dès la spécification, le concepteur peut implémenter la mauvaise fonctionnalité. Si les caractéristiques globales de la spécification sont erronées ou incomplètes, l'architecture globale du système dérivée de la spécification peut être incapable de répondre aux besoins de la réalisation.

5.3 Conception d'architecture :

La spécification ne dit pas comment le système fait les choses, seulement ce qu'il fait. Décrire comment le système implémente ces fonctions est le but de l'architecture. L'architecture est un plan pour la structure globale du système qui sera utilisée plus tard pour concevoir les composants de l'architecture. Nombreux concepteurs considèrent La création de l'architecture la première phase de la conception.

Pour comprendre ce qu'est une description d'architecture, nous prenons un exemple d'architecture de carte de déplacement GPS. La **Figure 2.3** montre l'architecture de système sous la forme de schéma bloc qui montre les principales opérations et le flux de données entre eux.

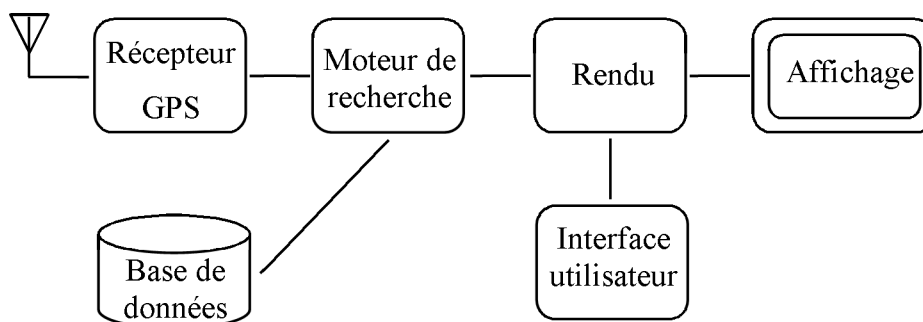


Figure 2.3 Schéma bloc de la carte de déplacement GPS

Ce schéma fonctionnel est encore assez abstrait, nous n'avons pas encore précisé quelles opérations seront effectuées par le logiciel en cours d'exécution sur l'unité centrale de traitement, ce qui sera fait par le matériel à usage spécial, etc. Cependant Le schéma fait un grand pas vers la description de la façon d'implémenter les fonctions décrites dans la spécification. Par exemple nous voyons clairement, que nous devons rechercher dans la base de données topographique et rendre (dessiner) les résultats pour l'affichage. Nous avons choisi de séparer ces fonctions afin que nous puissions les faire en parallèle en effectuant le rendu séparément de la recherche dans la base de données, ce qui peut nous aider à mettre à jour l'écran de manière plus souple.

Seulement après avoir conçu une architecture initiale qui n'est pas trop détaillée que nous devons affiner ce diagramme en deux schéma blocs, une pour le matériel et une autre pour le logiciel. Ces deux diagrammes sont présentés à la **Figure 2.4**.

Le schéma bloc matériel montre clairement que nous avons besoin d'un processeur entouré par mémoire et périphériques d'E/S. En particulier, dans cet exemple il y'a deux mémoires : une mémoire tampon pour les pixels à afficher et une mémoire de programme / données séparé pour l'utilisation générale par la CPU.

Le schéma bloc logiciel suit le diagramme du système, mais on trouve un timer qui contrôle la lecture des boutons d'interface utilisateur et la restitution des données sur l'écran.

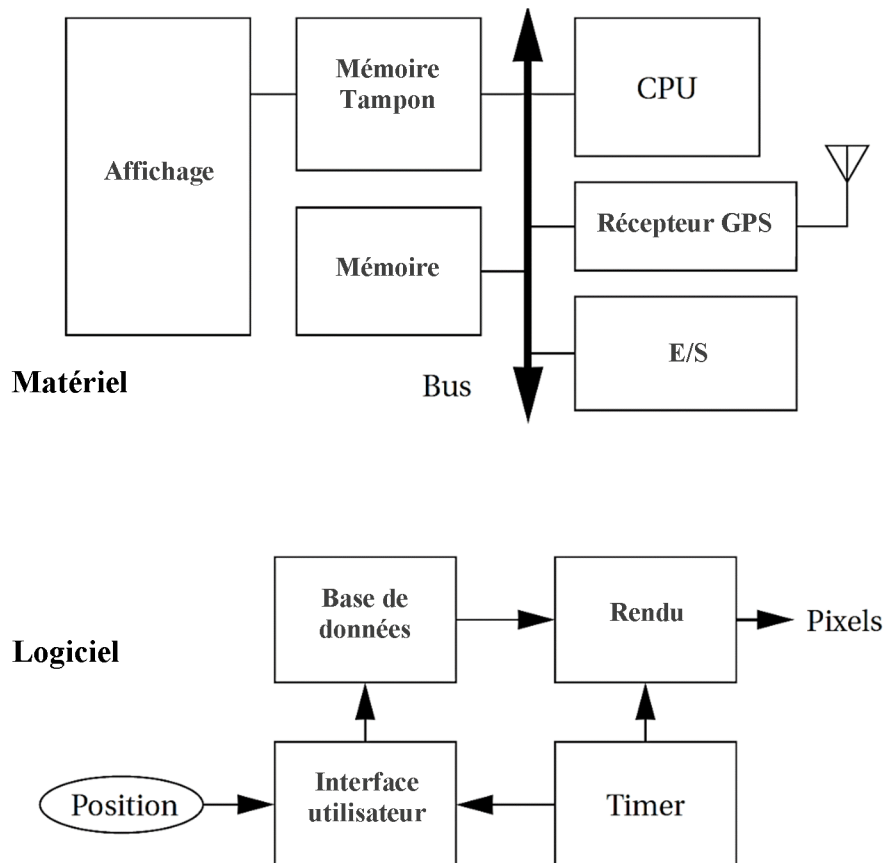


Figure 2.4 Architectures matérielles et logicielles pour carte de déplacement GPS

Les descriptions architecturales doivent être conçues pour satisfaire à la fois les exigences fonctionnelles et non fonctionnelles. Non seulement toutes les fonctions requises doivent être présentes, mais nous devons aussi répondre aux coûts, à la vitesse, à la puissance et à d'autres contraintes non fonctionnelles.

Commencer par une architecture système et affiner celle des architectures matérielles et logicielles est un bon moyen de s'assurer que nous répondons à toutes les spécifications.

Comment savons-nous que nos architectures matérielles et logicielles répondent aux contraintes de vitesse, de coût, etc. ?

Nous devons en quelque sorte être capable d'estimer les propriétés des composants des schémas bloc, Une estimation précise dérive de l'expérience, à la fois de l'expérience de conception générale et de l'expérience particulière avec des systèmes similaires.

5.4 Conception des composants matériel et logiciel :

La description de l'architecture nous dit de quels composants nous avons besoin. La conception des composants construit ces composants en conformité avec l'architecture et les spécifications. Les composants incluront généralement à la fois matériel (FPGA, cartes, etc.) et logiciels.

Certains des composants seront prêts à l'emploi. Par exemple le processeur, sera un composant standard dans presque tous les cas, tout comme les puces de mémoire et de nombreux autres composants. Dans la carte de déplacement GPS, le récepteur GPS est un bon exemple d'un composant spécialisé qui sera néanmoins un composant standard prédéfini.

Nous pouvons également utiliser des modules logiciels standard. Un bon exemple est la base de données topographique. Des bases de données topographiques standard existent, et nous utilisons probablement des routines standard pour accéder à la base de données. L'utilisation d'un logiciel standard pour ces fonctions d'accès nous permet non seulement de gagner du temps de conception, mais aussi de mettre en œuvre plus rapidement des fonctions spécialisées telles que la phase de décompression des données.

Nous devons concevoir nous-même certains composants. Même si nous n'utilisons que des circuits intégrés standard, nous devons peut-être concevoir la carte de circuit imprimé qui les connecte. Nous devons également faire beaucoup de programmation personnalisée. Lors de la

création de ces modules logiciels intégrés, nous devons assurer que le système fonctionne correctement en temps réel et qu'il ne prend pas plus d'espace mémoire que ce qui est autorisé. Nous devons faire attention à la lecture et l'écriture de la mémoire pour réduire la consommation d'énergie. Par exemple, les accès mémoire étant une source importante de consommation d'énergie, les transactions mémoire doivent être soigneusement planifiées pour éviter de lire plusieurs fois les mêmes données.

5.5 L'intégration du système :

Seulement après que les composants sont construits, nous pouvons les mettre ensemble et de voir un système fonctionnel. Bien sûr, cette phase consiste généralement beaucoup plus que simple branchement. Les bogues (défauts) sont typiquement trouvés pendant l'intégration de système, et une bonne planification peut nous aider à trouver les problèmes rapidement. Par l'exécution des tests correctement choisis pendant la phase de construction du système, nous pouvons souvent trouver des défauts plus facilement. Pendant les phases de conception de l'architecture et des composants nous devons assurer que l'assemblage du système et le test des fonctions sont relativement indépendants.

L'intégration du système est difficile car elle révèle généralement les problèmes. Il est souvent difficile d'observer le système avec suffisamment de détails pour déterminer exactement ce qui ne va pas, les possibilités de débogage pour les systèmes embarqués sont généralement beaucoup plus limitées que celles que l'on trouverait sur les systèmes de bureau. Par conséquent, déterminer pourquoi les choses ne fonctionnent pas correctement et comment elles peuvent être réparées est un défi en soi. Une attention particulière portée à l'insertion d'outils de débogage pendant la conception peut faciliter les problèmes d'intégration du système, mais la nature des systèmes embarqués signifie que cette phase sera toujours un défi.

6. Méthodologies de conception :

Pourquoi méthodologies de conception ?

Le processus de conception est important parce que sans ce dernier, nous ne pouvons pas livrer de manière fiable les produits que nous voulons créer. Penser à la séquence des étapes nécessaires pour construire quelque chose peut sembler superflu, mais le fait est que tout le monde a son propre

processus de conception, même s'il ne l'exprime pas. Si vous concevez vous-même des systèmes embarqués, il est bon d'avoir vos propres habitudes de travail, Mais lorsque plusieurs personnes travaillent ensemble sur un projet, ils doivent se mettre d'accord sur qui va faire les choses et comment ils vont se faire. Par conséquent, puisque de nombreux systèmes embarqués sont trop complexes pour être conçu et construit par une seule personne, nous devons penser à des processus de conception.

L'objectif évident d'un processus de conception est de créer un produit qui fait quelque chose d'utile. Les spécifications typiques pour un produit comprendront des fonctionnalités, coût de fabrication, performance, consommation d'énergie, ou d'autres propriétés. Bien sûr, un processus de conception a plusieurs objectifs importants au-delà de la fonction, de la performance et de la puissance. Deux de ces objectifs sont résumés ci-dessous :

- *Time-to-Market* :

Les clients veulent toujours de nouvelles fonctionnalités. Le produit qui sort en premier peut gagner le marché, même en définissant les préférences des clients pour les générations futures du produit. [9]

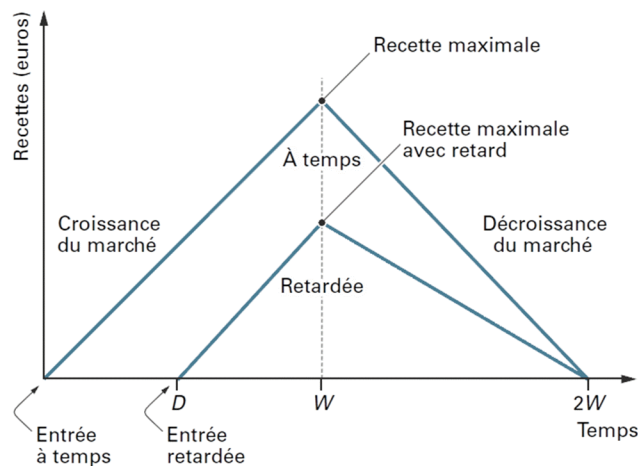


Figure 2.5 Influence d'un retard à l'arrivée sur le marché sur les recettes d'un produit.

- *Coût de conception :*

De nombreux produits de consommation sont très sensibles aux coûts. Les coûts de conception du système sont distincts des coûts de fabrication, le coût des salaires des ingénieurs, des ordinateurs utilisés dans la conception, etc. doit être réparti entre les unités vendues. Dans certains cas, une ou quelques copies d'un système embarqué peuvent être construites, de sorte que les coûts de conception peuvent dominer les coûts de fabrication. Les coûts de conception peuvent également être importants pour les systèmes grand public lorsque les contraintes de *Time-to-Market* augmentent la taille d'équipes de conception.

Les processus de conception évoluent avec le temps. Ils changent en raison des forces externes et internes. Les clients peuvent se changer, les exigences changent, les produits changent et les composants disponibles changent. Une bonne méthodologie est essentielle pour construire des systèmes qui fonctionnent correctement. La fourniture de systèmes défectueux aux clients provoque toujours l'insatisfaction. Mais dans certaines applications, telles que les systèmes médicaux et automobiles, les bogues créent de sérieux problèmes de sécurité qui peuvent mettre en danger la vie des utilisateurs.

6.1. Flux de conception :

Un flux de conception est une séquence d'étapes à suivre lors d'une conception [6], [16]. Certaines étapes peuvent être réalisées par des outils, tels que des compilateurs ou des systèmes de CAO, d'autres étapes peuvent être effectuées manuellement.

Les systèmes embarqués impliquent souvent la conception de matériel ainsi que de logiciels, même si vous ne concevez pas une carte électronique, vous pouvez sélectionner des cartes et branchez ensemble plusieurs composants puis écrire un code.

La **Figure 2.6** présente une méthodologie de conception pour un projet où le matériel et logiciel sont combinés. Les activités préliminaires telles que les spécifications et l'architecture considèrent simultanément les aspects matériels et logiciels. De même, l'intégration - et les tests- prennent en compte l'ensemble du système. Le développement de composants matériels et logiciels peut aller d'une manière indépendante, tandis que le test de l'un nécessitera la présence de l'autre, la plupart du travail matériel et logiciel pouvant s'effectuer d'une manière relativement indépendante. [16]

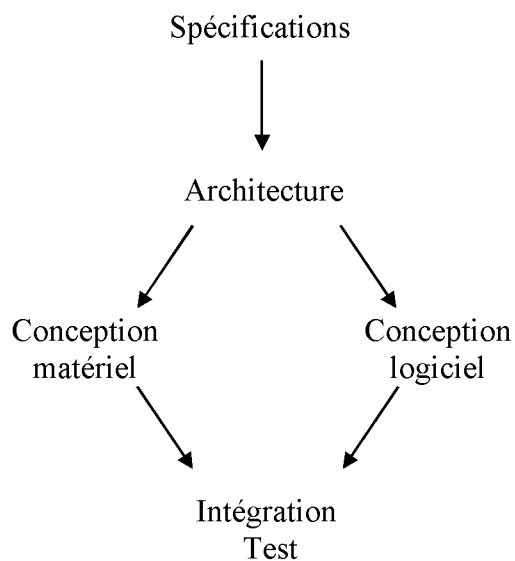


Figure 2.6. Méthodologie de conception simple de système embarqué.

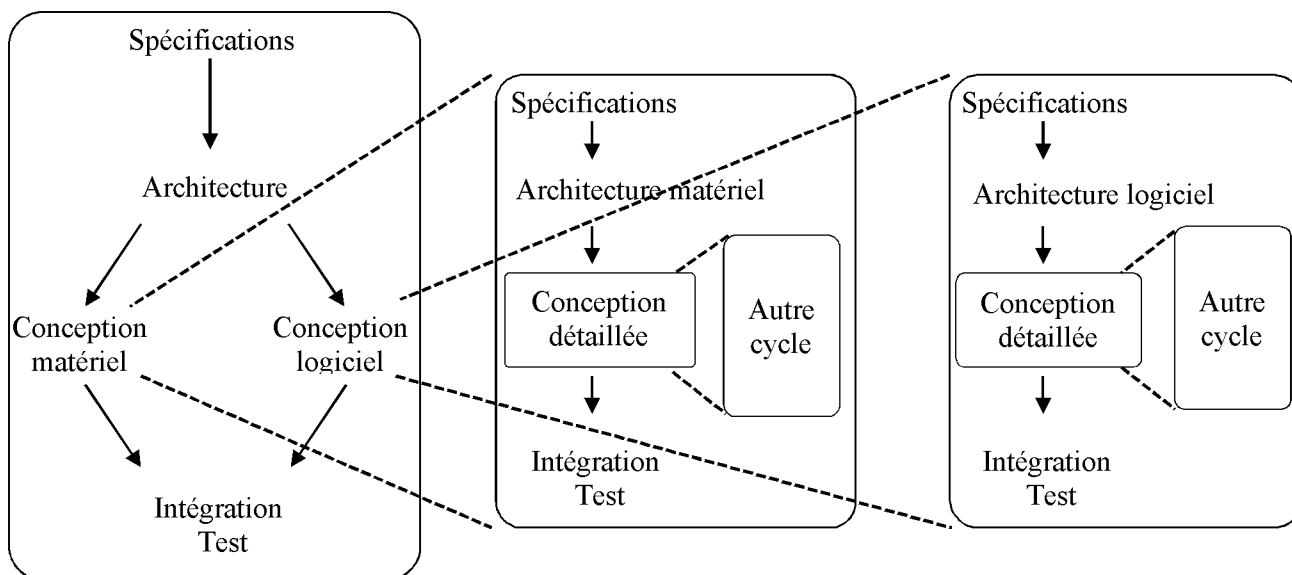


Figure 2.7. Flux de conception hiérarchique pour système embarqué.

En fait, de nombreux systèmes embarqués complexes sont construits avec des conceptions plus petites. Le système complet peut nécessiter la conception de composants logiciels importants, d'une logique personnalisée, etc., et ceux-ci peuvent à leurs tours être construits à partir de composants plus petits qui doivent être conçus. Le flux de conception suit les niveaux d'abstraction dans le système, depuis les flux de conception de système complets aux flux de conception les plus abstraits jusqu'aux composants individuels. Le flux de conception de ces systèmes complexes ressemble au flux illustré à la **Figure 2.7**. La phase d'implémentation d'un flux est elle-même un flux complet de la spécification à travers les tests. Dans un projet aussi complexe, chaque flux sera probablement géré par des personnes ou des équipes distinctes. Les équipes doivent s'appuyer sur les résultats de l'autre. Les équipes de composants répondent aux exigences de l'équipe qui gère le niveau d'abstraction le plus élevé, et l'équipe de niveau supérieur s'appuie sur la qualité de la conception et des tests effectués par l'équipe de composant. Une bonne communication est vitale dans de tels grands projets. [2], [16]

La **Figure 2.8** montre une méthodologie de conception de raffinement successive. Dans cette approche, le système est construit plusieurs fois. Un premier système est utilisé comme prototype et les modèles successifs du système sont affinés. Cette méthodologie est logique lorsque vous n'êtes pas familier avec le domaine d'application pour lequel vous construisez le système. Affiner le système en construisant plusieurs systèmes de plus en plus complexes vous permet de tester l'architecture et les techniques de conception. [16]

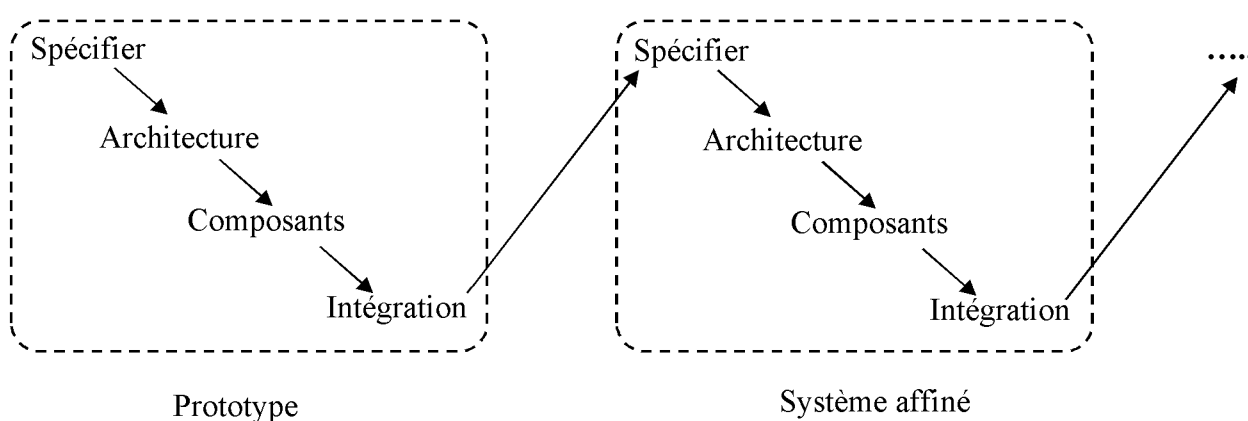


Figure 2.8 Modèle de conception raffinement successif.

1. Introduction :

Les Systèmes Embarqués sont de nos jours de plus en plus communicant s'appuyant nécessairement sur des réseaux de communication.

Nous présentons dans ce chapitre les principaux protocoles et supports de communication utilisés dans les systèmes embarqués communicants.

2. Systèmes embarqués communicants

Les systèmes embarqués communicants incluent des dispositifs tels que des assistants personnels numériques (PDA), des voitures intelligentes, etc. soit fixes ou mobiles ils sont tous le temps connectés aux réseaux au moyen de liens filaires ou sans fil. L'embarqué communicant connaît une évolution importante grâce aux nouvelles technologies de l'information et de la communication.

3. Les protocoles de communication série :

3.1. Inter-Integrated Circuit (I2C) :

Introduction :

Le bus I2C est développé par Philips au début des années 80 pour optimiser l'efficacité du matériel et la simplicité du circuit. L'interface I2C est de type maître / esclave simple. La simplicité du système I2C est principalement due à la conception bidirectionnelle à deux fils, une ligne de données série (SDA) et une autre d'horloge série (SCL), ainsi qu'au format de protocole.

I2C est approprié pour l'interfaçage avec des périphériques sur une seule carte et peut être étendu sur plusieurs cartes à l'intérieur d'un système fermé (distance courte). [8]

Le bus I2C :

I2C est un bus de communication série synchrone à deux fils, comme le montre la Figure 3.3. Il n'a pas besoin de chip select, ce qui le rend peu coûteux et simple à implémenter dans un système.

Le bus I2C se compose physiquement de deux fils actifs et d'une connexion à la masse. Les fils actifs, appelés SDA et SCL, sont tous deux bidirectionnels :

SCL : serial clock qui est l'horloge de cadencement des communications (unidirectionnel)

SDA : serial data permet les échanges bidirectionnels entre le maître et un esclave.

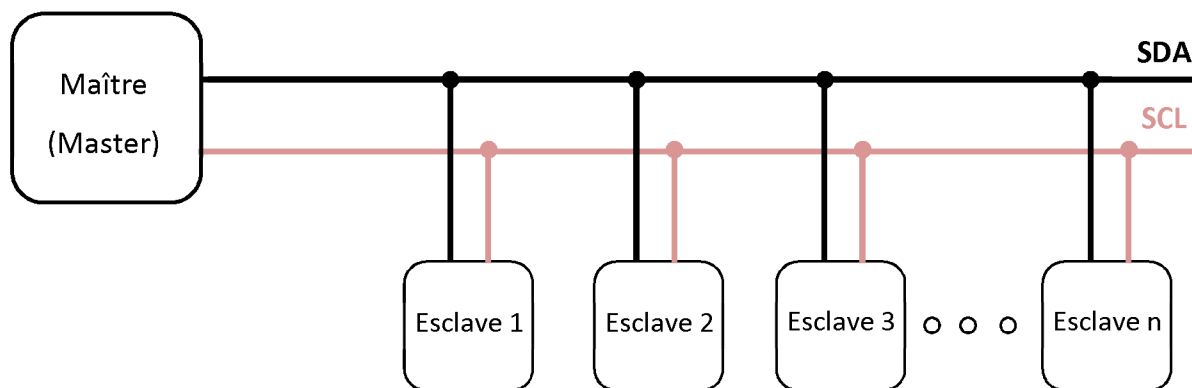


Figure 3.1 Vue Générale du Bus I2C

La fréquence de transfert est comprise entre 100KHz et 400KHz. Le bus est piloté par un maître (MASTER) qui génère l'horloge de communication. Tous les autres circuits sont esclaves (SLAVES), ils reçoivent tous la même horloge issue du maître.

Le protocole I2C prend en charge plusieurs maîtres, mais la plupart des conceptions de systèmes en incluent une seule. Il peut y avoir un ou plusieurs esclaves dans le bus.

Chaque périphérique esclave compatible-I2C est livré avec une adresse prédéfinie dont les bits à poids faible peuvent être configurés au niveau de la carte. Le maître transmet l'adresse de l'esclave au début de chaque transaction, chaque esclave est chargé de surveiller le bus et de répondre uniquement à sa propre adresse. Ce schéma d'adressage limite le nombre de dispositifs esclaves identiques pouvant exister sur le même bus I2C.

Sur le bus I2C le niveau logique dominant est le (0), le niveau récessif est le (1). En absence de commande le niveau logique (repos) est donc 1. Les interfaces sont constituées de transistors MOS drains ouverts, ce qui élimine tout risque de court-circuit électrique (mais pas de court-circuit logique). Le bus doit être équipé de deux résistances de pull-up (4K à 10K)

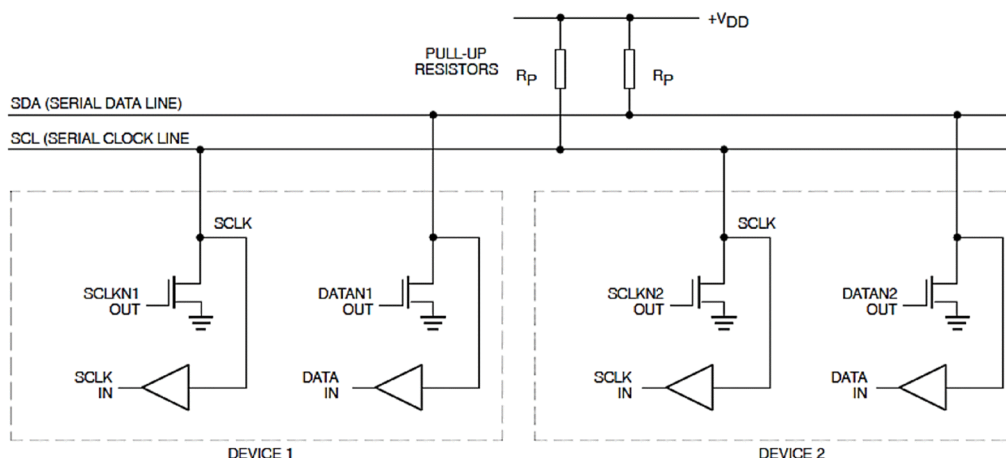


Figure 3.2 Connexion des dispositifs au bus I2C

Comme entrée et sortie se font sur un même fil, un circuit peut vérifier l'état récessif de la ligne. Il place un 1 logique (transistor bloqué) et vérifie que la ligne est bien à 1, dans le cas contraire, c'est qu'un autre circuit est en train de placer un 0. L'émetteur peut donc vérifier l'émission effective de chaque bit.

Protocole I2C :

Quelque soit le périphérique (MCU, afficheur LCD ou une mémoire) connecté au bus, il a son propre adresse unique. Chacune de ces périphérique peut agir comme un récepteur et / ou un émetteur, en fonction de la fonctionnalité. De toute évidence, un afficheur LCD est seulement un récepteur, tandis qu'une mémoire peut être à la fois émetteur et récepteur.

Tout d'abord, le Maître émet une condition START. Cela agit comme un signal «Attention» à tous les appareils connectés. Tous les périphériques du bus écoutent le bus pour les données entrantes.

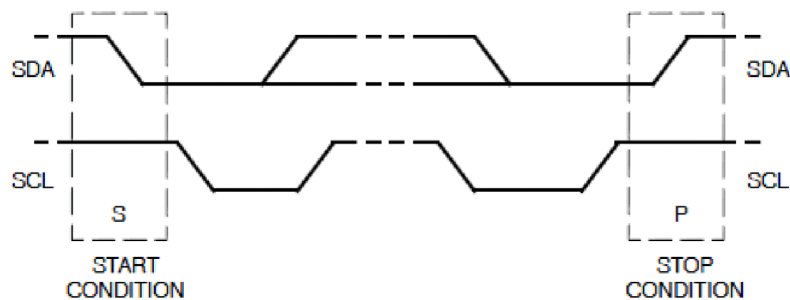


Figure 3.3 Les conditions START and STOP

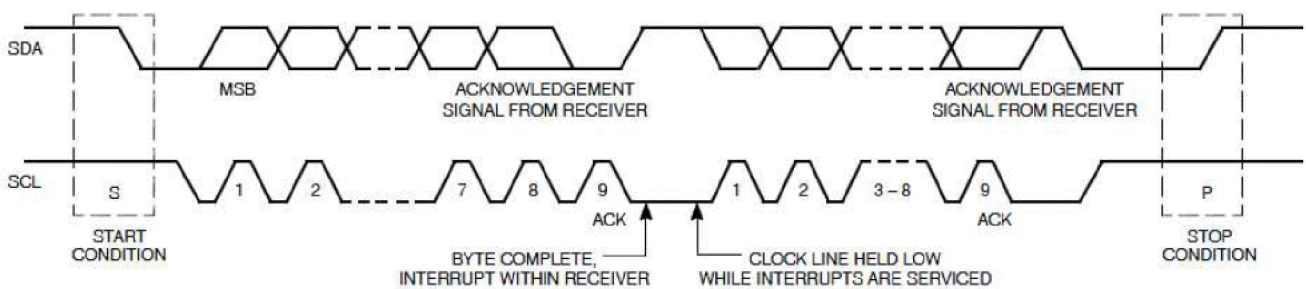


Figure 3.4 Transfert des données sur le bus I2C

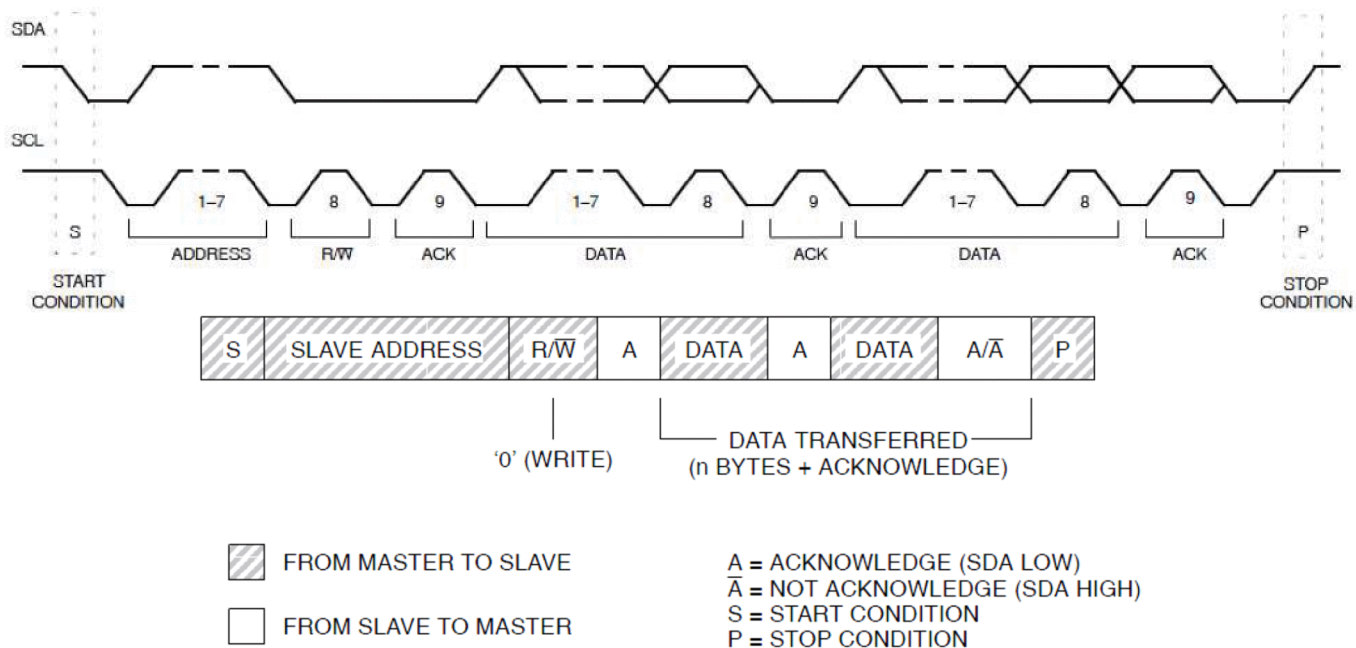
Ensuite, le Maître envoie l'ADRESSE du périphérique auquel il veut accéder, avec une indication si l'accès est une opération de lecture ou d'écriture. Ayant reçu l'adresse, tous les périphériques vont la comparer avec leur propre adresse. Si elle ne correspond pas, ils attendent

simplement que le bus soit libéré par la condition STOP (**Figure 3.3**). Si l'adresse correspond, cependant, le périphérique produira une réponse appelée le signal ACKNOWLEDGMENT.

Une fois que le Maître a reçu l'accusé de réception, il peut commencer à émettre ou à recevoir les données. Lorsque tout est terminé, le maître émet la condition STOP. Ceci est un signal que le bus a été libéré.

Exemple d'échange sur un bus I2C :

Dans cet exemple le maître transmet l'adresse d'écriture d'un esclave suivi de deux octets de données.



Avantages et inconvénients du bus I2C :

Avantages	Inconvénients
Utilisation de 2 lignes seulement pour transmettre les informations.	En plus d'être limité par les adresses, le nombre d'équipement maximum est limité par la charge capacitive du bus maximale (400pF).
Possibilité de configuration multi - maîtres	Faible débit d'échange de données (de 100 à 400 kbits)
Possibilité de connecter 128 équipements.	
Pré-intégré dans les circuits programmables.	
Possibilité d'ajouter ou de supprimer l'une des périphériques sans affecter les autres circuits sur le bus	

Tableau 3.1 Avantages et inconvénients de la liaison SPI

3.2 Serial Peripheral Interface (SPI) :

Introduction :

SPI est une interface série synchrone à usage général. Pendant un transfert SPI, les données d'émission et de réception sont décalées simultanément et en série. Une ligne d'horloge série synchronise le décalage des informations sur les deux lignes de données série.

Motorola a créé le port SPI au milieu des années 1980 pour l'utiliser dans ses familles de microcontrôleur. Le SPI est principalement utilisé pour permettre à un microcontrôleur de communiquer avec des périphériques tels que les EEPROM.

Les périphériques SPI communiquent en utilisant une relation maître-esclave. En raison d'absence d'adressage de périphérique intégré, SPI nécessite plus de matérielles que I2C lorsque plus d'un esclave est impliqué. Mais SPI tend à être plus simple et plus efficace que I2C dans les applications point à point (maître unique, esclave unique).

Le bus SPI :

SPI est un standard de bus série établi par Motorola et pris en charge dans les produits de divers fabricants. Les interfaces SPI sont disponibles sur la plupart des processeurs et des microcontrôleurs. Il s'agit d'une liaison de données série synchrone fonctionnant en Full Duplex.

La communication entre les périphériques est de type maître / esclave, dans laquelle le maître initie la trame de données. Lorsque le maître génère une horloge et sélectionne un périphérique esclave, les données peuvent être transférées dans l'une ou l'autre direction ou dans les deux directions simultanément. En effet, en ce qui concerne SPI, les données sont toujours transférées dans les deux sens. Il appartient aux périphériques maître et esclaves de savoir si l'octet reçu est significatif ou non.

SPI spécifie quatre signaux :

SCK : Horloge.

SI : Sortie de données maître, entrée de données esclaves.

SO : Entrée de données maître, sortie de données esclave.

CS : Chip select.

La **Figure 3.6** montre ces signaux dans une configuration à un seul esclave, **SCK** est généré par le maître et envoyé à tous les esclaves. **SI** transmet les données de maître à esclave. **SO** transporte les données de l'esclave au maître. Un périphérique esclave est sélectionné lorsque le maître affirme son signal **CS**.

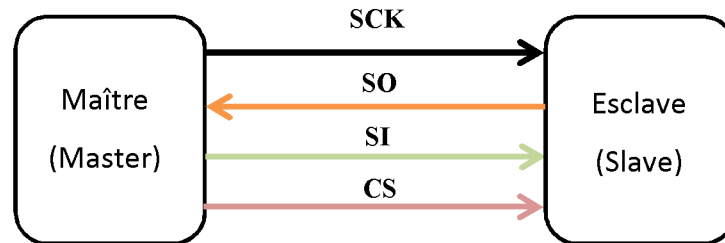


Figure 3.6 liaison SPI Maître - Esclaves

Si plusieurs périphériques esclaves existent, le maître génère un signal de sélection pour chaque périphérique esclave. Cette configuration est illustrée dans la **Figure 3.7**.

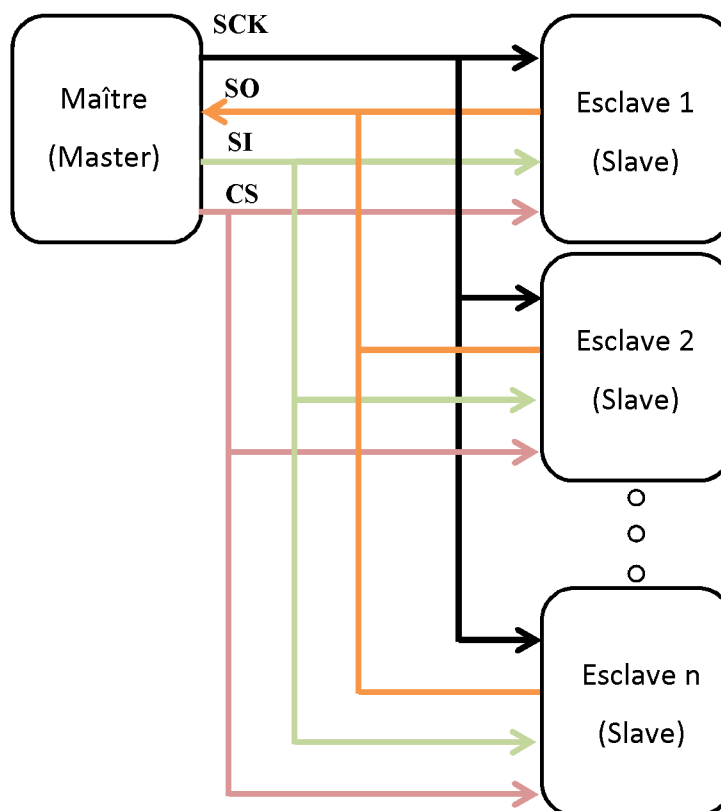


Figure 3.7 liaison SPI Maître - Multi Esclaves

Protocole SPI :

Toutes les liaisons SPI respectent le chronogramme général ci-dessous, mais quatre variantes existent au niveau de la prise en compte des données qui sont échangées :

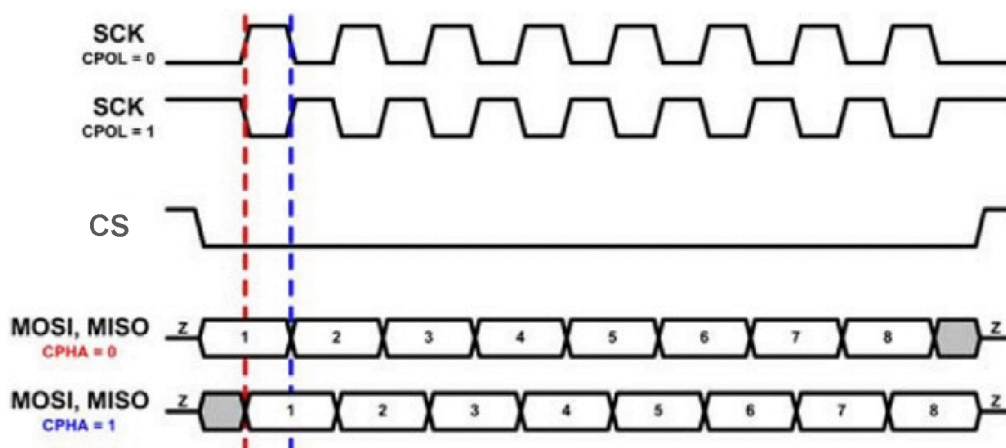


Figure 3.8 Chronogramme général d'un échange sur une liaison SPI

Ces variantes sont parfois appelées modes SPI et sont alors numérotées de 0 à 3. En présence d'un circuit à interface SPI, il vous suffira de comparer sa fiche technique avec le chronogramme pour savoir dans quel « mode » il fonctionne. Ces quatre modes sont définis par la polarité de l'horloge, appelée CPOL (Clock POLarity), qui peut-être égale à 0 ou à 1, et par la phase de cette même horloge, appelée CPHA (Clock PHAse), par rapport aux données, qui peut être aussi égale à 0 ou à 1. Cela conduit aux quatre possibilités, qui indiquent sur quel front du signal d'horloge les données sont échantillonnées et peuvent donc être considérées comme valides. Par défaut, le mode le plus courant est appelé mode 0, c'est-à-dire celui où CPOL et CPHA sont tous les deux nuls.

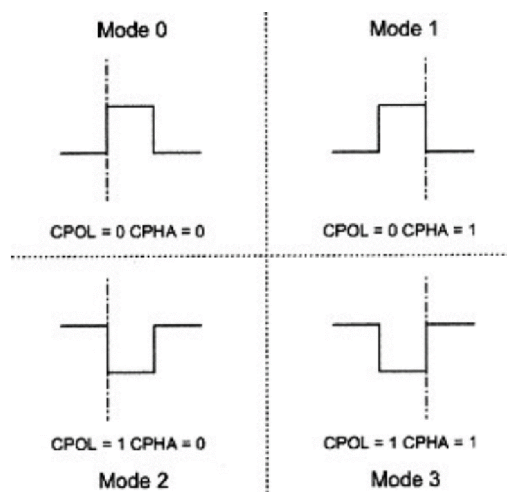
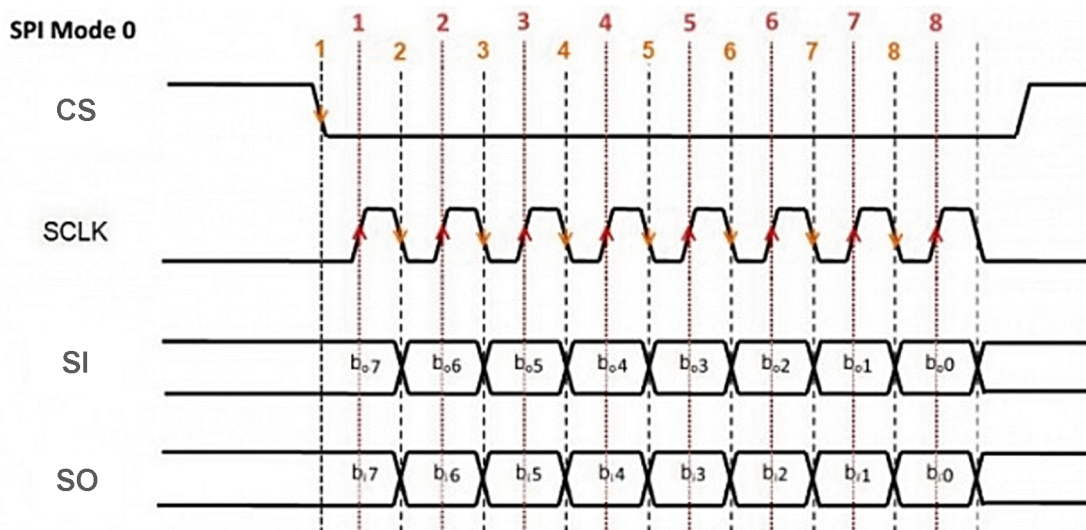


Figure 3.9 Les modes de prise en compte des données de la liaison SPI

Exemple d'échange sur un bus SPI :

Nous prendrons comme exemple l'échange d'une donnée au format d'un octet.



- Début de la communication : Le maître du bus SPI, met le signal CS à 0 de façon à indiquer à l'esclave que la communication commence (Slave Select est actif à l'état bas).
- Acquisition et transfert des bits : Le maître fera ensuite basculer huit fois le signal d'horloge. Sur les fronts montants, l'esclave enregistre les données provenant du signal SO et le maître mémorise sur la ligne SI la valeur de chaque bit provenant de l'esclave, c'est le Full duplex.
- Fin de la communication : Le maître tire la ligne CS à l'état haut pour indiquer que le transfert est terminé.

Avantages et Inconvénients du bus SPI :

Avantages	Inconvénients
Communication en Full Duplex	Pas d'adressage possible
Indépendant du nombre de bits à transmettre	Utilisation sur très courte distance (même carte)
Pas de collision possible	Nécessite plus de fils que I ² C
Les esclaves utilisent l'horloge du maître pas de problème de précision de quartz	Pas d'acquittement (le maître ne sait pas s'il est écouté)
Beaucoup plus rapide que I ² C en mode standard	
Possibilité de configuration à plusieurs maîtres	

Tableau 3.2 Avantages et inconvénients de la liaison SPI

4. Technologie de communication sans fil :

La technologie de communication sans fil repose sur les réseaux sans fil. Un réseau sans fil est un réseau dans lequel au moins deux terminaux peuvent communiquer sans liaison filaire. Les réseaux sans fil sont basés sur une liaison utilisant des ondes radioélectriques. L'installation de tels réseaux ne demande pas de lourds aménagements des infrastructures existantes. [31]

4.1. Généralités sur les réseaux sans fils :

4.1.1. Catégories des réseaux sans fils :

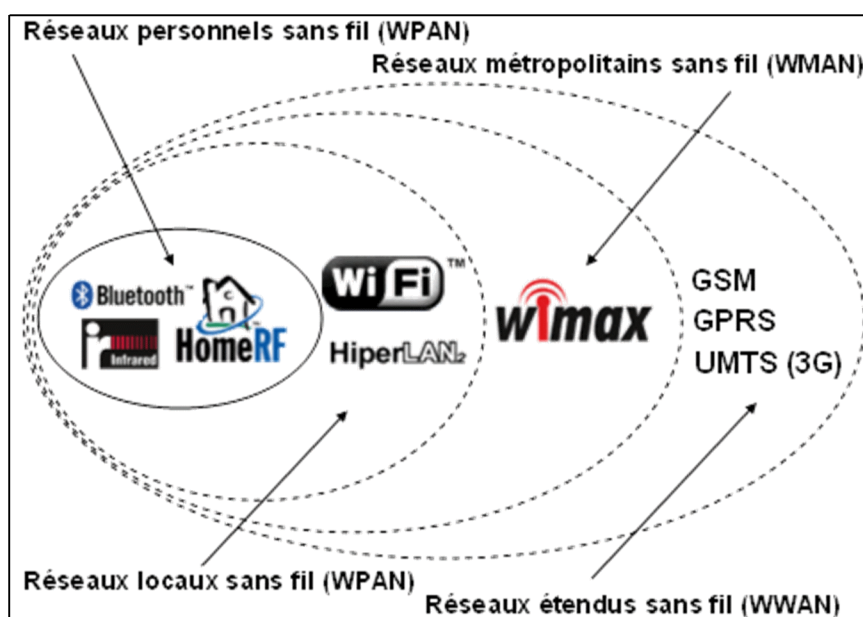


Figure 3.10 Les différentes catégories des réseaux sans fils

Les réseaux sans fils sont répartis en plusieurs catégories, selon le périmètre géographique offrant la connectivité appelée : « zone de couverture » (figure 3.9).

- **WPAN** : [*Wireless Personal Area Network*], réseau personnel sans fil, appelé également réseau individuel sans fil, concerne les réseaux sans fils d'une faible portée, de l'ordre de quelques dizaines de mètres. Ce type de réseau a recours aux technologies Bluetooth, HomeRF (*Home Radio Frequency*), ZigBee, et infrarouge. [2].
- **WLAN** : [*Wireless Local Area Network*], réseau local sans fil, permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de

mètres. Il permet de relier entre les terminaux présents dans la zone de couverture. le *Wi-Fi* est la technologie la plus connue dans ce type de réseaux.

- **WMAN** : [*Wireless Metropolitan Area Network*], réseau métropolitain sans fil, connu sous le nom de boucle local radio (*BLR*). Les *WMAN* sont fondés sur la norme *IEEE 802.16*. La norme de réseau métropolitain sans fil la plus connue est le *WIMAX*, permettant d'obtenir des débits de l'ordre de 70Mbit/s sur un rayon de plusieurs kilomètres.
- **WWAN** : [*Wireless Wide Area Network*], réseau étendu sans fil : également connu sous le nom de réseau cellulaire mobile. Il s'agit des réseaux sans fils les plus répandus puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fil. Les principales technologies sont *GSM (Global System for Mobile Communication. Groupe Spécial Mobile)*, *GPRS (General Packet Radio Service)* et *UMTS (Universal Mobile Telecommunications System)*.

Quelques types de réseaux, et ses technologies apparentées vont être étudié en détail par la suite, du plus petit au plus grand en termes de distances couvertes.

2.1. Les Réseaux personnels sans fil (WPAN) :

Le réseau personnel sans fil, noté *WPAN* pour *Wireless Personal Area Network*, concerne les réseaux sans fil d'une faible portée, de l'ordre de quelques dizaines de mètres. Ce type de réseau sert généralement à relier des périphériques (imprimante, téléphone portable, appareils domestiques, ...) ou un assistant personnel (PDA) à un ordinateur sans liaison filaire ou bien à permettre la liaison sans fil entre deux machines très peu distantes. Il existe plusieurs technologies utilisées pour le *WPAN* [1]

4.1.2. Le ZigBee :

- **Présentation de la technologie**

La technologie *ZigBee* aussi connue sous la norme *IEEE 802.15.4* permet d'obtenir des liaisons sans fil à très bas prix et avec une très faible consommation d'énergie, ce qui la rend particulièrement adaptée pour être intégrée dans de petits appareils électroniques. Le protocole *ZigBee* n'utilise que quatre couches sur les sept du modèle OSI. Il comprend une couche Physique qui contient la fréquence radio de l'émetteur-récepteur ainsi que son mécanisme de contrôle de bas niveau, et une sous couche *MAC (Medium Access Control)* qui donne accès au canal physique pour tous les types de transfert. Les couches supérieures sont constituées d'une couche réseau *NWK*

(*Network Layer*) qui fournit la conjuration du réseau, la manipulation et le routage des messages et d'une couche application *APL* (*Application Layer*) qui est dotée de 2 sous couches *APS* (*Application support sublayer*), qui lui est chargé de gérer la table de correspondance permettant d'associer les équipements entre eux et l'entité *ZDO* (*ZigBee Device Object*) qui est chargée de définir le rôle d'un objet dans le réseau et de sécuriser les relations entre les équipements.

- **Le principe de fonctionnement de ZigBee :**

Le réseau ZigBee est un réseau maillé sans fil. Chaque nœud (module) doit s'identifier avant de rejoindre le réseau. La norme IEEE 802.15.4 met en place deux topologies.

Topologie étoile :

Dans cette configuration, la pièce maîtresse, est le coordinateur. Il joue en quelque sorte le rôle de passerelle. Si un dispositif **A** souhaite dialoguer avec un dispositif **B**, la communication doit impérativement passer par le coordinateur. Dans cette topologie, les dispositifs sont très simples et n'embarquent qu'une couche simplifiée du protocole ZigBee. Le coordinateur est en général fixe et alimenté pour garantir le fonctionnement du réseau (**Figure 3.11**).

Topologie Point à Point :

Les dispositifs sont capables de dialoguer directement entre eux s'ils sont à proximité ou bien d'utiliser le coordinateur pour contacter un dispositif à plus longue distance. Dans cette topologie, les dispositifs sont plus compliqués et intègrent entièrement le protocole ZigBee.

Une mécanique de tables de liaison permet également à chaque dispositif de jouer le rôle de coordinateur, permettant à deux dispositifs éloignés de dialoguer par l'intermédiaire d'un élément à portée (**Figure 3.12**).

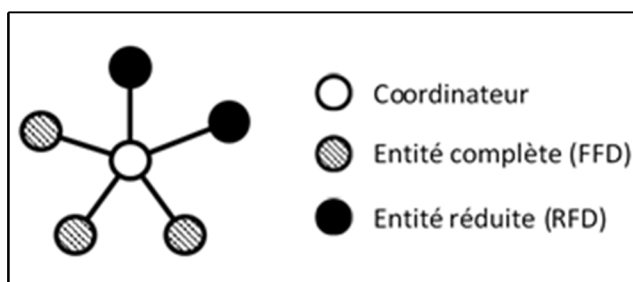


Figure 3.11 Représentation de la topologie en étoile.

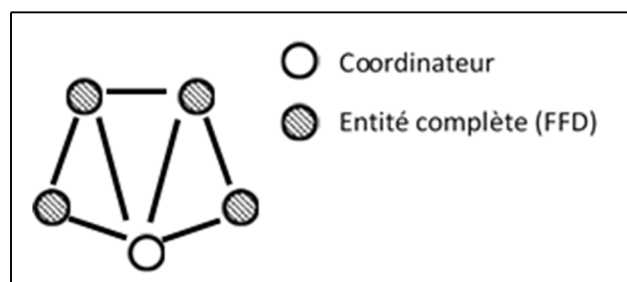


Figure 3.12 Représentation de la topologie point à point.

4.1.3. Le Bluetooth :

- **Présentation de la technologie**

Le *Bluetooth* (déposé à l'IEEE, sous le nom de 802.15) constitue donc une technologie de réseau personnel ou *PAN (Personal Area Network)*. C'est une technologie non protégée gérant les connexions sans fil de type onde radio utilisant la bande des 2,45 GHz, d'un débit de 1 Mbps, d'une portée d'environ 30 m et offrant un balayage de 360°, tout en s'affranchissant des obstacles les plus courants. Cette technologie concurrence fortement *IrDA*.

Ce système radio à courte distance permet à la fois les échanges voix et données. En effet, un appareil Bluetooth peut fonctionner en mode commutation de paquets IP (sous forme de données avec un débit montant de 57,6 kbit/s et en descendant de 721 kbit/s : connexion asynchrone) ou commutation de circuit (sous forme de voix avec un débit de 64 kbit/s : connexion synchrone) et même les deux simultanément [4].

De plus la zone de réception du signal, extrêmement limitée, constitue une sécurité plus importante que celle du *Wireless Fidelity* qui nécessite la mise en place de moyens plus importants de contrôle d'accès à l'information.

L'objectif du *Bluetooth* est de permettre de transmettre des données ou de la voix entre des équipements possédant des circuits radio de faible coût, sur un rayon de l'ordre d'une dizaine de mètres à un peu moins d'une centaine de mètres et avec une faible consommation d'énergie. Ainsi, la technologie *Bluetooth* est principalement prévue pour relier entre des périphériques sans utiliser une liaison filaire.

Classes	Puissance (affaiblissement)	Portée (m)
1	100 mW (20 dbm)	~ 100
2	2.5 mW (4 dbm)	~ 15-20
3	1 mW (0 dbm)	~ 10
4	0.5 mW (-3 dbm)	~ 3

Tab 3.3 Les classes d'émetteurs du standard Bluetooth

Le *Bluetooth* permet d'obtenir des débits de l'ordre de 1Mbps correspondant à 1600 échanges par seconde en full-duplex, avec une portée d'une dizaine de mètres environ avec un émetteur de classe II, et un peu moins d'une centaine de mètres avec un émetteur de classe I. Le standard *Bluetooth* définit en effet trois classes d'émetteurs proposant des portées différentes en fonction de leurs puissances d'émission (Tableau 3. 3).

- **Le principe de fonctionnement du Bluetooth :**

Le standard *Bluetooth* est basé sur un mode de fonctionnement Maître/Esclave (**Figure 3.13**). Ainsi on appelle pico-réseau (en anglais : piconet) le réseau formé par un périphérique, ou tous les périphériques présents dans son rayon de portée.

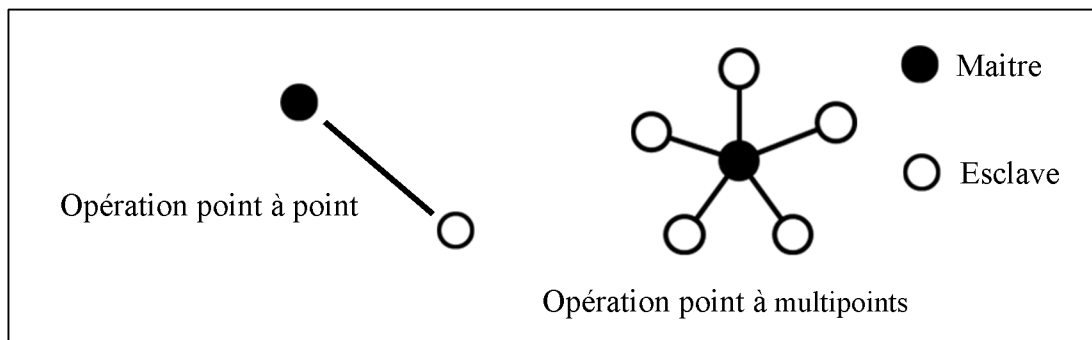


Figure 3.13 Diagramme topologique d'un réseau de type Maître/Esclave

Un réseau Bluetooth est capable de gérer aussi bien des connexions point-à-point que des connexions point-multi-points. Chaque piconet (c'est le nom donné à un réseau composé de plusieurs entités Bluetooth) possède une séquence de sauts de fréquences particulière et chaque participant d'un piconet est synchronisé sur la même séquence.

Le standard Bluetooth prévoit la possibilité de relier deux piconets entre eux afin de former un réseau élargi, appelé réseau chaîné (en anglais Scatternet) grâce à certains périphériques faisant office de pont entre les deux piconets (**Figure 3.14**).

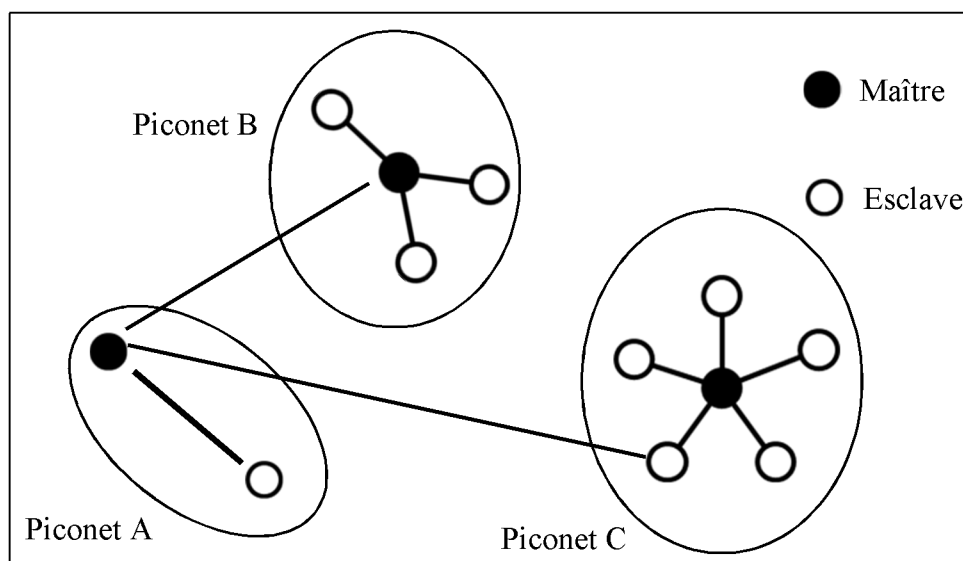


Figure 3.14 Diagramme topologique d'un réseau chaîné

Comme on le voit sur le diagramme ci-dessus, une unité peut être maître d'un piconet et esclave d'un autre piconet (c'est le cas du maître du piconet **A**, esclave du piconet **C**).

4.2. Les réseaux locaux sans fil (WLAN) :

Un réseau local sans fil *WLAN* (*Wireless Local Area Network*) est un réseau permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de mètres, il permet de relier entre eux les terminaux présents dans la zone de couverture, il existe plusieurs technologies concurrentes.

4.2.1. Le Wifi :

- **Présentation de la technologie :**

Le Wi-Fi (IEEE 802.11) est en réalité la norme initiale offrant des débits de 1 ou 2 Mbit/s. Des révisions ont été apportées à la norme originale à fin d'améliorer le débit ou de spécifier des détails de sécurité. Cette norme s'attache à définir les couches du modèles OSI pour une liaison sans fil utilisant des ondes électromagnétiques :

- **La couche physique** : proposant trois types de codage de l'information, définit la modulation des ondes radioélectriques et les caractéristiques de signalisation pour la transmission de données.
- **La couche liaison de données** : constitué de deux sous-couches, et définit l'interface entre le bus de la machine et la couche physique.

La technologie *Wi-Fi* permet de bâtir trois types principaux de réseaux, les réseaux domestiques, les ponts sans fil et les réseaux d'entreprise. Son principal attrait est donc de supprimer le câblage. En intérieur la portée radio du Wi-Fi est capable de traverser des murs en béton armé ou des étages (3 maximum) mais dans ces conditions, le débit est ralenti jusqu'à 1Mbps avant la perte du signal, et la portée peut descendre à 15mètres. La plupart du temps en intérieur on compte une portée de 25 mètres dans un environnement dense (de gros obstacles) et 60 mètres s'il n'y a que de petites cloisons à traverser.

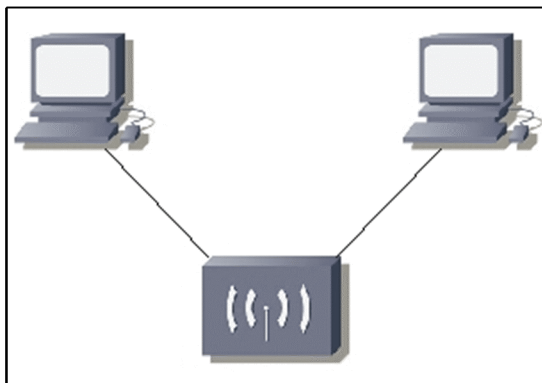
- **Les modes de fonctionnement de la norme IEEE 802.11 :**

- **Le mode infrastructure :**

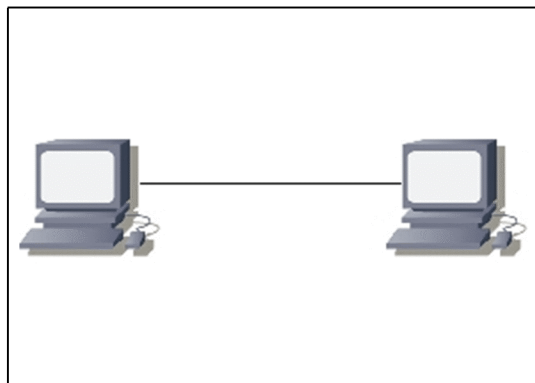
L'utilisation privilégiée de la norme **802.11**, où des stations de bases sont reliées entre elles par un réseau filaire assurant la couverture d'une certaine zone et prennent en charge les mobiles de leur voisinage (**Figure 3.15**).

- **Le mode ad-hoc :**

Qui consiste au fait simplement à autoriser les communications entre deux mobiles à portée l'un de l'autre sans interventions de stations ou d'autres mobiles extérieurs (**Figure 3.16**) [5].



**Figure 3.15 : Réseau 802.11
en mode infrastructure.**



**Figure 3.16 : Réseau 802.11
en mode ad-hoc.**

4.3. Les réseaux étendus sans fil (WWAN) :

Vous vous servez probablement quotidiennement d'un réseau étendu sans fil ou *WWAN* (*Wireless Wide Area Network*), également nommé réseau cellulaire étendu, Il s'agit des réseaux sans fil les plus répandus puisque tous les téléphones mobiles sont connectés à un réseau étendu sans fil. Les principales technologies sont les suivantes :

- **GSM** (*Global System for Mobile Communication* ou *Groupe Spécial Mobile*) : Ce standard a été prévu initialement pour les applications de transport de la parole, de transmission de données à bas débits (9600 bits/s) autour des fréquences 935MHz - 960MHz ou 890 - 915MHz. Cette norme a permis l'essor des communications mobiles d'aujourd'hui.
- **GPRS** (*General Packet Radio Service*) : Cette technologie est la première évolution de la norme GSM pour pouvoir atteindre des débits compatibles avec des applications à débits moyens de type multimédia.
- **UMTS** (*Universal Mobile Telecommunication System*) : Cette norme UMTS a pour but de développer une meilleure couverture radio et d'augmenter le nombre d'abonnés par unité de surface et promouvoir l'acheminement des services 3ème Génération (3G). L'UMTS permet un débit théorique jusqu'à 2 Mbit/s.

4.3.1. Le GSM :

- **Présentation du GSM :**

Le réseau *GSM (Global System for Mobile communications)* constitue au début du 21^{ème} siècle le standard de téléphonie mobile le plus utilisé en Europe. Il s'agit d'un standard de téléphonie dit « de seconde génération » (2G) car, contrairement à la première génération de téléphones portables, les communications fonctionnent selon un mode entièrement numérique. Baptisé « Groupe Spécial Mobile » à l'origine de sa normalisation en 1982, il est devenu une norme internationale nommée « Global System for Mobile communications » en 1991.

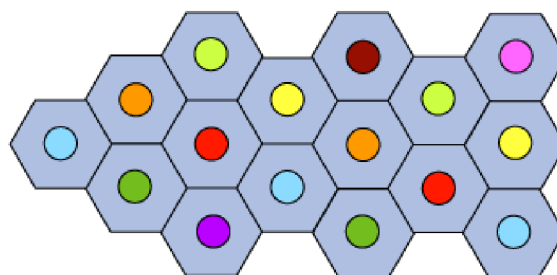
En Europe, le standard GSM utilise les bandes de fréquences 900 MHz et 1800 MHz. Aux Etats-Unis par contre, les bandes de fréquences utilisées sont les bandes 850 MHz et 1900 MHz. Ainsi, on qualifie de **tri-bande** (parfois noté *tribande*), les téléphones portables pouvant fonctionner en Europe et aux Etats-Unis et de **bi-bande** ceux fonctionnant uniquement en Europe.

La norme GSM autorise un débit maximal de 9,6 kbps, ce qui permet de transmettre la voix ainsi que des données numériques de faible volume, par exemple des messages textes (**SMS**, pour *Short Message Service*) ou des messages multimédias (**MMS**, pour *MultiMedia Message Service*).

- **Notion de réseau cellulaire**

Les réseaux de téléphonie mobile sont basés sur la notion de **cellules**, c'est-à-dire des zones circulaires se chevauchant afin de couvrir une zone géographique.

Figure 3.17 Structure cellulaire



Les réseaux cellulaires reposent sur l'utilisation d'un émetteur-récepteur central au niveau de chaque cellule, appelée « **station de base** » (en anglais *Base Transceiver Station*, notée **BTS**).

Plus le rayon d'une cellule est petit, plus la bande passante disponible est élevée. Ainsi, dans les zones urbaines fortement peuplées, des cellules d'une taille pouvant avoisiner quelques centaines de mètres seront présentes, tandis que de vastes cellules d'une trentaine de kilomètres permettront de couvrir les zones rurales.

Dans un réseau cellulaire, chaque cellule est entourée de 6 cellules voisines (c'est la raison pour laquelle on représente généralement une cellule par un hexagone). Afin d'éviter les interférences, des cellules adjacentes ne peuvent utiliser la même fréquence. En pratique, deux cellules possédant la même gamme de fréquences doivent être éloignées d'une distance représentant deux à trois fois le diamètre de la cellule.

- **Architecture du réseau GSM**

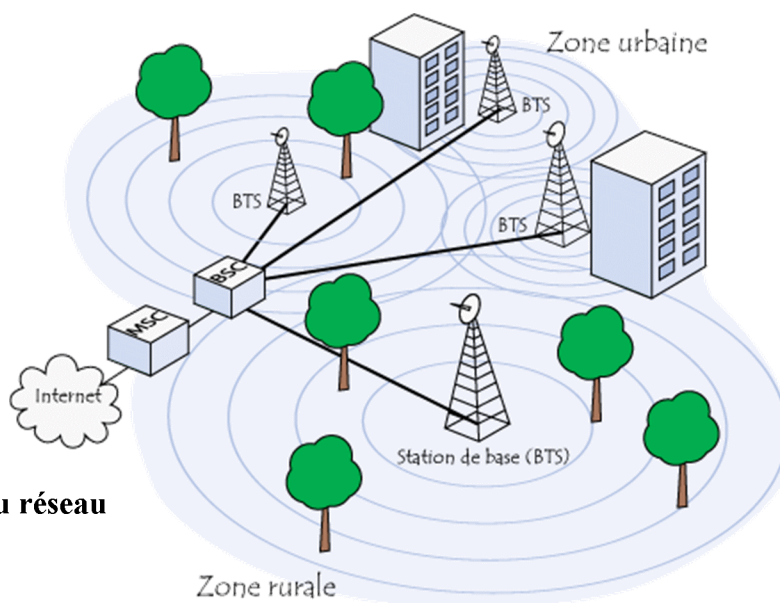


Figure 3.18 Architecture du réseau GSM

Dans un réseau GSM, le terminal de l'utilisateur est appelé **station mobile**. Une station mobile est composée d'une carte **SIM** (*Subscriber Identity Module*), permettant d'identifier l'utilisateur de façon unique et d'un terminal mobile, c'est-à-dire l'appareil de l'utilisateur (la plupart du temps un téléphone portable).

Les terminaux (appareils) sont identifiés par un numéro d'identification unique de 15 chiffres appelé **IMEI** (*International Mobile Equipment Identity*). Chaque carte SIM possède également un numéro d'identification unique (et secret) appelé **IMSI** (*International Mobile Subscriber Identity*). Ce code peut être protégé à l'aide d'une clé de 4 chiffres appelés *code PIN*.

La carte SIM permet ainsi d'identifier chaque utilisateur, indépendamment du terminal utilisé lors de la communication avec une station de base. La communication entre une station mobile et la station de base se fait par l'intermédiaire d'un lien radio, généralement appelé **interface air** (ou plus rarement *interface Um*). L'ensemble des stations de base d'un réseau cellulaire est relié à un **contrôleur de stations** (en anglais *Base Station Controller*, noté **BSC**), chargé de gérer la répartition des ressources.

L'ensemble constitué par le contrôleur de station et les stations de base connectées constituent le **sous-système radio** (en anglais **BSS** pour *Base Station Subsystem*).

Enfin, les contrôleurs de stations sont eux-mêmes reliés physiquement au **centre de commutation du service mobile** (en anglais **MSC** pour *Mobile Switching Center*), géré par l'opérateur téléphonique, qui les relie au réseau téléphonique public et à internet. Le MSC appartient à un ensemble appelé **sous-système réseau** (en anglais **NSS** pour *Network Station Subsystem*), chargé de gérer les identités des utilisateurs, leur localisation et l'établissement de la communication avec les autres abonnés.

Le **MSC** est généralement relié à des bases de données assurant des fonctions complémentaires :

- Le **registre des abonnés locaux** (noté **HLR** pour *Home Location Register*) : il s'agit d'une base de données contenant des informations (position géographique, informations administratives, etc.) sur les abonnés inscrits dans la zone du commutateur (MSC).
- Le **Registre des abonnés visiteurs** (noté **VLR** pour *Visitor Location Register*) : il s'agit d'une base de données contenant des informations sur les autres utilisateurs que les abonnés locaux. Le VLR rattaché les données sur un nouvel utilisateur à partir du HLR correspondant à sa zone d'abonnement. Les données sont conservées pendant tout le temps de sa présence dans la zone et sont supprimées lorsqu'il la quitte ou après une longue période d'inactivité (terminal éteint).
- Le **registre des terminaux** (noté **EIR** pour *Equipment Identity Register*) : il s'agit d'une base de données répertoriant les terminaux mobiles.
- Le **Centre d'authentification** (noté **AUC** pour *Authentication Center*) : il s'agit d'un élément chargé de vérifier l'identité des utilisateurs.

Le réseau cellulaire ainsi formé est prévu pour supporter la mobilité grâce à la gestion du *handover*, c'est-à-dire le passage d'une cellule à une autre.

- **Carte SIM**

Une carte SIM contient les informations suivantes :

- Numéro de téléphone de l'abonné (*MSISDN*)
- Numéro d'abonné international (*IMSI, international mobile subscriber identity*)

- Etat de la carte *SIM*
- Code de service (*opérateur*)
- Clé d'authentification
- Code *PIN* (*Personal Identification Code*)
- Code *PUK* (*Personal Unlock Code*)

A- Conception du système :

1. Exigences et spécification :

Pour le système de suivi préventif d'un château d'eau on peut définir les exigences suivantes :

- Le système doit assurer la mesure de quelques paramètres du château d'eau.
- Le système doit fonctionner en temps réel.
- Le système doit fournir les mesures pour le personnel chargé d'exploitation des châteaux d'eau sur cite (dispositif fixe), et hors site (dispositif mobile).
- Le dispositif fixe doit être capable à afficher les valeurs des paramètres et d'envoyer un sms en cas états critiques.
- Le dispositif mobile doit être capable à afficher les valeurs des paramètres et de les représenter graphiquement.
- L'appareil mobile doit fonctionner sur batterie.
- Les deux dispositifs fixe et mobile communique entre eux par liaison radiofréquence.

Pour résumer les choses nous créons un formulaire d'exigences :

Nom	Système de surveillance d'un château d'eau
Objectif	Le suivi préventif des paramètres d'un château d'eau
Entrée	Les données issues de capteurs (température, niveau et débit)
Sorties	<ul style="list-style-type: none"> - Afficheur LCD alphanumérique 16x2 (dispositif fixe). - Afficheur LCD graphique 128x64(dispositif mobile). - Module GSM.
Fonctions	<ul style="list-style-type: none"> - Mesurer, afficher en permanence la température, le niveau, le débit - Envoi un sms situations dangereuses - Envoi les données au dispositif mobile.
Performance	<ul style="list-style-type: none"> - Met à jour les valeurs toutes les 15 secondes. - Envoi un sms immédiatement en cas situation dangereuse.
Taille et poids	<ul style="list-style-type: none"> - Le dispositif mobile il doit tenir confortablement dans la main et ne dépasse pas les 500 gramme de poids.

2. Architecture :

La spécification ne dit pas comment le système fait les choses, seulement ce qu'il fait. Décrire comment le système implémente ces fonctions est le but de l'architecture. La **Figure 4.1** montre l'architecture de système sous la forme de schéma bloc qui montre les principales fonctions et le flux de données entre eux.

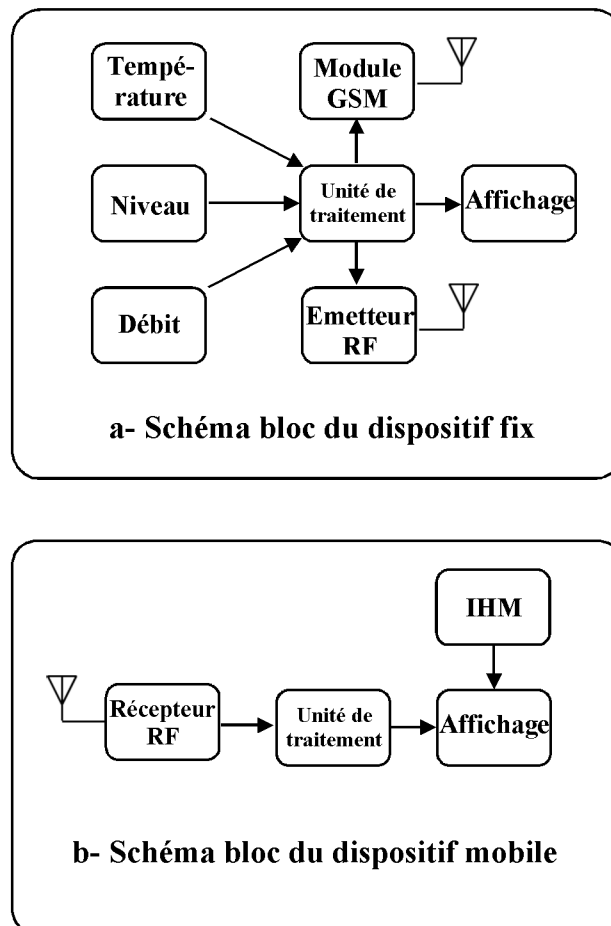


Figure 4.1 Schéma bloc du Système de surveillance d'un château d'eau

Le système de surveillance d'un château d'eau est constitué de deux dispositifs, un fixe et l'autre est mobile. Le schéma bloc du dispositif fixe nous montre que l'unité de traitement reçoit les valeurs des paramètres, elle les traite, ensuite elle les affiche localement et envoie ces données par émetteur Radiofréquence ou module Gsm.

Pour le dispositif mobile, le récepteur reçoit les données et l'unité de traitement les affiche.

Après avoir conçu cette architecture initiale on peut affiner ce diagramme en deux schéma blocs, un pour le matériel et un autre pour le logiciel. Ces deux diagrammes sont présentés à la **Figure 4.2** et **Figure 4.3**.

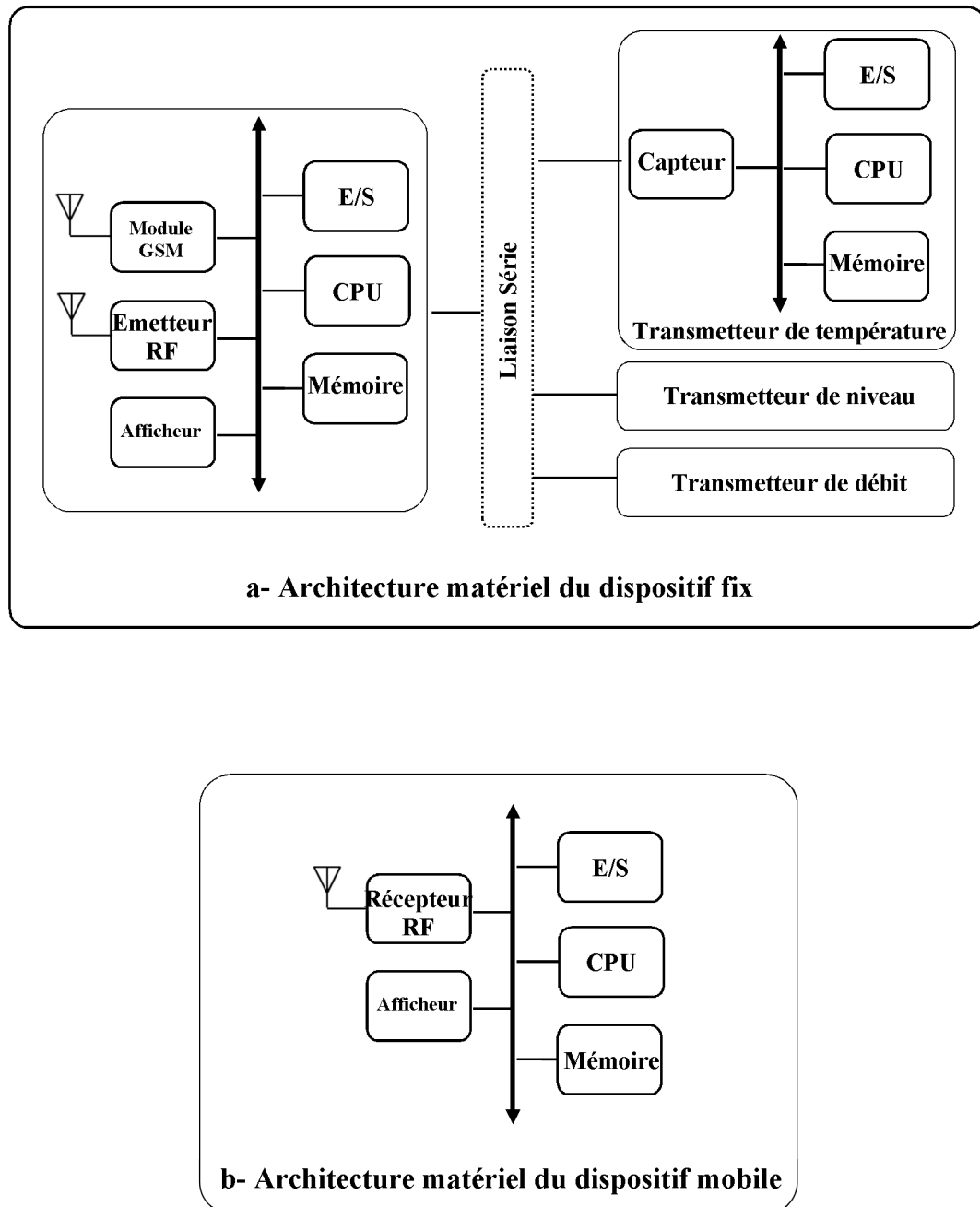


Figure 4.2 Architecture matérielle du Système de surveillance d'un château d'eau

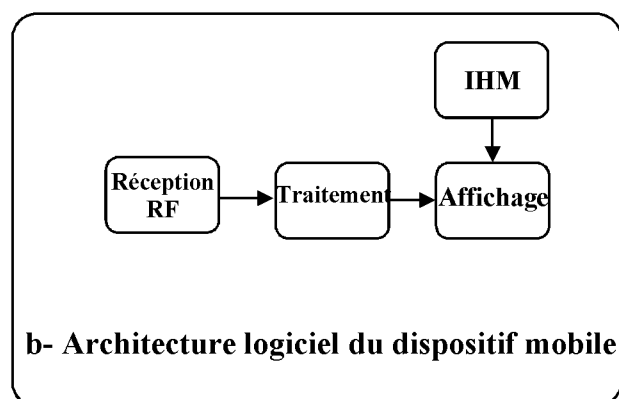
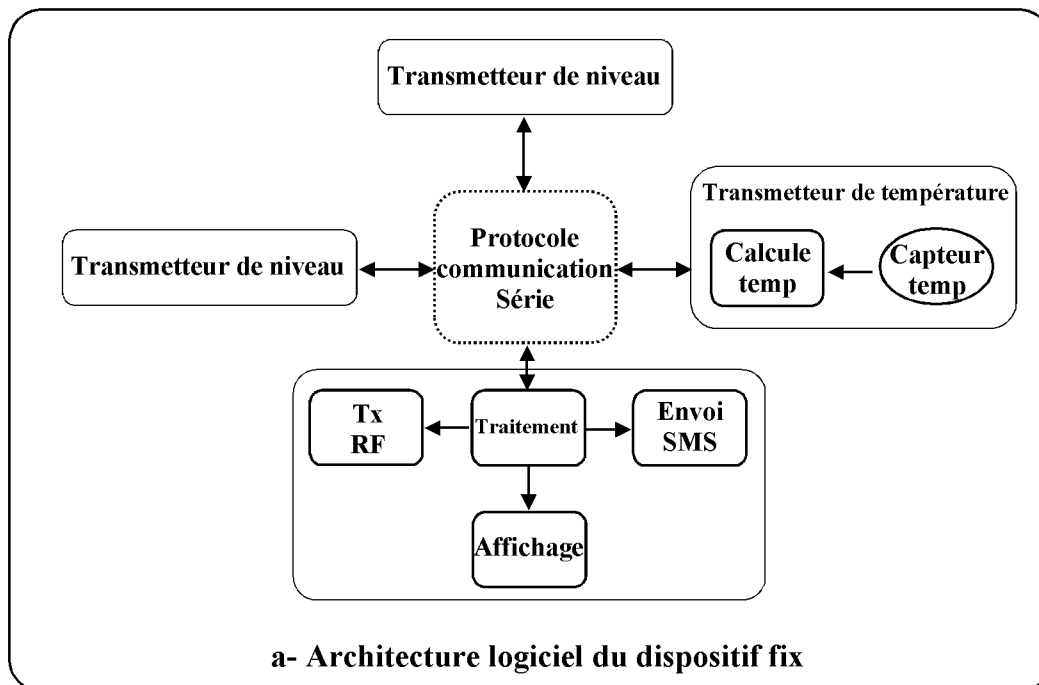


Figure 4.3 Architecture logiciel du Système de surveillance d'un château d'eau

3. Composants matériels et logiciel :

3.1. Matériel utilisé :

Plateforme Arduino :

Arduino est une plate-forme électronique open-source basée sur du matériel et des logiciels faciles à utiliser. Les cartes Arduino sont capables de lire les entrées - lumière sur un capteur, un doigt sur un bouton ou un message Twitter - et de le transformer en sortie - activer un moteur, allumer une LED, publier quelque chose en ligne. Vous pouvez indiquer à votre carte ce qu'il faut faire en envoyant une série d'instructions au microcontrôleur sur la carte. Pour ce faire, vous utilisez le langage de programmation Arduino et le logiciel Arduino (IDE).

Pourquoi Arduino ?

Il existe de nombreux autres microcontrôleurs et plates-formes de microcontrôleurs disponibles (Parallax Basic Stamp, Netmedia's BX-24, Phidgets). Tous ces outils prennent les détails désordonnés de la programmation du microcontrôleur et les envelopper dans un paquet facile à utiliser. Arduino simplifie également le processus de travail avec les microcontrôleurs, mais il offre un certain avantage intéressé par rapport à d'autres systèmes :

- *Le prix :*

En vue des performances qu'elles offrent, les cartes Arduino sont relativement peu coûteuses, ce qui est un critère majeur pour les concepteurs.

- *Matériel open source :*

Les plans des cartes Arduino sont publiés sous une licence libre, afin que les concepteurs de circuits expérimentés puissent créer leur propre version du module, l'étendre et l'améliorer.

- *Logiciel open source :*

Le logiciel Arduino est publié en tant qu'outil open source, Le langage peut être développer via les bibliothèques C ++.

Toutes ces avantages nous encourager à choisir la plate-forme Arduino, ou nous utilisons deux types de plate-forme :

- Arduino Mega2560.
- Arduino Nano.

Arduino Mega 2560 :

Il s'agit d'une carte électronique basée autour d'un microcontrôleur *ATmega2560* du fabricant *Atmel*, (**Figure 4.4**) dont le prix est relativement bas pour l'étendue possible des applications. Voilà à quoi ressemble la carte que nous allons utiliser :

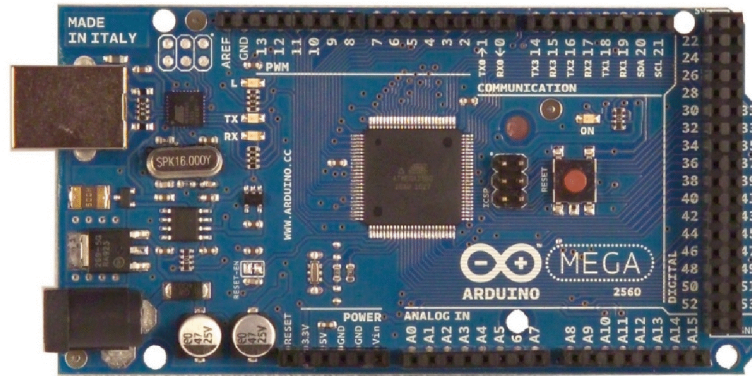


Figure 4.4 Carte Arduino "Mega 2560"

Les caractéristiques techniques de cette carte :

Microcontrôleur	ATmega2560
Tension de fonctionnement	5 V
Tension d'alimentation (recommandée)	7- 12 V
Tension d'alimentation (limites)	6 - 20V
Nombre d'E/S	54 (dont 14 pouvant générer des signaux PWM)
Nb ports "Analogique/Numérique"	16
Courant max. par E/S	40 mA
Courant pour broches 3.3 V	50 mA
Mémoire Flash	256 KB (ATmega328) dont 8 KB utilisé par le bootloader
SRAM	8 KB (ATmega328)
EEPROM	4 KB (ATmega328)
Vitesse horloge	16 MHz

Tableau 4.2 Caractéristiques techniques d'Arduino "Mega 2560 "

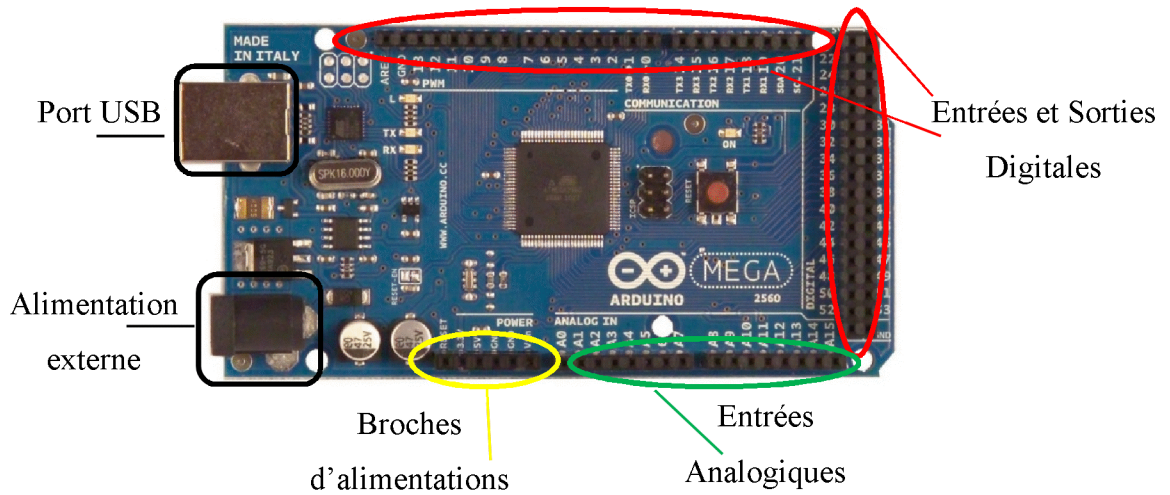


Figure 4-5 Brochage de la Carte Arduino "Mega 2560"

Arduino Nano :

L'Arduino Nano est une petite complète carte basée sur l'ATmega328P du fabricant *Atmel*, Il a plus ou moins la même fonctionnalité de l'Arduino méga avec un facteur de taille plus petit. **Figure 4.6**

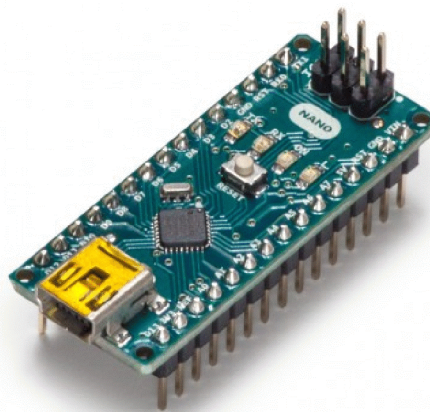


Figure 4.6 Carte Arduino "Nano"

Les caractéristiques techniques de cette carte le tableau 4.2 :

Microcontrôleur	ATmega328
Tension de fonctionnement	5 V
Tension d'alimentation (recommandée)	7- 12 V
Nombre d'E/S	22 (6 PWM)
Nb ports "Analogique/Numérique"	8
Courant max. par E/S	40 mA
Mémoire Flash	32 KB (ATmega328)
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Vitesse horloge	16 MHz

Tableau 4.3 Caractéristiques techniques d'Arduino "Nano "

Les capteurs :

Capteur de température :

Aujourd'hui, il existe diverses références de capteurs de température numériques, de grande précision et à sortie linéaire, et faciles à l'emploi, par exemple le LM35. Celle que nous utilisons dans notre projet est le capteur **DS18B20**.

Le capteur DS18B20 existe dans le commerce en deux versions **Figure 4.7** :

- Boitier TO-92 (format transistor) pour des utilisations standards en intérieur.
- Format "sonde étanche" pour des applications en milieu humide / extérieur.



DS18B20 en boitier TO-92



DS18B20 en format Sonde étanche

Figure 4.7 Les formats du capteur de température DS18B20

Le DS18B20 est un capteur de température numérique intégrant tout le nécessaire requis pour faire la mesure :

- Capteur analogique.
- Convertisseur analogique / numérique.
- Électronique de communication et alimentation.

Il communique via un bus 1-Wire et possède une résolution numérique programmable de 12 bits avec une plage de mesure de -55°C à $+125^{\circ}\text{C}$. La précision analogique du capteur est de $0,5^{\circ}\text{C}$ entre -10°C et $+85^{\circ}\text{C}$, ce qui rend ce capteur très intéressant pour une utilisation normale.

Le bus 1-Wire permet de relier plusieurs capteurs au microcontrôleur sur un seul bus sans aucun risque de conflit puisque, chaque capteur dispose d'une adresse unique gravée lors de la fabrication. Un bus 1-Wire est composé classiquement des trois fils : Un fil de masse, un fil d'alimentation et un fil de données.

Capteur de niveau :

La mesure de niveau continue détermine le niveau de produit en permanence. Il s'agit en fait d'une mesure de longueur.

Les ultrasons constituent une solution éprouvée et économique pour la mesure de niveau. Ce principe de mesure se caractérise par une planification et un montage simple, une mise en service rapide et sûre, une durée de vie longue et une maintenance réduite. Les ondes ultrasoniques sont émises par un émetteur, réfléchies par la surface du produit et détectées par un récepteur. A partir du temps de parcours de l'impulsion, on détermine la distance entre l'émetteur et la surface, la vitesse de propagation étant connue. Il est ensuite facile de calculer le niveau en tenant compte de la hauteur du réservoir.

Dans notre projet on utilise le capteur ultrasons **HC-SR04 (Figure 4.8)** pour mesure le niveau.



Figure 4.8 Le capteur ultrasons HC-SR04

Principe de mesure de niveau :

Un microcontrôleur avec le capteur HC-SR04 forme un transmetteur de niveau qui utilise des impulsions ultrasons réfléchies par la surface du produit par la différence de densité entre l'air et le produit. La durée entre l'émission et la réception de l'impulsion est mesurée et analysée par le microcontrôleur et donne une valeur directe pour la distance entre la membrane du capteur et la surface du produit.

Le déclenchement de mesure se fait par l'envoi sur l'entrée Trigger d'une impulsion High de durée 10 µs. Le capteur émet alors une série de 8 impulsions ultrasoniques à 40 kHz. Puis il attend le signal réfléchi, lorsque celui-ci est détecté, une impulsion de durée proportionnelle à la distance mesurée est envoyée sur la sortie "Echo".

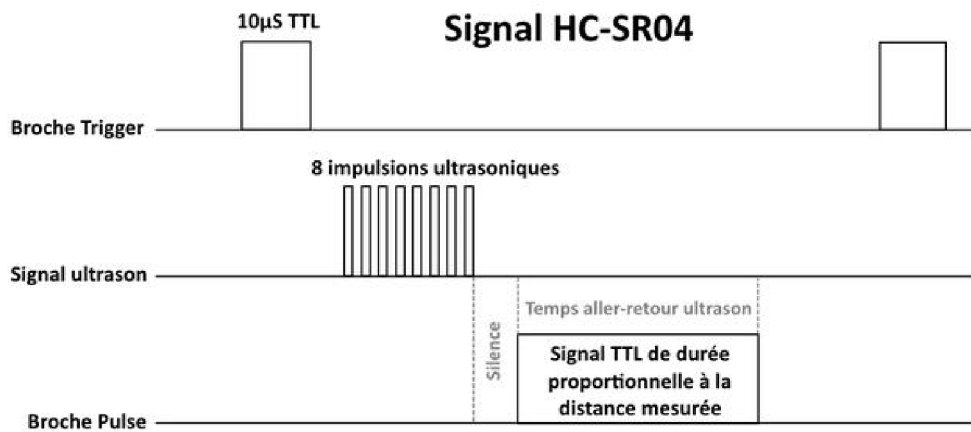


Figure 4.10 Illustration du signal TRIGGER et ECHO

La distance de l'objet détecté est égale au temps de l'impulsion de sortie divisée par deux (le signal fait un aller-retour) et multipliée par la vitesse du son dans l'air (340m/s).

Distance parcourue par le son = (vitesse du son (340 m/s) * temps aller-retour du son) / 2

$$\text{Distance} = (340 \text{ m/s} * T \text{ en sec}) / 2$$

$$\text{Distance} = (34000 \text{ cm} / 1000000\mu\text{s} * T \text{ en } \mu\text{s}) / 2$$

$$\text{Distance} = 17 / 1000 * T$$

$$T = \text{Distance} * 1000 / 17 = D * 58,82 \text{ en } \mu\text{s}$$

Distance en cm = temps propagation en µs / 58

Capteur de débit :

Pour le calcul du débit nous utilisons un capteur de débit **YFS 201** à effet hall **Figure 4.10**.



Figure 4.9 Le capteur de débit YFS 201

Ce capteur de débit d'eau est composé d'un corps en plastique, d'un rotor actionné par le passage de l'eau et d'un capteur à effet hall. La vitesse de rotation du rotor varie en fonction du débit d'eau et est mesurée par le capteur à effet hall qui sort le signal correspond sous la forme d'un train d'impulsion dont il suffit de mesurer la fréquence. On déduit le débit de ces impulsions, en sachant que chaque impulsion correspond à 2,25 millilitres.

Spécifications :

Tension de fonctionnement	5V - 24V
Courant max	15 mA
Poids	43 g
Plage de débit	1 – 30 L/min
Pression de fonctionnement	< 1.75 Mpa
Température de fonctionnement	0°C~80°C
Température du liquide	< 120°C

Tableau 4.3 Spécification Capteur ultrason YFS 201

Les modules de communication :

Le système de surveillance d'un château d'eau est un système communicant, il communique via deux modules de communication :

- Module GSM
- Module Radiofréquence 433 MHz.

Le module GSM :

Comme l'envoi d'un SMS est une exigence, l'utilisation d'un module GSM est indispensable, pour cette raison on a choisi le module SIM900a mini **Figure 4.12**

Le SIM900A est un modem à double bande GSM / GPRS de SIMCOM. Cela fonctionne sur les fréquences 900/1800 MHz.



Figure 4.11 Module GSM SIM900A mini

Tension de fonctionnement : 5 V
Dual-Band 900/ 1800 MHz
Control via AT commands
Dimensions : 24*24*3 mm
Poids : 3.4 g
Opération température : -40°C to +85 °

Tableau 4.4 Spécification techniques 'SIM900a'

Le module RF 433MHz

Le transmetteur-récepteur RF 433MHz doit assurer la communication entre le dispositif fixe et l'autre mobile.

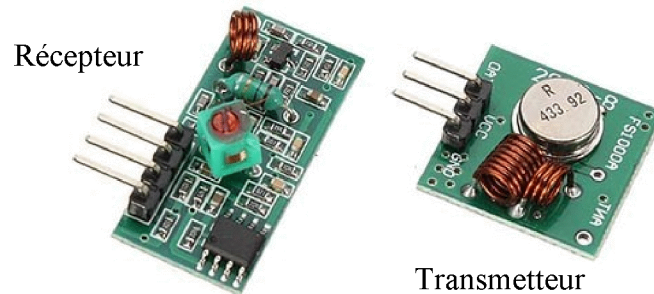


Figure 4.12 Le module RF433MHz

Ce module a une spécification pour :

Transmetteur :

Tension de fonctionnement	3V - 12V
Courant de fonctionnement	9mA - 40mA
Modulation	ASK
Fréquence	433 MHz
Vitesse de transmission de données	10Kbps
Couverture	3 m – 100 m

Tableau 4.5 Spécification Transmetteur RF 433MHz

Récepteur :

Tension de fonctionnement	3V - 12V
Courant de fonctionnement	Max 5.5 mA
Modulation	ASK
Fréquence	433 MHz
Vitesse de transmission de données	10Kbps

Tableau 4.5 Spécification Récepteur RF 433MHz

4. Intégration du système :

4.1. Transmetteur de niveau :

Le transmetteur de niveau assure la mesure de niveau d'eau dans le château d'eau, il est constitué d'un Arduino nano et un capteur ultrasons HC-SR04. Schéma électrique est représenté dans la **Figure 4.13** :

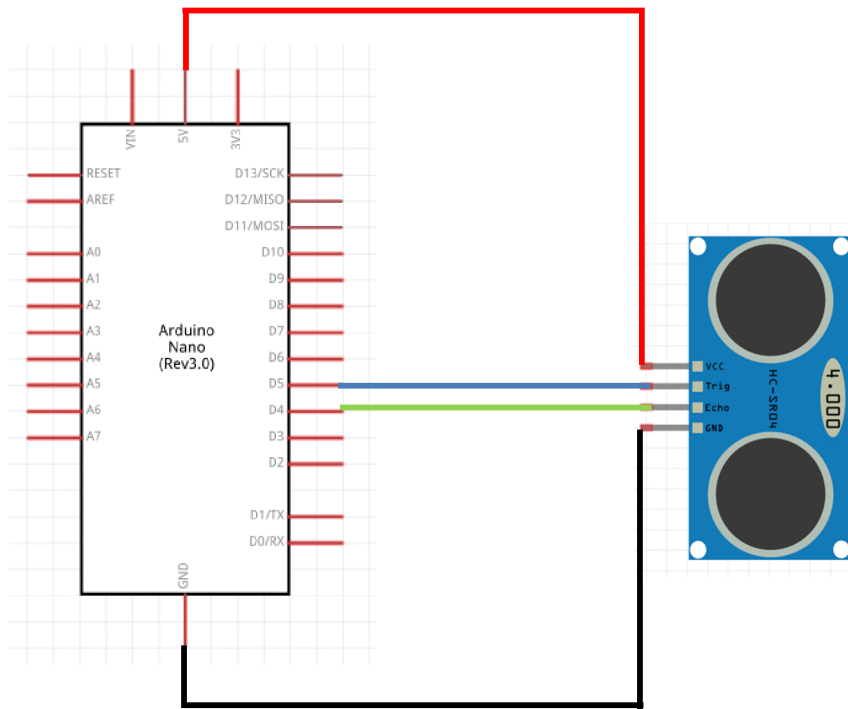


Figure 4.13 Schéma du transmetteur de niveau

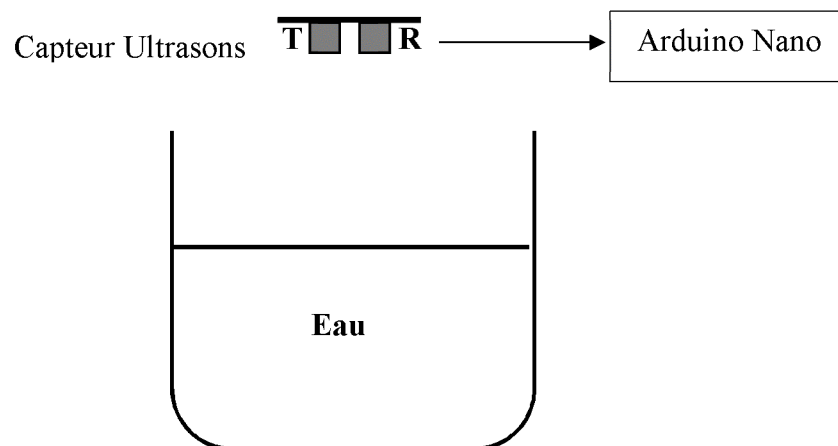


Figure 4.14 Configuration du transmetteur de niveau d'eau

Et le programme utilisé est le suivant :

```
/* Level Transmitter */
/* 22 May 2018 */
/* Arduino nano + Module Ultra son HC-SR04 */
/* LAKEHAL Radouane */

const int trigger=3;
const int echo=4;

float distance;
float distance1;
float water_level;
float level;
char send_level_string [10];

void setup() {

  Serial.begin (9600);
  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);
}
void loop () {

  distance1 = getDistance(trigger,echo);

  water_level = 0.250 - distance1;

  level = (water_level / 0.20) * 100;

  Serial.println(level);

  dtostrf(level, 0, 2, send_level_string);      // convert to char string - floatVariable,
string_length_incl_decimalpoint, number_decimals, array_to_store

  Serial.print(send_level_string);              // send level (string) to Serial2 of Arduino Mega

  delay (200);

  memset(send_level_string, '\0', 10);          // empty sendstring_array after serial print
}      // End loop

/***/ Fonction Distance ***/

float getDistance (int trigger, int echo)
{
  digitalWrite(trigger,LOW);
  delayMicroseconds(5);
  digitalWrite(trigger,HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger,LOW);
  distance=pulseIn(echo,HIGH);
  distance=distance*0.0001657;
  return distance;
}
```

4.2. Transmetteur de température :

Le transmetteur de température assure la mesure de température d'eau dans le château d'eau, il est constitué d'un Arduino nano et une Sonde étanche DS18B20. Schéma électrique est représenté dans la **Figure 4.15** :

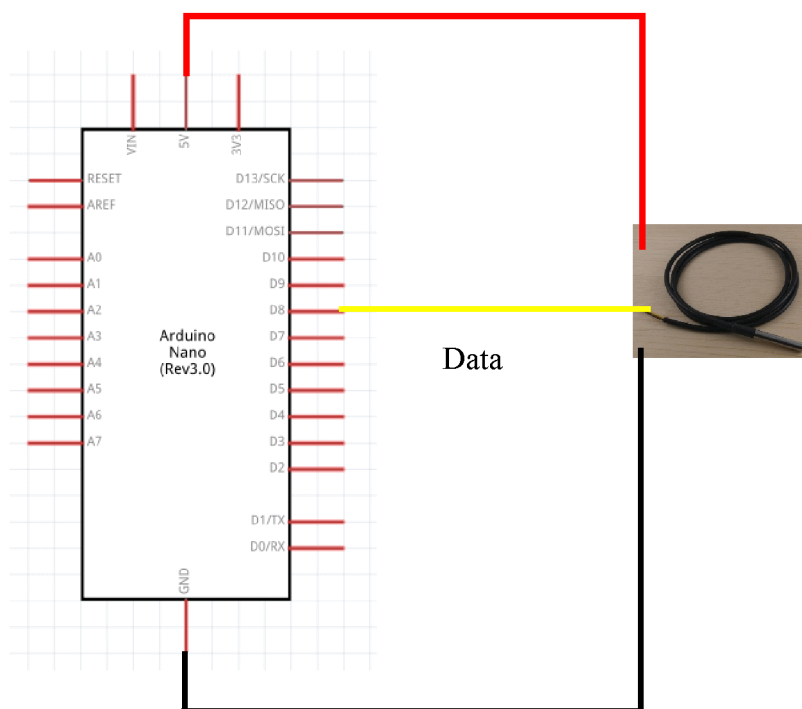


Figure 4.15 Schéma du transmetteur de température

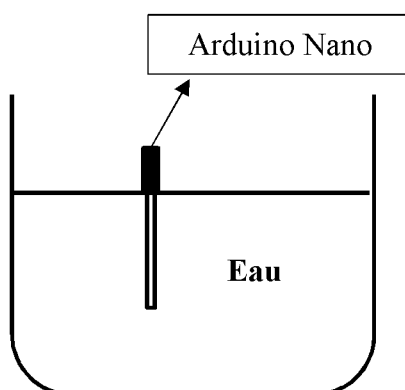


Figure 4.16 Configuration du transmetteur de température

Et le programme utilisé est le suivant :

```

/* Temperature Transmitter */
/* 22 May 2018 */
/* Arduino nano + DS18B20 */
/* LAKEHAL Radouane */

#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 8
#define DS18B20 0x28
#define NB_SAMPLE 12

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);

float temperature;
char tempstring_arr [10];
char sendstring [10];

void setup() {
  Serial.begin(9600);
  sensors.begin();
}

void loop() {
  static float tab[NB_SAMPLE] = {0};
  static int tab_index = 0;
  float somme = 0;
  float Tm = 0;

  sensors.requestTemperatures();
  temperature = sensors.getTempCByIndex(0);    // Calculate temperature
  dtostrf(temperature, 0, 2, tempstring_arr);  // convert to char string -
floatVariable, string_length_incl_decimalpoint, number_decimals, array_to_store
  strcat (sendstring, tempstring_arr);        // add tempstring_array to
sendstring_array
  Serial.print(sendstring);                   // send temperature (string) to Serial1 of
Arduino Mega
  delay (200);
  memset(sendstring, '\0', 10);               // empty sendstring_array after serial print
  delay (4800);
  memset(sendstring, '\0', 10);               // empty sendstring_array after serial print
}

```

4.3. Transmetteur de débit :

Le transmetteur de débit assure la mesure de débit d'eau dans le château d'eau, il est constitué d'un Arduino nano et un capteur de débit YFS 201. Schéma électrique est représenté dans la **Figure 4.17** :

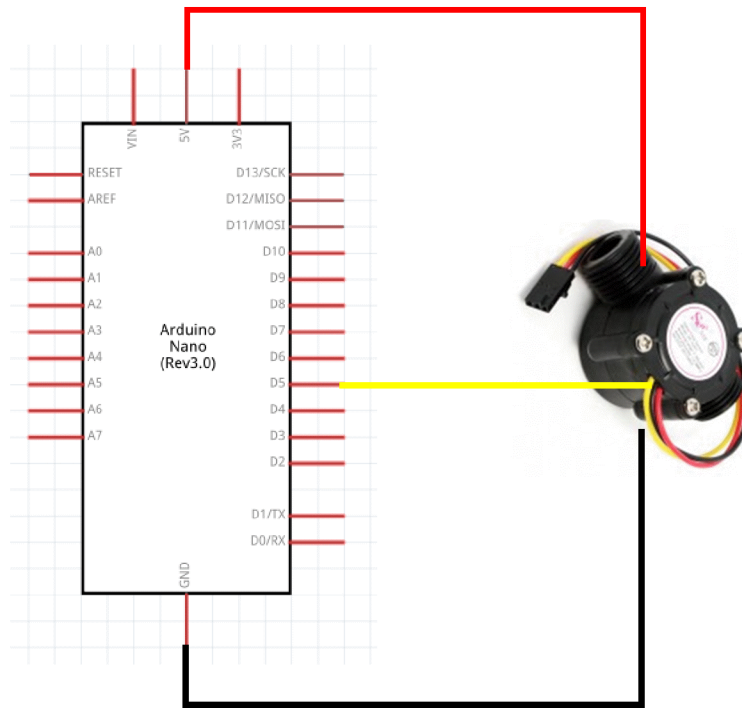


Figure 4.17 Schéma du transmetteur de Débit

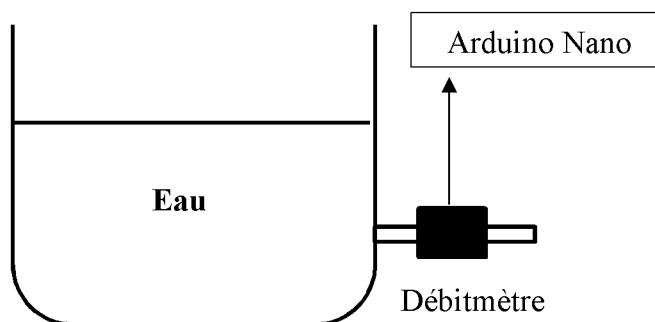


Figure 4.18 Configuration de transmetteur de débit d'eau

Le programme utilisé est le suivant :

```
/* Flow Transmitter */
/* 17 May 2018 */
/* Arduino nano + Flow Sensor YF-S201 */
/* LAKEHAL Radouane */

const int sensor = 2;
float liters_Hour;
float liters_Minute;
volatile int pulses = 0
unsigned long tempAnterior = 0;
unsigned long pulses_Acumu = 0;
char liters_arr [10];
char liters_Hour_arr [10];
char liters_Minute_arr [10];
char sendstring [10];
void flow()
{
  pulses++;
}
void setup()
{
  Serial.begin(9600);
  pinMode(sensor, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(sensor), flow, RISING);
  tempAnterior = millis();
}
void loop ()
{
  if(millis() - tempAnterior > 1000)
  {
    tempAnterior = millis();
    pulses_Acumu += pulses;
    liters_Hour = (pulses * 60 / 7.5);
    liters_Minute = (pulses / 7.5);
    liters = pulses_Acumu*1.0/450;
    pulses = 0;
    //Send data via Serial port
    dtostrf(liters, 0, 2, liters_arr);
    dtostrf(liters_Hour, 0, 2, liters_Hour_arr);
    dtostrf(liters_Minute, 0, 2, liters_Minute_arr);
    strcat (sendstring, liters_arr);
    strcat (sendstring, liters_Minute_arr);
    Serial.println(sendstring);
    memset(sendstring, '\0', 10);
  }
}
```

4.4. Transmetteur - Récepteur RF 433MHz :

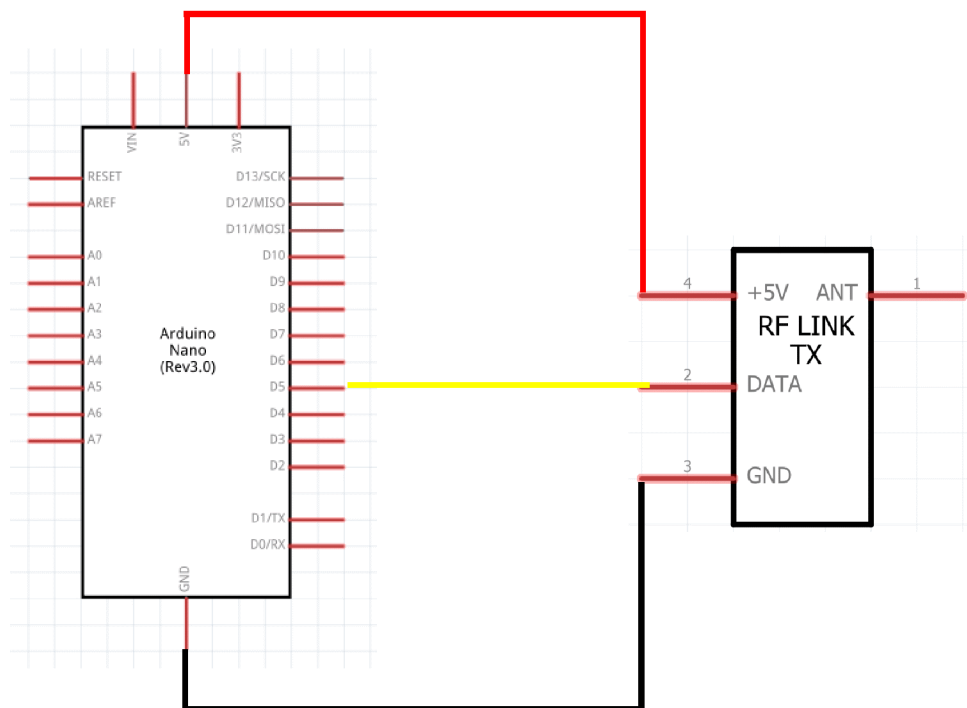


Figure 4.19 Schéma du transmetteur RF 433MHz

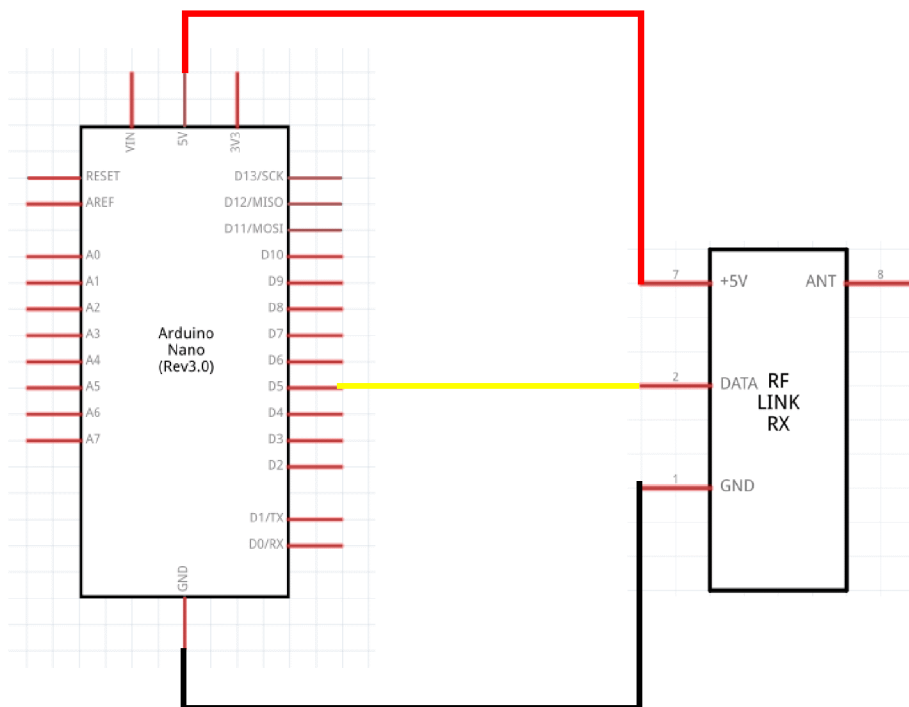


Figure 4.20 Schéma du récepteur RF 433MHz

La **Figure 4.21** représente la configuration générale du système de surveillance d'un château d'eau d'eau.

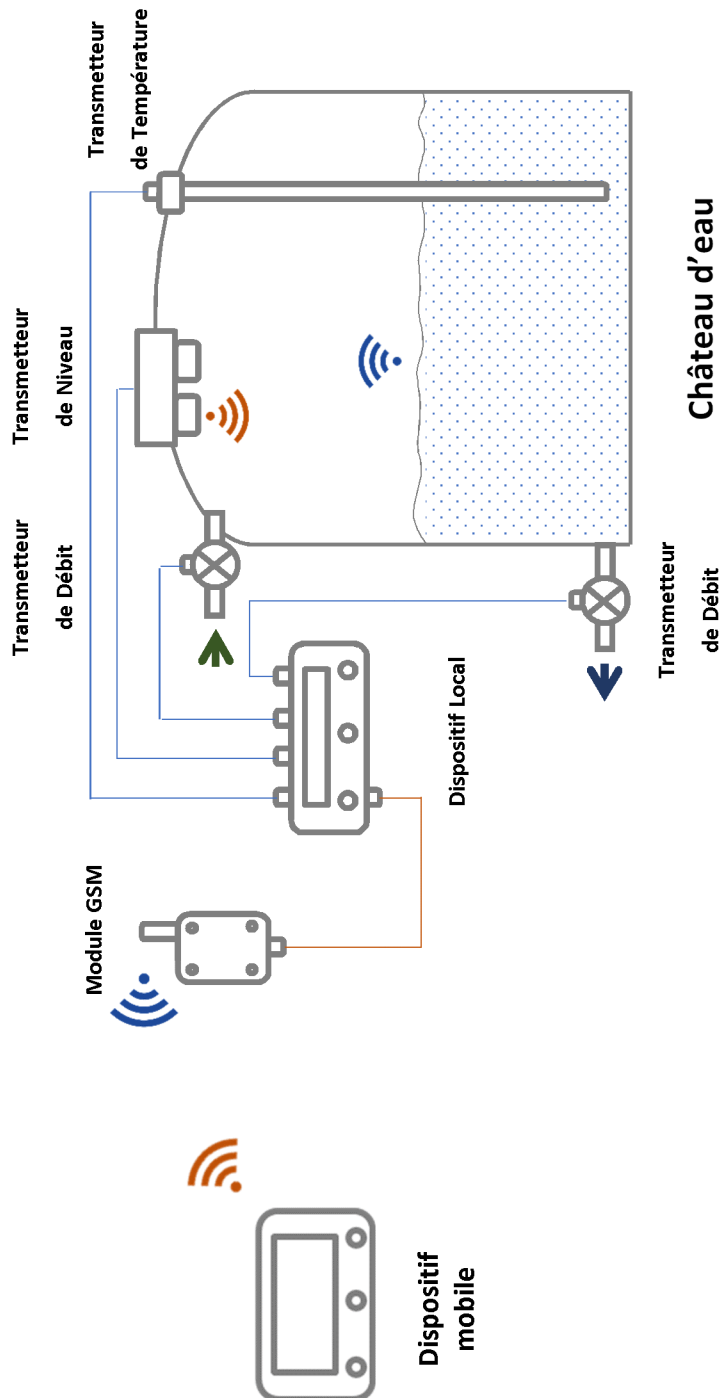


Figure 4.21- Configuration générale du Système de surveillance d'un château d'eau

B- Mise en œuvre et résultats :

Dans cette partie nous décrivons le système réalisé, Nous commençons par la première maquette qui représente le dispositif local lié au château d'eau (**Figure 4.22**) :

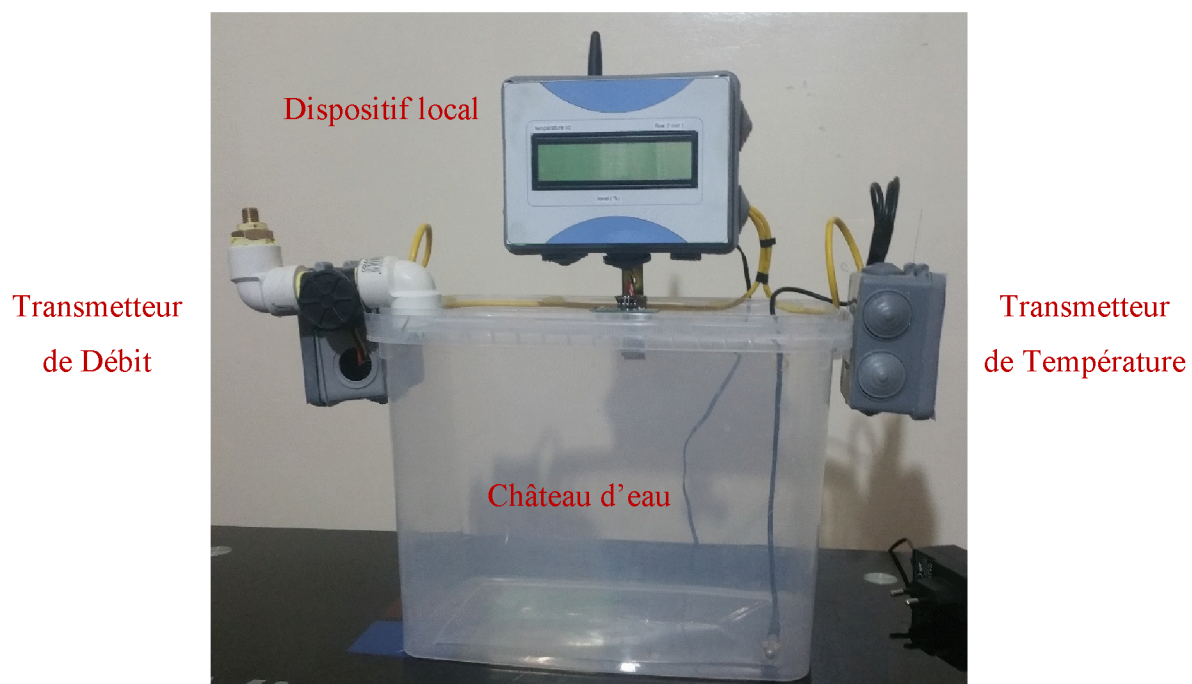


Figure 4.22 Maquette du système réalisé

Etude des cas :

I- Niveau :

1^{ère} situation : Château d'eau vide

- Affichage local : 'Empty' (Figure 4.24)
- Affichage mobile : 'Empty', '0.00 L', '< 5%' (Figure 4.25)
- SMS : 'Attention !!! Château d'eau VIDE' (Figure 4.26)

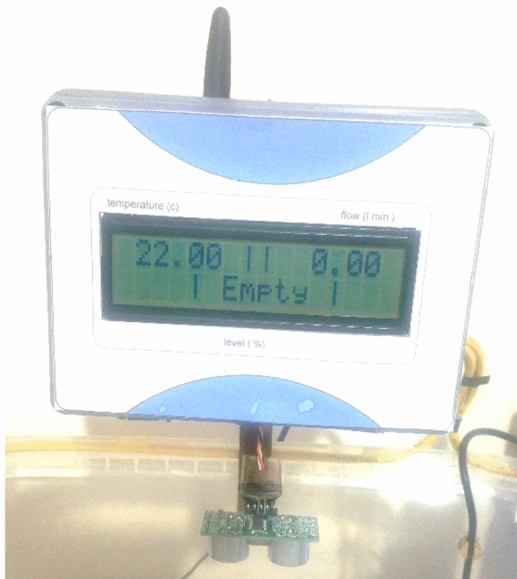


Figure 4.23 Affichage local de niveau Château d'eau vide



Figure 4.24 Affichage mobile de niveau Château d'eau vide

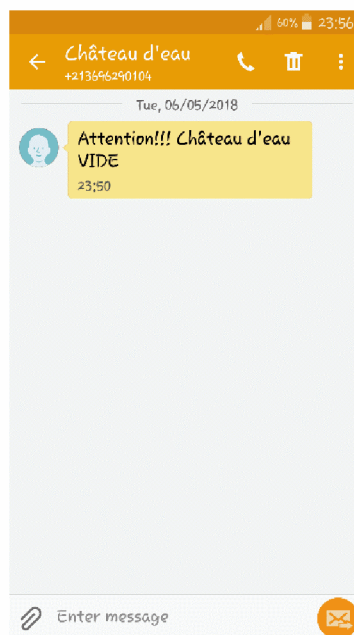


Figure 4.25 SMS Château d'eau vide

2^{ème} Situation : Château d'eau semi-plein

- Affichage local : '50 %' (Figure 4.27)
- Affichage mobile : '50 %', '2,5 L', 'Affichage graphique' (Figure 4.28)
- SMS : Pas d'envoi de SMS.



Figure 4.26 Affichage local de niveau Château d'eau demi plein

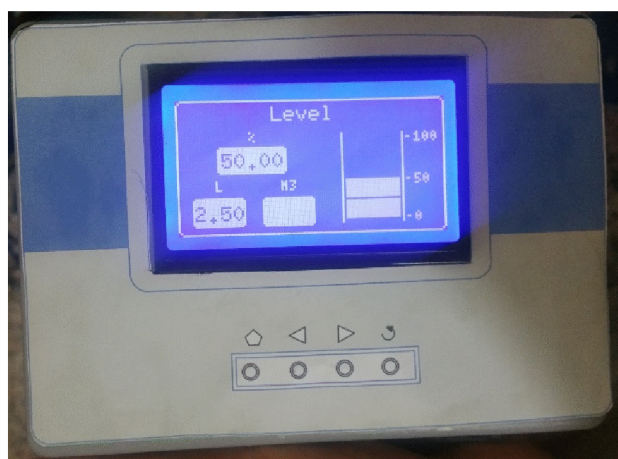


Figure 4.27 Affichage mobile de niveau Château d'eau demi plein

3^{ème} Situation : Château d'eau plein

- Affichage local : '100 %' (Figure 4.29)
- Affichage mobile : 'FULL', '5L', 'Affichage graphique' (Figure 4.30)
- SMS : Pas d'envoi de SMS.



Figure 4.28 Affichage local de niveau Château d'eau plein

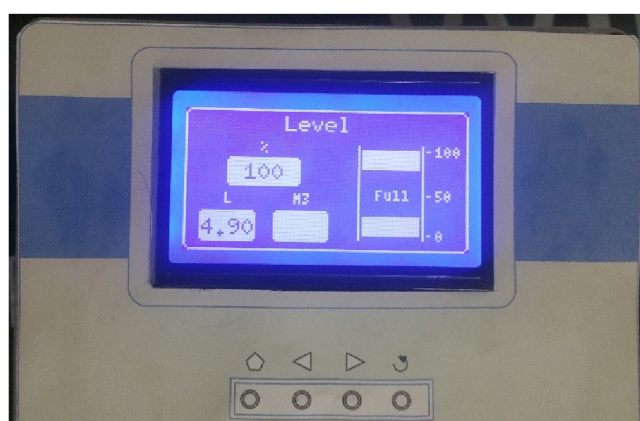


Figure 4.29 Affichage mobile de niveau Château d'eau plein

4^{ème} Situation : Château d'eau trop plein (débordement)

- Affichage local : ' O-FULL' (Figure 4.31)
- Affichage mobile : '> 100%', '> 3L', 'Affichage graphique' (Figure 4.32)
- SMS : ' Attention !!! Château d'eau TROP PLEIN ' (Figure 4.33)



Figure 4.30 Affichage local de niveau Château d'eau trop plein

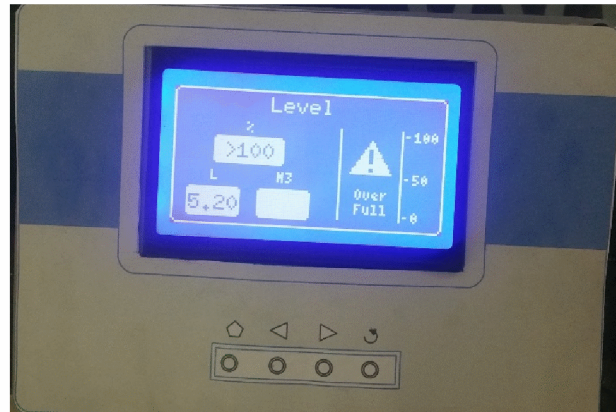


Figure 4.31 Affichage mobile de niveau Château d'eau trop plein

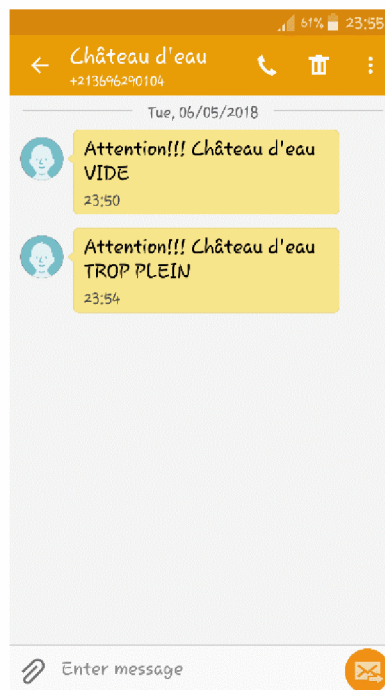


Figure 4.32 SMS Château d'eau trop plein

II- Débit :

1ère situation : Pas de remplissage

- Affichage local : " 0 L/min" (Figure 4.34)
- Affichage mobile : "0 L/min, "Affichage graphique" (Figure 4.35)
- SMS : Pas d'envoi de SMS



Figure 4.33 Affichage local du débit Pas de remplissage

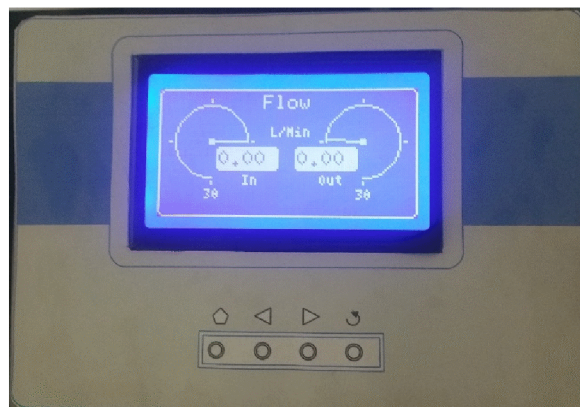


Figure 4.34 Affichage mobile du débit Pas de remplissage

2ème situation : Remplissage

- Affichage local : " x L/min" (Figure 4.36)
- Affichage mobile : "x L/min, "Affichage graphique" (Figure 4.37)
- SMS : Pas d'envoi de SMS

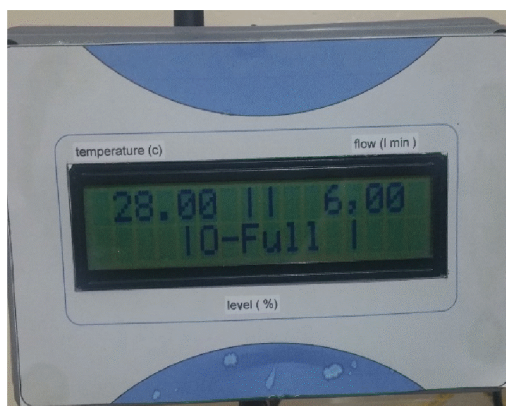


Figure 4.35 Affichage local du débit remplissage

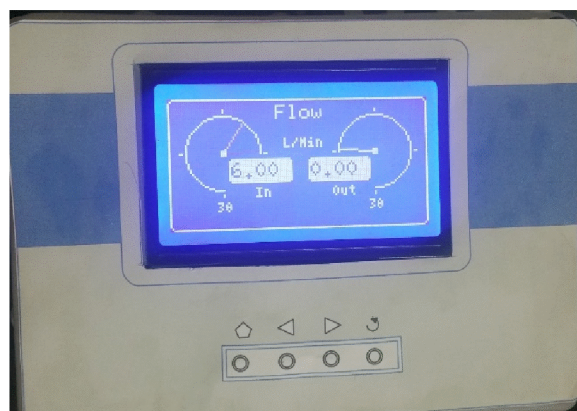


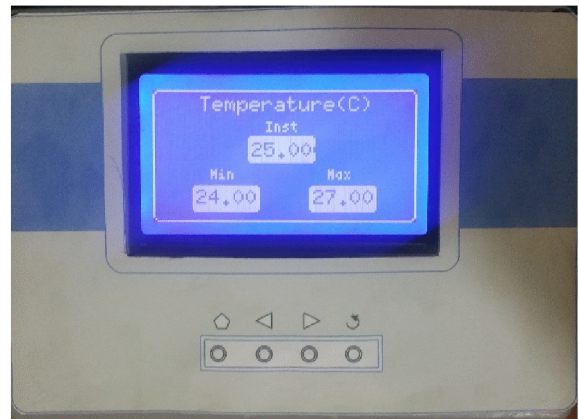
Figure 4.36 Affichage mobile du débit remplissage

III- Température :

- Affichage local : ‘‘ 25 °C’’ (Figure 4.38)
- Affichage mobile : ‘‘T inst 25°C, ‘‘T min 23°C’’, ‘‘T max 26°C’’ (Figure 4.39)
- SMS : Pas d’envoi de SMS



**Figure 4.37 Affichage local
température**



**Figure 4.38 Affichage mobile
température**

Ce mémoire présente un système embarqué communicant mobile de surveillance de l'état d'un château d'eau. Il constitue l'un des systèmes les plus prometteurs qui intègre la technologie de l'électronique numérique et les réseaux sans fils.

Le système réalisé est composé de deux sous-systèmes. Le premier est un sous-système local contenant les capteurs numériques et la cellule locale de traitement communicante par GSM et par radiofréquence. Le deuxième est un sous-système distant pour le suivi à distance des paramètres du château d'eau. Il sera à la disposition du personnel chargé de l'exploitation de château d'eau. Un tel système, qui combine efficacité et efficacité, offre des avantages certains, citons particulièrement :

- Un suivi en temps réel, et même à distance.
- Amélioration de l'accessibilité aux châteaux d'eau à l'aide de réseaux de communication sans fils.
- Diminution des coûts de suivi d'état des châteaux d'eau.

Nous formulons quelques perspectives à notre travail qui pourraient se développer dans plusieurs directions :

- Possibilité d'intégrer plusieurs d'autres paramètres de suivi (qualité chimique et microbiologique essentiellement).
- Possibilité d'agir à distance dans le cadre de l'urgence surtout, mais également dans certaines autres considérations (Fermeture / ouverture à distance, transfert de l'eau vers d'autres sites, etc.)
- Possibilité de suivre plusieurs châteaux d'eau en même temps par la même équipe.
- Possibilité de le relier aux tablettes et phablettes.

Bibliographie

- [1] : Francine KRIEF, « *Les systèmes embarqués communicants : mobilité, sécurité, autonomie* », Hermes Science Publications, Série réseaux et télécoms, ISBN : 978- 2746218734, Lavoisier librairie (Ed.), 2008.
- [2] : Rob Toulson Tim Wilmshurst. « *Fast and Effective Embedded Systems Design* », Elsevier 2012, 366 P
- [3] : KADIONIK P. « *Les systèmes embarqués : une introduction* », *Linux Magazine*, vol. 24, 2006.
- [4] : ZURAWSKI R., « *Embedded Systems Handbook* », CRC Press, New York, 2006.
- [5] : Wilmshurst, T, « *An Introduction to the Design of Small-Scale Embedded Systems* ». ISBN 0-333-92994-2, Palgrave, (2001)
- [6] : Wayne WOLFE, « *Computers as components, Principles of Embedded System Design* », Morgan Kaufmann, (2001).
- [7] : VAHID Franck et GIVARGIS Tony, « *Embedded System Design : a Unified Hardware-Software Definition* », John Wileys and Sons, (2002).
- [8] : Manuel JIMENEZ, Rogelio PALOMERA, Isidoro COUVERTIER. « *Introduction to Embedded Systems Using Microcontrollers and the MSP430* », Springer, (2014).
- [9] : Jack GANSSELE, « *The Art of Designing Embedded Systems* », 2nd Edition, (2008).
- [10] : Peter MARWEDEL, « *Embedded System Design* », Springer, 2006
- [11] : Daniel ETIEMBLE., « *Introduction aux systèmes embarqués* », Techniques de l'Ingénieur, (2010).
- [12] : Tammy NOERGAARD., « *Embedded Systems Architecture* », Elsevier Inc., (2005).
- [13] : Rahul DUBEY., « *Introduction to Embedded System Design Using Field Programmable Gate Arrays* », (2009).
- [14] : J. Valvano. « *Introduction to Embedded Systems, Interfacing to the FREESCALE 9S12* ». Cengage Learning, (2009).
- [15] : Lovic GAUTHIER., « *Génération de système d'exploitation pour le ciblage de logiciel multitâche sur des architectures* », thèse de doctorat, INPGrenoble, (2001).
- [16] : Stéphane LECOMTE., « *Méthodologie de conception de haut niveau orientée modèles pour les équipements de radio logicielle* », thèse de doctorat, Université De Rennes 1, (2011).
- [17] : Meikang QIU, Jiayin Li., « *Real-Time Embedded Systems Optimization, Synthesis, and Networking* », (2011).
- [18] : Rabaey J.M., Pedram M., « *Low Power design Methodologies* », Edition Kluwer Academic Publishers (1996).

- [19] : Raphaël SCHOENIG., « Définition d'une méthodologie de conception des systèmes mécatroniques surs de fonctionnement », Thèse de doctorat, l'Institut National Polytechnique de Lorraine, (2004)
- [20] : Fabrice LEMAINQUE, « *Tout sur les réseaux sans fils* », ISBN : 9782100525690, édition Dunod, (2009).
- [21] : Dominique DHORTAUT, « *Étude du standard IEEE 802.11 dans le cadre des réseaux ad-hoc : de la simulation à l'expérimentation.* », Thèse de Doctorat, Institut national de sciences appliquées de Lyon, (2003).
- [22] : Nadeen SALAMEH., « *Conception d'un système d'alerte embarqué basé sur les communications entre véhicules* », Thèse de doctorat, L'INSA DE ROUEN, (2011).
- [23] : R. Dömer, D. Gajski, J. Zhu., « *Specification and Design of Embedded Systems* », Department of Information and Computer Science University of California, Irvine.
- [24] : Steve Heath. « *Embedded Systems Design* » 2nd Edition, Newnes, 2003.
- [25] : Adnan Shaout, Ali H. El-Mousa., and Khalid Mattar. « *Specification and Modeling of HW/SW CO-Design for Heterogeneous Embedded Systems* ». Procédons of the World Congress on Engineering (2009) Vol I, London, U.K.
- [26] : « *Les microcontrôleurs dans les systèmes embarqués* », Haute école spécialisée bernoise, (2012).
- [27] : Z. Mammeri, « *Systèmes temps réel et embarqués* », université de Toulouse II
- [28] : FabriceK., Jérôme H., Agusti C., Alain D., « *Modélisation et analyse de systèmes embarqués* », Hermes (2012).
- [29] : Guillaume Normand, Benjamin Bonny, Loïc Raucy, « *Communications dans les systèmes embarqués* », (2011).
- [30] : KopetzH., « *Real-Time Systems: Design Principles for Distributed Embedded Applications* », Springer New York Dordrecht Heidelberg London.
- [31] : SalvadorC.H., CarrascoM.P., deMingoM.A.G., CarreroA.M., MontesJ.M., MartinL.S., Cavero, LozanoI.F., MonteagudoJ.L., « *Bluetooth Telemedicine Processor for Multichannel Biomedical Signal Transmission Via Mobile Cellular Networks* », IEEE Trans, Biomedicine, vol.9, Mar (2005).
- [32] : Fouquet P. et Bouchy A., « *Les Réservoirs d'eau* », Dunod, 1963, 165 p