
ASSISTANCE GLOBALE POUR AIDE AU PARKING

Rapport intermédiaire de projet

Cinquième année département

Informatique, Microélectronique et Automatique

Auteurs :

Mathieu GERIER

Céline LY

Année :

2014/2015

Nom de la matière :

Projet de fin d'études

À l'attention de :

M. Alexandre BOÉ

M. Thomas VANTROYS

Remerciements

Nous tenons tout d'abord à remercier toute l'équipe pédagogique de Polytech Lille et les responsables de la formation Informatique, Microélectronique et Automatique de nous avoir enseigné les bases pour réaliser ce projet dans de bonnes conditions.

Nous tenons aussi à remercier et à témoigner toute notre reconnaissance à Monsieur Alexandre BOÉ et Monsieur Thomas VANTROYS, nos responsables de projet, pour leur soutien permanent et leur disponibilité tout au long du projet.

Nous souhaitons également remercier Monsieur Xavier REDON et Monsieur Thierry FLAMEN pour nous avoir aidés sur certains aspects du projet lorsque nous en avons besoin.

Sommaire

| | |
|---|-----------|
| Remerciements | 1 |
| Sommaire..... | 2 |
| Introduction | 4 |
| I. Présentation du projet | 5 |
| 1. Présentation générale | 5 |
| a. Contexte | 5 |
| b. Présentation du projet..... | 5 |
| 2. Matériel et outils nécessaires | 7 |
| a. Matériel utilisé..... | 7 |
| b. Outils nécessaires..... | 8 |
| 3. Étapes du projet | 9 |
| II. Avancement du projet..... | 11 |
| 1. Partie réseau de capteurs | 11 |
| a. Choix des solutions liées aux capteurs | 11 |
| b. Choix des solutions liées au réseau..... | 14 |
| c. Système d'exploitation | 15 |
| d. Tests réalisés..... | 15 |
| 2. Partie applicative | 16 |
| a. Différentes études menées | 17 |
| b. Tests réalisés..... | 20 |
| 3. Partie commune : gestion de la communication entre le module capteur et le serveur web 21 | |
| III. Bilan de la situation actuelle..... | 22 |
| 1. Difficultés rencontrées | 22 |
| 2. Récapitulatif | 23 |
| IV. Perspectives futures..... | 24 |
| 1. Futurs objectifs et démarches..... | 24 |
| a. Partie réseau de capteurs | 24 |

| | |
|--------------------------------|-----------|
| b. Partie applicative | 25 |
| c. Projet global..... | 25 |
| 2. Planning prévisionnel | 26 |
| Conclusion..... | 27 |
| Annexes | 28 |

Introduction

Les réseaux de capteurs sans fil font de plus en plus partie de la vie quotidienne. En effet, on les retrouve dans les domaines du militaire, de la sécurité civile ou encore dans le transport. Ils sont de plus en plus présents afin de nous faciliter la vie et d'améliorer notre confort. Le but du projet est d'utiliser une telle technologie afin d'aider les automobilistes dans la recherche de places de parking. Dans un premier temps, nous verrons les objectifs fixés par le cahier des charges, puis dans un second temps, le déroulement du projet. Ensuite, nous ferons un bilan de la situation actuelle et enfin finirons par évoquer les perspectives futures.

I. Présentation du projet

1. Présentation générale

a. Contexte

De nos jours, de par l'augmentation du nombre de véhicules circulant, il est de plus en plus stressant de se rendre à son lieu de travail ou dans des grandes surfaces. Les embouteillages, les accidents sont des facteurs de stress pour les automobilistes. De plus, lors de l'arrivée de ces derniers dans les parkings, ils perdent de précieuses minutes à chercher des places pour se garer. Ceci peut augmenter le stress, la pollution et l'énerverment. C'est dans ce contexte que nous proposons une solution aux automobilistes afin de les aider à trouver des places de parking plus aisément et dans de bonnes conditions.

b. Présentation du projet

L'objectif de ce projet est de concevoir un prototype répondant à la problématique.

Afin d'aider au mieux les automobilistes, nous avons pensé à utiliser une application qui affiche des informations comme par exemple, le nombre de places disponibles, le nombre de places totales ou encore les types de places qu'il y a dans un parking. L'application pourra également guider l'utilisateur vers des places disponibles en présentant un plan du parking.

Un réseau de capteurs est nécessaire pour connaître des données telles que l'emplacement, l'état de chaque place de parking. Un capteur est un dispositif capable de mesurer une grandeur physique. Dans le cadre du projet, il sert à détecter la présence d'un obstacle (une voiture en

l'occurrence). Le réseau de capteurs permet d'établir la communication entre les différents capteurs afin de remonter les informations à l'application (cf. point précédent) pour qu'elle puisse se mettre à jour et interagir avec les automobilistes.

Pour que l'application récupère les données du parking, nous utiliserons des balises permettant de récupérer les informations des places grâce aux capteurs : ce sont les concentrateurs. Il y aura un concentrateur général, où toutes les données seront acheminées à partir des autres concentrateurs que l'on appellera concentrateurs relais et des capteurs ; le concentrateur général permettra de communiquer les données à l'application grâce à Internet. Ces dernières seront stockées dans une base de données qui permet d'enregistrer des données de façon efficace et organisée.

Ainsi, le projet s'articule autour de trois points :

- Le réseau de capteurs permettant de récupérer les informations comme la disponibilité de certaines places, le type de place ou encore la morphologie.
- L'application permettant d'interagir avec l'automobiliste et d'afficher les informations indispensables à l'optimisation de son temps pour le parage.
- L'association des deux parties précédentes afin de mettre à jour l'application grâce au réseau de capteurs. Il s'agit de la communication entre l'application et le réseau de capteurs.



Fig. 1 : *Vue globale des parties du projet*

2. Matériel et outils nécessaires

a. Matériel utilisé



Fig. 2 : Carte CC430

Afin de réaliser au mieux notre projet, nous avons choisi d'utiliser plusieurs cartes CC430 réalisées par Monsieur BOÉ et Monsieur VANTROYS. Il s'agit de cartes électroniques comprenant un module RF (*Radio Frequency*) permettant d'envoyer et de recevoir des informations en communication sans fil à une certaine fréquence. Elles disposent également d'un microcontrôleur dont l'utilité, dans notre projet, est de

connaître la disponibilité d'une place de parking grâce à un capteur connecté à ce dernier et de gérer les échanges de données entre les différents modules (concentrateur ou capteur). Une carte serait présente sur chaque place de parking avec un capteur, de même une carte fait partie de chaque concentrateur afin d'échanger les informations grâce à la communication sans fil. On désignera désormais un module capteur comme l'association d'une carte et d'un capteur.

L'utilisation de ces cartes n'est pas la plus adaptée pour le projet mais il est intéressant de s'en servir afin de concevoir un prototype. En effet, elle est plus performante que ce dont nous avons réellement besoin (mémoire, puissance de calcul, etc.). Cependant, nous pourrions déterminer les capacités réelles que demande le prototype et nous pourrions ainsi utiliser une solution plus adaptée pour le projet.

Par ailleurs, afin d'assurer une bonne communication sans fil, nous avons décidé d'utiliser la bande de fréquence 868 MHz. Grâce à cette fréquence, la communication peut être réalisée à une plus grande distance par rapport à une fréquence de 2,4 GHz et ce, pour une même taille d'antenne.

Nous avons également besoin du MSP430 LaunchPad. Il s'agit d'une carte comprenant un programme permettant d'envoyer un programme sur une carte CC430 à partir d'un ordinateur (cf. Annexe 1). Le programme permet, dans ce cas précis, de récupérer l'information du capteur et d'envoyer et de recevoir des informations venant du réseau de capteurs.

Pour nos phases de tests, nous allons utiliser un interrupteur pour chaque carte où il y a besoin d'un capteur. L'interrupteur permet de remplacer le capteur. En effet, ils ont la même fonction, c'est-à-dire, que si un obstacle apparaît devant le capteur, ce dernier changera d'état. Il en va de même pour l'interrupteur, où l'on va pouvoir simuler un obstacle en changeant de valeur l'interrupteur.

b. Outils nécessaires

Certains outils ont été nécessaires quant à la réalisation du projet. Tout d'abord, un choix dans l'utilisation un serveur web a été fait. Il s'agit d'un serveur informatique utilisé pour publier des sites web sur Internet. L'objectif de l'application est de transmettre des informations à tous les utilisateurs et pour cela nous allons utiliser un serveur web qui pourra informer les automobilistes qui ont accès à Internet. Cet accès est indispensable car l'état des places peut changer à tout moment. L'application, stockant ces informations dans une base de données, doit être capable de les récupérer à tout moment pour en informer l'utilisateur afin d'être tout le temps à jour. Il existe des hébergeurs de site webs gratuits. Cependant, ils présentent quelques contraintes comme l'utilisation des langages PHP et MySQL que nous évoquerons ultérieurement.

Afin de réaliser une page web, nous avons besoin de certains outils comme les langages HTML et CSS. Ces derniers sont des langages de balisage permettant de structurer une page web. Ce sont eux qui affichent le texte désiré d'une certaine manière. En effet, on peut définir à quel endroit d'une page web va se situer le texte, la taille, les couleurs, etc. Ils permettent d'agencer la page de sorte à ce qu'elle soit attractive.

PHP est un langage de programmation utilisé pour le développement de pages web ; c'est celui qui est le plus utilisé dans ce domaine. Il est gratuit et ne nécessite aucune licence d'utilisation. Il existe un nombre incommensurable de bibliothèques et de tutoriels.

Pour que l'application fonctionne correctement, le site web doit stocker les informations relatives aux parkings renseignés. Pour cela nous utilisons une base de données. Il s'agit d'un outil capable de stocker et de récupérer des informations comme défini précédemment. Le réseau de capteurs renseignera la base de données permettant à l'application d'aller chercher les informations dans cette dernière et d'afficher celles souhaitées par l'utilisateur. Nous utiliserons le langage MySQL qui est un langage de programmation permettant la gestion de base de données (ajout, mise à jour ou suppression d'informations). Ce langage présente quelques avantages : il est gratuit et présente beaucoup de fonctionnalités. Il est également très répandu et il est très facile de rechercher des informations en cas de problème.

3. Étapes du projet

Afin de mener à bien le projet, différents objectifs ont été posés :

- Compréhension du sujet de projet : première appréhension du sujet avec des échanges entre nos tuteurs et nous afin de réellement connaître les attentes de ce projet ;
- Étude des solutions à choisir : établissement d'un cahier des charges précis, et étude théorique des solutions qui pourraient être appliquées dans le cas où le prototype est fonctionnel ;
- Prise en main du matériel et familiarisation avec les outils choisis : familiarisation avec le matériel et ses fonctionnalités ainsi que découverte et approfondissement des connaissances et des possibilités des outils choisis ;

- Premières phases de tests pour le réseau de capteurs : réalisation de premiers tests basiques pour s'approprier le matériel (carte CC430 et MSP430 LaunchPad) et prise en main des fonctionnalités nécessaires au projet ;
- Premières phases de tests pour la partie application : conception d'une page web assez basique avec base de données et affichage des données souhaitées à l'utilisateur ;
- Réalisation d'un prototype simple : prototype répondant au cahier des charges et ne prenant pas forcément en compte toutes les erreurs et problèmes qui peuvent survenir ;
- Amélioration de ce prototype : prise en compte les erreurs et problèmes rencontrés lors du premier prototypage en situation réelle et d'apporter de nouvelles fonctionnalités.

II. Avancement du projet

1. Partie réseau de capteurs

Cette partie du projet permet de récupérer les informations des capteurs (disponibilité de certaines places).

On peut y distinguer deux phases : une première phase d'initialisation qui consiste à connaître la position de chaque capteur ; et une seconde phase qui consiste à remonter l'information de disponibilité de la place grâce à l'identifiant du capteur (chaque capteur est distingué par un identifiant unique). Cette information remontée servira à mettre à jour la base de données.

Dans cette partie, nous verrons, dans un premier temps, les choix technologiques que nous avons réalisés, tels que les solutions liées aux capteurs. Nous évoquerons, dans un second temps, la topologie du réseau retenue. Ensuite, nous argumenterons sur le fait d'utiliser ou non un système d'exploitation. Enfin, nous présenterons les tests réalisés, ainsi que les résultats obtenus.

a. Choix des solutions liées aux capteurs

Il est à noter qu'avant de concevoir le réseau de capteurs, nous avons dû passer par une première étape d'étude du sujet. En réalisant cette étude, nous sommes parvenus à des choix technologiques qui sont les solutions que l'on aurait appliquées dans une seconde phase de prototypage, et qui auraient été plus adaptées aux besoins. En effet, pour notre projet, la première phase de prototypage consiste à faire fonctionner le réseau de capteurs avec le matériel déjà à disposition. Ce matériel est, dans notre cas, surdimensionné, mais cela nous a permis de commencer à travailler concrètement sur des solutions même si elles ne sont pas les plus adaptées aux besoins de projet.

Une première phase a consisté à réfléchir sur la solution liée aux capteurs que nous allons utiliser pour le projet. Nous nous sommes demandé s'il était possible de récupérer les informations de plusieurs places de parking grâce à un seul capteur. Cette solution aurait permis de ne pas avoir de contraintes de consommation d'énergie car le capteur aurait été placé sur un lampadaire et il aurait été alimenté par ce dernier. Cependant, dans un souci de rendre notre projet générique, c'est-à-dire adaptable à n'importe quel parking, nous avons conclu que cette solution ne pouvait pas être utilisée. En effet, chaque parking est différent : certains sont intérieurs, d'autres extérieurs et chacun a un agencement "unique" (arbres, trottoirs, etc.). Nous avons donc retenu la solution d'un capteur par place, ce qui signifie que le module capteur permet de savoir si une voiture est en stationnement ou non sur une place de parking uniquement. Cette information sera alors par la suite remontée à l'application. Les inconvénients de cette solution sont que le module capteur devra être alimenté par une batterie ou une pile. Il faudra ainsi limiter la consommation énergétique le plus possible afin d'éviter d'avoir trop souvent recours à la maintenance pour remplacer ou recharger un module capteur.

Nous avons également réfléchi sur les endroits où l'on pouvait placer les modules capteurs afin d'éviter des erreurs potentielles comme des erreurs de détection d'obstacles ou de mauvaise communication sans fil. Ils auraient pu être placés en hauteur, sous ou derrière les places de parking, etc. Nous avons pensé à placer les capteurs au milieu des places de parking car il est plus facile de repérer une voiture en cas de pluie ou de brouillard. Ce choix pourra être changé par la suite en cas de tests non concluants.

En ce qui concerne le placement des concentrateurs qui devront se charger de récupérer les données des modules capteurs, nous avons établi l'hypothèse que leur position devra être connue à l'avance.

Chaque concentrateur sera placé :

- De telle sorte à garantir, du mieux possible, l'homogénéité de l'émission et de la réception des données. La consommation énergétique sera plus équitable pour chaque module

capteur car plus un module capteur est loin d'un concentrateur, plus il va dépenser de l'énergie afin de communiquer entre eux ;

- Sur un lampadaire afin d'être alimenté à une source d'énergie "infinie", c'est-à-dire sans problème de décharge. Il pourra ainsi envoyer et recevoir des données sans tomber en panne d'énergie.

De plus, en connaissant la distance entre les concentrateurs, nous pourrions utiliser une technique comme la triangulation pour déterminer l'emplacement des modules capteurs. Une phase de test sera nécessaire afin de déterminer si la technique de localisation est fiable ou non et dans le cas où elle ne l'est pas, nous adapterons notre solution. Lorsque la localisation sera définie, le module capteur aura un identifiant assigné afin que l'application web puisse déterminer à quelle place correspond quel module capteur. L'application pourra alors afficher un plan représentant le parking avec toutes ses places et leur état.

Une étude a été consacrée à la recherche de capteurs que l'on aurait pu utiliser pour le projet. Deux solutions ont été retenues. La première est l'utilisation d'un capteur à ultrasons : son coût est faible mais présente un inconvénient non négligeable, à savoir son incapacité à détecter des obstacles en cas d'obstruction au niveau de l'antenne. La deuxième solution est d'utiliser un capteur électromagnétique : il est plus fiable que la solution précédente, mais il est plus cher aussi.

b. Choix des solutions liées au réseau

Principe de la topologie choisie : *Cluster-Tree*

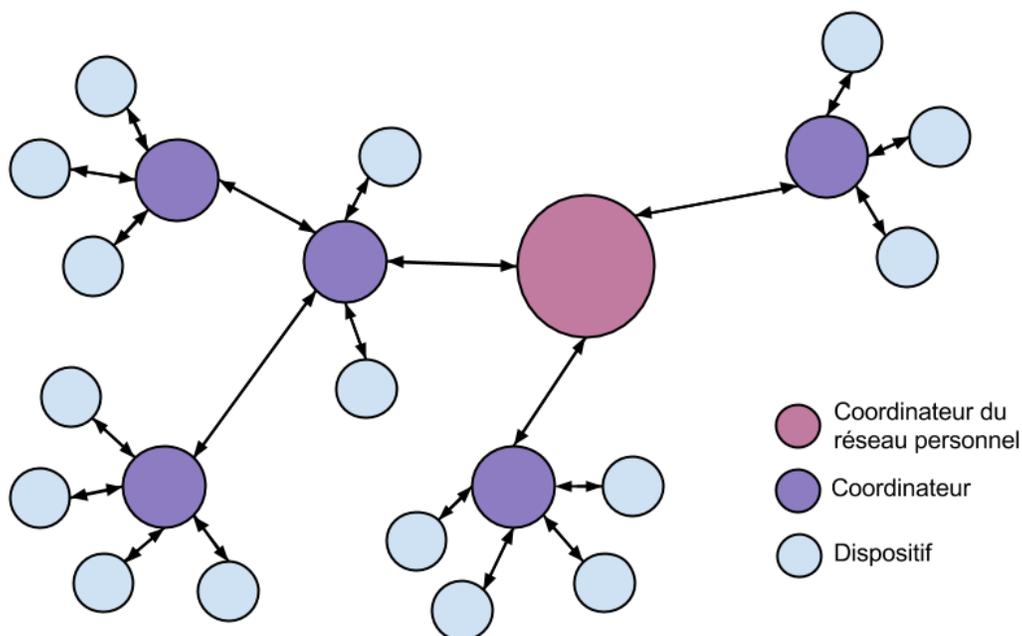


Fig. 3 : Topologie *Cluster-Tree* adoptée pour le projet

Ce schéma montre que le réseau de capteurs est la partie qui va permettre de remonter les informations liées aux places de parking disponibles. Toute la communication réalisée est sans fil. Suivant la taille du parking, la communication des données ne peut être réalisée avec un unique concentrateur, qui regrouperait toutes les données. Ces informations doivent tout d'abord être remontées à un premier concentrateur, qui s'occupe de relayer les informations liées à une zone spécifique d'un parking. Une fois que ces informations sont récupérées par ce premier concentrateur, si la distance vers le concentrateur général le permet, les informations sont communiquées à ce concentrateur général, sinon elles passent par un autre concentrateur relais jusqu'à atteindre le concentrateur général. Ce principe suit la topologie *Cluster-Tree*, que nous avons choisie. Cette topologie possède aussi l'avantage d'être statique : en effet, les

chemins de communication sont prédéfinis. Ceci permet une implémentation plus facile, et par la même occasion une diminution de la consommation d'énergie. En effet, s'il n'y a pas de recherche dynamique à réaliser avant d'envoyer des informations, les microcontrôleurs associés aux capteurs auront moins de calculs à réaliser, réduisant leur temps de travail et donc leur consommation d'énergie. Cependant, il est à noter qu'il existe une première phase dynamique qui consiste à définir l'agencement du réseau (les liens entre module capteur et concentrateur).

c. Système d'exploitation

Nous avons pondéré la question de l'utilité d'un système d'exploitation dans notre cas. En effet, un système d'exploitation permet d'assurer la liaison entre les ressources matérielles, l'utilisateur et les applications. Son implémentation sur un microcontrôleur confère un accès plus pratique pour la modification et donc l'écriture de programmes, une certaine robustesse au niveau du traitement des données, une possibilité de gestion en temps réel, etc. Cependant, dans notre cas, nous pouvons effectuer ces opérations sans système d'exploitation même si la mise en œuvre en sera plus difficile. Par ailleurs, l'utilisation d'un système d'exploitation nécessite une certaine performance du microcontrôleur et nous limite donc dans le choix du matériel à utiliser. Ainsi, nous n'utiliserons pas de système d'exploitation afin de choisir le matériel le plus adapté au projet pour la suite.

d. Tests réalisés

Une première partie consistait à l'appréhension du LaunchPad et des cartes CC430. Lors de cette phase, nous avons effectué des tests basiques sur le LaunchPad, tels qu'allumer une LED par appui sur bouton poussoir ou la faire clignoter à une certaine fréquence. Pour cela, nous devons programmer le microcontrôleur implanté sur le LaunchPad. Pour qu'il fonctionne d'une certaine manière, il est indispensable de paramétrer des registres. Ainsi, cette première approche

a permis de découvrir les registres de base qui assignent aux ports leur type (entrée ou sortie) et leur état.

Une fois ces tests réalisés, la programmation sur carte CC430, à l'aide du LaunchPad, a commencé. Afin d'effectuer des tests basiques comme par exemple connaître l'état d'un interrupteur, nous avons soudé un fil sur les cartes CC430 afin de l'y connecter. Ce dernier permet de simuler l'état des capteurs de détection d'obstacle. En effet, lorsque l'interrupteur est dans une certaine position, il simule que le capteur détecte un obstacle (la place est occupée) et lorsqu'il est dans l'autre position, le capteur détecte qu'il n'y a pas d'obstacle (la place est libre). Cette deuxième phase de tests a donc consisté en la programmation des cartes CC430. De nouveaux registres ont été paramétrés : il a fallu paramétrer les registres d'interruption, à savoir l'autorisation d'interruption et le type d'interruption (front descendant).

Ces tests ont ainsi permis de découvrir les registres de bases d'entrée et de sortie, ainsi que les registres d'interruption. Dans un second temps, il a fallu appréhender les registres de la communication RF et ceux de la communication série. Une première série de tests a été basée sur des morceaux de codes fournis par Monsieur VANTROYS afin de mieux comprendre le fonctionnement de la communication RF.

2. Partie applicative

Cette partie du projet permet de récupérer les données envoyées par le réseau de capteurs et de mettre à jour la base de données en conséquence. Elle permet ensuite d'afficher les informations nécessaires à l'utilisateur pour qu'il puisse savoir s'il peut ou non se garer dans un parking choisi ainsi que de connaître les caractéristiques de ce dernier. Enfin, il est possible qu'un opérateur puisse mettre à jour la base de données, rajouter de nouvelles informations ou enlever. Il pourra de plus, créer grâce à une interface, un parking virtuel représentant le parking réel qu'il souhaite ajouter. Ce dernier sera visualisable par un utilisateur de l'application afin d'avoir un aperçu des places disponibles où il peut se garer.

a. Différentes études menées

Une première phase d'étude a consisté à chercher le type d'application que nous allons utiliser (application Android, application web, etc.). Afin que tout le monde puisse avoir accès à l'application, nous avons choisi de faire une application web. En effet, dans le cas d'une application Android, certains utilisateurs de téléphones comme Apple ou Microsoft n'auraient pas pu y avoir accès. Le fait d'utiliser une application web permet à toute personne ayant une connexion Internet, qui est indispensable pour le projet dans un souci de mise à jour des informations, d'utiliser l'application.

Pour que tous les utilisateurs puissent y avoir accès, nous devons mettre l'application sur un serveur web.

Voici le schéma illustrant l'architecture du système :

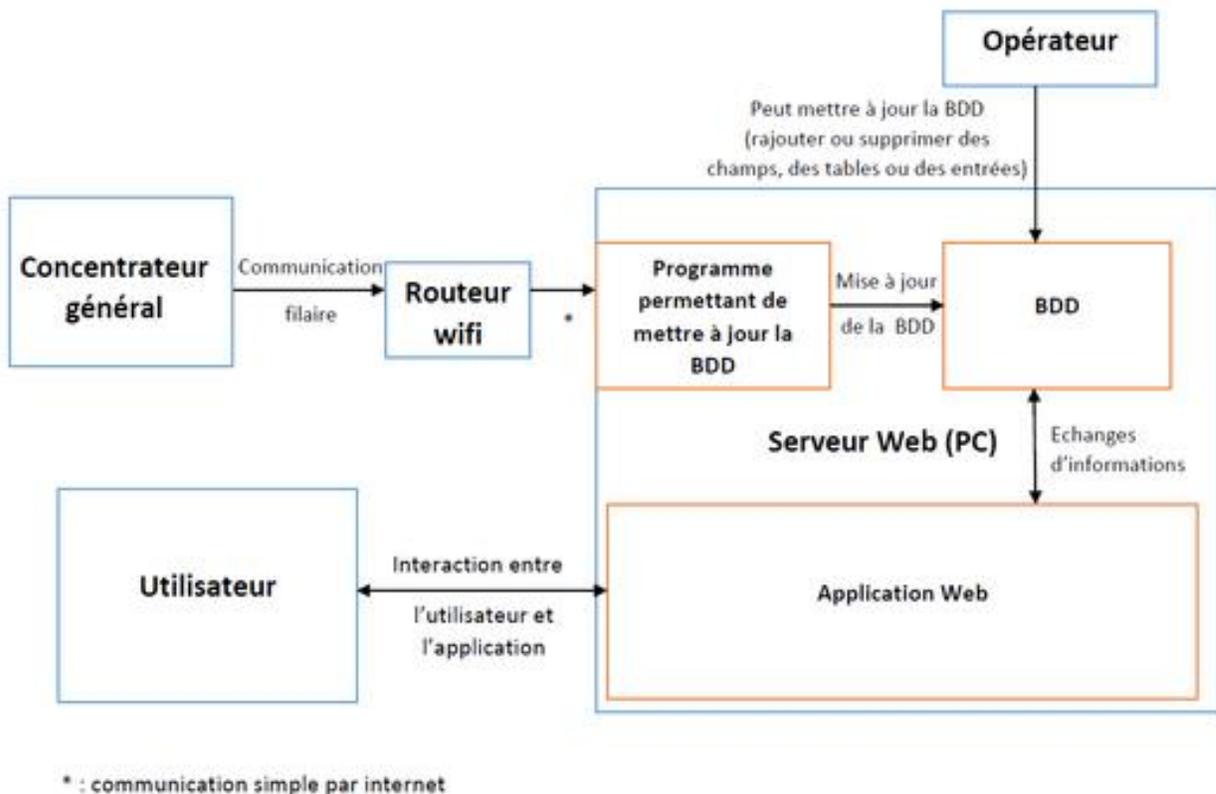


Fig. 4 : Architecture du système

Comme précisé dans la partie C.2., une phase du projet a consisté à choisir quels outils nous allons utiliser pour l'application.

Afin de connaître l'état des différents capteurs appartenant à un parking, nous avons réfléchi sur la conception de la base de données que nous devons utiliser. Le schéma ci-contre présente la base de données créée :

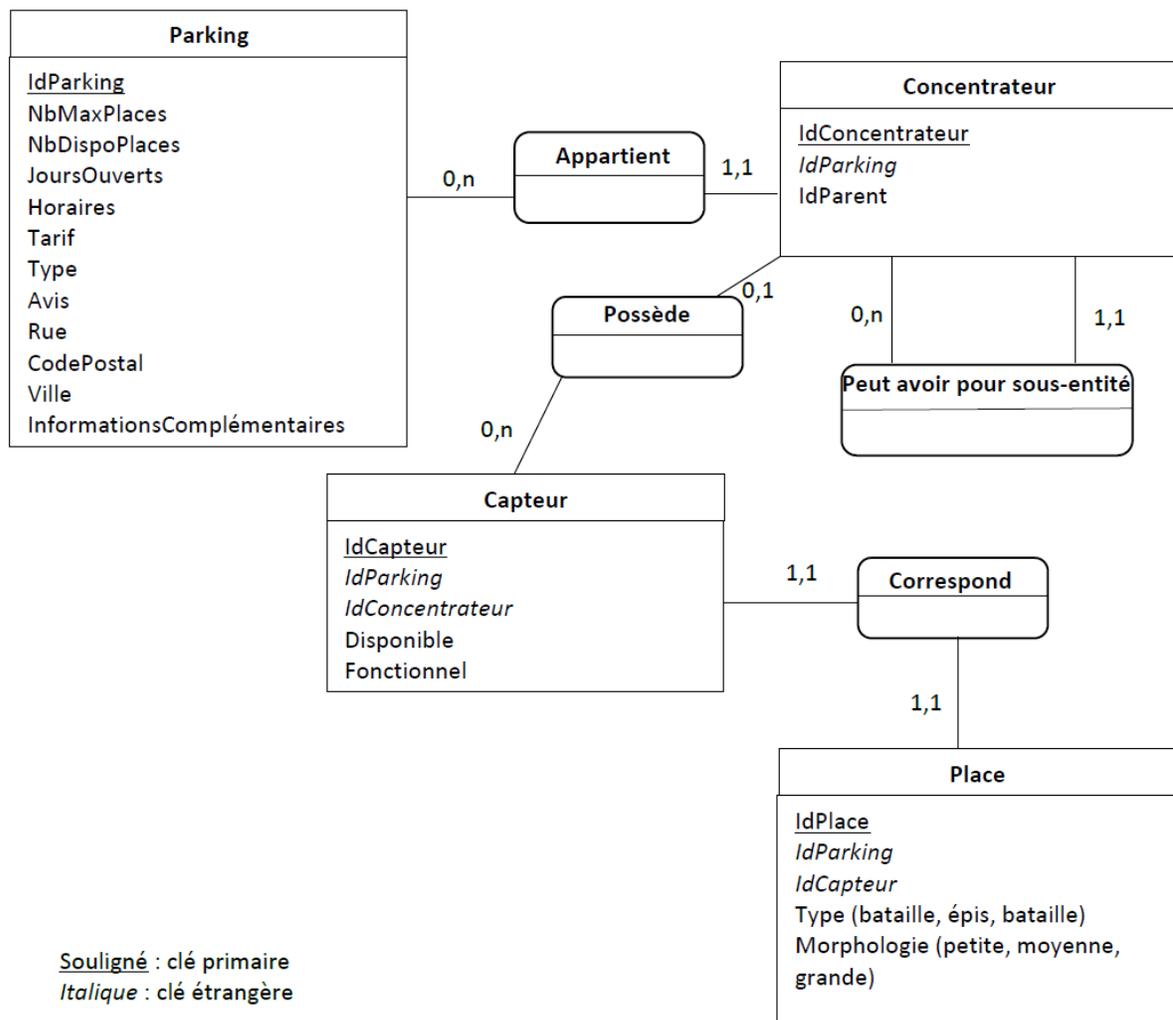


Fig. 5 : Représentation de la base de données utilisée pour le projet

Par exemple, sur le schéma ci-dessus, on peut voir qu'il existe un parking où il peut y avoir plusieurs concentrateurs et chaque concentrateur peut être relié à zéro ou plusieurs capteurs. La base de données permettra de renseigner sur une page web de l'application combien il reste de places disponibles sur un nombre total de places. Il pourra aussi renseigner les horaires du parking, les tarifs, le type (intérieur ou extérieur) ou encore la morphologie et les types de places de chaque parking.

Nous avons ensuite réfléchi sur les informations à transmettre à l'utilisateur et comment nous allons les afficher. Par exemple, un utilisateur souhaite accéder à un parking particulier, nous avons pensé à lui afficher les différents parkings répertoriés. Cet affichage sera dans un premier temps très basique (liste déroulante) puis dans un second temps, il sera plus attractif et plus intuitif (exemple d'une carte GPS). Les informations telles que l'affichage de la capacité totale du parking, le nombre de places disponibles, les types ou la morphologie de chaque place seront mentionnées. Il est très important de transmettre les informations de manière intuitive pour que l'utilisateur ait envie de s'en servir souvent. En effet, dans la suite de la conception du site web, nous pourrons afficher à l'écran (ordinateur, tablette ou téléphone) le parking avec un plan qui mentionnera où sont les places disponibles pour l'aider à se repérer dans l'espace.

Enfin, une dernière phase a consisté à réfléchir sur la manière dont un opérateur pouvait créer un plan de parking. Le but de ce dernier est qu'il puisse être affiché à un utilisateur en renseignant si une place est libre ou non ; en effet, chaque capteur est associé avec une place de parking. L'interface utilisée se présente sous la forme suivante.

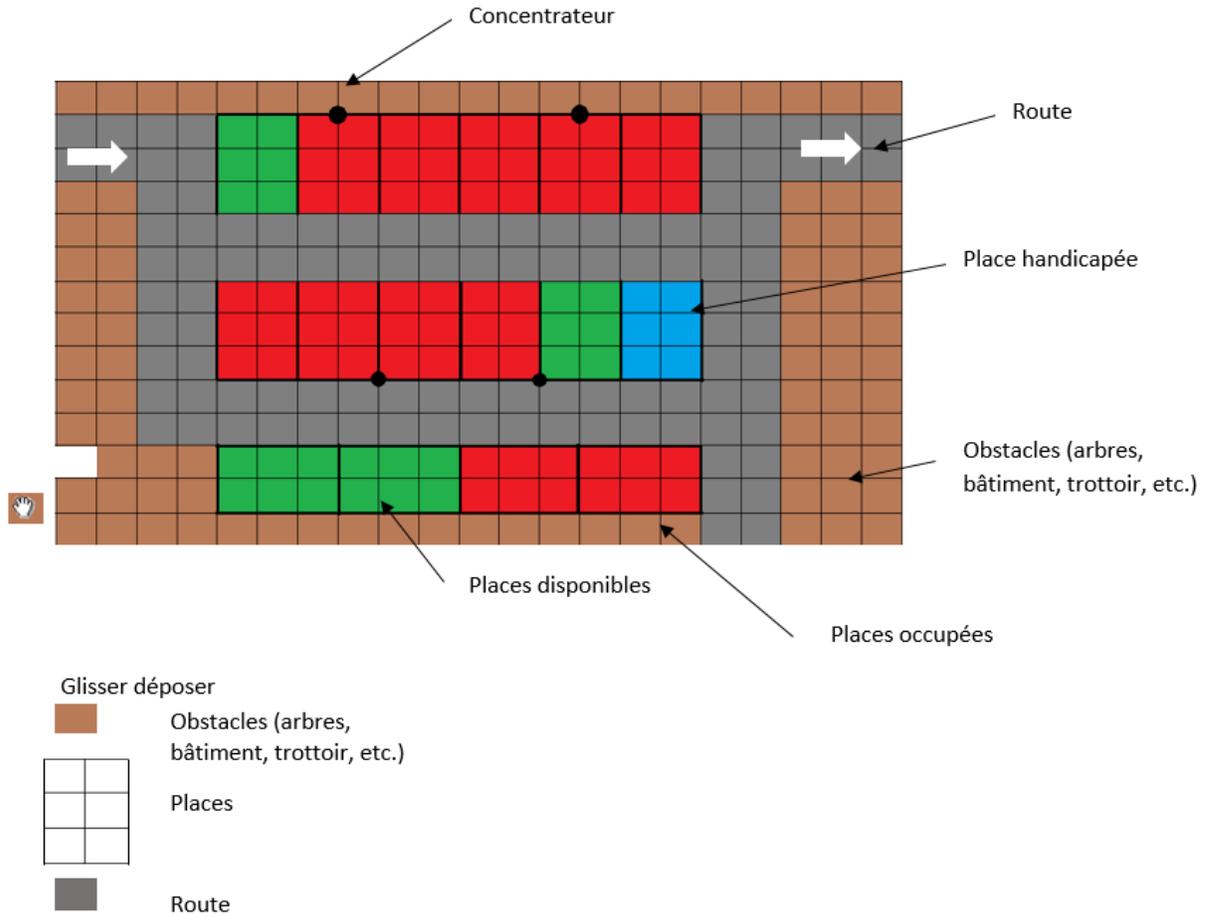


Fig. 6 : Interface de création de plan de parking

Le réseau de capteurs permet, dans une phase d'initialisation, de connaître le temps de transmission entre un module capteur et un concentrateur. Cette information est alors remontée à l'application, qui associera les places de parking sur le plan à des capteurs placés sur le vrai parking. Ce traitement sera réalisé à l'aide d'un algorithme.

b. Tests réalisés

Tout d'abord, des tutoriels ont été réalisés afin de se familiariser avec les outils nécessaires au projet. Ensuite, nous avons géré les types d'utilisateurs qu'il peut y avoir sur un

site web. Il peut y en avoir trois : un membre, un configurateur et un administrateur. Un configurateur pourra créer, grâce à une interface, le plan du parking destiné à être affiché à tous les utilisateurs. Un administrateur pourra changer le statut de chaque utilisateur du site. Un utilisateur pourra se connecter et en fonction de son statut, il pourra effectuer diverses tâches.

Une troisième phase a consisté à introduire la base de données précédemment citée sur le serveur web. Dans une première phase de tests, un ordinateur fera office de serveur web. Il sera amené à être implanté sur un hébergeur gratuit pour la suite.

3. Partie commune : gestion de la communication entre le module capteur et le serveur web

Cette partie concerne la communication entre le réseau de capteurs et l'application web. Les informations, recueillies par le réseau de capteurs, doivent être stockées dans la base de données situées sur le serveur web.

Dans un premier temps, les informations sont transmises entre les concentrateurs et les modules capteurs. Or, pour que deux cartes CC430 puissent communiquer correctement, elles doivent suivre un protocole de communication. Ce dernier est la spécification d'un ensemble de règles permettant d'établir un type de communication particulier. Il en va de même pour la communication entre l'application et une carte. C'est une manière de recevoir et de comprendre ce qui a été envoyé. Dans un second temps, il était indispensable de connaître l'ordre des informations envoyées par le réseau de capteurs afin de mettre à jour la base de données sans souci d'erreur. Par exemple, l'état d'un certain capteur a changé et il faut que la base de données reconnaisse ce capteur afin d'en changer la valeur d'état.

III. Bilan de la situation actuelle

1. Difficultés rencontrées

Tout au long du projet, nous avons dû faire face à plusieurs problématiques :

- La compréhension du sujet a été la tâche la plus complexe du projet. En effet, il est très important de comprendre en tout point le sujet donné afin de répondre au mieux à la problématique du projet. Nous avons pris rendez-vous avec nos tuteurs de projet afin de connaître précisément son contexte et de savoir les tâches indispensables à réaliser.
- La maîtrise de nouveaux procédés : l'utilisation de nouveau matériel comme les cartes CC430, de nouveaux outils tels les langages de programmation. La recherche d'informations via des livres, Internet, des articles rédigés à cet effet ainsi que des échanges avec des membres de l'équipe pédagogique de Polytech Lille ont été nécessaires afin de pallier les problèmes rencontrés. Il était indispensable d'assimiler ces procédés afin d'avoir un bon déroulement de projet.
- La recherche et les choix des solutions à adopter. Cette partie a été très problématique. En effet, nous n'avons pas beaucoup d'expérience dans les domaines dans lesquels nous avons travaillé et il a été difficile, de trouver les meilleures solutions et d'en choisir une. Afin de résoudre ceci, nous avons discuté avec des personnes plus expérimentées et nous avons pu échanger nos idées, les argumenter afin de retenir les meilleures.
- Le respect du planning s'est révélé pour cette moitié de projet difficile à suivre du fait de la survenue de problèmes ponctuels qu'il a fallu résoudre au fur et à mesure. Il a été nécessaire de trouver des solutions rapidement afin de remplir, dans les temps impartis, les objectifs fixés. Il est important de ne pas sous-estimer la tâche à effectuer afin de respecter les délais définis (gestion de projet).

2. Récapitulatif

Au cours de ces derniers mois, nous nous sommes intéressé au sujet et aux solutions que nous allions apporter en définissant un cahier des charges précis. Cette étape nous a permis de mieux comprendre le sujet, les attentes et ainsi prévoir un premier planning prévisionnel. Cependant, comme dit précédemment, ce premier planning n'a pas été respecté à cause des difficultés rencontrées. Il a été revu et réadapté tout au long du projet. Pour la seconde moitié du projet, un planning prévisionnel a été défini (cf. partie IV - B) et prend donc en compte nos avancements et nos retards.

Au niveau de ce qui a été concrètement réalisé, en ce qui concerne la partie réseau de capteurs, des tests basiques ont été réalisés permettant de détecter un changement d'état d'un interrupteur, représentant un changement d'état de capteur. Ces changements sont représentés par un clignotement d'une des LEDs présentes sur la carte CC430. Une seconde phase de tests en lien avec la communication RF est actuellement en cours.

En ce qui concerne la partie applicative, une phase d'étude a été entreprise afin de définir concrètement les fonctionnalités de cette partie. Une deuxième phase a permis de se familiariser avec les outils utilisés afin d'être plus efficace par la suite en terme de programmation. Enfin, une partie de conception a été réalisée comme l'implémentation de la base de données et la gestion des statuts des utilisateurs.

IV. Perspectives futures

1. Futurs objectifs et démarches

a. Partie réseau de capteurs

Dans cette partie, il s'agira de mettre en place un prototype de réseau de capteurs, à l'aide de multiple cartes CC430. Un premier test consistera à mettre en place un petit réseau de deux ou trois capteurs (avec les cartes CC430, donc) qui devront communiquer en RF, suivant le protocole de communication défini, à un premier concentrateur relai (représenté lui aussi par une carte CC430). Une fois cette première étape réalisée, la deuxième étape aura pour objectif de réaliser plusieurs petits réseaux de capteurs, qui posséderont chacun un concentrateur relai ; chaque concentrateur devra alors communiquer avec un concentrateur "général", qui s'occupe de traiter toutes les données d'un parking. Enfin, dans une dernière étape, il serait bien d'adapter ce précédent réseau afin de pouvoir transmettre les données d'un concentrateur relai à un autre avant de les transmettre au concentrateur général. En effet, ceci permettrait de gérer le cas où les concentrateurs relais ne seraient pas assez proches du concentrateur général pour pouvoir communiquer directement avec celui-ci.

Par ailleurs, lorsqu'un prototype sera établi, une phase de tests devra être effectuée afin de savoir s'il est fonctionnel dans l'environnement réel souhaité. Ces tests permettront aussi de déterminer si les informations envoyées par les modules capteurs prennent un chemin unique, ou si elles passent par plusieurs intermédiaires. Dans le cas où les hypothèses ne seraient pas valides, il faudra adapter la solution en fonction des résultats obtenus.

b. Partie applicative

Pour la partie applicative, il sera nécessaire de créer une interface pour qu'un opérateur puisse réaliser les plans de parking sur l'application. Ensuite, une seconde phase permettra d'associer un capteur à une place de parking. Ceci permettra d'indiquer à un utilisateur de visualiser le plan de parking et de connaître les places disponibles avec leurs caractéristiques (sa morphologie, son type). Pour réaliser cette association, nous devons récupérer les informations envoyées du réseau de capteurs et concevoir une fonction permettant de connaître l'emplacement d'un capteur sur le plan créé par l'opérateur.

L'agencement du site web ainsi que les informations à y afficher seront d'autres tâches à effectuer. En effet, nous pourrons rajouter un plan GPS pour indiquer aux automobilistes les parkings qu'il y a à proximité d'un lieu spécifique.

De plus, il est possible que la base de données soit modifiée pour que l'application puisse intégrer de nouvelles fonctionnalités.

c. Projet global

Il s'agira ici de réaliser le protocole de communication permettant de faire communiquer les modules afin de remonter les informations à l'application qui pourra mettre à jour la base de données.

Dans une première étape, plusieurs concentrateurs récupéreront le temps de transmission entre un module capteur et chaque concentrateur. Ils transmettront les informations au concentrateur général qui va remonter les informations à l'application. Cette dernière calculera grâce aux temps récupérés, par des calculs très simples, la distance entre les concentrateurs et les modules capteurs. Elle permettra ainsi de positionner un capteur, par rapport aux concentrateurs, dans le plan du parking et associera donc une place de parking au module capteur. Dans la seconde phase, les modules capteurs communiqueront seulement avec le concentrateur le plus proche pour lui communiquer leur état. Pour cela, l'application comparera les temps reçus, sélectionnera

le temps le plus faible et choisira donc le concentrateur le plus proche du module capteur. Ensuite, cette information sera enregistrée dans la base de données et sera transmise au concentrateur général. Enfin, ce dernier communiquera l'information au concentrateur choisi et il pourra renseigner au module capteur que ce dernier devra communiquer ses informations seulement à lui.

2. Planning prévisionnel

Le planning prévisionnel a été établi de telle sorte à anticiper sur des points où nous sommes susceptibles de rencontrer des difficultés.

| | Partie réseau de capteurs | Partie application web | Communication entre les deux parties |
|--------------------------------|--|---|---|
| Début janvier (rentrée) | Communication RF | Finalisation de l'affichage des informations | |
| Mi-janvier | Envoi et réception des trames Gestion de l'initialisation des modules capteurs | | |
| Fin janvier | Communication entre modules capteurs et concentrateurs | | |
| Début février | Finalisation des communications réseau avec implémentation de la topologie <i>Cluster-Tree</i> | Interface homme-machine permettant de créer des plans de parking | |
| Mi-février | | Gestion du design (Affichage d'une carte géographique avec les informations sur le parking) | Tests avec le prototype complet en situation réelle |
| Fin février | | | Améliorations en fonction des tests Finalisation du projet |

Conclusion

Lors de cette première moitié de projet, nous avons principalement étudié le sujet et recherché les solutions appropriées. Ces premiers mois étaient donc en grande partie axés sur la théorie ; en effet, une fois les solutions trouvées, nous nous sommes, dans chaque partie traitée, familiarisés avec le matériel et les outils nécessaires au projet. Dans la deuxième moitié, des premiers tests concluants ont été réalisés mais il reste beaucoup de tâches à accomplir.

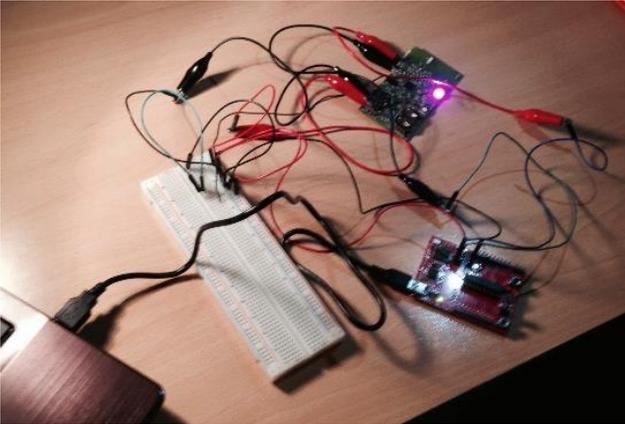
Cependant, avec les connaissances que nous avons acquises dans cette première phase de projet et notre capacité de faire des choix justes croissant, nous devrions avancer plus vite pour la suite.

Annexes

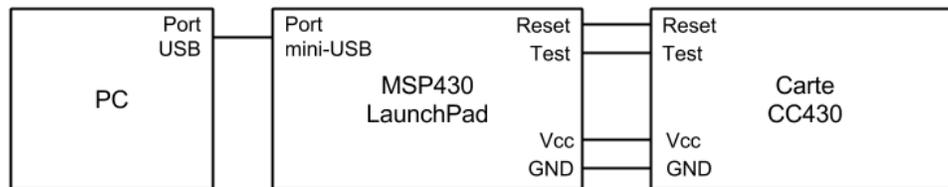
Sommaire

| | |
|--|-----------|
| SOMMAIRE | 29 |
| ANNEXE 1 : Branchement du MSP430 LaunchPad afin de programmer une carte CC430 | 30 |
| ANNEXE 2 : Présentation d'un protocole de communication | 31 |

Annexe 1 : Branchement du MSP430 LaunchPad afin de programmer une carte CC430

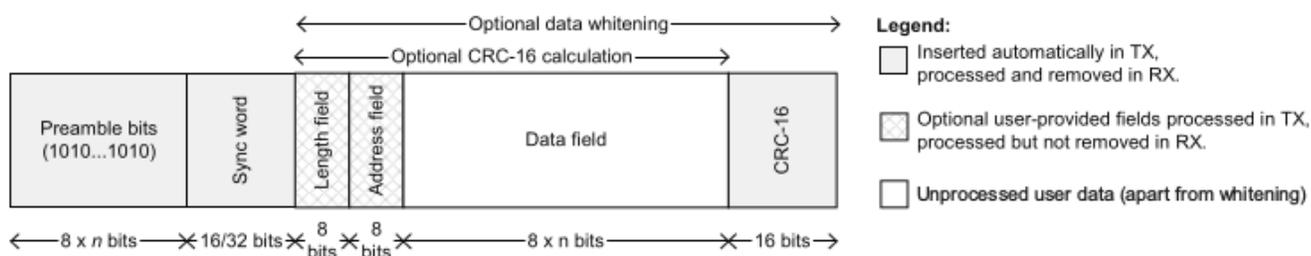


Les branchements réalisés entre le LaunchPad et les cartes CC430 est comme indiqué sur la photo. Comme le LaunchPad est utilisé uniquement en tant que programmeur, il n'est pas utile de devoir brancher toutes les cartes CC430 simultanément. Le schéma présente les différents branchements à faire afin de programmer la carte avec un PC.



Annexe 2 : Présentation d'un protocole de communication

Dans les cartes CC430, des fonctionnalités permettant de traiter les protocoles de communication à base de trames à envoyer (et recevoir) existent déjà. Une trame est une succession de données ordonnées de manière à ce que les entités qui communiquent connaissent la provenance, la quantité d'informations à traiter, le but de l'échange et les informations à traiter. Les trames sont paramétrables, et suivent le schéma suivant :



Source: *CC430 User's Guide*, de Texas Instruments

Ce que nous avons dû alors définir dans la trame que nous allons utiliser sont les données que nous souhaitons échanger. Dans les données, nous précisons :

- le type de données :
 - initialisation du capteur ;
 - information sur l'état du capteur ;
 - *acknowledgement*.
- l'identifiant du capteur ;
- l'information à transmettre.