

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

Département Electronique

Mémoire de projet de fin d'études  
pour l'obtention du diplôme d'ingénieur d'état en électronique

# Conception et développement d'une solution Courants Porteurs en Ligne dédiée au Smart Lighting

M Khaled BOUBEKEUR  
M Youva MANI

Sous la direction de M .Rabah SADOUN Pr  
Présenté et soutenu publiquement le (30/06/2019)

**Composition du jury :**

Président	M .Mourad ADNANE,	Dr ENP
Promoteur	M .Rabah SADOUN,	Pr ENP
Examineur	Mme .Nesrine BOUADJENEK,	Dr ENP

ENP 2019



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Ecole Nationale Polytechnique



المدرسة الوطنية المتعددة التقنيات  
Ecole Nationale Polytechnique

Département Electronique

Mémoire de projet de fin d'études  
pour l'obtention du diplôme d'ingénieur d'état en électronique

# Conception et développement d'une solution Courants Porteurs en Ligne dédiée au Smart Lighting

M Khaled BOUBEKEUR  
M Youva MANI

Sous la direction de M .Rabah SADOUN Pr  
Présenté et soutenu publiquement le (30/06/2019)

**Composition du jury :**

Président	M .Mourad ADNANE,	Dr ENP
Promoteur	M .Rabah SADOUN,	Pr ENP
Examineur	Mme .Nesrine BOUADJENEK,	Dr ENP

ENP 2019

## *Dédicace*

*On dédie ce travail à nos chers parents,*

*A nos familles,*

*A nos frères et soeurs,*

*A nos amis,*

*Et à toute personne ayant contribué de près ou de loin  
d'une manière directe ou indirecte à l'avancement  
scientifique.*

# *Remerciements*

*Nous souhaitons avant tout remercier DIEU tout puissant de nous avoir donné le courage et la possibilité de terminer notre parcours académique dans les conditions les plus favorables qui soit.*

*Nous remercions nos parents, sans qui rien ne serait possible, et nous tenons à préciser qu'aucun mot ne pourrait exprimer notre gratitude envers eux, nous leurs somme reconnaissant de nous avoir soutenus, et surtout 'supportés' durant toute notre vie.*

*Nous tenons à remercier tous nos ami(e)s, qui sans le savoir nous ont permis de surmonter les moments de doutes et d'incertitudes, en nous ayant montré sympathie et compréhension.*

*Nos vifs remerciement à notre encadrant, Monsieur **Rabah SADOUN**, de nous avoir aidé dans notre travail, et surtout d'avoir cru en nos compétences.*

## ملخص

إن التقدم التكنولوجي الذي نشهده اليوم، خاصة إنترنت الأشياء، يدفع الباحثين والعلماء إلى تطوير حلول أكثر تعقيدا تقدم خدمات تنافسية بتكلفة أقل مع تقليص أوقات التنفيذ واحترام البيئة.

من هذا المنطق، نقترح إنشاء شبكة من الأشياء المتصلة، بناءً على تقنية التيار الحامل التي تعتمد على التركيبات الكهربائية كوسيلة نقل بهدف أتمتة الإضاءة في المدن الذكية. وبذلك الجمع بين المكاسب الاقتصادية والبيئية مع تقديم العديد من الخدمات.

**الكلمات الدالة :** الإضاءة الذكية، إنترنت الأشياء، التيار الحامل.

## ABSTRACT

The technological progress we are witnessing today, particularly the Internet of Things, is driving researchers and scientists to develop increasingly complex solutions that offer competitive services at lower cost, with reduced implementation times and respect for the environment.

In this logic, we propose to set up a network of connected objects, based on Power Line Communication technology as a transmission medium, with the aim of smart lighting in smart cities. Combining economic, ecological and implementation time gains, while offering a multitude of services.

**Key words :** Power Line Communication, Internet of Things, Smart Lighting.

## RESUME

L'avancement technologique auquel nous assistons aujourd'hui, notamment l'Internet des objets, pousse les chercheurs et les scientifiques à développer des solutions de plus en plus complexes offrant des services compétitifs à moindre coût, avec des temps de mise en place réduits et le respect de l'environnement.

Dans cette logique, nous nous proposons de mettre en place un réseau d'objets connectés, se basant sur la technologie des Courants Porteurs en Ligne comme support de transmission, ayant comme finalité l'automatisation des éclairages (éclairage intelligent) dans les villes intelligentes. Alliant ainsi gain économique, écologique et temps de mise en œuvre, tout en offrant une multitude de services.

**Mots clés :** Courants Porteurs en Ligne, Internet des objets, éclairage intelligent.

# Table des matières

Liste des tableaux

Liste des figures

Liste des abréviations

Introduction générale 14

**I État de l’art et concepts 16**

**1 CPL pour IoT 17**

1.1 Introduction . . . . . 18

1.2 Internet des Objets . . . . . 18

1.2.1 Objets Connectés (OC) . . . . . 18

1.2.2 Ecosystème IoT . . . . . 19

1.2.3 Normalisation des modèles d’architecture de l’IoT . . . . . 20

1.2.3.1 Architecture IEEE . . . . . 21

1.2.3.2 Architecture CISCO . . . . . 22

1.2.4 Supports de transmission . . . . . 23

1.3 Courants porteurs en ligne . . . . . 23

1.3.1 Présentation . . . . . 23

1.3.2 CPL : Bref historique . . . . . 24

1.3.3 Concepts théoriques . . . . . 25

1.3.3.1 Classification fréquentielle des communications par CPL 25

1.3.3.2 Propriétés et généralités du canal CPL-BE . . . . . 26

1.3.3.3 Fonction de transfert du canal CPL . . . . . 27

1.3.4 Bruit sur le canal CPL-BE . . . . . 31

1.3.5 Techniques de modulation les plus répandues dans l’usage du  
CPL . . . . . 32

1.3.5.1 Modulation OFDM . . . . . 32

1.3.5.2 Modulation à étalement de spectre (Spread Spectrum) 33

1.3.6 CPL Indoor et Outdoor . . . . . 35

1.3.6.1 CPL Indoor . . . . . 35

1.3.6.2 CPL Outdoor . . . . . 36

1.3.6.3 Cadre juridique . . . . . 37

1.3.7	Normes protocolaires . . . . .	37
1.3.7.1	Introduction . . . . .	37
1.3.7.2	Tests et résultats expérimentaux . . . . .	37
1.3.7.3	Comparaison . . . . .	38
1.4	Conclusion . . . . .	39
<b>2</b>	<b>Pile protocolaire G3-PLC</b>	<b>40</b>
2.1	Introduction . . . . .	41
2.2	Modèle en couches – Positionnement par rapport au modèle OSI . . . .	41
2.3	Spécification de la couche physique . . . . .	42
2.3.1	Architecture en bloc de la couche PHYSIQUE . . . . .	42
2.3.1.1	Brouilleur (Scrambler) . . . . .	43
2.3.1.2	Codage FEC (Forward Error Correction) . . . . .	43
2.3.1.3	Bloc de mappage . . . . .	45
2.3.1.4	Génération de symboles OFDM (IFFT et CP) . . . . .	45
2.3.1.5	Fenêtrage . . . . .	45
2.3.1.6	Mappage de tonalités adaptatif et contrôle de puissance d’émission . . . . .	46
2.3.1.7	Passage par un transformateur . . . . .	46
2.3.1.8	Détection de la phase . . . . .	47
2.4	Couche liaison . . . . .	48
2.4.1	Structure d’une trame G3-PLC . . . . .	48
2.5	Couche d’adaptation 6LoWPAN . . . . .	49
2.5.1	IPv6 . . . . .	50
2.5.1.1	En-tête IPv6 . . . . .	50
2.5.1.2	Adressage IPv6 . . . . .	51
2.5.2	6LoWPAN . . . . .	52
2.6	Comparaison de solutions dédiées à l’implémentation de G3-PLC . . .	55
2.7	Conclusion . . . . .	56
<b>3</b>	<b>Circuit PL360</b>	<b>57</b>
3.1	Introduction . . . . .	58
3.1.1	Caractéristiques du PL360 . . . . .	58
3.2	Architecture système d’une application PL360 . . . . .	59
3.3	Interfaces de communication . . . . .	60
3.3.1	Serial peripheral interface (SPI) . . . . .	60
3.3.1.1	Description . . . . .	60
3.3.1.2	Caractéristiques intégrées . . . . .	61
3.3.2	Circuits de couplage CPL . . . . .	61
3.3.2.1	Étage de transmission . . . . .	62
3.3.2.2	Étage de filtrage . . . . .	62
3.3.2.3	Étage de couplage . . . . .	62
3.3.2.4	Étage de réception . . . . .	62
3.3.3	Chemin de transmission . . . . .	63
3.3.3.1	Description . . . . .	63
3.3.3.2	Caractéristiques intégrées . . . . .	63
3.3.4	Chemin de réception . . . . .	64



3.3.4.1	Description . . . . .	64
3.3.4.2	Caractéristiques intégrées . . . . .	64
3.3.4.3	Description fonctionnelle . . . . .	64
3.3.4.4	Dumping Buffer . . . . .	66
3.4	Bootloader . . . . .	66
3.4.1	Description . . . . .	66
3.4.2	Caractéristiques intégrées . . . . .	67
3.4.3	Description fonctionnelle . . . . .	67
3.5	Implémentation de G3-PLC sur PL360 . . . . .	68
3.5.1	Architecture générale . . . . .	69
3.5.2	Couche PHY . . . . .	69
3.5.3	Physical Abstraction Layer (PAL) . . . . .	69
3.5.4	Couche MAC . . . . .	70
3.5.5	Couche ADP (Adaptation Sublayer) . . . . .	70
3.5.6	Couches de sérialisation et interface série universelle (USI) . . . . .	70
3.5.7	Couche applicative . . . . .	71
3.6	Conclusion . . . . .	71

## II Implémentation d'une solution pour le Smart Lighting 72

<b>4</b>	<b>Conception et Développement</b>	<b>73</b>
4.1	Introduction . . . . .	74
4.2	Ecosystème IoT pour le Smartlighting . . . . .	74
4.3	Spécifications du protocole applicatif . . . . .	76
4.3.1	Principe Publication/Abonnement . . . . .	76
4.3.2	Principe de topic (sujet) . . . . .	77
4.3.3	Langage de description des messages . . . . .	77
4.3.4	Choix du protocole applicatif . . . . .	78
4.4	Protocole MQTT . . . . .	79
4.4.1	Format des paquets . . . . .	79
4.4.2	Architecture système . . . . .	82
4.4.2.1	Broker . . . . .	82
4.4.2.2	Client . . . . .	82
4.5	Outils de développement . . . . .	83
4.5.1	Cartes d'évaluation . . . . .	83
4.5.1.1	Kit d'évaluation PL360MB . . . . .	83
4.5.1.2	SAM4CMS16C . . . . .	83
4.5.1.3	Carte SAMA5D4 . . . . .	85
4.5.2	Kits de développement software . . . . .	85
4.5.2.1	Pour le nœud périphérique . . . . .	85
4.5.2.2	Pour la passerelle . . . . .	86
4.6	Flot de développement . . . . .	88
4.7	Conception/développement du nœud périphérique . . . . .	88
4.7.1	Flot de développement . . . . .	89
4.7.2	Conception/Développement en utilisant la carte PL360EK . . . . .	89
4.7.2.1	Pile protocolaire . . . . .	89

4.7.2.2	Couche applicative . . . . .	90
4.7.2.3	Couche session . . . . .	92
4.7.2.4	Couche transport . . . . .	92
4.7.2.5	Couche réseau . . . . .	93
4.7.2.6	Couche Liaison . . . . .	94
4.7.3	Préparation et publication des messages . . . . .	95
4.8	Conception/Développement de la Passerelle . . . . .	97
4.8.1	Flot de développement . . . . .	97
4.8.2	Modèle de la couche applicative . . . . .	97
4.8.3	Système d'exploitation pour SAMA5D4 . . . . .	98
4.8.4	Couche Device Driver . . . . .	98
4.8.5	Device Tree (DT) . . . . .	99
4.8.6	Préparation de la carte PL360MB . . . . .	100
4.8.7	Test et validation du fonctionnement de la passerelle . . . . .	101
4.8.7.1	Approche de test . . . . .	102
4.8.7.2	Résultats et discussion . . . . .	103
4.9	Etude de la portabilité sur Raspberry Pi . . . . .	104
4.9.1	Intérêt . . . . .	104
4.9.2	Approche d'adaptation . . . . .	104
4.9.3	Contraintes . . . . .	105
4.9.4	Solutions retenues . . . . .	105
4.10	Conclusion . . . . .	106
	<b>Conclusion générale</b>	<b>107</b>
	<b>Bibliographie</b>	<b>109</b>

# Liste des tableaux

1.1	Ecosystème IoT . . . . .	20
1.2	Paramètres de la fonction de transfert hybride . . . . .	31
1.3	Résultats de tests en l'absence d'interférences . . . . .	37
1.4	Résultats de tests en présence d'interférences . . . . .	38
1.5	Comparaison théorique entre G3-PLC et PRIME . . . . .	38
2.1	Composition d'une trame G3-PLC . . . . .	48
2.2	Comparatif des modems CPL implémentant G3-PLC . . . . .	55
3.1	Description de la trame de base . . . . .	68
4.1	Table de valeurs QOS . . . . .	80
4.2	Types de messages MQTT . . . . .	81

# Table des figures

1.1	Chaîne complète d'un système IoT . . . . .	21
1.2	Architecture IoT selon IEEE . . . . .	22
1.3	Les différentes couches de l'Internet des Objets selon Cisco . . . . .	22
1.4	Illustration de la segmentation du réseau . . . . .	29
1.5	Bruit sur le canal CPL-BE . . . . .	32
1.6	Technique de modulation OFDM . . . . .	33
1.7	Exemple d'étalement d'un signal de données binaires . . . . .	35
1.8	Réseau CPL INDOOR . . . . .	36
1.9	Réseau CPL OUTDOOR . . . . .	36
2.1	Modèle en couche . . . . .	41
2.2	Architecture en bloc de la couche PHYSIQUE . . . . .	42
2.3	Brouilleur de données . . . . .	43
2.4	Codeur convolutif . . . . .	44
2.5	Détecteur de passage par zéro . . . . .	47
2.6	Structure d'une trame G3-PLC . . . . .	49
2.7	En-tête IPv6 . . . . .	50
2.8	Pile protocolaire . . . . .	52
2.9	Trame 802.15.4 . . . . .	53
3.1	Circuit PL360 . . . . .	58
3.2	Diagramme en bloc du PL360 . . . . .	59
3.3	Schéma d'application du PL360 . . . . .	60
3.4	Circuit de couplage CPL . . . . .	61
3.5	Diagramme en block du chemin de transmission . . . . .	63
3.6	Diagramme en bloc du chemin de réception(1) . . . . .	64
3.7	Diagramme en bloc du chemin de réception(2) . . . . .	65
3.8	Passage par zéro du signal . . . . .	66
3.9	Schéma fonctionnel . . . . .	67
3.10	Schéma fonctionnel transfert de données . . . . .	68
3.11	Schéma synoptique de la pile G3-PLC de Microchip . . . . .	69
4.1	Réseau CPL Internet . . . . .	74
4.2	Ecosystème IoT . . . . .	75
4.3	Format des paquets MQTT . . . . .	79
4.4	Fixed Header structure . . . . .	79

4.5	Architecture système MQTT . . . . .	82
4.6	Architecture de la carte PL360MB . . . . .	83
4.7	Diagramme en Bloc de la série SAM4CMS16C . . . . .	84
4.8	Carte SAMA5D4 Xplained Ultra . . . . .	85
4.9	Interface de configuration . . . . .	87
4.10	Interface de configuration du noyau . . . . .	87
4.11	Modèle en couche de Cyclone TCP . . . . .	90
4.12	Machine d'état du noeud périphérique . . . . .	90
4.13	Publication d'un message . . . . .	96
4.14	Modèle de la couche applicative . . . . .	97
4.15	Déclaration des signaux de contrôle du PL360 sur SAMA5D4 . . . . .	99
4.16	Réservation d'un pool dans la securam . . . . .	100
4.17	Modifications du connecteur XPLAINED PRO pour une connexion di- recte au PL360 . . . . .	101
4.18	Etat initial, après modification . . . . .	101
4.19	Réseau test . . . . .	102
4.20	Approche de validation du fonctionnement de la passerelle . . . . .	102
4.21	PLC Sniffer . . . . .	103
4.22	Détection trame G3-PLC . . . . .	104
4.23	Déclaration signaux de commande du PL360 sur Raspberry . . . . .	105

# Liste des abréviations

<b>6LoWPAN</b>	: IPv6 LoW Power wireless Area Networks
<b>ADC</b>	: Analog to Digital Converter
<b>AES</b>	: Advanced Encryption Standard
<b>AGC</b>	: Automatic Gain Control
<b>ARIB</b>	: Association of Radio Industries and Businesses
<b>BT</b>	: Basse tension
<b>CENELEC</b>	: Comité Européen de Normalisation en Electrotechnique
<b>CP</b>	: Cyclic Prefix
<b>CPL</b>	: Courants Porteurs en Ligne
<b>CPL-BE</b>	: Courants Porteur en Ligne à Bande Etroite
<b>CPL-BUE</b>	: Courants Porteur en Ligne à Bande Ultra Etroite
<b>CPL-LB</b>	: Courants Porteur en Ligne à Large Bande
<b>CSMA/CA</b>	: Carrier Sense Multiple Access with Collision Avoidance
<b>UNEP</b>	: United Nations Environment Programm
<b>FCC</b>	: Federal Communications Commission
<b>FEC</b>	: Forward Error Correction
<b>FCH</b>	: Frame Control Header
<b>FSC</b>	: Frame Control Structure
<b>FFT</b>	: Fast Fourier Transform
<b>HAN</b>	: Home Area Network
<b>HT</b>	: Haute Tension
<b>IEEE</b>	: Institute of Electrical and Electronics Engineers
<b>IFFT</b>	: Inverse Fast Fourier Transform
<b>IPv6</b>	: Internet Protocol version 6

<b>MAC</b>	: Media Access Control
<b>MT</b>	: Moyenne Tension
<b>MTU</b>	: Maximum Transmit Unit
<b>NAN</b>	: Neighbour Area Network
<b>OC</b>	: Objets Connectés
<b>OFDM</b>	: Orthogonal Frequency-Division Multiplexing
<b>SG</b>	: Smart Grid
<b>SNR</b>	: Signal to Noise Ratio
<b>SPI</b>	: Serial Peripheral Interface
<b>TEM</b>	: Transverse Electro Magnétique

# Introduction générale

Parmi les nombreuses problématiques qui se posent dans la gestion des villes intelligentes, la supervision des installations lampadaire, qui par leur forte consommation énergétique (15 % de la consommation d'électricité mondiale selon l'UNEP) induisent des complications majeures (enjeux financier, pollution . . .). Ainsi il est impératif de développer de nouvelles technologies palliant toutes ces contraintes, sous condition de respecter l'environnement, et si possible, à des coûts réduits.

Par conséquent, nous proposons l'étude et la mise en œuvre d'un réseau d'objets connectés dédié au smart lighting, qui aura comme support de transmission les installations électrique préexistantes. Avec, la conception et le développement des différents éléments composant la chaîne complète. Partant d'un nœud périphérique qui sera installé sur les lampadaires et qui sera doté de toutes les ressources nécessaires à la collecte et transmission de données directement sur les lignes électrique. Pour ensuite atteindre une passerelle qui fera le lien entre les nœuds périphériques et internet. Permettant donc l'envoi des données recueillies vers le cloud. Répondant ainsi à la contrainte de gestion des installations d'éclairage, sans avoir à redévelopper ou réinstaller une nouvelle infrastructure pour la transmission des données.

Ainsi ce mémoire comporte quatre chapitres. Les trois premiers exposent les principes de base des objets connectés, et des transmissions sur courant porteur en ligne, avec un dernier chapitre qui montre le flot de conception et de développement de toute la chaîne. Notamment :

**Chapitre 1** : Une brève présentation de la notion d'objets connectés, avec une initiation aux CPL (canal de transmission, bruit, standards de transmission).



**Chapitre 2 :** Présentation de la norme protocolaire G3-PLC adoptée pour la transmission des données sur le canal CPL.

**Chapitre 3 :** Présentation des caractéristiques du circuit PL360 choisi pour la mise en place d'une connexion sur le réseau électrique.

**Chapitre 4 :** Mise en évidence des différentes parties permettant l'implémentation du système complet, partant du nœud périphérique, passant par le réseau CPL jusqu'à la passerelle.

# Première partie

## État de l'art et concepts

# Chapitre 1

## CPL pour IoT

## 1.1 Introduction

L'internet des objets est la 4<sup>e</sup> révolution technologique dans le domaine de l'informatique et des télécommunications. L'IoT fait référence à une variété d'équipements et de systèmes d'informations, tels que les réseaux de capteurs, les dispositifs de lecture (RFID, code à barres), de systèmes de localisation et de communication courte portée basés sur la communication machine à machine (M2M), à travers le réseau Internet pour former un réseau plus grand et plus intelligent.

Cette révolution est basée sur une évolution constante de l'Internet, des technologies, des logiciels, des protocoles de communication et des capteurs embarqués qui ne cessent d'être améliorés, des objets physiques de plus en plus intelligents et capables de fournir des informations et de percevoir en temps réel leurs environnement.

## 1.2 Internet des Objets

Le « cluster des projets européens de recherche sur l'internet des objets » définit l'internet des objets comme : « une infrastructure dynamique d'un réseau global. Ce réseau global a des capacités d'auto-configuration basée sur des standards et des protocoles de communication interopérables. Dans ce réseau, les objets physiques et virtuels ont des identités, des attributs physiques, des personnalités virtuelles et des interfaces intelligentes, et ils sont intégrés au réseau d'une façon transparente ».

Cette définition met en évidence les deux principaux atouts de l'internet des objets (temporel et spatial), en d'autres termes une connexion peut être établie de n'importe où et à n'importe quel moment à travers ces objets connectés pour éventuellement, la surveillance, la commande ou bien la collectes de données (tout peut se connecter à n'importe quel moment et de n'importe où).

### 1.2.1 Objets Connectés (OC)

Dispositif matériel permettant l'envoi et la réception de données (simple ou complexe) se basant sur les technologies et protocoles de communication préexistant (Wifi,

Bluetooth, CPL, etc. . .) dont le principal but étant la surveillance, le contrôle et la collecte de données.

Un objet connecté dispose de ressources matérielles et logicielles lui permettant d'interagir avec le monde physique de manière indépendante, toutefois certaines contraintes sont à prendre en considération tel que la mémoire, la bande passante, la consommation énergétique, etc.

Un OC à trois éléments clés :

- Les données produites ou reçues (stockées ou transmises).
- Les algorithmes pour traiter ces données.
- L'écosystème dans lequel il va interagir et s'intégrer.

### 1.2.2 Ecosystème IoT

Les OC sont au cœur de l'IoT, mais il est important de pouvoir connecter l'ensemble de ces objets, les faire échanger des informations et interagir au sein d'un environnement. La mise en place d'un écosystème IoT passe globalement par les étapes suivantes :

- Identification
- Installation de capteurs
- Connexion des objets entre eux
- Intégration
- Mise en réseau

Identifier	Capter	Connecter	Intégrer	Mettre en réseau
Rendre possible l'identification de chaque élément connecté.	Mise en place de dispositif nous rapprochant du monde réel. Les fonctions de base des objets (le capteur de température pour la thermométrie par exemple).	Établir une connexion entre tous les objets afin qu'ils puissent dialoguer et s'échanger des données.	Disposer d'un moyen de communication rattachant les objets au monde virtuel.	Relier les objets et leurs données au monde informatique via un réseau (internet par exemple).
IPv4, IPv6, 6LoWPAN	MEMS, RF MEMS, NEMS	SigFox, LoRa...	RFID, NFC, Bluetooth, Bluetooth LE, ZigBee, WIFI, CPL, réseaux cellulaires,	CoAP, MQTT, AllJoyn, REST http ...

TABLE 1.1: Ecosystème IoT

### 1.2.3 Normalisation des modèles d'architecture de l'IoT

Vu le développement rapide de l'IoT, il devenait nécessaire d'avoir une architecture de référence qui permettrait d'uniformiser la conception des systèmes et favoriserait l'interopérabilité et la communication entre les différents écosystèmes de l'IoT (la figure 1.1 présente la chaîne d'un système IoT/M to M).

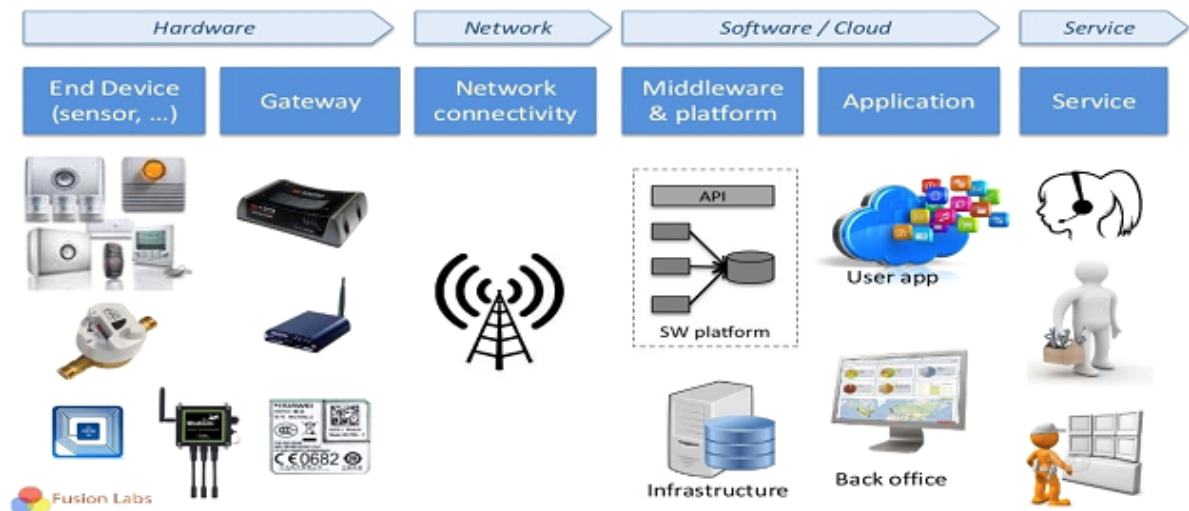


FIGURE 1.1: Chaîne complète d'un système IoT

### 1.2.3.1 Architecture IEEE

L'organisme de standardisation IEEE a proposé un modèle à trois niveaux qui prend en compte les relations, les interactions et les éléments d'architectures commun pour divers domaines d'application :

1. **Applications** : concerne les applications et les services proposés aux clients.
2. **Cloud Computing** : concerne les plateformes de services à qui sont destinées les données. Ce niveau permet d'établir le lien entre les capteurs, le réseau de plateformes et le logiciel de traitement de données.
3. **Réseaux de capteurs** : le niveau le plus bas correspond aux capteurs et à la communication entre eux (le Machine-to-Machine). Il s'agit d'un réseau de capteurs qui génèrent les données.

Ce modèle est dit « cloud-centric » du fait qu'il est en grande partie basé sur le cloud. L'IEEE considère le cloud Computing comme élément central pour le développement de l'IoT.

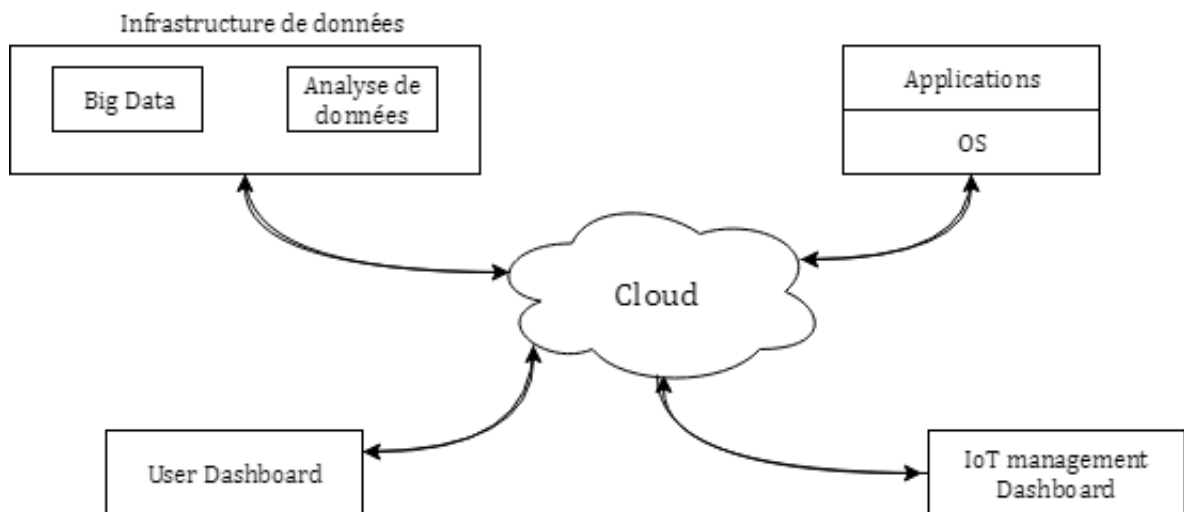


FIGURE 1.2: Architecture IoT selon IEEE

### 1.2.3.2 Architecture CISCO

D'autres entreprises ont proposé des architectures de couches superposées comme l'entreprise américaine Cisco. En octobre 2013, Jim Green présente « Building the Internet of Things », le modèle envisagé par son entreprise pour l'IoT est composé de 7 couches (figure 1.3).

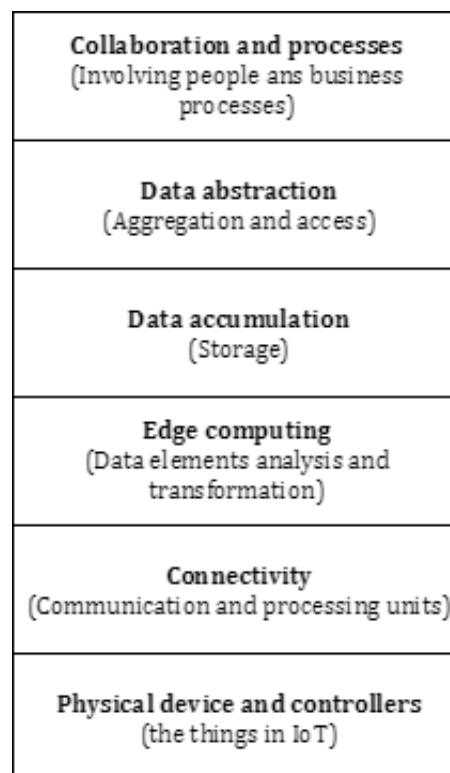


FIGURE 1.3: Les différentes couches de l'Internet des Objets selon Cisco



Ces modèles montrent l'engouement des entreprises pour les développements des écosystèmes IoT ouvert et interopérable pour être acceptés par les acteurs des marchés.

Malgré ces architectures, il reste des efforts à faire pour proposer un modèle de référence globale qui prend en compte les spécificités de l'IoT.

#### **1.2.4 Supports de transmission**

Comme indiqué dans le tableau 1.1 de nombreux supports de transmissions peuvent être utilisés pour la mise en place d'un réseau d'objets connectés (Wifi, Bluetooth, réseau cellulaire, CPL ...), ces dernières années les chercheurs et les industriels se sont de plus en plus intéressés aux CPL pour la mise en œuvre d'infrastructures IoT, ce qui semble légitime vu le déploiement des réseaux électriques en INDOOR qu'en OUTDOOR, et du gain non seulement économique qu'écologique qu'apporte cette technologie, dans ce projet nous proposons l'installation d'un réseau d'objets connectés à travers le support CPL.

### **1.3 Courants porteurs en ligne**

Comme indiqué dans le paragraphe précédent ce mémoire propose la mise en place d'un réseau d'objets connectés via le réseau électrique, ces derniers n'ayant pas besoin de débits trop élevés et de bandes large nous nous intéressons ici aux techniques de transmission CPL à bandes étroites (CPL-BE).

#### **1.3.1 Présentation**

Les CPL (Courants Porteurs en Ligne) sont une technologie de communication, qui utilise le réseau électrique pour fournir des services de télécommunications.

Le principe consiste à superposer au courant électrique 50-60 Hz un signal de plus haute fréquence de faible énergie portant de l'information, ce qui fait qu'elle réutilise les installations déjà existantes dont le principal but est de fournir une alimentation électrique pour construire un réseau de communication.

### 1.3.2 CPL : Bref historique

La technique des CPL n'est pas récente dans son principe. Dès 1838, en Angleterre, Edward Davy a proposé une solution permettant de mesurer à distance les niveaux de batterie des sites éloignés du système télégraphique entre Londres et Liverpool. En 1897, il présentait le premier brevet (British Patent N : 24833) d'une technique de mesure à distance des compteurs du réseau électrique communiquant sur les câbles électriques.

Appelés Ripple Control, les premiers systèmes CPL ont été élaborés puis déployés sur les réseaux électriques moyenne tension et basse tension en 1950. La fréquence porteuse était alors comprise entre 100 Hz et 1 kHz. Il s'agissait d'établir des communications monodirectionnelles via des signaux de commande pour l'allumage et l'extinction à distance des éclairages publics ou encore pour des changements tarifaires.

Les premiers systèmes industriels sont apparus en France en 1960 sous le nom de Pulsadis. Les puissances mises en jeu avoisinent la centaine de kilovoltampères (kVa).

Ce n'est qu'ensuite qu'apparurent les premiers systèmes CPL de la bande dite CE-NELEC, s'étendant de 3 à 148,5 kHz et permettant des communications bidirectionnelles sur le réseau électrique BT (basse tension) afin, par exemple, de pratiquer des relevés de compteurs (télé relève), ainsi que bon nombre d'applications relevant du domaine de la domotique (alarme d'intrusion, détection d'incendie, détection fuite de gaz, etc.). Les puissances injectées étaient beaucoup moins importantes que les précédentes, puisque réduites à des niveaux de l'ordre d'une centaine de milliwatts.

L'expression « courants porteurs en ligne », communément abrégée CPL, est apparue à la fin de la Seconde Guerre mondiale, en 1945. À l'époque, beaucoup de lignes téléphoniques et électriques étaient détruites, mais il restait davantage de lignes électriques d'infrastructure que de lignes téléphoniques. Pour des besoins de communication, des systèmes ont été conçus afin de transmettre des données sur les câbles

haute tension ou moyenne tension, en s'inspirant des télé-relèves déjà effectuées sur les lignes électriques.

L'intérêt pour ce médium de communication s'est accru dans les années 80 et s'est intensifié surtout pendant les années 90. Et plus récemment depuis 2007 on assiste à la montée des technologies CPL multi-porteuses à bande étroite pour des applications smart grid.

### 1.3.3 Concepts théoriques

La modélisation du canal de transmission étant essentielle pour l'élaboration de systèmes de communications fiables et optimaux, dans ce qui suit nous allons présenter sans trop de détails quelques modèles du canal CPL-BE qui serviront au calcul de la fonction de transfert, une succincte présentation des bruits rencontrés sur la canal CPL, ainsi que la classification des plages de fréquences utilisées pour les communications CPL, sans omettre les principales propriétés de ce derniers, avec une introduction aux standards G3-PLC et PRIME utilisé dans les liaison CPL-BE.

#### 1.3.3.1 Classification fréquentielle des communications par CPL

Les réseaux CPL sont généralement classés en fonction des plages de fréquences de fonctionnement. Ainsi nous distinguons :

- Les Courant Porteur en Ligne à Bande Ultra Étroite (CPL-BUE) qui opèrent dans la bande de fréquence [0.3-3] kHz et fournissent de l'ordre de 100 bits/s sur de longues distances (plus de 150 km). Elles offrent un débit insuffisant pour les futures communications Smart Grid (SG).
- Les CPL-BE (Bande Étroite) qui utilisent la gamme de fréquences [3-500] kHz. En fonction des pays, les organismes de normalisation attribuent des bandes différentes. La bande en dessous de 150 kHz étant la bande CENELEC (Comité Européen de Normalisation en Électrotechnique) en Europe. Pour les États-Unis, la commission fédérale des communications (Federal Communications Commission (FCC)) alloue la bande de 10 à 490 kHz. Pour le Japon, l'association japonaise des industries et entreprises de radio (Association of Radio

Industries and Businesses (ARIB)) utilise la bande [10-450] kHz. Et enfin la Chine utilise la bande [3-500] kHz.

Les CPL-BE sont en mesure de livrer quelques kbit/s en utilisant les communications mono-porteuses, ou plusieurs centaines de kbit/s en utilisant des communications multi-porteuses. Les CPL-BE et à haut débit ont suscité énormément d'intérêt en tant que moyen de communication pour les applications de télémétrie et pour les SG en général.

- Pour les CPL à Large Bande (CPL-LB) et à haut débit, la borne inférieure de la bande de fréquence est typiquement de 1 MHz, et la borne supérieure peut aller jusqu'à 30 ou 60 MHz. L'utilisation de fréquences supérieures, de l'ordre de centaines de MHz a fait l'objet d'étude et d'implémentation [1]. Nous distinguons également dans cette section plusieurs types de réseaux télécoms supportés par le réseau électrique en fonction de leur localisation et de leur taille :
  - Les réseaux privés domestiques Home Area Network (HAN) qui se situent derrière le compteur électrique et relient les appareils et les capteurs sur les lignes électriques à l'intérieur des bâtiments.
  - Les réseaux de quartier Neighbour Area Network (NAN) qui relient les compteurs intelligents aux concentrateurs de données du réseau électrique.
  - Les réseaux étendus Wide WAN qui sont en fait des liens de communication acheminant le trafic entre les concentrateurs de données et les centres de contrôle du réseau électrique.

### 1.3.3.2 Propriétés et généralités du canal CPL-BE

Avant d'entamer la modélisation du canal CPL il paraît opportun de rappeler les principales caractéristiques de ce dernier :

- Une propriété importante du canal CPL qui a été prouvée mathématiquement et expérimentalement [2] est sa symétrie.
- Une autre caractéristique du canal CPL est son comportement variant dans le temps. Les consommateurs d'électricité, dans le cadre d'une utilisation normale du réseau, et leurs appareils électriques commutent les charges d'une manière

incontrôlée et imprévisible, provoquant des fluctuations d'impédance soudaines et à grande dynamique. Ces changements entraînent des variations lentes du canal. En outre, le canal CPL présente des variations synchrones à la fréquence du secteur car les paramètres haute fréquence des appareils électriques qui sont connectés au réseau dépendent de l'amplitude instantanée de la tension du réseau. Le bruit injecté dans la ligne CPL par les appareils connectés au réseau dépend lui aussi de l'amplitude instantanée de la tension du réseau. Par conséquent, le bruit et la réponse fréquentielle du canal présentent un comportement cyclo-stationnaire en temps (périodiquement stationnaire).

- Les caractéristiques des canaux CPL-BE dépendent des pratiques de câblage, de la topologie et de la densité des charges connectées au réseau électrique. Ces paramètres varient d'un pays à un autre et pour un même pays ils changent en fonction du segment du réseau considéré (HT, MT ou BT), de l'environnement dans lequel l'électricité est distribuée (rural ou urbain, résidentiel ou industriel) et indépendamment de ces facteurs également avec le temps. Malheureusement, il n'existe pas de canaux de référence pour les communications CPL-BE dans la partie basse tension. Le canal CPL-BE peut être complètement décrit par sa fonction de transfert et du bruit additif qui l'affecte.

### 1.3.3.3 Fonction de transfert du canal CPL

Les effets de distorsion du canal comprennent les aspects usuels d'atténuation mais aussi de sélectivité fréquentielle à cause des désadaptations rencontrées sur les lignes de transmission. Dans cette section, nous introduisons trois approches populaires pour la modélisation de la fonction de transfert des canaux CPL.

#### **Modèle paramétrique « top-down » :**

Elle a d'abord été proposée par Phillips dans [3] et plus tard améliorée et validée par Zimmermann et Dostert dans [4]. L'idée ici étant de fournir une fonction paramétrique qui décrit tous les aspects du comportement du canal CPL : ses caractéristiques passe-bas, son atténuation, sa sélectivité fréquentielle.

La fonction de transfert paramétrique s'écrit :

$$H(f) = \sum_{i=0}^N g_i(f) \cdot e^{-\alpha(f)l_i} \cdot e^{-2j\pi f\tau_i} \quad (1.1)$$

Où  $N$  est le nombre de trajets dit distinguables,  $g_i$  est le gain complexe sur chaque trajet,  $l_i$  la longueur du  $i$ -ème trajet,  $\tau_i$  le délais sur le  $i$ -ème trajet et  $a(f)$  représente l'atténuation en fonction de la fréquence.

$$\alpha(f) = \alpha_0 + \alpha_1 \cdot f^k \quad (1.2)$$

Où  $\alpha_0$ ,  $\alpha_1$  et  $k$  sont les paramètres de l'atténuation,  $a(f)$  prend en compte les effets de peau et de perte diélectrique [4]. Le modèle nécessite des campagnes de mesures de la réponse du canal pour déterminer les paramètres  $N$ ,  $a(f)$  et  $(g_i, l_i, \tau_i)$ . Ce modèle nécessite beaucoup de paramètres, spécialement quand  $N$  est grand et est sensible à la taille de  $N$  en termes de précision. Par exemple, des valeurs de  $N$  comprises entre 15 et 44 sont nécessaires pour un ajustement d'un lien de 110m [4, 5]. Enfin, le modèle est peu lié à la réalité physique du réseau qui souvent peut être complexe. Et une fois défini il est peu flexible, en d'autres termes pour un réseau donné il est sensible à tout changement de topologie. Cependant, cette approche est utile lorsque la topologie du réseau n'est pas disponible.

### **Modélisation structurelle : « bottom-up »**

« Bottom-up » en anglais se traduit littéralement de bas en haut pour indiquer que la fonction de transfert s'obtient depuis les caractéristiques basses (physiques) du réseau électrique. Cette approche est basée sur la théorie des lignes de transmission à deux ou à plusieurs conducteurs. Celle-ci nécessite une connaissance de la structure complète du réseau (les paramètres linéiques des câbles électriques, la topologie du réseau, les impédances connectées au réseau...). La théorie des lignes de transmission permet l'analyse de la propagation d'ondes en milieu filaire. Elle se base sur la décomposition de la ligne en plusieurs éléments infinitésimaux décrits chacun par les équations des télégraphistes [6]. La théorie des lignes de transmission à deux conducteurs est incapable de décrire parfaitement tous les phénomènes physiques se produisant dans les câbles électriques dans la pratique (comme le couplage de mode).

Cependant, en première approche, elle donne un aperçu sur une grande majorité des caractéristiques du canal de propagation CPL [7].

Dans cette partie, nous décrivons la fonction de transfert des liens CPL en suivant la théorie des lignes de transmission à deux conducteurs. Pour une analyse complète des réseaux CPL suivant la théorie des lignes de transmission multi-conducteurs, voir [8]. Pour les fréquences considérées dans les CPL-BE, le mode principal se propageant sur les lignes électriques est le mode Transverse Electro Magnétique (TEM) ou quasi-TEM [6]. En fonction de la technique utilisée, la propagation des signaux sur les lignes de transmission peut être décrite par les paramètres S, le rapport de tension, les paramètres ABCD. L'approche du rapport de tension, permet de calculer la fonction de transfert  $H(f)$  comme étant le rapport entre la tension du récepteur  $V_{Rx}$  et la tension de l'émetteur  $V_{Tx}$ .

$$H(f) = \frac{V_{Rx}(f)}{V_{Tx}(f)} \quad (1.3)$$

La fonction de transfert entre un émetteur et un récepteur situés sur la ligne électrique se calcule de la façon suivante :

- D'abord il faut trouver le chemin (backbone) le plus court qui relie l'émetteur et le récepteur. Diviser la ligne en plusieurs segments élémentaires le long de ce chemin, de préférence au niveau des discontinuités du réseau,

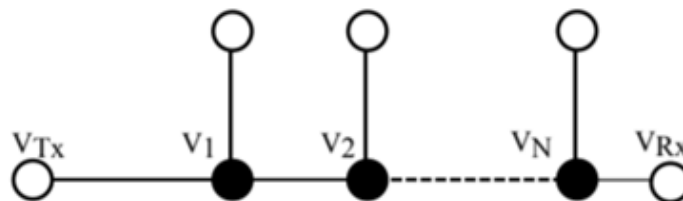


FIGURE 1.4: Illustration de la segmentation du réseau[7]

- Une seconde étape consiste à décrire la propagation des signaux dans un segment élémentaire du réseau électrique sur la base de la théorie des lignes de transmission. Chaque segment est supposé avoir des caractéristiques homogènes. On

note la fonction de transfert de chaque segment par :

$$H_n(f) = \frac{V_{n+1}(f)}{V_n(f)} \quad (1.4)$$

qui est le rapport entre la tension d'entrée et de sortie du segment. Les segments finaux sont ceux associés à l'émetteur et au récepteur ( $V_{n+1} = V_{Rx}$  et  $V_0 = V_{Tr}$ ).

- Enfin la fonction de transfert entre l'émetteur et le récepteur est calculée en mettant en cascade les différentes fonctions de transfert des segments élémentaires intermédiaires. La fonction de transfert totale peut s'écrire :

$$H_n(f) = \frac{V_{N+1}(f)}{V_0(f)} = \prod_{n=0}^N H_n(f) \quad (1.5)$$

L'avantage de cette approche est qu'elle permet de rendre compte de la corrélation des fonctions de transfert pour des nœuds (émetteurs ou récepteurs) situés sur la même ligne de transmission.

### Modélisation hybride

La forme paramétrique définie plus haut peut donner lieu à une description statistique du canal en affectant à ses paramètres des lois de distribution statistiques. Ceci a été surtout accompli pour les canaux CPL-LB [9, 10]. Dans un modèle hybride, on pourrait définir un ensemble de topologies réalistes qui peuvent être considérées comme représentatives de la majorité de ce qui peut être trouvé dans la réalité. Sur la base de ces topologies et pratiques courantes, et à l'aide de la modélisation structurelle, on peut définir un ensemble de fonctions de transfert pour les canaux CPL-BE dans la partie BT. Les lois de variations des paramètres de la fonction de transfert  $H(f)$  défini dans (1.1) sont données par l'organisme de standardisation IEEE dans [11] et définis comme :

- $l_i$  est une variable aléatoire gaussienne de moyenne  $l_a$  et de déviation standard  $l_s$  qui représente la longueur des trajets de propagation.
- $\nu$  représente la vitesse de propagation de l'onde.
- $a_0$  et  $a_1$  sont les paramètres d'atténuation qui dépendent des caractéristiques physiques de la ligne de transmission.



- $g_i$ , le gain sur le trajet  $i$ , est une variable aléatoire gaussienne de moyenne nulle et de variance 1.
- $N$  est le nombre de trajets significatifs. Les paramètres  $l_i, l_a, l_s, l_m$  sont présentés dans le tableau suivant, et sont uniquement pertinents pour les réseaux BT.

Paramètre	valeurs	description
$l_a$	1000	Trajet moyen entre Trx et Rx
$l_s$	400	Déviation standard des trajets entre Trx et Rx
$l_m$	100	Distance minimale des $l_i$
V	$3 \cdot \frac{10^4}{4}$	Vitesse de propagation de l'onde
K	1 ( $0.5 < k < 1$ )	Pente de l'atténuation en fonction de la fréquence
$a_0$	$e^{-3}$	Dépend des caractéristiques des lignes
$a_1$	$2.5 \cdot e^{-9}$	Dépend des caractéristiques des lignes
N	$5 < N < 50$	Nombre de trajets significatifs

TABLE 1.2: Paramètres de la fonction de transfert hybride

### 1.3.4 Bruit sur le canal CPL-BE

Le bruit sur les CPL-BE provient d'une part de la somme de plusieurs formes d'onde produites et émises sur les lignes par des appareils connectés au réseau électrique. Et d'autre part, du rayonnement d'autres sources opérant aux mêmes fréquences que les systèmes CPL-BE, qui ne sont pas connectés au réseau mais dont les émissions sont couplées aux systèmes CPL-BE. Le bruit résultant est non-blanc avec un contenu spectral qui présente une décroissance en fonction de la fréquence en raison de la diminution de la concentration des sources de bruit avec la fréquence [7]. De façon générale, le bruit affectant les communications CPL est classé en trois catégories comme présenté dans la figure ci-dessous : le bruit de fond coloré, le bruit impulsif et le bruit à bande étroite. Le bruit impulsif est généralement classé en trois catégories : le bruit impulsif périodique asynchrone à la fréquence du secteur principalement causé par les alimentations commutées. Le bruit impulsif périodique synchrone à la fréquence du réseau

causé par la commutation des diodes de redressement d'alimentations. Et le bruit impulsif asynchrone principalement causé par des événements de commutation.

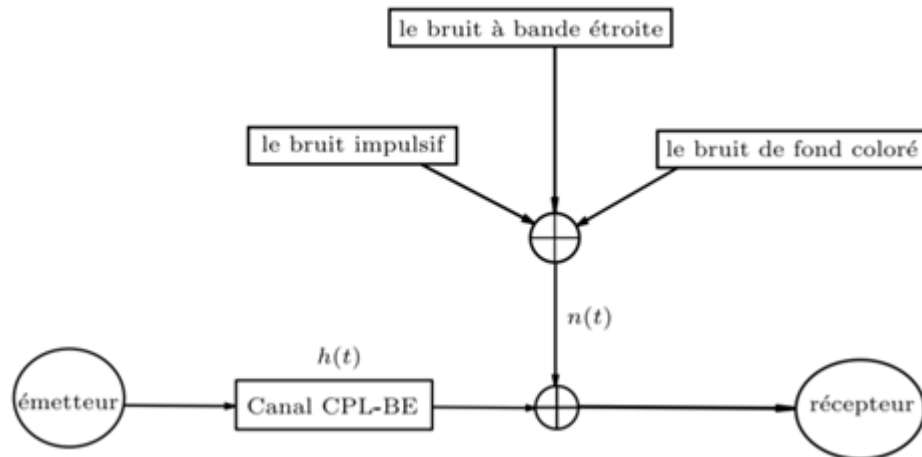


FIGURE 1.5: Bruit sur le canal CPL-BE [7]

### 1.3.5 Techniques de modulation les plus répandues dans l'usage du CPL

Le but de la modulation est de garantir un débit suffisant, tout en gardant des niveaux d'émissions les plus faibles possibles vu les contraintes qui sont imposées aux CPL en puissance d'émission.

Parmi les solutions actuelles, deux techniques ont montré de bonnes performances dans des environnements complexes. Il s'agit de la technique de modulation OFDM (Orthogonal Frequency Division Multiplexing) et la technique de modulation à étalement de spectre SS (Spread Spectrum).

Ces deux techniques utilisent les méthodes de modulation classiques qui consistent à modifier une caractéristique (amplitude, fréquence ou phase) d'une onde porteuse en fonction de la donnée à transmettre.

#### 1.3.5.1 Modulation OFDM

C'est la technique la plus répandue pour les systèmes CPL. Elle a de ce fait été retenue par les constructeurs pour les applications indoor.

Par rapport à d'autres techniques de modulation, cette méthode offre une bonne efficacité spectrale et est moins sensible aux perturbations.

La méthode OFDM est une technique de modulations multi porteuses, son principe consiste à envoyer en parallèle (simultanément) l'information sur N sous porteuses puis à leur appliquer une transformée de Fourier. Le fait d'envoyer le même signal sur plusieurs fréquences à la fois, permet de récupérer l'information même dans des environnements très perturbés. En effet lorsque le signal rencontre un obstacle sur une des porteuses, il suffit qu'il réussisse à passer sur une autre, de plus si le signal est affaibli, le récepteur peut parvenir à reconstituer le message complet. Les sous porteuses utilisées par ce type de modulation sont espacés par une valeur constante, et constituent une base orthogonale. Grâce à cette orthogonalité des sous porteuses, le spectre du signal OFDM présente une occupation optimale de la bande allouée.

La figure suivante, tirée de la spécification " power packet " établie par le fabricant Intellon, et reprise par le consortium Home Plug Alliance, montre la façon dont cette technique de modulation est mise en œuvre.

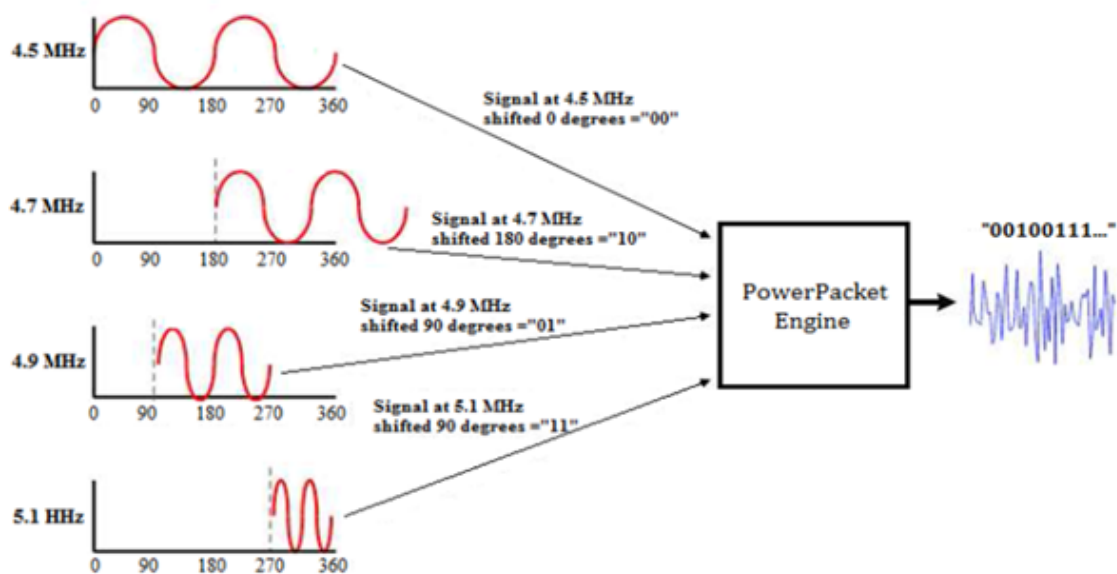


FIGURE 1.6: Technique de modulation OFDM

### 1.3.5.2 Modulation à étalement de spectre (Spread Spectrum)

Les techniques de modulation à étalement de spectre sont apparues dans les années 40 grâce à la théorie de l'information développée par N. Wiener et C. E. Shannon. Son principe peut être expliqué par la relation de C. E. Shannon qui exprime la capacité

d'un canal de propagation gaussien en fonction du rapport signal/bruit et de sa largeur de bande pour un canal perturbé par un bruit additif gaussien :

$$C = B * \log_2\left(1 + \frac{P_s}{P_b}\right) \quad (1.6)$$

Où :

- **C** : est la capacité du canal en bit/s
- **B** : la bande occupée par le signal émis en Hz
- **Ps** : la puissance du signal en W
- **Pb** : la puissance du bruit en W

D'après cette relation, pour assurer sans erreur la transmission d'une quantité d'information donnée, il est envisageable d'utiliser soit une large bande B et un faible rapport signal sur bruit, soit une bande étroite et un fort rapport signal sur bruit.

Les techniques de modulation Spread Spectrum (SS) consistent à émettre un signal sur une bande largement supérieure à celle du signal utile afin de combattre les interférences et les distorsions liées à la propagation.

La technique de modulation à étalement de spectre la plus utilisée pour certaines solutions CPL est la modulation CDMA (Code Division Multiple Access). Elle utilise généralement une séquence pseudo-aléatoire créée par une porte logique XOR appelée code d'étalement afin de disperser l'information sur une bande beaucoup plus large que celle nécessaire. Un code est assigné à chacun des usagers pour permettre le décodage à l'arrivée comme le montre la figure suivante :

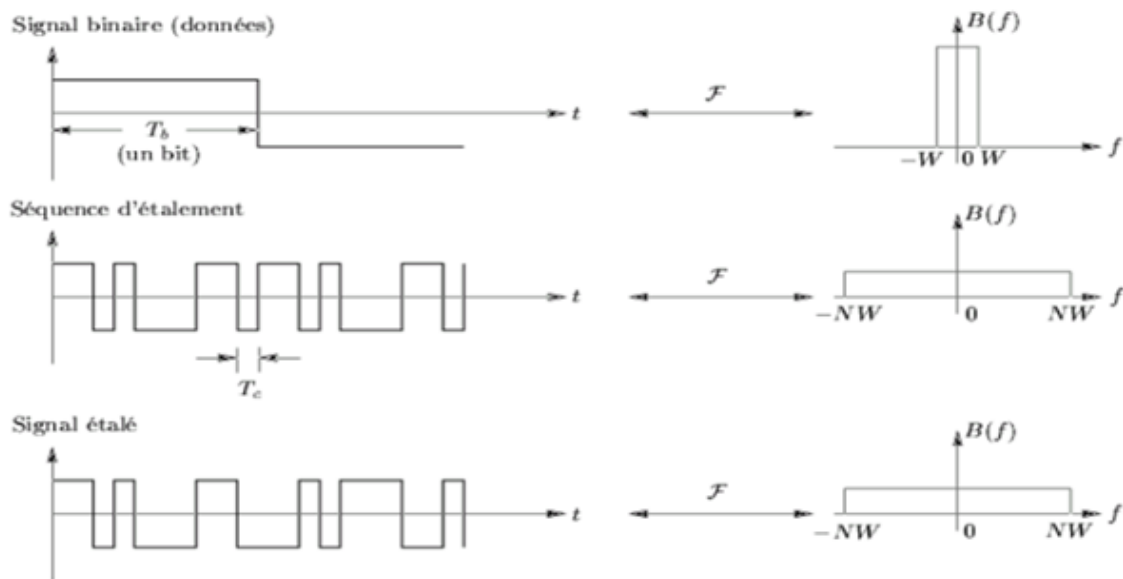


FIGURE 1.7: Exemple d'étalement d'un signal de données binaires

A noter que les deux types de modulation OFDM et à étalement de spectre SS présentent des intérêts différents. En effet, elles s'attaquent à deux problèmes distincts, la distorsion du signal et la montée en débit pour la première technique, tandis que la deuxième permet de lutter contre le bruit.

### 1.3.6 CPL Indoor et Outdoor

Le majeur intérêt des communications CPL étant le large étalement des réseaux de distributions électriques sur le globe terrestre, cette propriété rend cette technologie très intéressante pour la réalisation d'un réseau de desserte. D'une part l'accès haut débit à internet pour les zones enclavés et d'une autre part la possibilité de mise en place de systèmes de contrôles et de gestion des installations électriques à bas coût et à performance élevées, ce qui est sans équivoque un gain non seulement économique qu'écologique [1].

#### 1.3.6.1 CPL Indoor

La technologie CPL résidentielle dite indoor, est utilisée sur le réseau électrique privé de l'abonné en aval du compteur électrique (voir figure 1.8). Elle permet de partager une connexion existante, ou de constituer un réseau informatique local (LAN) personnel ou professionnel à l'intérieur des bâtiments tels les hôpitaux, les collèges ou

les immeubles administratifs. En particulier pour les monuments et les musées, il n'est plus possible légalement d'installer des lignes téléphoniques ou de construire un réseau câblé classique. Les CPL ont ici un avantage indéniable puisque le réseau électrique est préexistant.

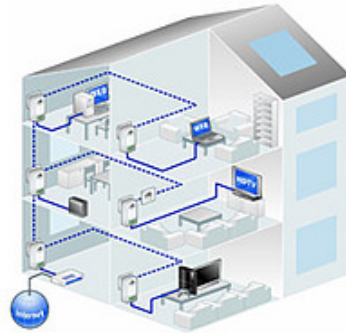


FIGURE 1.8: Réseau CPL INDOOR [1]

### 1.3.6.2 CPL Outdoor

L'accès outdoor profite de l'infrastructure électrique publique moyenne et basse tension (MT/BT) en amont du compteur électrique (voir figure 1.9) afin d'acheminer un flux haut débit depuis un point d'accès préexistant. Pour ces installations extérieures, les expérimentations sont en cours, avec un couplage entre le réseau électrique et l'arrivée Internet haut débit via un transformateur HT/BT pour la création d'une boucle locale électrique.

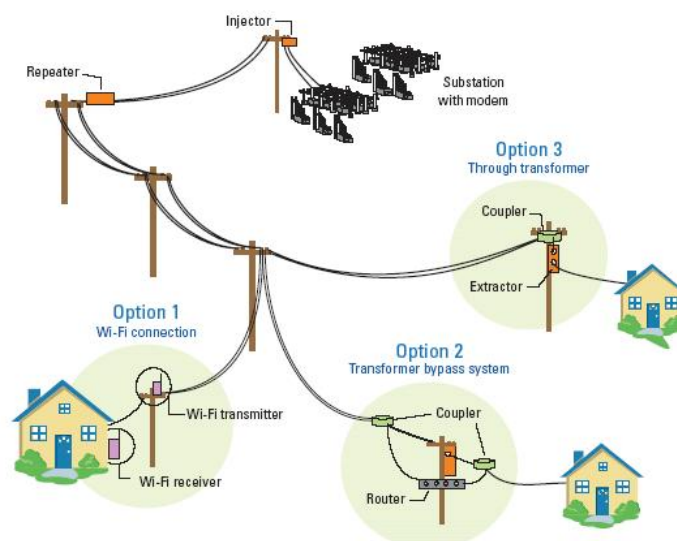


FIGURE 1.9: Réseau CPL OUTDOOR [1]

### 1.3.6.3 Cadre juridique

Concernant les installations en aval d'un compteur privé (réseau Indoor), la mise en place est libre sous condition de ne pas générer de perturbations (perturbation du réseau de distribution électrique, émission d'onde électromagnétique susceptible d'interférer avec les liaisons radioélectriques, ou bien nocives à la santé, ...). Par contre pour les réseaux Outdoor, il n'est que récemment que les autorités en Europe, aux USA et au Japon ont levé le statut expérimental qui encadre le déploiement des CPL. Ces autorités considèrent que les contraintes qui avaient justifié le statut expérimental de la mise en place des CPL n'étaient plus pertinentes.

## 1.3.7 Normes protocolaires

### 1.3.7.1 Introduction

PRIME et G3-PLC sont les deux nouveaux protocoles des CPL-BE les plus répandus, ils ont d'abord été développés en tant que solution libre par un petit groupe d'industries des télécommunications et des réseaux électriques, qui ont ensuite développé de grande alliances avec un nombre croissant d'entreprises, bien qu'ils aient été tous deux développés pour les réseaux intelligents, ils présentent des différences considérables [12].

Dans ce qui suit, nous présentons une comparaison, ainsi que les résultats de tests expérimentaux réalisés sur ces protocoles.

### 1.3.7.2 Tests et résultats expérimentaux

	DQPSK	DBPSK	RDQPSK	RDBPSK
BER PRIME(%)	25	7	0	0
Trames reçues PRIME(%)	98	96	100	100
BER G3-PLC(%)	0	0	0	0
Trames reçues G3-PLC(%)	100	100	100	100

TABLE 1.3: Résultats de tests en l'absence d'interférences[12]

	DQPSK	DBPSK	RDQPSK	RDBPSK
BER PRIME (FEC OFF) (%)	16	2	0	0
Trames PRIME (FEC OFF) reçues (%)	100	100	100	100
BER PRIME (FEC ON) (%)	2	0	0	0
Trames PRIME (FEC ON) reçues (%)	100	100	100	100
BER G3-PLC (%)	0	0	0	0
Trames G3-PLC reçues (%)	100	100	100	100

TABLE 1.4: Résultats de tests en présence d'interférences[12]

Le mécanisme FEC (Forward Error Correction) du protocole PRIME est composé d'un codeur convolutif associé à un brouilleur, un entrelaceur et un codeur de répétition. Pour le G3-PLC, en mode normal, le FEC est composé d'un codeur Reed-Solomon, d'un codeur convolutif et d'un entrelaceur. En mode robuste, un codeur de répétition (RC4) supplémentaire est utilisé pour répéter les bits quatre fois. En mode super robuste, ces bits sont répétés six fois avec un codeur de répétition (RC6). Plus de détails sur le FEC sont donnés dans le chapitre 2.

### 1.3.7.3 Comparaison

	PRIME	G3-PLC
Plage de fréquence	42 – 89 KHz	35 - 91 KHz
Fréquence d'échantionnage	250 KHz	400 KHz
Espacement des sous porteuses	488.28125 Hz	1562.5 Hz
Nombre de sous porteuses de données	84 (header) 96 (payload)	36
FFT intervalle	2048 $\mu$ s	640 $\mu$ s
Préfixe cyclique	192 $\mu$ s	75 $\mu$ s
Durée d'un symbole OFDM	2240 $\mu$ s (2048 $\mu$ s + 192 $\mu$ s)	715 $\mu$ s
Période du préambule	2048 $\mu$ s (PRIME 1.3.6) 8192 $\mu$ s (PRIME 1.4)	6080 $\mu$ s

TABLE 1.5: Comparaison théorique entre G3-PLC et PRIME



## 1.4 Conclusion

Les développeurs de systèmes IoT se basant sur le canal CPL-BE sont mis au défi par les caractéristiques très défavorables de ce canal de transmission. Il est nécessaire de lui adjoindre des techniques de traitement de signal avancées comme le codage correcteur d'erreurs pour fiabiliser les transmissions. La conception et la mise en œuvre de ces infrastructures devra tenir compte des caractéristiques particulières du canal CPL-BE, ainsi que des différentes contraintes comme la puissance du signal à émettre qui est très limité.

Bien que les standards G3-PLC et PRIME est étaié développés pour les mêmes finalités, il présente cependant des différences considérables, PRIME privilégie les débits de données élevés dans des conditions favorables, tandis que G3-PLC offre une meilleure fiabilité dans des conditions défavorables même sans utiliser le mode robuste, on note aussi que PRIME offre une fiabilité similaire au G3-PLC en mode ROBUSTE mais cela se fait au prix d'une forte réduction du débit de transmission [12].

Dans le chapitre suivant nous présenterons les spécifications des couche physiques, liaison et d'adaptation de la pile protocolaire G3-PLC qui semble la plus adéquate pour la mise en place de réseaux d'objets connectés ne demandant pas des débits trop élevés mais imposant une haute fiabilité de transmission.

# Chapitre 2

## Pile protocolaire G3-PLC

## 2.1 Introduction

Le canal de courant porteur étant un support de communication très hostile, ces caractéristiques varient selon la fréquence, l'équipement branché dessus, l'heure, l'emplacement, sans négliger les multitudes de bruits qui y règnent, il paraît donc opportun de définir un protocole de communication fiable, pour assurer une transmission robuste, sans pour autant négliger la contrainte qu'est un débit acceptable.

Le standard G3-PLC a été conçu spécifiquement pour répondre à ces contraintes, basé sur la technologie OFDM il vise principalement les objectifs suivants :

- Assurer une communication robuste,
- Fournir un débit de transmission élevée, autour de 20 kbps en mode normal,
- Capacité d'enlever(masquer) des fréquences sélectionnées, autorisant la cohabitation avec d'autre protocole de communication CPL, et assurant une transmission sur les fréquences les moins atténuées par le canal.

Dans ce qui suit vont être présentées les trois couches (PHYSIQUE, LIASON, ADAPTATION) qui forment la pile protocolaire G3-PLC.

## 2.2 Modèle en couches – Positionnement par rapport au modèle OSI

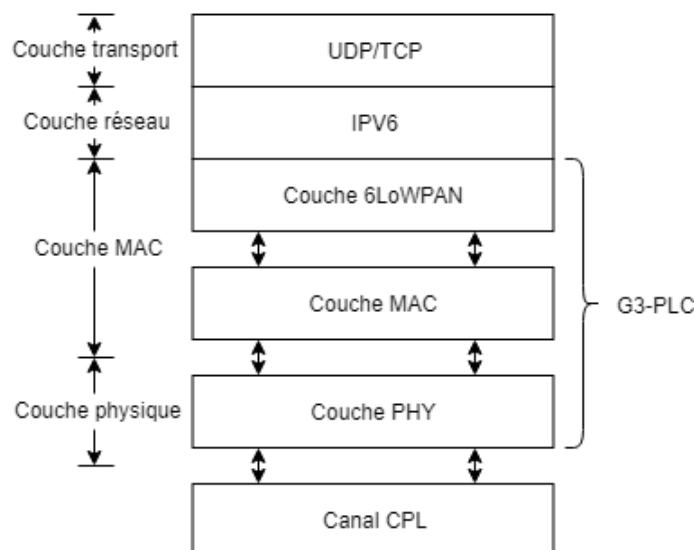


FIGURE 2.1: Modèle en couche [13]

La spécification G3-PLC définit la couche PHYSIQUE et la couche LIAISON du modèle OSI, avec une couche d'adaptation 6LoWPAN qui se situe entre la couche LIAISON et la couche RESEAU, cette dernière combiné à IPv6 forme la couche réseau, elle permet de s'affranchir des contraintes d'IPv6 sur les objets connectés du fait de l'importante taille de ces en-têtes.

## 2.3 Spécification de la couche physique

Cette section présente le bloc de la couche physique basé sur la technologie OFDM.

### 2.3.1 Architecture en bloc de la couche PHYSIQUE

La figure suivante présente le diagramme en bloc du système OFDM permettant la transmission de trame G3-PLC.

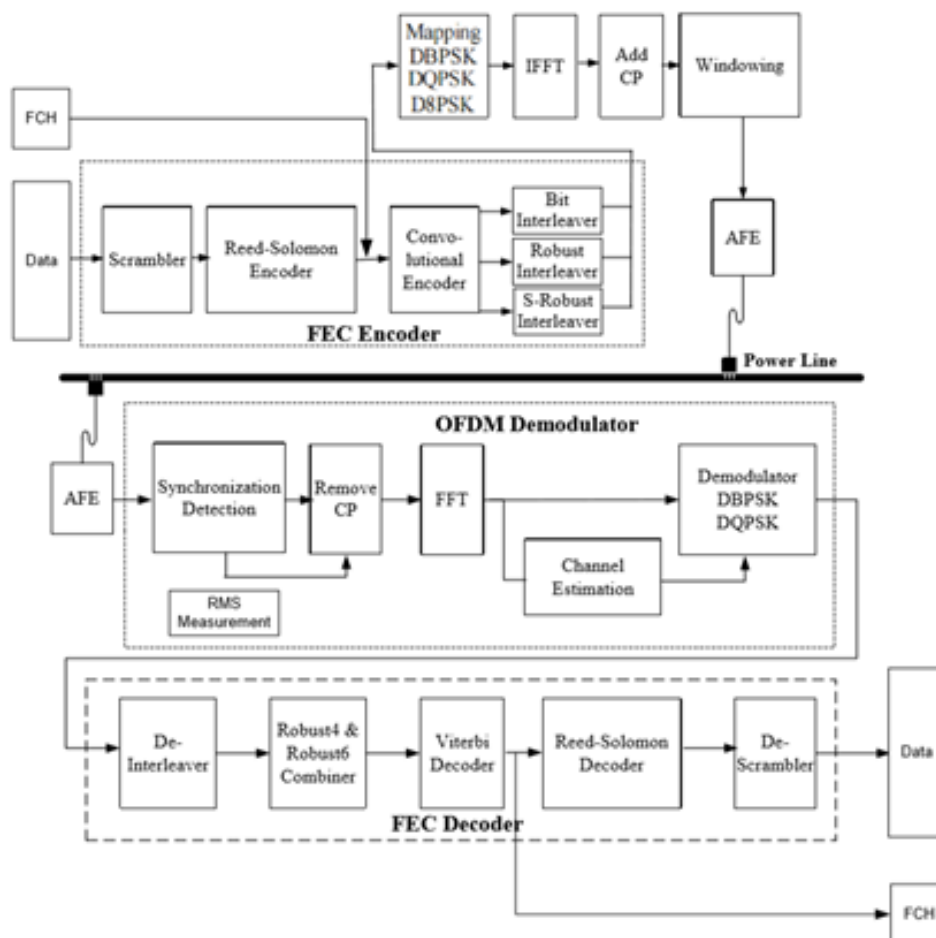


FIGURE 2.2: Architecture en bloc de la couche PHYSIQUE [13]

### 2.3.1.1 Brouilleur (Scrambler)

Le bloc brouilleur permet de fournir aux données une distribution aléatoire évitant ainsi une trop longue succession de un ou zéro, utilisant le polynôme générateur suivant :

$$S(x) = x^7 \oplus x^4 \oplus 1 \quad (2.1)$$

Ceci est illustré à la Figure 2.3. Les bits contenus dans le brouilleur sont tous initialisés à un au début du traitement de chaque trame PHY.

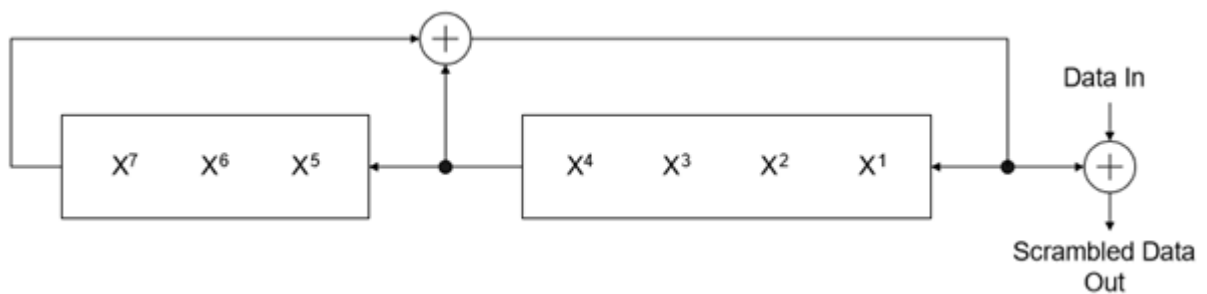


FIGURE 2.3: Brouilleur de données [13]

### 2.3.1.2 Codage FEC (Forward Error Correction)

Le codeur FEC est composé d'un codeur Reed-Solomon suivi d'un codeur convolutif. En mode Robuste, un codeur supplémentaire, intitulé RC (Code de Répétition), est utilisé après le codeur convolutif afin de répéter quatre, ou six fois les bits à la sortie du codeur convolutif selon le mode robuste ou super robuste.

1. **Codeur Reed-Solomon** : Le code de Reed-Solomon est un code correcteur dont le principe est de construire un polynôme à partir des symboles à transmettre et de le sur-échantillonner. Le résultat est alors envoyé, au lieu des symboles originaux. La redondance de ce sur-échantillonnage permet au récepteur du message encodé de reconstruire le polynôme même s'il y a eu des erreurs pendant la transmission.
2. **Codeur Convolutif** : Le codeur convolutif utilisé à un taux de 1/2, avec une contrainte  $K=7$ , avec les polynômes générateurs  $G_1 = (1,1,1,1,0,0,1)$  et  $G_2=(1,0,1,1,0,1,1)$ . Après réception du dernier bit de données le codeur convolutif insère six bits de queue nécessaires pour le retour à "l'état zéro".

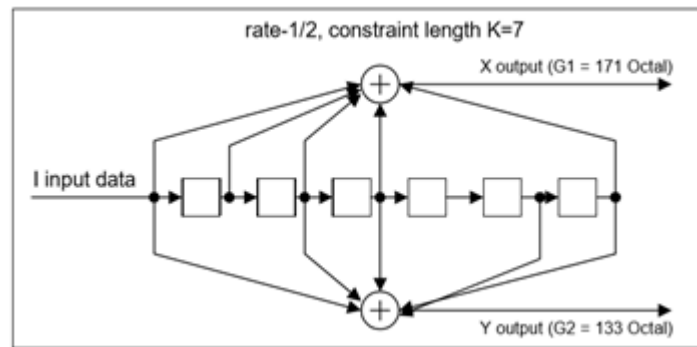


FIGURE 2.4: Codeur convolutif [13]

3. **Codage de répétition par 4** : En mode Robuste, le paquet résultant est répété quatre fois par le dispositif d'entrelacement. Ce codeur n'est activé qu'en mode Robuste. La répétition est réalisée en répétant quatre fois la trame. Pour le FCH, on utilise le mode Super Robuste. Ce mode fait appel à une répétition de 6, ce qui accroît la robustesse.
4. **Codage de répétition par 6** : En mode Super Robuste, le paquet résultant est répété six fois par le dispositif d'entrelacement. La répétition s'effectue en répétant six fois la trame.
5. **Dispositif d'entrelacement** : Le dispositif d'entrelacement a été conçu d'une manière à assurer une protection contre deux types d'erreurs :
  - Une erreur burst qui corrompt quelques symboles OFDM consécutifs.
  - Une fréquence fortement atténuée qui corrompt des fréquences adjacentes sur un grand nombre de symboles OFDM.

Pour lutter contre ces deux problèmes en même temps, l'entrelacement se fait en deux étapes. Dans une première étape, chaque colonne est décalée circulairement un nombre différent de fois. Par conséquent, un symbole OFDM corrompu, est réparti sur différents symboles. Dans une deuxième étape, chaque ligne est circulairement déplacée un nombre différent de fois, ce qui empêche une fréquence très atténuée de perturber l'ensemble de la colonne.

### 2.3.1.3 Bloc de mappage

Le bloc de mappage a pour fonction de s'assurer que le signal émis est en conformité avec la carte de tonalités et le masque de tonalités fournis. Les carte et masque de tonalités sont des concepts de la couche MAC. Le masque de tonalités est un paramètre prédéfini et étendu au système qui établit les fréquences de démarrage, d'arrêt et d'encoche. La carte de tonalités est un paramètre adaptatif qui, basé sur une estimation du canal, contient la liste des porteuses utilisées pour une communication particulière entre deux modems. Par exemple, des porteuses subissant des évanouissements peuvent être ignorées et donc aucune information n'est émise sur ces porteuses.

### 2.3.1.4 Génération de symboles OFDM (IFFT et CP)

Les symboles OFDM sont générés avec une transformée de Fourier inverse, conformément à l'équation suivante :

$$S = \sum_{K=0}^{N-1} C_k \cdot e^{2j\pi f_k t} \quad (2.2)$$

- $N$  : Taille IFFT.
- $C_k$  : Symbole numérique.
- $f_k$  : Fréquence de la sous porteuse  $k$ .

En suite un préfixe cyclique (CP) est ajouté pour s'assurer que le canal soit vide avant l'émission du prochain symbole.

Le préfixe cyclique étant égal au plus grand retard causé par le canal, ceci permet avant l'émission d'un symbole de s'assurer que le symbole précédent et toutes ses versions retardés ne soit plus dans le canal, et donc éliminer toute erreur de transmission qui serait dû à ce retard.

### 2.3.1.5 Fenêtrage

Afin de réduire l'émission hors bande, une fenêtre en forme de cosinus surélevé est appliquée pour réduire la puissance des lobes secondaires.

### 2.3.1.6 Mappage de tonalités adaptatif et contrôle de puissance d'émission

La technologie G3-PLC doit évaluer le SNR du signal reçu et sélectionner de façon adaptative les tonalités utilisables, la modulation optimale ainsi que le taux de transmission du code pour assurer une communication fiable sur le canal de courants porteurs. Elle doit également spécifier le niveau de puissance que le transmetteur à distance doit utiliser ainsi que des valeurs de gain à appliquer pour les diverses parties du spectre.

Le mappage adaptatif de tonalité vise à permettre au récepteur G3-PLC d'obtenir le plus grand débit possible que les conditions existantes du canal puissent offrir. Pour réaliser cet objectif, le récepteur doit informer le transmetteur distant sur les tonalités à utiliser pour l'envoi des bits de données et sur celles qui serviront pour l'envoi de bits de données fictives que le récepteur doit ignorer. Le récepteur doit également informer le transmetteur distant sur le volume d'amplification ou d'atténuation à appliquer sur chacune des tonalités.

### 2.3.1.7 Passage par un transformateur

Le système G3-PLC doit être en mesure de communiquer sur les lignes électriques basse tension (généralement INDOOR), ainsi que sur les lignes électriques haute tension (généralement OUTDOOR). En fonctionnement sur les lignes électriques hautes tension, le système doit être capable de communiquer avec les lignes basses tension. En d'autres termes, le récepteur côté basse tension doit être capable de détecter le signal émis après que celui-ci ait subi une importante atténuation résultant du passage par un transformateur HT/BT. Lorsque le signal passe par le transformateur, celui-ci subit une importante atténuation globale de son niveau de puissance ainsi qu'une atténuation dépendante de la fréquence qui atténue particulièrement les fréquences plus élevées. Le transmetteur et le récepteur doivent tous les deux posséder un mécanisme de compensation de cette atténuation. L'émetteur doit avoir la capacité d'ajuster son niveau de signal ainsi que la forme de son spectre de puissance, tandis que le récepteur doit avoir un AGC (Automatic Gain Control) à la fois analogique et numérique afin d'avoir suffisamment de gain pour compenser l'atténuation.



### 2.3.1.8 Détection de la phase

Il s'agit d'une approche de base pour détecter la phase associée à un modem . Cependant, la conception optimale dépend de la topologie du réseau. Ces informations sont principalement utiles au niveau du système pour contrôler les pertes imprévues sur la ligne.

Les trois phases sur le secteur sont des formes d'onde sinusoïdales avec un déphasage de  $120^\circ$  l'une de l'autre, chaque demi-cycle étant égal à 10 ms à 50 Hz et à 8,3 ms à 60 Hz. Un détecteur de passage par zéro délivre une impulsion de sortie à chaque passage de la tension secteur par zéro.

Il est utilisé pour synchroniser le transmetteur et le récepteur. Le transmetteur génère un horodatage basé sur un compteur interne à l'instant où un paquet doit être transmis. Le récepteur fournit son propre horodatage et le délai entre le transmetteur et le récepteur fournit la différence de phase. La procédure pour obtenir la différence de phase entre émetteur et récepteur est la suivante :

- Tous les périphériques doivent posséder un temporisateur interne et être synchronisés avec le détecteur de passage par zéro,
- Tous les périphériques doivent avoir un détecteur de passage par zéro qui délivre une impulsion de sortie de telle sorte que la largeur d'impulsion soit égale 5% de la période totale. Les caractéristiques du détecteur de passage par zéro sont présentées à la Figure suivante.

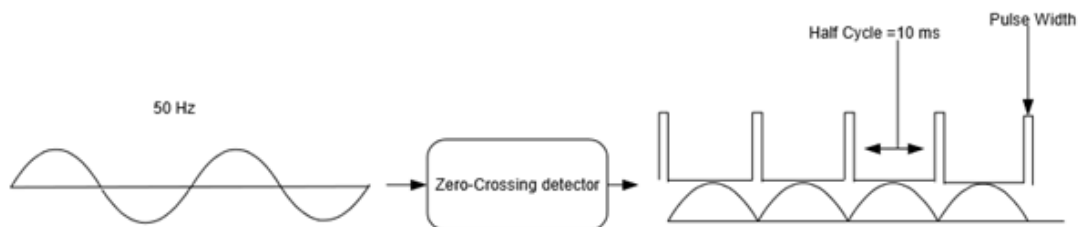


FIGURE 2.5: Détecteur de passage par zéro [13]

- Un compteur 8 bits fournit un horodatage placé sur une trame lors d'une transmission de données utiles.

Lors de la détection d'une trame, le récepteur doit calculer le retard, il s'agit de la différence entre le compteur 8 bits de l'émetteur et celui du récepteur.

## 2.4 Couche liaison

La couche liaison implémenté pour G3-PLC se base sur IEEE 802.15.4, elle fournit les moyens nécessaires pour l'acheminement de données, elle offre les procédés de détections et de corrections des erreurs, avec le protocole de cryptage AES [14] pour la sécurisation des données qui est natif pour cette couche, elle est, entre autre, responsable de :

1. L'accès au support : en utilisant l'algorithme CSMA / CA (accès multiple à détection de porteuse avec prévention des collisions), il permet aux périphériques de partager la bande passante du support CPL en évitant les collisions.
2. L'adressage du message : l'adresse source et l'adresse de destination permettent de filtrer les paquets indiqués à la couche supérieure.
3. Le chiffrement de tous les paquets : il utilise AES-CCM avec une clé de groupe et permet de protéger toutes les données échangées entre les périphériques du réseau contre les écoutes ou les modifications. La configuration de la clé de groupe est sous la responsabilité de la couche 6LoWPAN et de sa procédure d'amorçage.
4. Négociation de la modulation et de la liste des porteuses (carte de tonalité) à utiliser par la couche physique lors de la transmission de la trame.
5. La segmentation de la trame : elle permet d'adapter la taille des données fournies à la couche PHY pour l'aligner sur le PHY MTU (Maximum Transmit Unit), qui dépend de la modulation utilisée et de la carte de tons.

### 2.4.1 Structure d'une trame G3-PLC

Une trame G3-PLC est composée d'un PREAMBLE, HEADER et PAYLOAD, comme montré sur la figure ci-dessous.

PREAMBLE	HEADER	PAYLOAD
6,08 ms	9,295 ms et 13 symboles OFDM	Mx4xOFDM symboles

TABLE 2.1: Composition d'une trame G3-PLC

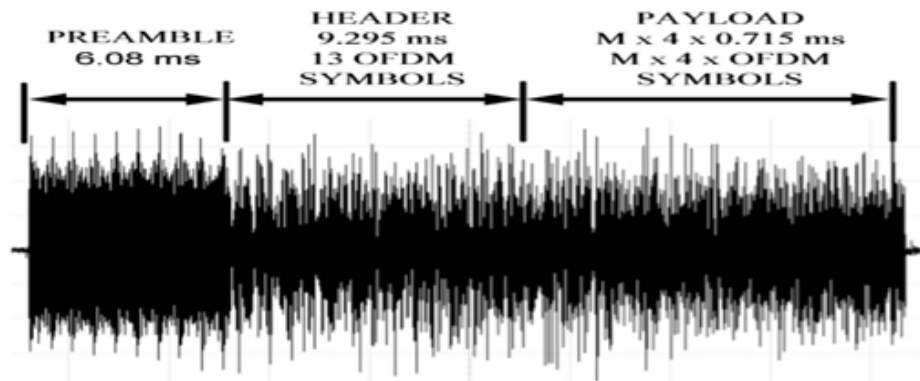


FIGURE 2.6: Structure d'une trame G3-PLC [12]

**PREAMBLE** : un seul symbole OFDM répété 9,5 fois, cela permet de réduire la probabilité qu'un récepteur ne reconnaisse pas un paquet, ces symboles sont également utilisés pour estimer le SNR, la synchronisation et l'adaptation AGC. Les phases des sous-porteuses précédentes sont prises comme référence pour détecter les phases des sous porteuses actuel.

**HEADER** : composé de 13 symboles OFDM, il contient l'information nécessaire au décodage de l'information du reste de la trame.

**PAYLOAD** : composé de groupes de 4 symboles OFDM, il est la charge utile, et peut être modulé en BPSK, QPSK ou 8PSK qui sont sélectionné sur la base du SNR, ou selon le choix de l'utilisateur.

## 2.5 Couche d'adaptation 6LoWPAN

IPv6 LoW Power wireless Area Networks, est une couche définie explicitement pour l'encapsulation et la compression d'entête IPv6, ce qui permet la mise en œuvre d'un réseau IPv6 [15] sur des systèmes contraints (ex : Objet connecté), cette couche d'adaptation se situe entre la couche liaison et la couche réseau du modèle OSI (la couche réseau étant IPv6), dans ce qui suit nous présenterons d'abord les notions de bases IPv6, pour ensuite présenter la couche d'adaptation 6LoWPAN [16].

## 2.5.1 IPv6

IP version 6 IPv6 est une nouvelle version du protocole Internet, conçu pour succéder à IP version 4 IPv4. Les modifications apportées d'IPv4 à IPv6 relèvent principalement de ce qui suit [17] :

- **Capacités d'adressage étendues**

IPv6 augmente la taille de l'adresse IP de 32 bits à 128 bits, ce qui fait une remarquable augmentation du nombre de nœud adressable (à titre de comparaison grossière :  $2^{32}$  de nœuds adressable pour IPv4 et  $2^{128}$  pour 6).

- **Simplification du format d'en-tête**

Certains champs d'en-tête IPv4 ont été supprimés ou rendus facultatifs pour réduire le coût de traitement des cas communs de traitement des paquets et limiter le coût de la bande passante de l'en-tête IPv6.

- **Prise en charge améliorée des extensions et des options**

Les modifications apportées à la manière dont les options d'en-tête IP sont codées permettent des transmissions plus efficaces, des limites de longueur moins strictes d'options et une plus grande flexibilité pour introduire de nouvelles options à l'avenir.

- **Capacité d'étiquetage de flux**

Une nouvelle fonctionnalité est ajoutée pour permettre l'étiquetage des paquets appartenant à un "flux" de trafic particulier pour lequel l'expéditeur demande un traitement spécial, tel que la qualité de service ou service "temps réel".

### 2.5.1.1 En-tête IPv6

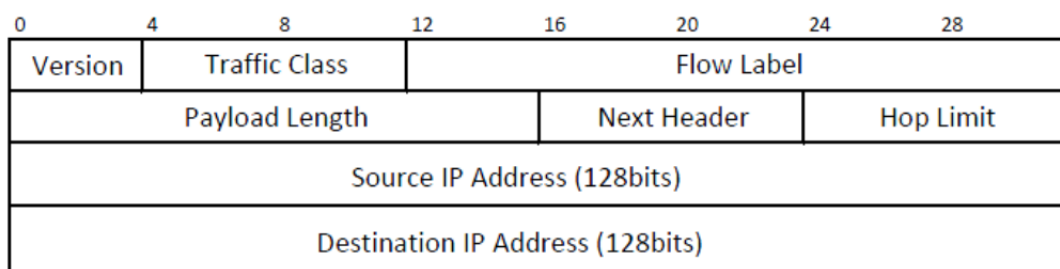


FIGURE 2.7: En-tête IPv6 [15]

— **Version** : 4 bits, numéro de version internet protocole = 6.

- **Traffic class** : Le champ Classe de trafic sur 8 bits de l'en-tête IPv6 peut être utilisé par les nœuds d'origine et / ou les routeurs de transmission pour identifier et différencier les différentes classes ou priorités de paquets IPv6.
- **Flow label** : Le champ Etiquette de flux 20 bits de l'en-tête IPv6 peut être utilisé par une source pour étiqueter les séquences de paquets pour lesquelles elle demande un traitement spécial par les routeurs IPv6, telle qu'une qualité de service autre que celle par défaut ou un service "en temps réel".
- **Payload Length** : Entier non signé 16 bits. Longueur de la charge utile IPv6, c'est-à-dire le reste du paquet suivant cet en-tête IPv6, en octets.
- **Next Header** : champ de 8 bits. Identifie le type d'en-tête qui suit immédiatement l'en-tête IPv6, pour s'avoir si elle doit être traitée ou pas.
- **Hop Limit** : Entier non signé de 8 bits. Décrémenté de 1 par chaque nœud qui transmet le paquet. Le paquet est détruit si Hop Limit atteint zéro.
- **Source Address** : 128 Bits, adresse de l'expéditeur du paquet.
- **Destination Address** : 128 bits, adresse du destinataire du paquet.

### 2.5.1.2 Adressage IPv6

Les adresses IPv6 sont des identifiants de 128 bits pour des interfaces et ensembles d'interfaces, Il y a trois types d'adresses :

- **Envoi individuel (unicast)** : identifiant pour une seule interface. Un paquet envoyé à une adresse d'envoi individuel est livré à l'interface identifiée par cette adresse.
- **Envoi à la cantonade (anycast)** : identifiant pour un ensemble d'interfaces (appartenant à des nœuds différents). Un paquet envoyé à une adresse d'envoi à la cantonade est livré à une des interfaces identifiées par cette adresse.
- **Envoi en diffusion groupée (multicast)** : identifiant pour un ensemble d'interfaces (appartenant à des nœuds différents). Un paquet envoyé à une adresse de diffusion groupée est livré à toutes les interfaces identifiées par cette adresse.

## 2.5.2 6LoWPAN

L'adressage complexe d'IPv6 apporte des avantages indéniables, mais ceci aux coups d'une pile protocolaire plus complexes [18] qui demande des ressources de traitements élevées (consommation énergétique, espace mémoire, ressource CPU ...) ce qui en fait des contraintes non négligeables sur les objets connectées, pour faire face à ces contraintes la couche d'adaptation 6LoWPAN a été développé, dans ce qui suit, va être présenté les contraintes jugées les plus pertinentes rencontrées lors de l'implémentation d'IPv6 sur des OC, aux quelles réponde les spécifications 6LoWPAN.

- Fortes contraintes sur la MTU (Maximum Transmit Unit). IPv6 requiert une MTU de 1280 octets et indique que tout lien ne remplissant pas cette contrainte doit implémenter de la fragmentation de niveau liaison [17].
- Coût de l'entête IPv6. Le protocole possède une taille d'en-tête minimum de 40 octets, en ajoutant à ça un en-tête TCP de 20 octets ou UDP de 8 octets, quand la couche IPv6 est superposé à la couche liaison IEEE 802.15.4 qui définit une PSDU (Physical layer Service Data Unit, « Unité de données service de la couche physique ») de 127 octets avec 25 octets pour la couche liaison elle-même, le chiffrement natif pour cette couche (AES128) 21 octets, finalement il ne restera plus que 21 octets pour TCP et 33 pour UDP.

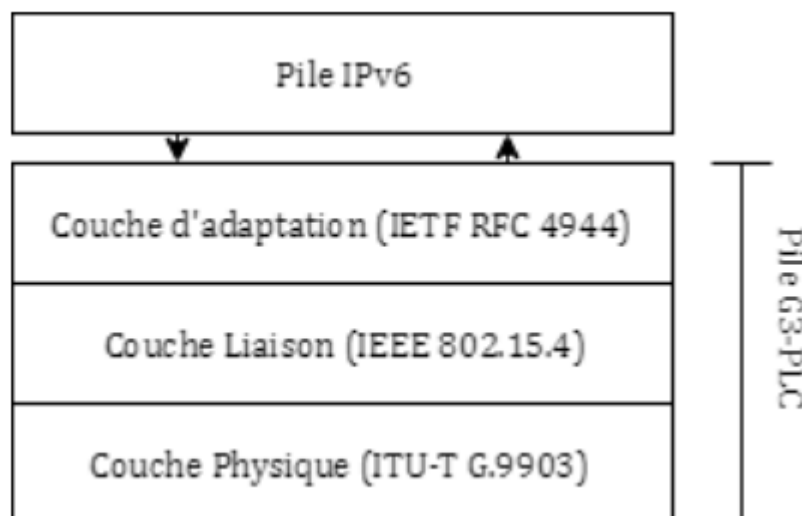


FIGURE 2.8: Pile protocolaire

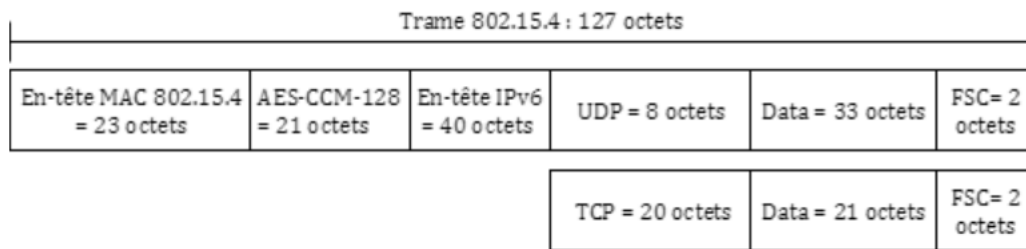


FIGURE 2.9: Trame 802.15.4

- Fragmentation de niveau liaison. L'obligation de fragmentation pour la couche réseau va conduire à une très forte charge de découpage/reconstruction pour des éléments n'en ayant clairement pas le potentiel.
- Loose Source Routing. IPv6 a réintroduit le support de cette technique de routage. C'est un moyen permettant à un hôte de spécifier explicitement le chemin que va emprunter son paquet. Cela conduit à un certain nombre de problèmes, et spécialement en termes de sécurité. L'exemple nous concernant est celui permettant à un hôte de pouvoir indiquer une boucle dans son paquet (échange continu entre un routeur R1 et un routeur R2) qui va conduire à une surcharge artificielle du réseau (particulièrement dans les réseaux de capteurs).

Pour répondre à ces contraintes 6LoWPAN [16] est livré avec deux mécanismes de transfert de fragments (Fragment Forwarding) : mesh under et route over. Le premier schéma fonctionne au niveau de la couche d'adaptation, tandis que le second fonctionne au niveau de la couche réseau.

1. **Mesh under** : le réseau (c'est-à-dire, IP) n'effectue aucun routage IP. En fait, la couche d'adaptation 6LoWPAN exécute le routage et la transmission sur le réseau. Pour transmettre un datagramme à une destination donnée, l'adresse EUI-64<sup>1</sup> bits ou l'adresse courte à 16 bits est utilisée [19]. Ainsi, pour transmettre la trame reçue, la couche 6LoWPAN utilise l'en-tête Mesh (Mesh Header) ainsi que l'adresse de la source et de destination incluse dans la couche liaison (802.15.4). Par conséquent, il n'est pas nécessaire de décompresser l'en-tête IPv6.

---

1. EUI-64 : Procédé permettant la génération d'une adresse IPv6 à partir de l'adresse MAC

2. **Route Over** : dans le schéma route-over, tous les routages et les opérations de transfert sont effectuées dans la couche réseau. Étant donné qu'IEEE 802.15.4 fonctionne sur des réseaux maillés, un protocole de routage fonctionne souvent au niveau de la couche 3. Par exemple, le protocole de routage IPv6 bien connu pour les réseaux à faible puissance et à perte (RPL) est l'un des protocoles les plus adoptés de routage des paquets dans un réseau multi-sauts. Dans RPL, chaque nœud peut servir de relais pour les autres. Il est évident que la couche d'adaptation 6LoWPAN traite les fragments reçus à chaque saut afin de réassembler le paquet de données original IPv6, puis à fragmenter à nouveau, avant de transférer au saut suivant, donc au lieu que le paquet remonte jusqu'à la couche IPv6 et qu'il y soit réassembler et/ou fragmenter de nouveau ce qui demande beaucoup de ressource, la couche 6LoWPAN s'occupe de cette tâche, avec des algorithmes plus optimisés.
3. **Compression d'en-tête IPv6** : les en-têtes IPv6 et UDP non compressés ont une longueur de 48 octets. La couche d'adaptation supprime les informations implicites et redondantes pour les réduire à 2 octets.



## 2.6 Comparaison de solutions dédiées à l'implémentation de G3-PLC

	PL360	ATPL250A	ST8500	NCN49597	R9A06G-037GNP
Fabriquant	Microchip	Microchip	ST Microelectronics	ON Semiconductor	Renesas Electronics
Standards	G3&PRIME	G3	G3&PRIME	G3	G3&PRIME
Contrôleur	Externe	Externe	Interne	Externe	Externe
Fréquence max	500	500	500	150	500
Prix (euro)	5,04	5.47	5,82	7.80	4.98
Bandes de Fréquences					
CENELEC	Oui	Oui	Oui	Oui	Oui
FCC	Oui	Oui	Oui	Non	Oui
ARIB	Oui	Oui	Oui	Non	Oui
Modulation					
PSK	Oui	Oui	Oui	Oui	Oui
FSK	Non	Non	Oui	Non	Non

TABLE 2.2: Comparatif des modems CPL implémentant G3-PLC

Le standards G3-PLC étant le plus adéquat pour l'application envisagée dans ce projet du fait de sa fiabilité de communication dans des conditions défavorable, de sa robustesse au bruit, et de sa couche 6LoWPAN qui permet une mise en place optimale d'IPv6 ce qui est le plus approprié pour les OC, le circuit PL360 de Microchip implémentant ce standard a été choisi pour le développement et la mise en œuvre de notre système.

## 2.7 Conclusion

Dans ce chapitre ont été présentées :

- les spécifications de la couche physique du protocole G3-PLC, en passant du codeur FEC, jusqu'aux modulations OFDM, ce qui justifie la robustesse de transmission de ce protocole.
- Un bref aperçu sur la structure de la trame G3-PLC, avec le chiffrement natif (AES128) pour cette couche.
- le protocole IPv6 qui, vue le nombre d'objets connectée prévu (environ 29 milliards en 2020 selon Siemens, Cisco et IDC) dans les années avenir, est sans concurrent.
- la couche d'adaptation 6LoWPAN qui fait abstraction des contraintes d'IPv6 sur des dispositifs ayant des ressources limitées.

Le chapitre suivant sera exclusivement dédié à la description du circuit PL360 de Microchip, avec un aperçu sur la pile protocolaire implémenté, répondant aux spécifications de G3-PLC.

# Chapitre 3

## Circuit PL360

## 3.1 Introduction

Le PL360 est un circuit programmable pour la communication sur Courants Porteur en Ligne à Bande Etroite (CPL-BE), capable d'exécuter n'importe quel protocole CPL dans la bande de fréquence inférieure à 500 kHz.

Ce circuit a été conçu pour être conforme aux normes FCC, ARIB, et CENELEC correspondant aux exigences des applications Internet des objets et Smart Grid. Le PL360 a été conçu pour être piloté par des périphériques hôtes externes, offrant ainsi un niveau de flexibilité supplémentaire du côté de l'hôte. Le périphérique hôte charge le microprogramme du protocole CPL approprié avant le fonctionnement du circuit et le contrôle en mode de fonctionnement normal.

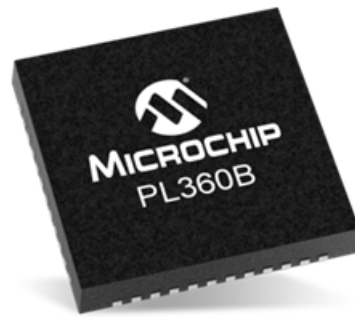


FIGURE 3.1: Circuit PL360[20]

### 3.1.1 Caractéristiques du PL360

1. Frontal PLC intégré :
  - Contrôle numérique du niveau de transmission.
  - Supporte deux branches de transmission indépendantes pour le signal CPL.
  - Largeur de bande du signal CPL jusqu'à 500 kHz.
2. Architecture
  - Architecture haute performance combinant CPU, co-processeurs spécifiques pour le signal numérique.
  - Mémoires SRAM dédiées pour le code et les données.
3. Système ARM Cortex M7 de 32-bit comme système de gestion de noyau du PL360 : Co-processeurs, Accélérateurs de matériel et Périphériques
  - Fréquence maximale de 216 MHz.

- 192 kb de SRAM pour les données et le code.
  - Bootloader permettant de charger des programmes simples ou cryptés.
  - 12 GPIO multiplexés.
  - 1 SPI, 1 UART, 2 PWM.
4. Moteur cryptographique et démarrage sécurisé :
- AES 128, 192, 256 pris en charge.
  - Contrôle de la programmation des fusibles pour le décryptage et l'authentification des clés de 128 bits.

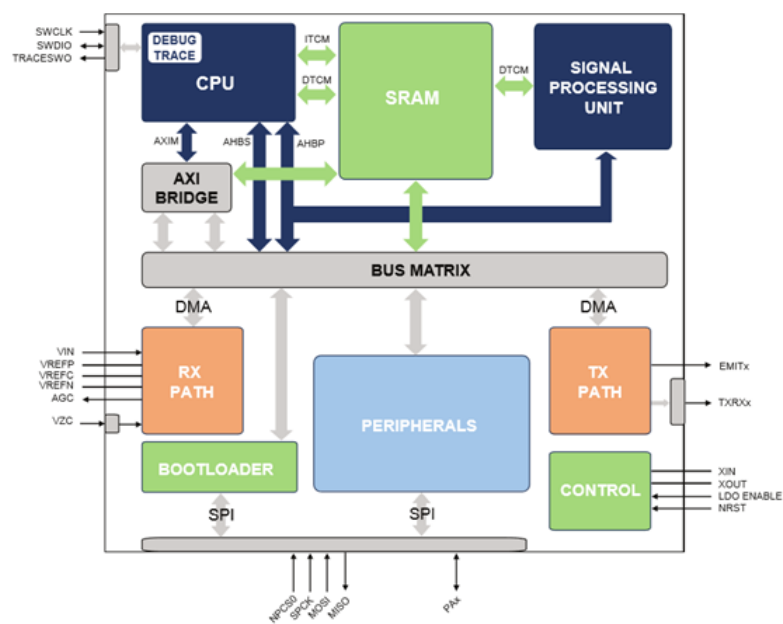


FIGURE 3.2: Diagramme en bloc du PL360[20]

## 3.2 Architecture système d'une application PL360

L'émetteur-récepteur PL360 a été conçu pour être facilement géré par un microcontrôleur externe via une interface série standard (SPI) à 4 lignes. Grâce au SPI, le microcontrôleur externe peut entièrement gérer et contrôler le PL360 en accédant aux registres de périphériques internes. L'hôte utilise deux signaux supplémentaires pour contrôler le PL360 : LDO\_en et NRST. Certains GPIO peuvent être utilisés en tant que signaux d'interruption du PL360 vers le microcontrôleur externe en fonction des exigences du protocole utilisé.

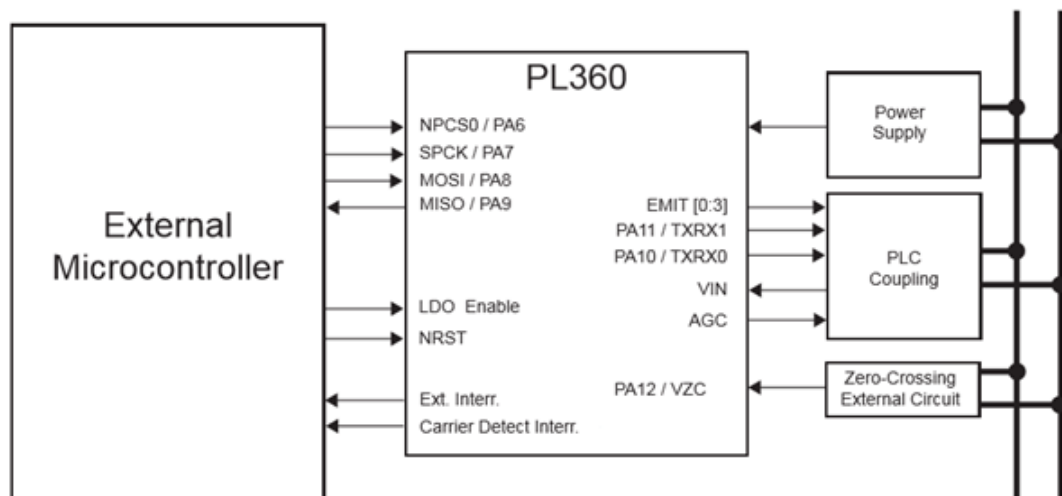


FIGURE 3.3: Schéma d'application du PL360 [20]

- **MOSI** : SPI Master Out Slave In
- **MISO** : SPI Master In Slave Out
- **LDO** : enable internal LDO regulator
- **INT** : external interruption
- **SPCK** : SPI Clock Signal
- **NPCS** : SPI Chip Select
- **NRST** : System Reset
- **CD** : Carrier Detect interrupt

### 3.3 Interfaces de communication

#### 3.3.1 Serial peripheral interface (SPI)

##### 3.3.1.1 Description

Le circuit SPI est une liaison de données série synchrone qui permet la communication avec des périphériques externes en mode esclave. L'interface périphérique série est essentiellement un registre à décalage qui transmet en série des bits de données à un dispositif maître. Pendant un transfert de données, le maître contrôle le flux de données. Le système SPI comprend deux lignes de données et deux lignes de contrôle :

- **Master Out Slave In (MOSI)** : cette ligne de données fournit les données de sortie du maître décalées vers l'entrée ou les entrées du ou des esclaves.
- **Master In Slave Out (MISO)** : cette ligne de données fournit les données de sortie d'un esclave à l'entrée du maître. Il peut ne pas y avoir plus d'un esclave transmettant des données lors d'un transfert particulier.

- **Horloge (SPCK)** : cette ligne de commande est pilotée par le maître et régule le flux des bits de données. Le maître peut transmettre des données à une variété de débits en bauds ; il y a une impulsion SPCK pour chaque bit transmis .
- **Chip Select (NPCS)** : cette ligne de contrôle permet aux esclaves d'être activés et désactivés par le matériel.

### 3.3.1.2 Caractéristiques intégrées

Interface de bus périphérique série esclave.

- Longueur de données de 8 bits à 16 bits.
- Le mode esclave fonctionne sur SPCK.

## 3.3.2 Circuits de couplage CPL

La technologie Microchip CPL est purement numérique et ne nécessite pas de convertisseur DAC/ADC externe, ce qui simplifie la circuiterie externe requise.

En règle générale, tous les modèles de référence de couplage CPL de Microchip sont généralement composés des mêmes sous-circuits :

- Etage de transmission,
- Etage de réception,
- Etage de filtrage,
- Etage de couplage.

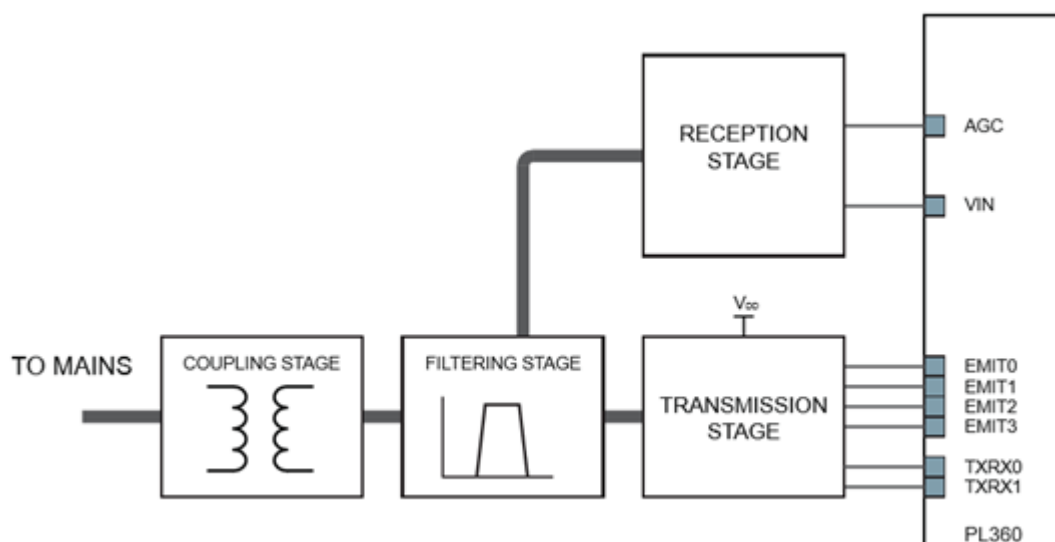


FIGURE 3.4: Circuit de couplage CPL[20]

### 3.3.2.1 Étage de transmission

L'étage de transmission adapte les signaux EMIT et les amplifie si nécessaire. Il peut être composé de :

- **Adaptateur** : il adapte les signaux EMIT soit pour contrôler l'amplificateur soit pour être filtré à l'étage suivant.
- **Amplificateur** : si nécessaire, un amplificateur de classe D générant une forme d'onde carrée de 0 à V<sub>dd</sub> est inclus.
- **Biais et protection** : fournit une composante continue et une protection contre les perturbations.

### 3.3.2.2 Étage de filtrage

A une réponse plate dans la bande de travail, il réduit les rayonnements non essentiels aux limites fixées par la régulation correspondante et bloque les interférences potentielles provenant d'autres canaux de transmissions. Cet étage a trois objectifs :

- Filtrage passe bande des composantes hautes fréquences de la forme d'onde carrée générée par l'étage de transmission.
- Adapte l'impédance d'entrée/sortie pour une réception/transmission optimale.
- Dans certain cas, un filtrage passe bande est effectué pour les signaux reçus, mais lorsque le système est destiné à un canal physique haute tension ou qui n'est pas référencé électriquement au même point, alors l'étage de filtrage doit toujours être suivi par un étage de couplage.

### 3.3.2.3 Étage de couplage

L'étage de couplage bloque la composante continue de la ligne vers/à partir de laquelle le signal est injecté/reçu (à savoir 50/60 Hz du secteur).

### 3.3.2.4 Étage de réception

Adapte le signal analogique reçu afin qu'il soit correctement capturé par la chaîne de réception interne du PL360, il consiste essentiellement en :

- Filtre anti repliement (filtre RC),
- Résistance d'atténuation pour circuit AGC (Automatic Gain Control),



- Convertisseur Analogique Numérique.

### 3.3.3 Chemin de transmission

#### 3.3.3.1 Description

Le chemin de transmission adapte le signal numérique d'entrée en un train de bits émis par les broches EMIT qui entre dans le circuit de couplage CPL. Le bloc de contrôle de la transmission gère le début et la fin de toute transmission.

Le bloc émetteur permet d'émettre un train de bits via deux paires de broches EMIT différentes, gérées indépendamment les unes des autres avec des broches TXRX différentes, indiquant la paire de broches EMIT en cours de transmission.

#### 3.3.3.2 Caractéristiques intégrées

- Capable de transmettre en utilisant deux branches de transmission indépendantes.
- Prise en charge de la commande d'un amplificateur externe (pilote externe).
- Prise en charge de la configuration sans pilote externe (pilote interne).

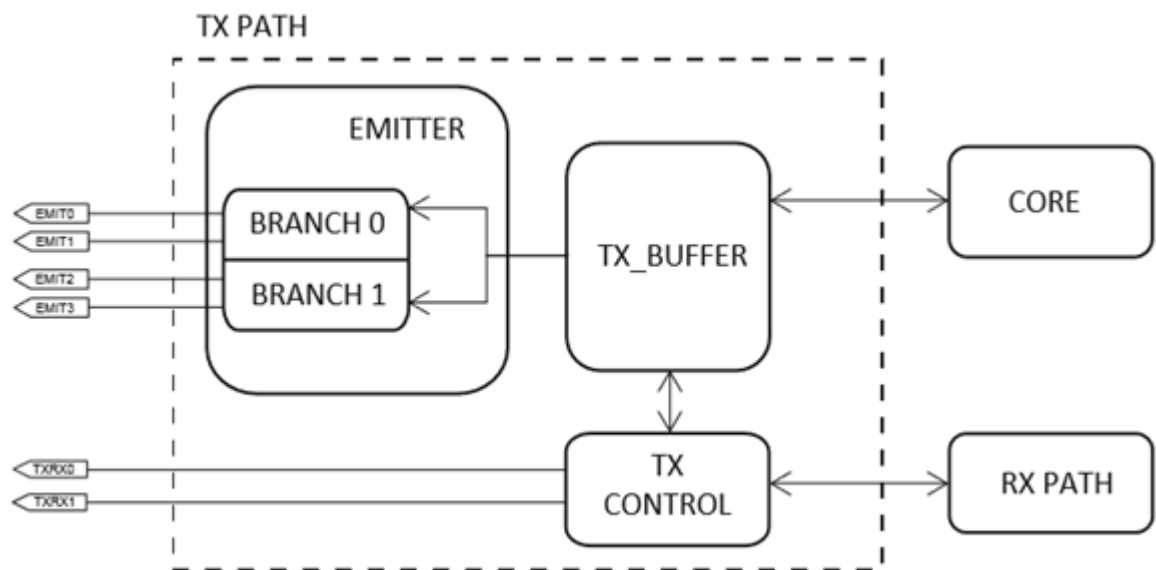


FIGURE 3.5: Diagramme en bloc du chemin de transmission[20]

### 3.3.4 Chemin de réception

#### 3.3.4.1 Description

Le chemin de réception a deux objectifs principaux :

- Adapter le signal d'entrée de l'étage de couplage externe pour échantillonner le signal reçu afin de récupérer les données.
- Calcul du passage par zéro à partir d'un signal adapté obtenu du secteur.

#### 3.3.4.2 Caractéristiques intégrées

- AGC pour maintenir un niveau d'énergie constant à l'entrée.
- Détection du passage par zéro de la tension secteur.

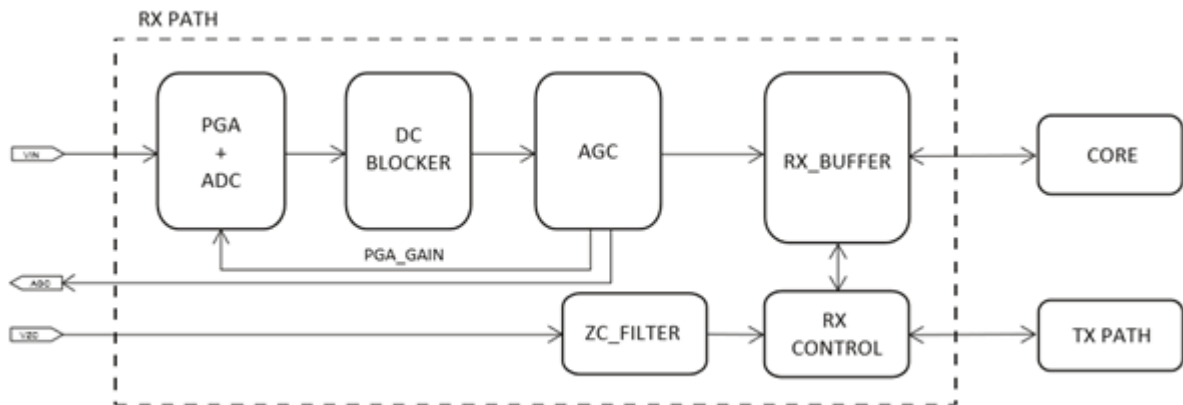


FIGURE 3.6: Diagramme en bloc du chemin de réception(1)[20]

#### 3.3.4.3 Description fonctionnelle

##### 1 - Contrôle automatique de gain (Automatic Gain Control AGC)

Le périphérique AGC adapte le signal d'entrée CPL pour qu'il soit compatible le avec PL360.

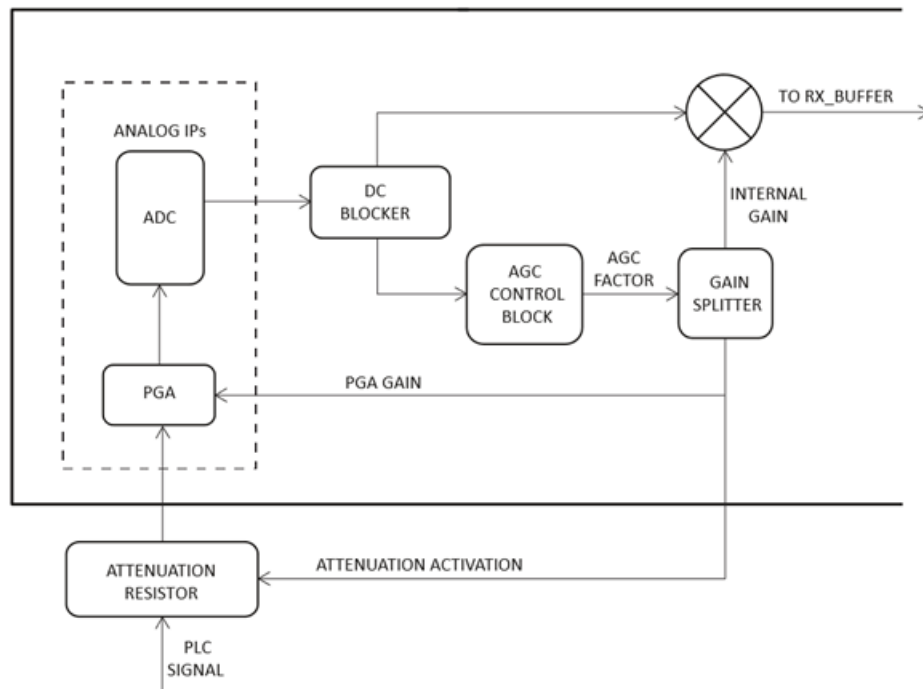


FIGURE 3.7: Diagramme en bloc du chemin de réception(2)[20]

DC Blocker : supprime l'éventuelle composante continues après l'ADC (Analog to Digital Converter), calcule la moyenne de cette composante et la soustrait au signal. Cela se fait au moyen d'un filtre passe-bas très étroit

Une fois que la composante continue a été retirée, le signal entre dans le bloc AGC. L'objectif du bloc AGC est de maintenir un niveau d'énergie constant à l'entrée du cœur. En cas de détection de bruit impulsif, l'AGC maintient le facteur d'amplification. Cependant, si l'amplitude du signal entrant diminue, l'AGC réagit rapidement et amplifie le signal.

Le signal de sortie de l'AGC est utilisé pour activer une résistance externe qui atténue le signal d'entrée. L'objectif est d'éviter la saturation de l'ADC.

## 2 - Détection de passage par zéro

Sur le PL360, la stratégie de détection du passage par zéro est basée sur un circuit de filtrage pour éliminer les transitions rapides après l'étape de passage par zéro. Les temps des fronts montants et descendants sont également mesurés par le matériel. Le

centre de l'entrée d'impulsion de niveau bas doit être aligné sur le pic de l'onde principale, bien que l'application puisse être ajustée pour corriger le délai entre l'impulsion et l'onde.

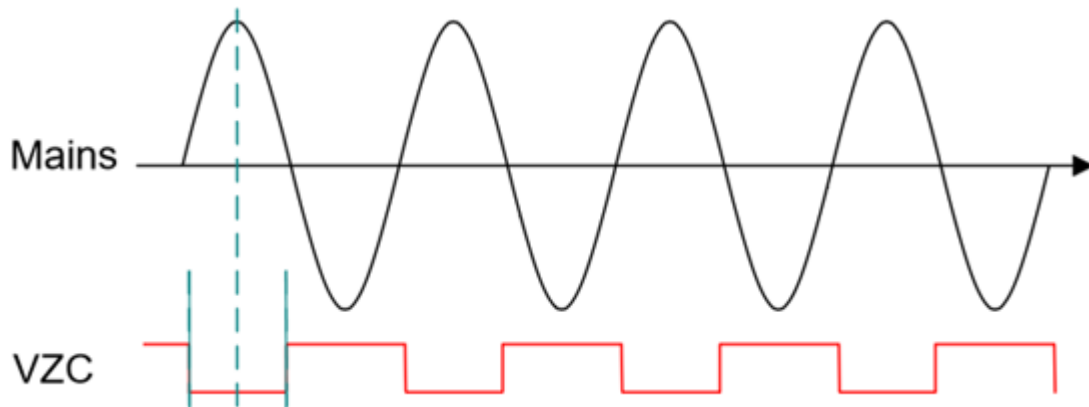


FIGURE 3.8: Passage par zéro du signal[20]

#### 3.3.4.4 Dumping Buffer

En parallèle avec RX Buffer, un tampon de vidage (DP Buffer) est implémenté. L'objectif principal du tampon DP est de stocker les échantillons à analyser aux fins de caractérisation du canal. Le contenu du tampon DP est accessible via SPI.

## 3.4 Bootloader

### 3.4.1 Description

Le bootloader charge le programme d'un maître externe dans la mémoire interne du PL360. Il permet le chargement de programmes simples ou sécurisés.

Lorsqu'un programme sécurisé est chargé, la longueur originale du programme doit être rembourrée pour devenir un multiple de 16 octets, et la longueur (nombre de blocs, où un bloc a un jeu de 16 octets) doit être spécifiée pour que la validation et le décryptage de la signature soient corrects.

Le décryptage du programme sécurisé utilise AES128. Une fois le transfert de données sécurisé a été sélectionné, le système ne démarre que lorsque la signature est apposée. Le bootloader permet également la programmation des clés de sécurité et des fusibles de contrôle de sécurité.

### 3.4.2 Caractéristiques intégrées

- Le Bootloader fonctionne sur SPCK (fréquence typique : 12Mhz, fréquence max  $\leq 16$ Mhz), de manière synchrone avec les horloges centrales et les horloges bus.
- Protocole de contrôle SPI phase fixe et polarité (SPI mode 0).
- Mot de passe pour déverrouiller le bootloader.
- Commande de programmation de fusibles.

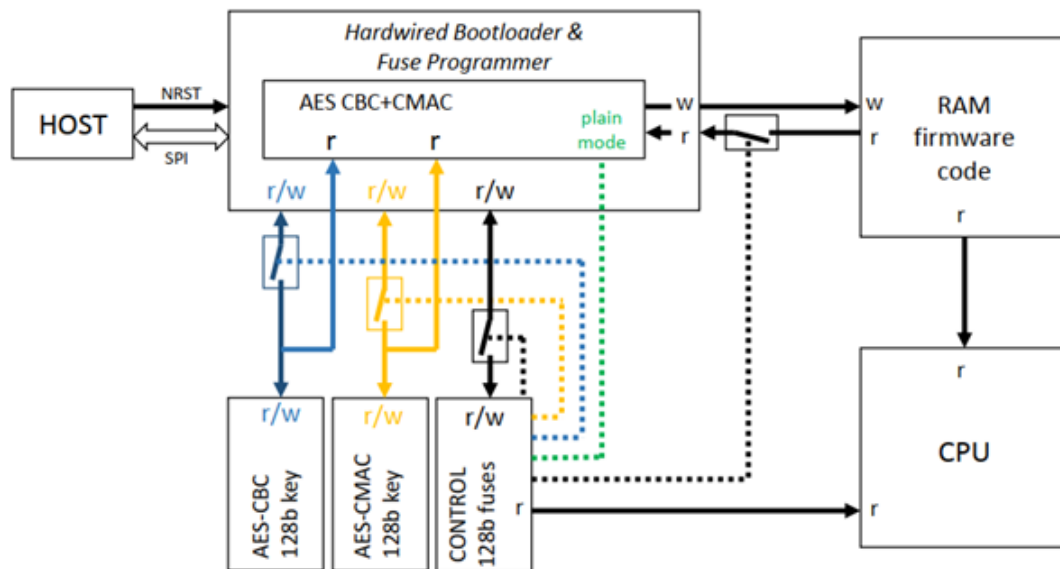


FIGURE 3.9: Schéma fonctionnel[20]

### Standard de cryptage avancé (Advanced Encryption Standard AES)

Le PL360 comprend un périphérique dédié pour effectuer des opérations AES. L'algorithme AES [14] est un chiffrement par bloc symétrique pouvant chiffrer et déchiffrer des informations. L'algorithme AES prend en charge l'utilisation de clés cryptographiques de 128, 192 et 256 bits pour chiffrer et déchiffrer des données par blocs de 128 bits.

### 3.4.3 Description fonctionnelle

Le bootloader charge le programme d'un maître externe dans la mémoire interne du PL360.

Le maître externe peut accéder à la mémoire d'instructions, à la mémoire de données

et aux registres par SPI. Le bootloader fonctionne en mode SPI 0 (CPHA=0 et CPOL=0).

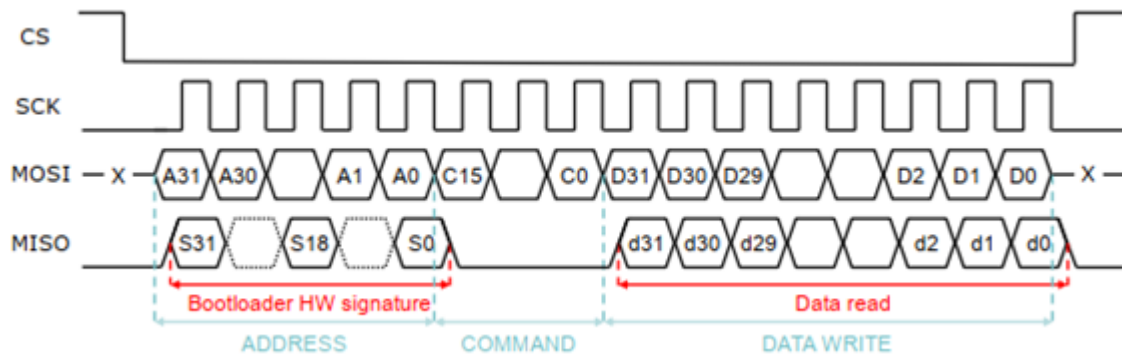


FIGURE 3.10: Schéma fonctionnel transfert de données [20]

La trame de base envoyée par le maître via le signal MOSI se compose de :

Adresse	commande	Donnée
Bloc de 32 bits	Bloc de 16 bits	N bloc de 32 bits

TABLE 3.1: Description de la trame de base

Tous les blocs de la trame de base envoyés par MOSI utilisent le format little-endian. Chaque trame reçue du maître fera l'objet d'un accusé de réception avec une signature de 32 bits par l'intermédiaire du signal MISO.

### 3.5 Implémentation de G3-PLC sur PL360

Le PL360 étant un circuit conçu pour être piloté par un périphérique hôte externe à travers des signaux de commande, SPI entre autre, la pile protocolaire suivante sera donc implémentée sur le périphérique hôte.

### 3.5.1 Architecture générale

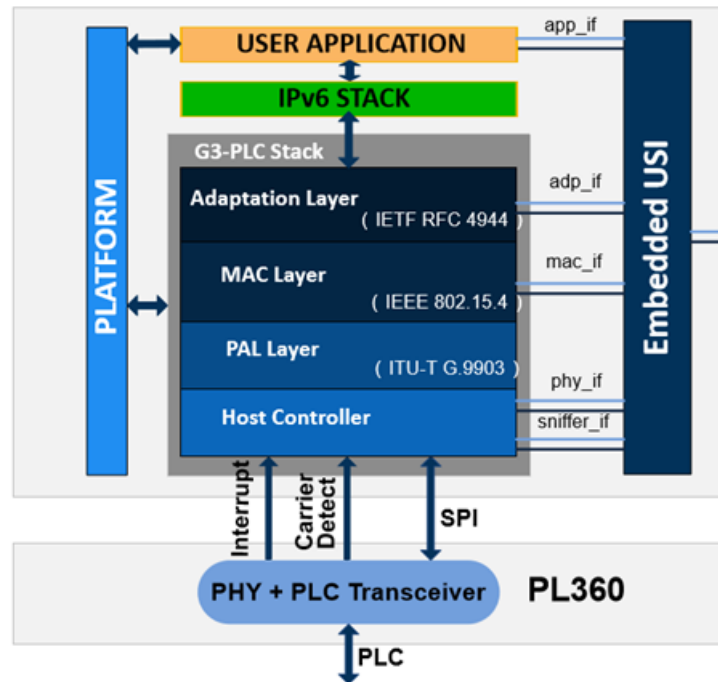


FIGURE 3.11: Schéma synoptique de la pile G3-PLC de Microchip[21]

### 3.5.2 Couche PHY

La couche PHY est chargée de la transmission et de la réception des trames. Cette couche est principalement alimentée par des interruptions avec des événements provenant du circuit PL360.

Dans le cas de l'architecture PL360, le microprogramme de la couche PHY est géré par le dispositif PL360. Par conséquent, les événements d'indication et de confirmation de données sont déclenchés par le module contrôleur hôte pour les couches supérieures.

Le PL360 étant un périphérique Flashless, son microprogramme doit être enregistré dans la mémoire du microcontrôleur hôte et chargé au démarrage.

### 3.5.3 Physical Abstraction Layer (PAL)

Cette couche fait l'abstraction de la couche PHY développée par Microchip et fournit une interface compatible avec la couche PHY G3-PLC.

Il implémente des primitives pour informer la couche MAC d'événements provenant du

canal de transmission, à savoir PhyDataConfirm, PhyAckConfirm, PhyDataIndication et PhyAckIndication.

### 3.5.4 Couche MAC

La couche MAC remplit les fonctions suivantes :

- Accès aux canaux avec CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance).
- Priorité élevée et normale.
- Segmentation de gros paquets.
- Accusé de réception de segment et retransmission de segments perdus.
- Analyse de réseau.
- Gestion de la robustesse de transmission.

### 3.5.5 Couche ADP (Adaptation Sublayer)

La couche d'adaptation (ADP) a les fonctions suivantes :

- Gestion de l'authentification du périphérique et de la distribution des clés de cryptage.
- Compression/décompression des en-têtes IPv6.
- Fragmentation/réassemblage des paquets IPv6.
- Contrôle de la propagation par diffusion/multidiffusion.
- Routage d'un message sur le réseau.
- Découverte et maintenance des routes sur le réseau.

### 3.5.6 Couches de sérialisation et interface série universelle (USI)

La sérialisation est disponible sur la couche PHY. La sérialisation de la couche PHY servira de couche API (Application Programming Interface) PHY, Ainsi, il aura les fonctions de rappel correspondantes pour gérer les événements provenant de la couche PHY. Il est utilisé pour créer des applications sur la couche PHY qui communiquent avec les outils PC fournis par Microchip.



### 3.5.7 Couche applicative

Les couches situées au-dessus de la couche ADP ne sont pas directement liées au protocole G3-PLC et ne sont donc pas incluses dans cette pile logicielle. Les utilisateurs doivent créer leurs applications en utilisant l'interface fournie par l'API ADP (celle décrite dans la spécification G3-PLC), où ils peuvent trouver des fonctions pour initialiser la pile, définir des rappels, envoyer des paquets de données, démarrer des processus de gestion et accéder à la base d'informations ADP et MAC .

## 3.6 Conclusion

Dans ce chapitre, ont été présentées les caractéristiques du PL360 :

- Liaison PL360 Secteur, avec tous les périphériques nécessaires pour la mise en forme du signal
- Communication PL360 contrôleur hôte, avec une liaison SPI
- Description interne du circuit PL360

Le PL360 étant conçue pour être contrôlé par un microcontrôleur externe, laissons ainsi le choix libre, selon l'application envisagée et les contraintes, sur le périphérique hôte. On peut donc utiliser la solution de Microchip implémentant la pile G3-PLC sur MCU présent sur la carte de développement PL360EK et l'adapter à l'application envisagée en y ajoutant une couche supplémentaire, ou bien prendre une autre carte de développement, et y implémenter les parties nécessaire pour l'établissement d'une liaison CPL avec le circuit PL360.

Le chapitre suivant présentera une solution implémentant une passerelle sur SAMA5D4, qui collecte les données sur le réseau CPL et les redirige ensuite vers internet, d'un nœud périphérique qui forme le client sur SAM4CMS16C, qui est le MCU présent de manière native sur la carte de développement PL360EK, avec l'étude de la portabilité du contrôleur hôte du PL360 sur Raspberry, ce qui permettra la mise en place d'un réseau d'objets connectés, avec une communication fiable et sécurisée.

## Deuxième partie

# Implémentation d'une solution pour le Smart Lighting

# Chapitre 4

## Conception et Développement

## 4.1 Introduction

Comme introduit dans le chapitre précédent ce chapitre sera exclusivement dédié à la présentation de la solution implémentée dans le cadre de ce projet de fin d'étude, et qui consiste en un réseau hybride (CPL, Ethernet et/ou Wifi) pour la gestion des installations d'éclairage, et la surveillance de leur consommation énergétique, pouvant être extrapolée pour la gestion d'autres installations électriques (ex : mise en place d'un réseau d'objets connectés sur le support CPL).

Notre solution consiste en un objet connecté, qui est un dispositif doté d'actionneurs, et de capteurs pour la collecte de données, d'une interface CPL pour la transmission de ces données vers une passerelle, qui à son tour fournit une interface par laquelle les données recueillies transiteront vers le réseau Internet, où elles auront comme destinataire une base de données munie d'algorithmes d'Intelligence Artificielle (IA) qui prendront des décisions adéquates avec ces dites données, pour ensuite envoyer les consignes nécessaires à l'OC, qui permettront de mettre l'actionneur dans l'état requis (ex : activation ou extinction du lampadaire).

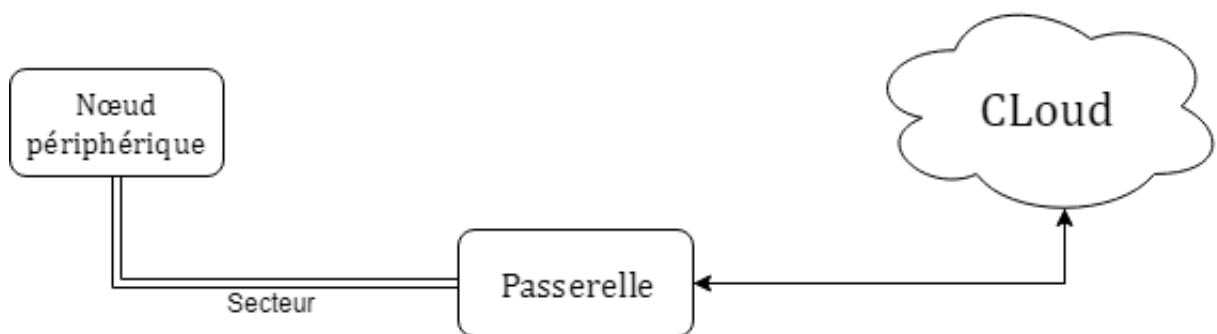


FIGURE 4.1: Réseau CPL Internet

## 4.2 Ecosystème IoT pour le Smartlighting

Ayant comme cible la gestion des installations d'éclairage dans le cadre des villes intelligentes, ce qui permettra entre autre, un gain non seulement économiques qu'écologique, le système illustré par la figure suivante a été implémenté :

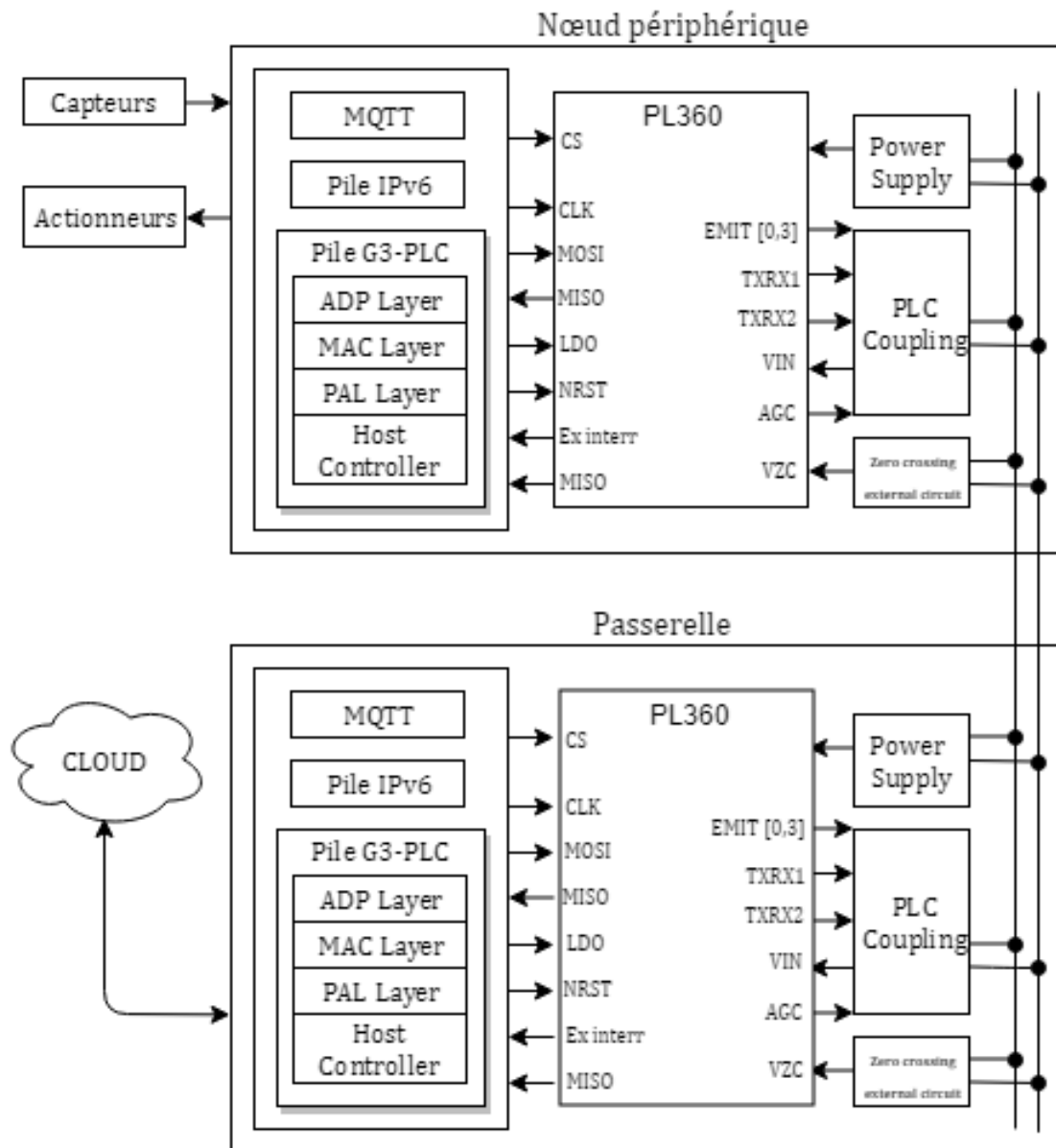


FIGURE 4.2: Ecosystème IoT

Le nœud périphérique consiste en un dispositif complet possédant toutes les ressources nécessaires pour l'accomplissement de sa tâche, qu'est la collecte et la transmission de données. Il se compose d'un circuit CPL PL360 avec tous les circuits décrits dans le chapitre 3 qui permettent l'établissement d'une liaison sur le support CPL, d'un microcontrôleur externe sur lequel a été implémentée une pile protocolaire dotée de la stack G3-PLC, IPv6 avec au niveau applicatif un protocole permettant l'envoi de message (MQTT).

Ainsi les données recueillies par ce nœud périphérique seront transmises à la passerelle via une liaison CPL, arrivé à cette passerelle dotée elle aussi de toutes les parties précédemment énoncées avec en plus une interface vers le réseau internet, par laquelle ces données transiteront vers le réseau Internet ou elles auront comme destinataire une base de données, où des algorithmes dédiés prendront des décisions sur l'état adéquat du dispositif d'éclairage, qui répondront par une consigne, transmise à son tour à la passerelle qui la dirigera vers le nœud périphérique, et qui permettra la mise en état de l'actionneur.

## 4.3 Spécifications du protocole applicatif

Le protocole applicatif adopté dans notre cas étant un protocole supportant la mise en place du concept de publication/abonnement, permettant l'envoi de messages entre les différents éléments composant le réseau d'objets connectés, avec le respect des contraintes de ressources limitées.

### 4.3.1 Principe Publication/Abonnement

La fonction de publication/abonnement est le mécanisme par lequel les abonnés peuvent recevoir des informations de diffuseurs de publications sous la forme de messages.

Un système de publication/abonnement standard implique plusieurs diffuseurs de publications et plusieurs abonnés. Une application peut à la fois être diffuseur de publications et abonné.

Le fournisseur des informations est appelé le diffuseur de publications. Les diffuseurs de publications fournissent des informations sur un sujet, sans avoir besoin de connaître les applications qu'elles intéressent. Les diffuseurs de publications génèrent ces informations sous la forme de messages, appelés publications. Ils indiquent aussi le sujet de ces messages.

L'utilisateur des informations est appelé l'abonné. Les abonnés souscrivent des abonnements qui indiquent les sujets auxquels ils s'intéressent. L'abonnement détermine

donc les publications transmises à l'abonné. Les abonnés peuvent souscrire différents abonnements et recevoir des informations de différents diffuseurs de publications.

Les informations publiées sont envoyées dans un message (ex : MQTT, HTTP ...) et leur sujet est identifié par sa rubrique. Lorsqu'il publie des informations, le diffuseur de publications les associe à une rubrique et l'abonné indique les rubriques à partir desquelles il souhaite recevoir des publications. L'abonné reçoit des informations uniquement des rubriques auxquelles il s'est abonné.

Le modèle de message de publication/abonnement offre les avantages suivants :

- Le diffuseur de publications n'a pas besoin de connaître l'abonné. Cela signifie que le diffuseur de publications n'a pas à s'inquiéter des informations de destination.
- L'abonné est découplé du diffuseur de publications. Ce découplage est utile car l'abonné ne reçoit de message que lorsqu'il s'agit de rubriques qui l'intéressent. L'abonné peut s'abonner ou se désabonner d'une rubrique à tout moment, sans affecter le diffuseur.
- La topologie de la messagerie est dynamique et flexible. Les diffuseurs et les abonnés peuvent utiliser la structure de la rubrique pour rapidement diffuser et recevoir des messages.
- Le modèle de publication/abonnement prend en charge le déploiement et l'intégration faciles.

### **4.3.2 Principe de topic (sujet)**

Techniquement, les sujets sont des files d'attente de messages. Les sujets supportent le modèle de publication/abonnement pour les clients. Logiquement, les sujets permettent aux clients d'échanger des informations avec une sémantique définie.

### **4.3.3 Langage de description des messages**

On distingue plusieurs langages de description de messages qui peuvent être mis en place par exemple : XML, CSV, JSON ... Certains de ces formats sont des formats ouverts qui favorisent l'interopérabilité, les données peuvent être réutilisées

indépendamment du logiciel qui a été utilisé pour les créer.

Notre choix s'est porté sur le format JSON du fait de sa simplicité de mise en oeuvre, et de sa compatibilité avec les langages de programmation (soit de manière native ou en y ajoutant des bibliothèques déjà existantes dans la majorité des cas)

### Décription du format JSON

JSON (JavaScript Object Notation) est un format de texte permettant la structuration de données sous un format hiérarchique, comparé à d'autres standards (CSV, XML ...) les documents JSON ont tendance à être beaucoup plus petits, conduisant à leur utilisation fréquente dans l'échange de données.

---

```
{
  {
    "Ville": "ALGER",
    "Lampadaire",
    [
      "Id:87650": "lamp1",
      "Quartier": "16200,Rue des Freres OUEDEK",
      "Etat": "on",
    ]
  }
}
```

---

LISTING 4.1: Exemple de fichier JSON

#### 4.3.4 Choix du protocole applicatif

Sans omettre les nombreuses contraintes rencontrées sur les objets connectés, le protocole de messagerie MQTT [22] a été choisi pour l'implémentation, ce qui se justifie par :

- Sa faible consommation énergétique [23],
- Particulièrement adapté pour utiliser une très faible bande passante [23],
- Permet une forte fiabilité si nécessaire [23],
- Nécessite peu de ressources (processeurs, mémoires ...) [23].



## 4.4 Protocole MQTT

MQTT est un protocole de messagerie publish/subscribe conçu de manière allégée de façon à répondre aux contraintes des objets connectés

Il implémente deux parties nécessaires à son bon fonctionnement, un client, et un serveur appelé communément Broker, par le quel tous les messages doivent transiter afin qu'il puisse les traiter et les rediriger ensuite vers leur destinataire, ce qui évite aux clients de traiter des messages qui ne leur sont pas destinés, et augmente la quantité de données surveillée du fait que toutes les données passent par le Broker.

### 4.4.1 Format des paquets

La structure du paquet MQTT comporte trois champs : **Fixed Header**, **Variable Header** et **Payload**. Le champ d'en-tête fixe est présent dans tous les paquets et sa taille est de 2 octets. Variable Length Header et Payload ne sont pas toujours présents dans les paquets, cela dépend du type de message.



FIGURE 4.3: Format des paquets MQTT

1. **Fixed Header** : Fixed Header est présent dans tous les paquets MQTT, sa structure est présentée ci-dessous

7	6	5	4	3	2	1	0	bit
Message type				Header Flags				1 <sup>st</sup> Byte
Remaining Length								2 <sup>nd</sup> Byte

FIGURE 4.4: Fixed Header structure

Message type : Le type de message représente un type de demande de connexion. Chaque valeur de type de message représente une demande de connexion différente. Le tableau 4.2 indique la valeur de la demande de message et sa signification.

Header Flags : formé de trois champs : DUP bit 3, QOS bit 1 et bit 2 et RETAIN bit 0.

- DUP : Si DUP = 0 cela signifie qu'il s'agit de la première tentative d'envoye du client. Si DUP = 1 cela signifie que le client essaie à nouveau d'envoyer le paquet précédemment transmis.
- QOS : La qualité de service (QOS) indique l'assurance de la remise du message. Il existe trois 3 QOS.

Valeur QOS	Bit 2	Bit 1	Description
0	0	0	au plus une fois ou livraison en cas de perte
1	0	1	au moins une fois ou un accusé de réception de livraison
2	1	0	une seule fois ou livraison garantie
Réserve	1	1	Implémentation future

TABLE 4.1: Table de valeurs QOS

- RETAIN : RETAIN = 1 signifie que le Broker ou le serveur stockera le paquet s'il n'y a pas d'abonné sur le même sujet que le sujet dans le paquet. Dès qu'il y aura un abonné, le Broker livrera le paquet stocké. RETAIN = 0 signifie que le Broker ne stockera pas le paquet

Remaining Length : Nombre d'octets restants dans le paquet en cours, y compris Variable Length Header et Payload

2. **Variable Header** : pas toujours présent dans les paquets MQTT. Sa structure est différente pour les différentes requêtes MQTT. Le tableau 4.2 indique dans quelle requête Variable Header est présent ou non.
3. **Payload** : données réelles à envoyer, pas toujours présentes dans les paquets MQTT. Le tableau 4.2 indique dans quelle demande la charge utile est requise ou non.

Type du message	Valeur	Description	Fixed Header	Variable Header	Payload
Réservé	0	Réservé	Présent	-	-
Connection	1	Requête de connexion du client au Broker	Présent	Présent	Présent
Connack	2	Acquittement de la requête de connexion	Présent	Présent	Non
Publish	3	Message de publication	Présent	Présent	Présent
Puback	4	Acquittement de publication	Présent	Présent	Non
Pubrec	5	Réception de la publication	Présent	Présent	Non
Pubrel	6	Publication de communiqué	Présent	Présent	Non
Pubcomp	7	Publication terminée	Présent	Présent	Non
Subscribe	8	Requête d'abonnement du client	Présent	Présent	Présent
Suback	9	Requête d'acquittement de l'abonnement	Présent	Présent	Présent
UNSubscribe	10	requête de désabonnement	Présent	Présent	Présent
UNSuback	11	Acquittement de la requête de désabonnement	Présent	Présent	Non
PINGreg	12	Requête PING	Présent	Non	Non
PINGresp	13	Réponse PING	Présent	Non	Non
Disconnect	14	Déconnexion du client	Présent	Non	Non
Réservé	15	Réservé	Présent	-	-

TABLE 4.2: Types de messages MQTT

## 4.4.2 Architecture système

Le protocole MQTT est défini selon la topologie de réseau en étoile, avec des nœuds périphériques qui s'abonne à des sujets, et un Broker qui forme le centre de cette topologie et par qui tous les paquets doivent transiter pour qu'il puisse les aiguiller vers leur destination finale.

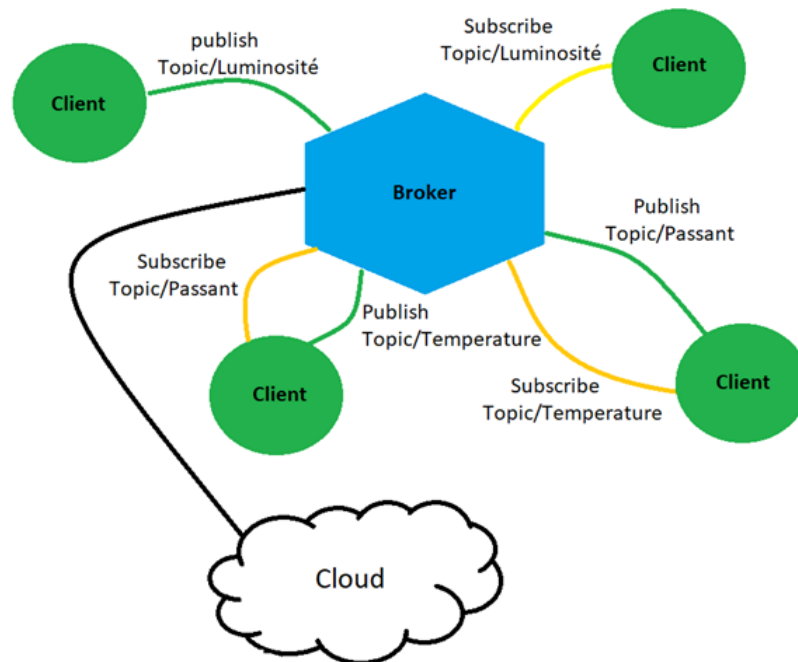


FIGURE 4.5: Architecture système MQTT

### 4.4.2.1 Broker

Le Broker MQTT est un serveur qui gère la transmission de tous les messages, un message d'un client MQTT destiné à un autre doit impérativement transiter par le Broker, pour établir une communication les clients doivent souscrire aux Topics qui « les intéressent », le broker MOSQUITTO qui est une solution open-source proposée par ECLIPSE a été choisi pour la mise en œuvre.

### 4.4.2.2 Client

Le protocole MQTT est basé sur le principe de la publication de messages et de l'abonnement à des sujets (Topic), ou "pub/sub". Plusieurs clients se connectent à un serveur et s'abonnent à des sujets qui les « intéressent ». Les clients se connectent également au serveur et publient des messages aux sujets. Les serveurs (Broker) MQTT

agissent comme une interface simple et commune à laquelle tout le monde peut se connecter, pour la mise en place du client, PAHO client qui aussi une solution libre proposé par Eclipse a été choisi.

## 4.5 Outils de développement

### 4.5.1 Cartes d'évaluation

#### 4.5.1.1 Kit d'évaluation PL360MB

PL360MB est une carte multi-usage pour les CPL basée sur l'émetteur-récepteur PL360 et sur le microcontrôleur SAM4CMS16C ARM Cortex M4. Cette carte de développement fournit une plate-forme complète pour développer un système de communication basé sur la technologie CPL et pour démontrer les fonctions de comptage d'énergie monophasées du dispositif SAM4CMS16C.

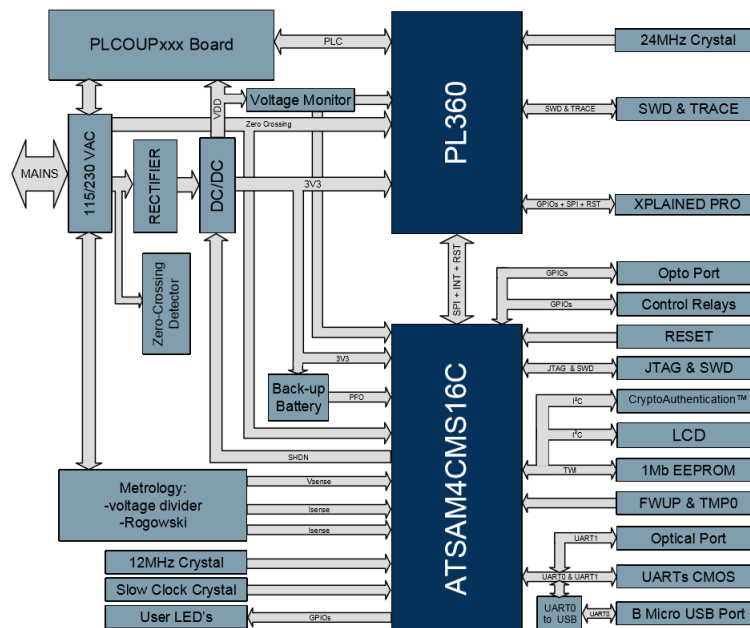


FIGURE 4.6: Architecture de la carte PL360MB [24]

#### 4.5.1.2 SAM4CMS16C

Le PL360MB est équipé d'un périphérique SAM4CMS16C. La série SAM4CM représente une famille de solutions système sur puce pour les applications de comptage résidentielles et polyphasées. Les appareils offrent une précision de métrologie allant

jusqu'à la classe 0,2 dans la plage de température industrielle.

Le jeu de périphériques comprend une référence de tension de précision, prenant en charge trois canaux de mesure de tension et quatre de courants, unité à virgule flottante (FPU), quatre USART, deux UART, deux TWI, quatre SPI, trois modules PWM 16 bits, convertisseur analogique-numérique 10 canaux, et un contrôleur LCD.

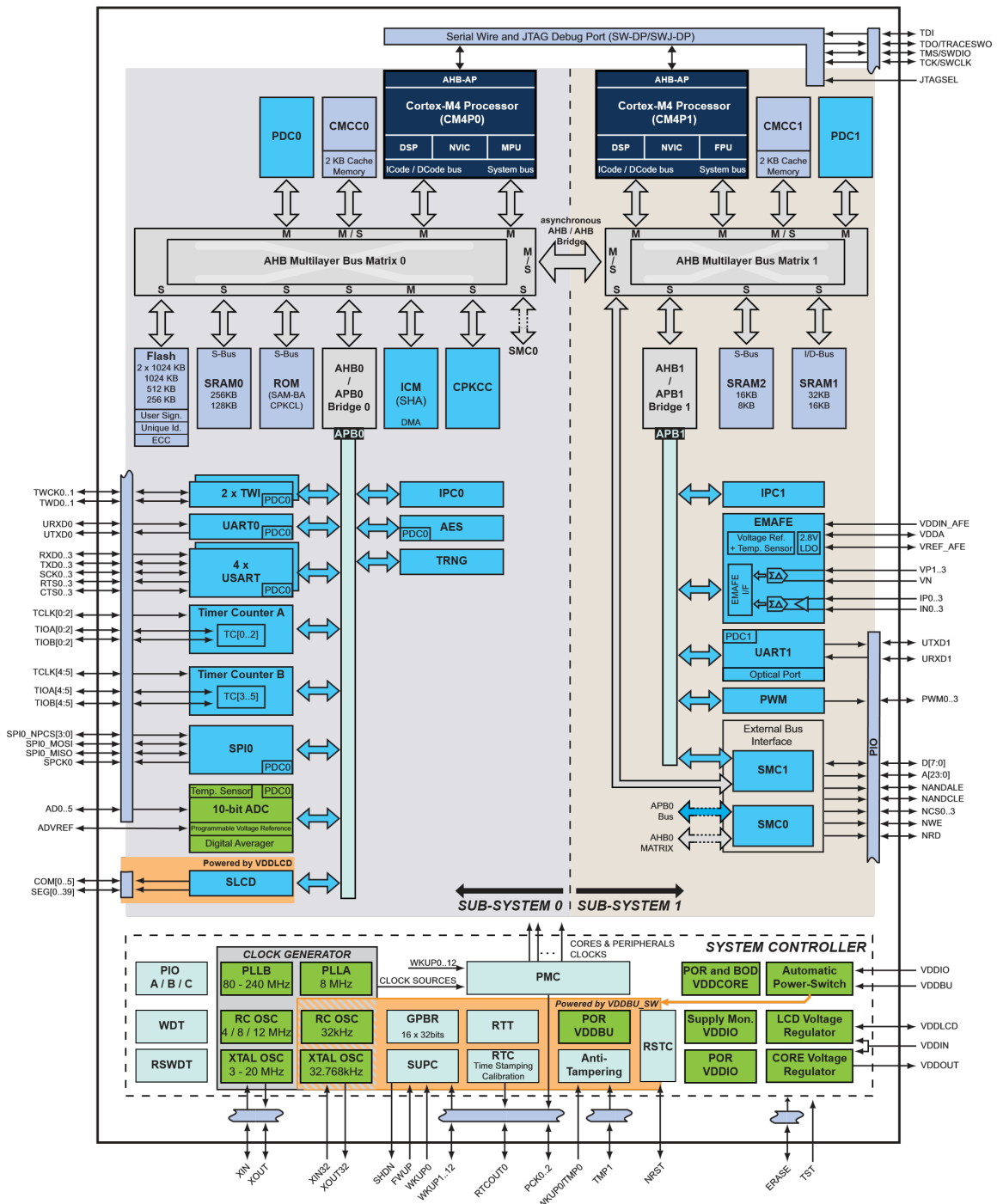


FIGURE 4.7: Diagramme en Bloc de la série SAM4CMS16C [24]

### 4.5.1.3 Carte SAMA5D4

Ce kit d'évaluation est une plateforme de prototypage basé sur un microprocesseur Arm Cortex-A5 hautes performances et éco énergétique, capable de fonctionner à 600 MHz. Il intègre le moteur Arm NEON SIMD<sup>1</sup> pour le traitement accéléré du signal, le multimédia et graphiques, ainsi qu'un cache L2<sup>2</sup> de 128 Ko pour des performances système optimales.

Le SAMA5D4 [25] offre une technologie transparente pour l'Internet des objets (IoT) et des applications grand public et industrielles. Cela comprend les panneaux de contrôle, les passerelles de communication, terminaux d'imagerie, etc.

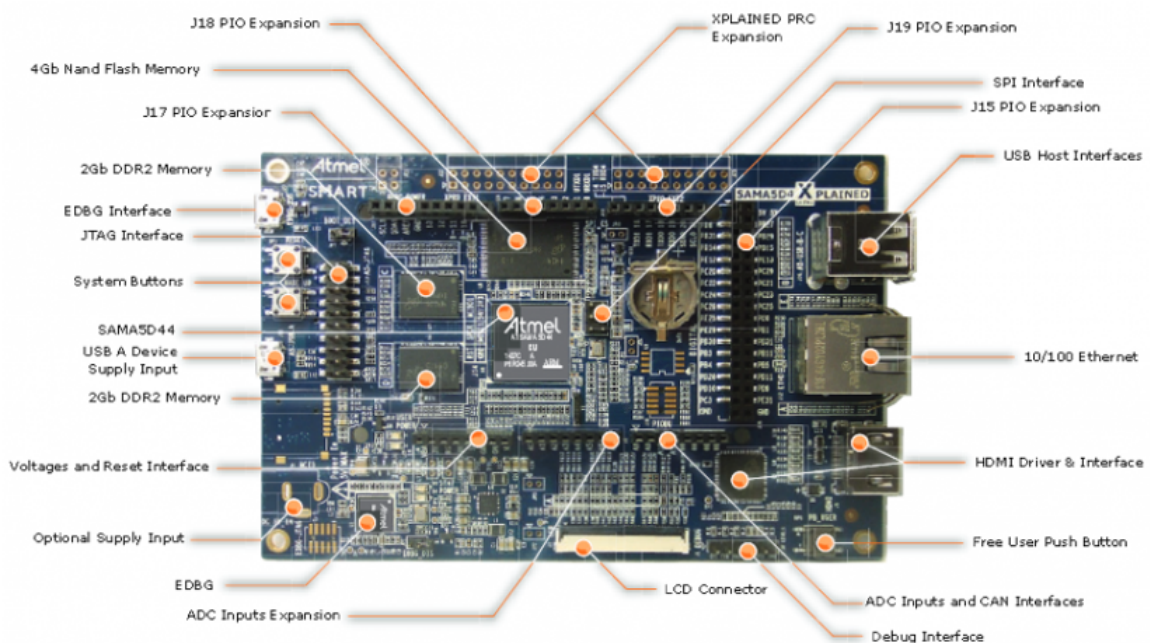


FIGURE 4.8: Carte SAMA5D4 Xplained Ultra [25]

## 4.5.2 Kits de développement software

### 4.5.2.1 Pour le nœud périphérique

la programmation et la compilation de la partie software pour le nœud périphérique a été faite à l'aide de keil SDK qui est un environnement de développement logiciel

1. Arm NEON SIMD :Unité de calcul de type SIMD, accélérant les calculs de type DSP  
 2. Cache L2 : mémoire tampon du processeur ce qui évite pendant les calcul de stocker dans la RAM.

complet pour une large gamme de microcontrôleurs basés sur Arm Cortex-M. SDK inclut l'IDE, le débogueur  $\mu$ Vision et le compilateur Arm C/C++.

#### 4.5.2.2 Pour la passerelle

##### **Buildroot :**

Buildroot [26] est techniquement un ensemble de Makefiles définissant, en fonction des options paramétrées par l'utilisateur, la manière de compiler chaque paquet sélectionné avec des options particulières. Il construit au final une distribution complète et prête à être installée sur l'environnement cible. Il fournit tous les outils nécessaires pour la compilation de toutes les briques principales composant un système d'exploitation :

- Toolchain : qui est l'ensemble des outils à compiler qui nous permettra ensuite d'avoir un environnement capable de faire de la compilation croisée (cross-compiler) depuis notre architecture hôte
- Bootloader : code permettant le chargement du système d'exploitation
- Noyau : cœur du système d'exploitation il s'occupe de fournir une interface à l'espace utilisateur pour l'exploitation de ressources matérielles
- Rootfs : systèmes de fichier (organisation hiérarchique des fichiers au sein d'un système d'exploitation) composant le système d'exploitation

La construction d'un OS avec Buildroot se fait principalement selon les étapes suivantes :

- Choix d'une configuration par default pour une cible (quand elle est disponible), ex : `make raspberrypi3_defconfig`
- Choix de la toolchain, Bootloader, version du noyau, fichier DT... grâce à l'interface de configuration accessible via la commande : `make menuconfig`



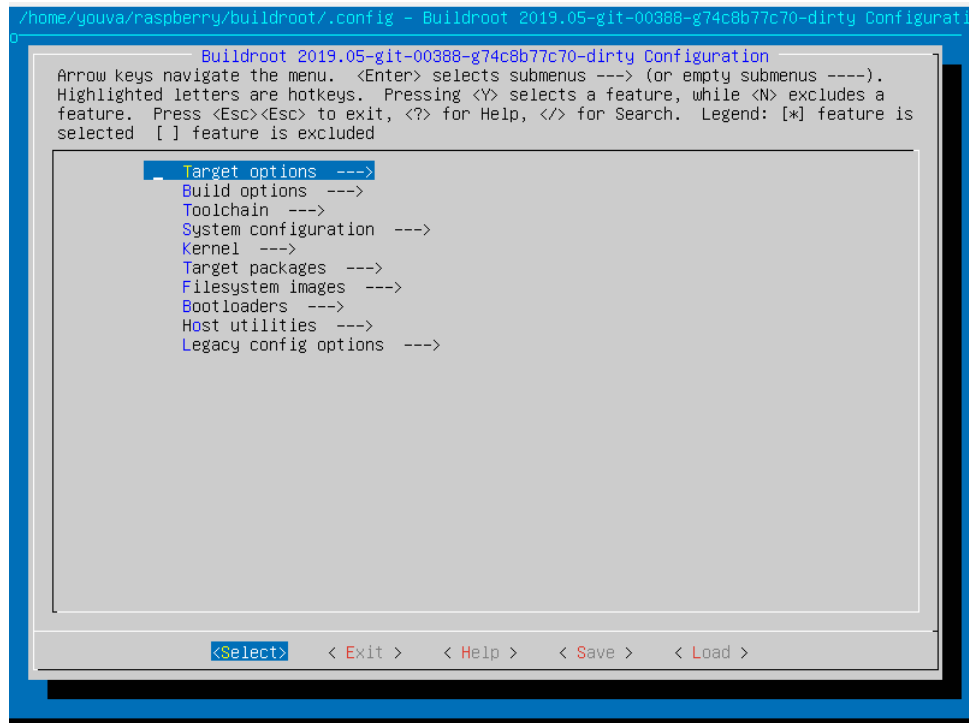


FIGURE 4.9: Interface de configuration

- Configuration du noyau via la commande : `make linux-menuconfig`

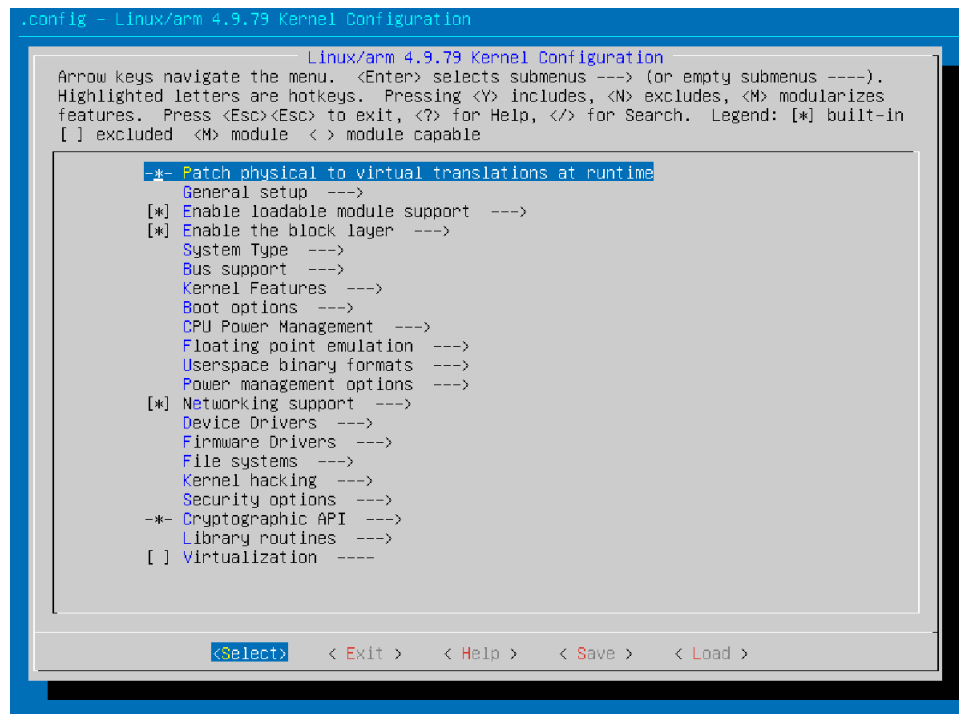


FIGURE 4.10: Interface de configuration du noyau

- Lancement de la compilation via la commande : `make`

- Après la fin de la compilation, les images finales à écrire sur la mémoire de la cible se trouve dans le dossier `output/images/`.
  - `uboot.bin` : Image du bootloader
  - `uImage` : Image du noyau Linux
  - `rootfs.tar` : système fichier complet

## 4.6 Flot de développement

Le developement et la compilation des sources s'est faite pour la passerelle à l'aide de la toolchain integrer de Buildroot qui permet une compilation croisé (applications, pilotes, noyau ...) pour une architecture cible, dans notre cas la toolchain étant Uclibc. Ajouté à cette toolchain, des fichier Makefile adapté au noyau servent de recette pour la génération des binaires.

Coté noeud périphérique elle s'est faite en se basant sur KEIL SDK pour le developement sur PL360EK, et avec Buildroot pour la portabilité sur Raspberry.

Ainsi ces outils de développement permettent une compilation à partir d'une architecture hôte, qui aura comme resultat des fichier binaire à chargés directement sur le périphérique cible(noeud périphérique, passerelle).

Exemple de mise en oeuvre utilisant la toolchain de Buildroot :

---

```
make ARCH=arm CROSS_COMPILE=~/.g3-linux/buildroot-at91/output/host/bin/  
arm-buildroot-linux-uclibc-gnueabihf
```

---

## 4.7 Conception/développement du noeud périphérique

Le client étant la partie responsable de la collecte et transmission des données à travers le réseau CPL vers la passerelle, il assure aussi la réception des consignes envoyées à travers la passerelle par les algorithmes responsable de la prise de décision. Pour la mise en place du client deux solutions ont été envisagées :

- Ajout à la pile protocolaire fournit par Microchip (G3-PLC, IPv6) pour le MCU SAMA4CMS16C présent sur le kit de développement PL360EK une couche MQTT client, avec un paramétrage adéquat pour les différentes couches
- Adaptation du travail fait pour la carte SAMA5D4 pour migrer sur un RASPBERRY, qui est une solution offrant plus de flexibilité, tant sur les ressources que les services offerts (voir section 4.9)

### 4.7.1 Flot de développement

Le développement et la compilation des sources pour le noeud périphérique s'est faite en se basant sur KEIL SDK, les binaires obtenu ont été chargé par la suite sur le microcontrôleur SAM4CMS16C du kit PL360-EK à l'aide d'un programmeur STLink.

### 4.7.2 Conception/Développement en utilisant la carte PL360EK

#### 4.7.2.1 Pile protocolaire

La carte de développement PL360EK étant dotée d'un microcontrôleur SAM4CMS16C [27], il semble judicieux de mettre en place un client MQTT [22] sur ce dispositif, pour ce faire des codes fournit par Cyclone TCP [28] ont été ajoutée à la stack G3-PLC, IPv6 déjà disponible pour ce dernier afin de former un client MQTT, dans la suite nous présentons quelques explication sur les différentes couches qui forme l'application finale.

Cyclone TCP propose une architecture générale de son modèle en couches qui correspond aux sept couches du modèle OSI avec les différents protocoles disponibles pour chacune d'elle (figure ci-dessous).

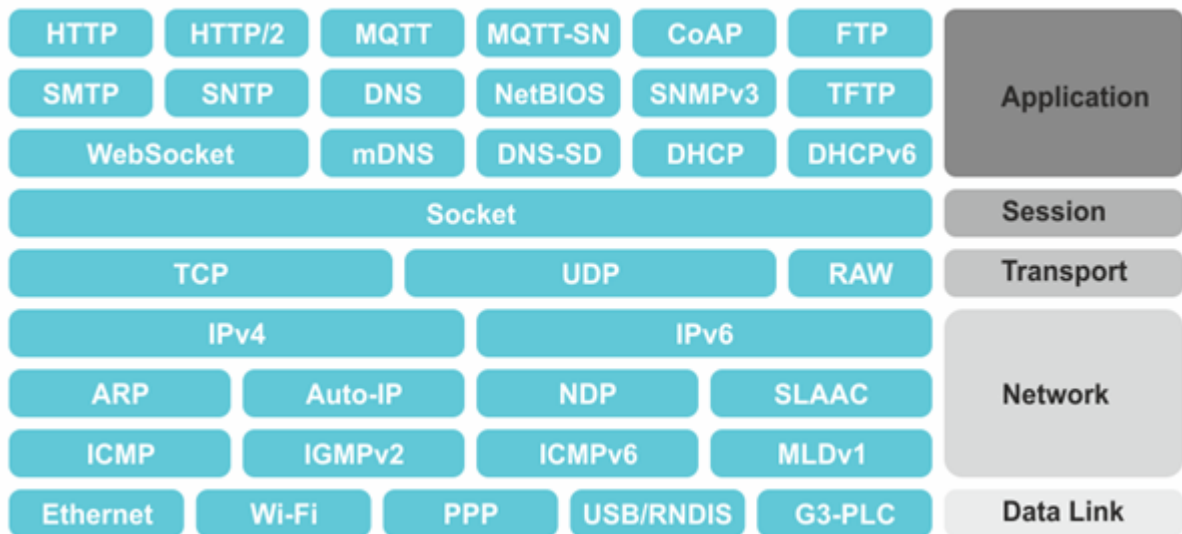


FIGURE 4.11: Modèle en couche de Cyclone TCP [28]

#### 4.7.2.2 Couche applicative

La figure suivante illustre la machine d'état implémentée pour la gestion du nœud périphérique.

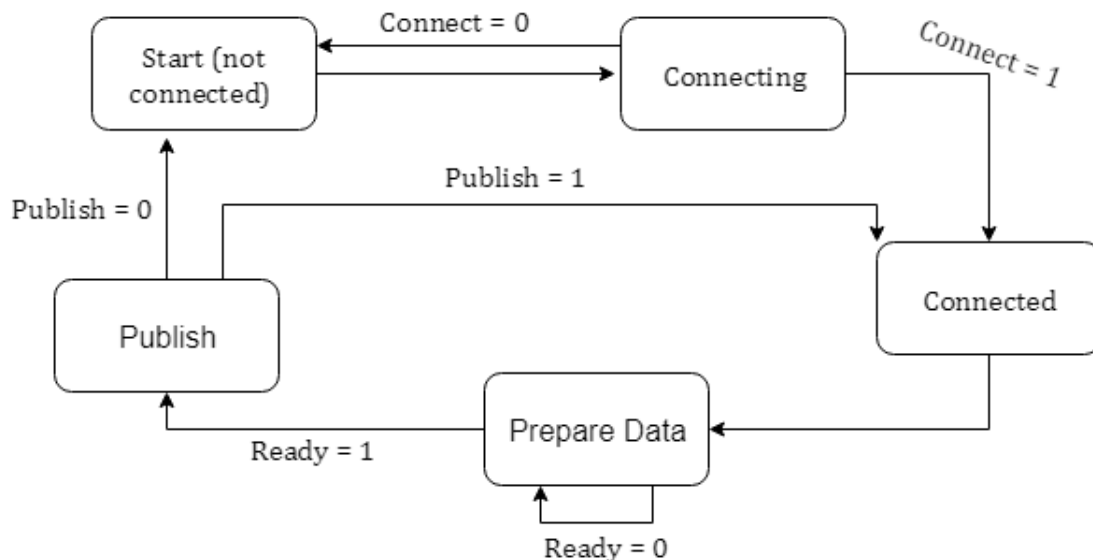


FIGURE 4.12: Machine d'état du nœud périphérique

---

```
void app_process(void)
{
    error_t error;
    char_t buffer[16];
    int light;
    while(1)
    {
        //Check connection state
        if(connectionState == APP_STATE_NOT_CONNECTED)
        {
            //Update connection state
            connectionState = APP_STATE_CONNECTING;
            //Try to connect to the MQTT server
            error = mqttConnect();
            //Failed to connect to the MQTT server?
            if(error)
            {
                //Update connection state
                connectionState = APP_STATE_NOT_CONNECTED;
                //Recovery delay
                osDelayTask(2000);
            }
            else
            {
                //Update connection state
                connectionState = APP_STATE_CONNECTED;
            }
        }
    }
}
```

---

LISTING 4.2: Code implémentant la machine d'état

L'état initial étant `APP_STATE_NOT_CONNECTED`, on bascule vers l'état `APP_STATE_CONNECTING` pour établir la connexion avec le broker.

### Formation des messages :

La formation des messages à transmettre se fait en appelant la primitive

---

```
mqttClientFormatPublish ( MqttClientContext *context, const char_t *topic,
const void *message, size_t length, MqttQosLevel qos, bool_t retain)
* Contexte : Pointeur vers le contexte du client MQTT.
* Topic : Nom du sujet.
* Message : Message Payload.
* length : Longueur du payload du message.
* qos : QoS niveau de qualite de service a utiliser lors de la publication du message.
* Retain : indique si le message doit etre conserve par le Broker.
```

---

### 4.7.2.3 Couche session

Ainsi la fonction `SocketConnect ( )` de la couche `Session` sera appelée pour transmettre l'ordre d'ouverture de la connexion vers les couches plus basses.

---

```
//TCP transport protocol
if(context->settings.transportProtocol == MQTT_TRANSPORT_PROTOCOL_TCP)
{
    //Set timeout
    error = socketSetTimeout(context->socket, context->settings.timeout);
    //Check status code
    if(!error)
    {
        //Connect to the MQTT server using TCP
        error = socketConnect(context->socket, serverIpAddr, serverPort);
    }
}
}
```

---

Le protocole MQTT supportant uniquement le Protocole TCP raison pour laquelle la couche session a été configurée pour utiliser les API du protocole TCP en ajoutant la ligne :

---

```
Setting.transportProtocol = MQTT_TRANSPORT_PROTOCOL_TCP
```

---

Et Pour établir la connexion avec la Passerelle, quelques paramètres ont été définis comme l'adresse IP et le port de cette dernière.

---

```
#define APP_SERVER_NAME "fd00:0:2:781d:1122:3344:5566:0" //"fe80::c8c1:7cb5:6aa8:655d"
#define APP_SERVER_PORT 1883 //MQTT over TCP
```

---

### 4.7.2.4 Couche transport

Au final un socket qui contient toutes les informations nécessaires pour atteindre la passerelle aura été formé, ainsi la couche transport a pu être appelée en utilisant l'API `tcpConnect ( )` qui associe l'adresse du nœud périphérique ou l'adresse de la source aux messages.

---

```
//Save port number and IP address of the remote host
socket->remoteIpAddr = *remoteIpAddr;
socket->remotePort = remotePort;
//Select the source address and the relevant network interface
//to use when establishing the connection
```

---

```
error = ipSelectSourceAddr(&socket->interface,
    &socket->remoteIpAddr, &socket->localIpAddr);
//Any error to report?
if(error) return error;
//Make sure the source address is valid
if(ipIsUnspecifiedAddr(&socket->localIpAddr))
    return ERROR_NOT_CONFIGURED;
//Get exclusive access
osAcquireMutex(&netMutex);
//Establish TCP connection
error = tcpConnect(socket);
```

---

Ensuite la fonction `tcpConnect ( )` met les messages en forme en ajoutant les entêtes spécifiques à cette couche, cette opération est illustrée par le bout de code présenté ci-dessous.

---

```
//Format TCP header
segment->srcPort = htons(socket->localPort);
segment->destPort = htons(socket->remotePort);
segment->seqNum = htonl(seqNum);
segment->ackNum = (flags & TCP_FLAG_ACK)htonl(ackNum):0;
segment->reserved1 = 0;
segment->dataOffset = 5;
segment->flags = flags;
segment->reserved2 = 0;
segment->window = htons(socket->rcvWnd);
segment->checksum = 0;
segment->urgentPointer = 0;
```

---

#### 4.7.2.5 Couche réseau

Une fois le message mis en forme, il est mis dans un buffer ensuite la couche IP est appelée.

##### Formation du paquet

---

```
//Format IPv6 pseudo header
pseudoHeader.destAddr = ipAddr->ipv6Addr;
pseudoHeader.length = htonl(length);
pseudoHeader.reserved = 0;
pseudoHeader.nextHeader = IPV6_ICMPV6_HEADER;
```

---

---

## Transmissin des paquets

---

```
//Form an IPv6 packet and send it
error = ipv6SendDatagram(interface, &pseudoHeader->ipv6Data, buffer, offset, ttl);
```

---

La primitive `ipv6SendDatagram` sert à envoyer un datagramme IPv6

---

```
* interface : Interface reseau sous-jacente
* pseudoHeader : pseudo en-tete IPv6
* tampon : Tampon en plusieurs parties contenant le payload
* offset : Offset au premier octet de la charge utile
* hopLimit (ttl) : valeur limite de sauts, valeur default utilisee
lorsque ce parametre est egal a zero.
```

---

### 4.7.2.6 Couche Liaison

Une fois toutes les étapes précédentes terminées, les paquets résultants ont été mis dans un buffer et en utilisant l'API fournis par la couche ADP de la pile G3-PLC, ces paquets sont envoyés vers le circuit PL360 à travers les couches de cette dernière (voir chapitre 3).

---

```
AdpDataRequest(us_data_length, adp_sdu, uc_nsdu_handle++, true, 0x00);
```

---

**Prototype de la fonction transmission et explication des arguments :** La primitive `AdpDataRequest` demande le transfert d'une PDU vers un ou plusieurs autres périphériques à travers la pile G3-PLC.

---

```
* u16NsduLength (us_data_length) : La taille du NSDU, en octets ; Jusqu'a 1280
* pNsdu (adp_sdu) : Le NSDU a envoyer doit etre un paquet IPv6 valide
* u8NsduHandle (uc_nsdu_handle++) : La poignee du NSDU a transmettre.
Ce parametre est utilise pour identifier le message.
* bDiscoverRoute (true) : Si VRAI, une procedure de decouverte d'itineraire
sera effectuee avant l'envoi de la trame.
* u8QualityOfService (0x00) : La qualite de service (QoS) demandee de la trame
a envoyer. Les valeurs autorisees sont :0x00 = priorite normale, 0x01 = priorite
elevee.
```

---



### 4.7.3 Préparation et publication des messages

Une fois la connexion établie, des données capteurs ont été récupérées et transmises à la passerelle.

---

```
//Initialize status code
error = NO_ERROR;
light = memread();
if( light > SEUIL ){
    strcpy(buffer, "jour");
}
else{
strcpy(buffer, "nuit");
}

//Send PUBLISH packet
error = mqttClientPublish(&mqttClientContext,
"board/buttons/1",buffer, strlen(buffer);
MQTT_QOS_LEVEL_1, TRUE, NULL);
//Failed to publish data?
if(error)
{
    //Close connection
    mqttClientClose(&mqttClientContext),
    //Update connection state
    connectionState = APP_STATE_NOT_CONNECTED;
    //Recovery delay
    osDelayTask(2000);
}
```

---

Comme pour la partie souscription et publication, les messages envoyés par la couche applicative passent à travers les différentes couches de la pile en effectuant des appels de fonction de la même façon qu’illustrée dans les parties précédentes pour l’établissement d’une connexion.

Ainsi une stack complète aura été formée, avec l’établissement d’une connexion au niveau applicatif, ouverture des sockets, formation de paquets IPv6, compression de ces derniers par 6LoWPAN et aiguillage vers le circuit PL360.

La figure suivante illustre les différents API utilisés pour la publication d’un message :

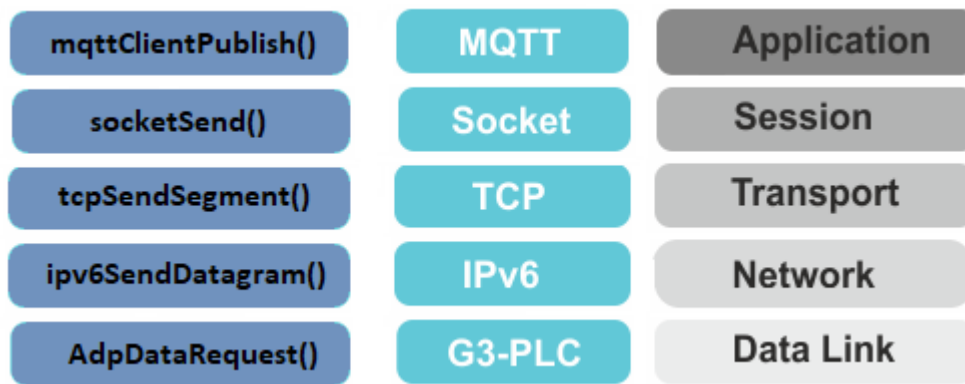


FIGURE 4.13: Publication d'un message

## 4.8 Conception/Développement de la Passerelle

Constitué d'une carte SAMA5D4 associé au circuit PL360, la section suivante sera dédiée à la description de ce dispositif.

### 4.8.1 Flot de développement

Comme indiqué plus haut le développement de la partie software pour la passerelle s'est fait à l'aide de Buildroot, et de la toolchain intégré à Buildroot qui permet la compilation croisé du noyau linux, des applications et de la partie Driver et Device Tree nécessaire à l'interfaçage du circuit PL360.

### 4.8.2 Modèle de la couche applicative

Le dispositif SAMA5D4 [25] offre deux modes de fonctionnement :

- Mode microcontrôleur
- Operating System (OS) mode

Dans la suite, seul le mode OS nous intéresse. Pour le mettre en place, un système d'exploitation adapté à ce dispositif a été préparé, compiler et installer sur la carte SAMA5D4.

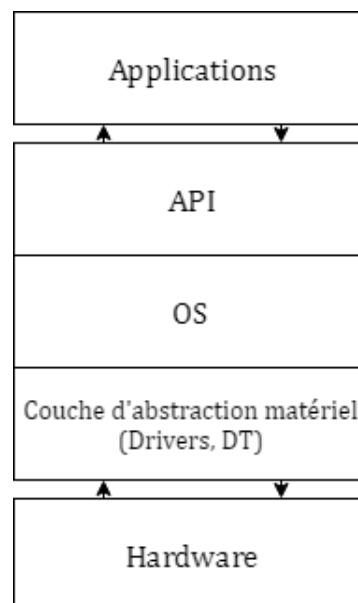


FIGURE 4.14: Modèle de la couche applicative

### 4.8.3 Système d'exploitation pour SAMA5D4

Pour ce faire Microchip fournit un OS compatible avec la carte SAMA5D4 avec une couche HAL (Hardware Abstraction Layer) qui permet la portabilité d'un noyau linux sur ce dispositif. La préparation de cette OS s'est faite, à l'aide de Buildroot, majoritairement selon les étapes suivantes :

- Choix de la bibliothèque C : conformément aux codes fournis par Microchip pour l'implémentation de la stack G3-PLC, la bibliothèque standard Uclibc a été choisie dans les options de configuration de Buildroot [26].
- Activation de l'interface TUN/TAP qui permet la simulation d'un réseau de virtuel pour l'envoi de paquet de l'espace utilisateur vers le noyau et vice-versa.
- Choix de construction du noyau avec le fichier Device Tree customisé [29]
- Installation d'un broker MQTT (Mosquitto)
- Installation d'un client MQTT(Paho)

Seuls les points jugés les plus pertinents seront abordés, les autres sortent du cadre de ce travail.

### 4.8.4 Couche Device Driver

Pour permettre le contrôle du PL360 à partir de la carte SAMA5D4 des pilotes (Drivers) ont été ajoutés au noyau linux installé, leurs rôles consistent entre autre :

- Interfaçage des signaux déclarés dans le fichier DT (section 4.8.5) (LDO\_en, Reset, Interrupt, Carrier Detect)
- Accès au pool mémoire réservé pour le stockage et le brouillage/décryptage des données, reçu ou à envoyer vers le circuit.
- Chargement des firmware pour le démarrage du circuit PL360 (Bande de fréquences)
- Configuration des options SPI (SPI.mode, SPI.bits\_per\_word ...)
- Envoi de commande pour le contrôle du bootloader (commande de boot, déblocage du bootloader, Reset)
- Vérification de l'identifiant de la puce PL360

Ces pilotes combinés au fichier DT décrit ci-dessous forme une couche d'abstraction matérielle ce qui permet l'interfaçage des caractéristiques matériel, de l'adressage ou du mappage des GPIOs, et donc un développement au niveau applicatif sans se soucier des contraintes matériel.

#### 4.8.5 Device Tree (DT)

Le Device Tree est un fichier décrivant les composants matériels d'un dispositif permettant l'installation d'un système d'exploitation, de sorte que le noyau du système d'exploitation puisse utiliser et gérer ces composants, y compris le ou les CPU, la mémoire, les bus et les périphériques, il fait partie de la couche d'abstraction matériel, d'une autre manière le noyau voie les ressource matériel à travers le fichier Device Tree. Pour la mise en œuvre de la solution proposée, il est nécessaire de modifier ce dit fichier afin que la carte SAMA5D4 puisse contrôler le circuit PL360. Pour ce faire, quatre signaux ont été déclarés dans ce fichier :

1. **LDO\_en** : signal permettant le contrôle du régulateur de tension nécessaire pour l'alimentation correcte du PL360.
2. **Reset** : signal permettant la remise à zéro du dispositif PL360 et périphériques.
3. **Interrupt** : signale d'interruption du PL360 vers le périphérique hôte.
4. **Carrier Detect** : indique la présence ou non d'une porteuse sur le réseau CPL.

```
spi2: spi@fc01c000 {
    cs-gpios = <&pioD 17 0>;
    status = "okay";
    pinctrl-0 = < &pinctrl_spi2 &pinctrl_pl360_gpio_xpro2>;
    pinctrl-names = "default";

    plcboard_pl360amb {
        compatible = "microchip,pl360";
        spi-max-frequency = <8000000>;
        reg = <0>;
        interrupt-parent = <&pioB>;
        interrupts = <14 0>;
        microchip,gpio-ldo_en = <&pioE 18 GPIO_ACTIVE_HIGH>;
        microchip,gpio-reset = <&pioE 19 GPIO_ACTIVE_HIGH>;
        microchip,gpio-carrier-detect = <&pioE 27 GPIO_ACTIVE_HIGH>;
    };
};
```

FIGURE 4.15: Déclaration des signaux de contrôle du PL360 sur SAMA5D4

Ajouté à ces quatre signaux un pool d'espace mémoire doté d'un brouilleur AES (Securam) a été réservé dans le fichier DT.

```
securam: sram@fc058000 {  
    compatible = "mmio-sram";  
    reg = <0xfc058000 0x2044>;  
    clocks = <&securam_clk>;  
    #address-cells = <1>;  
    #size-cells = <1>;  
    ranges = <0 0xfc058000 0x2044>;  
    status = "okay";  
  
    plcram@1000 {  
        reg = <0x1000 0x100>;  
        pool;  
    };  
};
```

FIGURE 4.16: Réserve d'un pool dans la securam

Après préparation du fichier de configuration de BUILDROOT [2] permettant la mise en place d'un noyau linux avec les caractéristiques précédemment décrites, la compilation et l'installation de cette image sur la carte SAMA5D4 ont été effectuées.

#### 4.8.6 Préparation de la carte PL360MB

La carte de développement PL360EK [24] permettant l'évaluation du circuit PL360 est initialement conçue pour que ce circuit soit piloté par le microcontrôleur fournie sur cette carte, quelques soudures doivent être effectuées pour permettre un contrôle à partir d'un hôte externe à cette carte de développement.

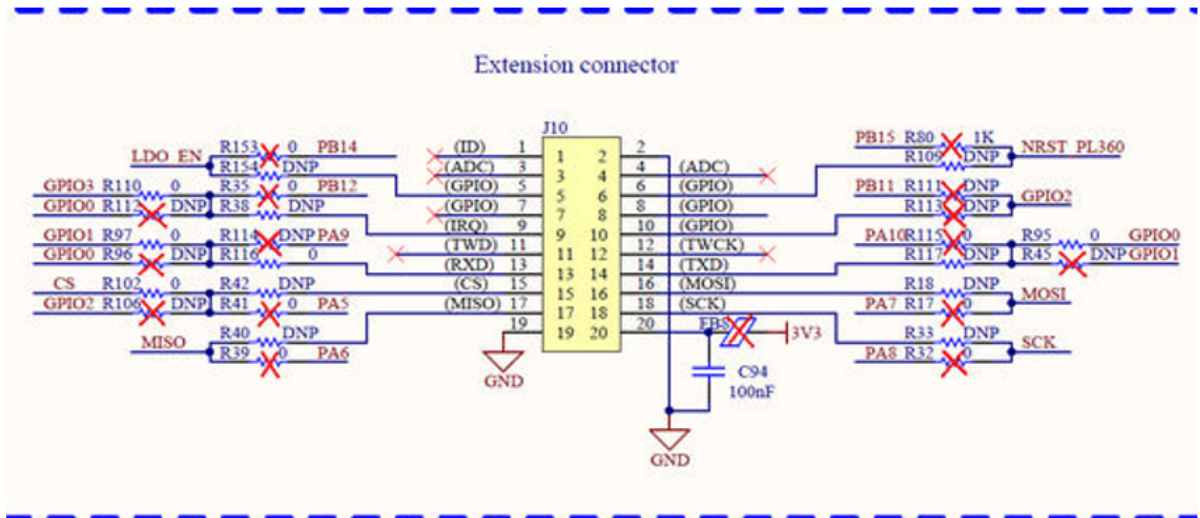


FIGURE 4.17: Modifications du connecteur XPLAINED PRO pour une connexion directe au PL360

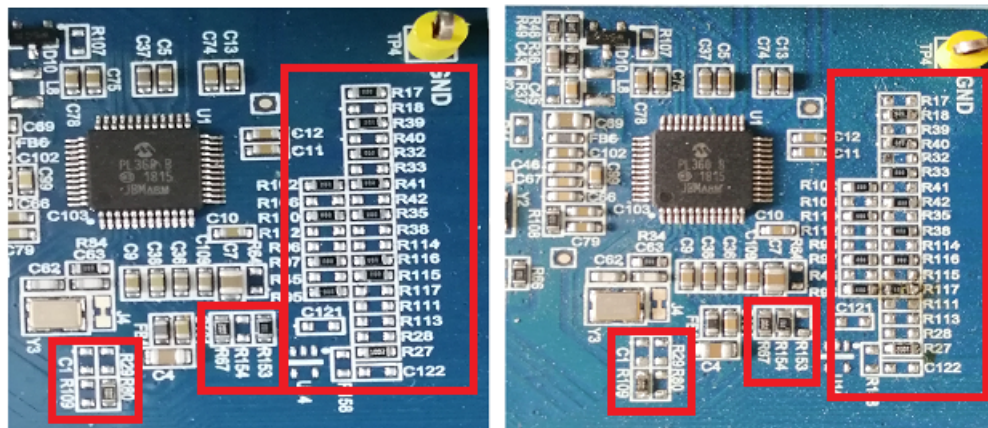


FIGURE 4.18: Etat initial, après modification

### 4.8.7 Test et validation du fonctionnement de la passerelle

Pour la validation du fonctionnement de la passerelle, le PLC Sniffer fournit par Microchip a été utilisé, il permet la vérification de la présence d'un trafic sur le réseau CPL, la figure ci-dessous montre le réseau test.

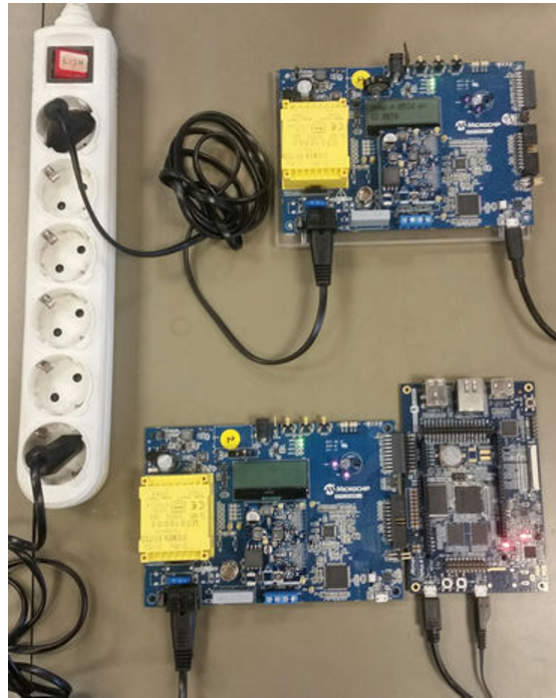


FIGURE 4.19: Réseau test

#### 4.8.7.1 Approche de test

Génération de paquet MQTT au niveau applicatif, ce qui permettra le test de toute la pile, aigüillage de ces derniers vers le PL360 qui les injectera sur le secteur, de l'autre côté un sniffer bas niveau a été utilisé, ce qui permettra la confirmation du fonctionnement dans le cas où il y aurai détection de trafic

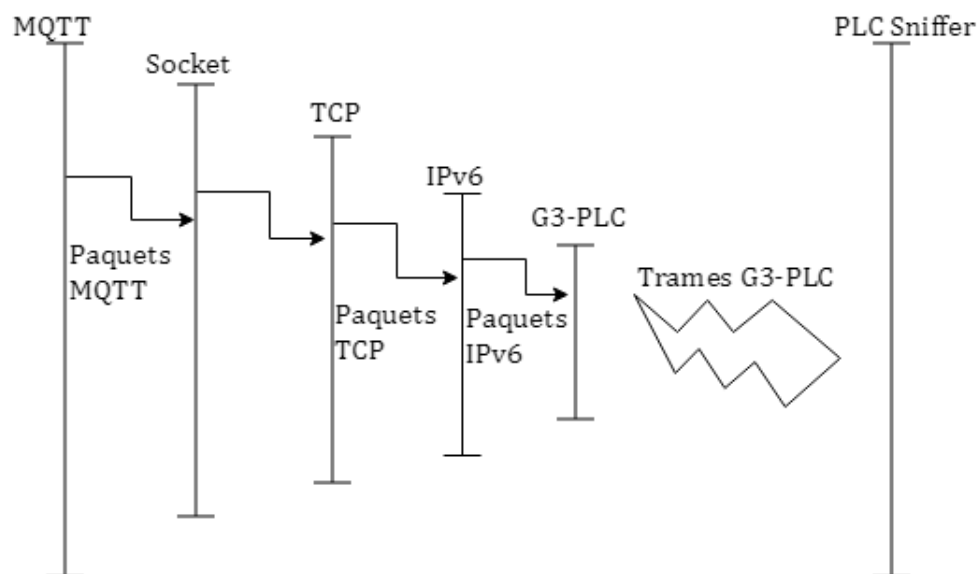


FIGURE 4.20: Approche de validation du fonctionnement de la passerelle



4.8.7.2 Résultats et discussion

PduType	ndTyp	SecurityEn	arityL	imeCc	eylnd	Num	omp	Addr	estPA	tAddr	addrM	rcPAI	SrcAddress	Length
-	-	-	-	-	-	-	-	-	-	-	-	-	-	9
DAT	LBP...	0	-	-	-	0x52	1	2	0x78...	0x00...	3	0x78...	0x1808301000230000	118
-	-	-	-	-	-	-	-	-	-	-	-	-	-	63
DAT	LBP...	0	-	-	-	0x53	1	2	0x78...	0x00...	3	0x78...	0x1808301000230000	37
-	-	-	-	-	-	-	-	-	-	-	-	-	-	63
DAT	LBP...	0	-	-	-	0x6A	1	3	0x78...	0x18...	2	0x78...	0x0000	63
-	-	-	-	-	-	-	-	-	-	-	-	-	-	9
DAT	LBP...	0	-	-	-	0x54	1	2	0x78...	0x00...	3	0x78...	0x1808301000230000	118
-	-	-	-	-	-	-	-	-	-	-	-	-	-	63
CMD	BREQ	0	-	-	-	0x55	0	2	0xFF...	0xFF...	0	-	-	13
CMD	BREQ	0	-	-	-	0x56	0	2	0xFF...	0xFF...	0	-	-	13
BCN	-	0	-	-	-	0xEA	0	0	-	-	2	0x78...	0x0000	18
DAT	LBP...	0	-	-	-	0x57	1	2	0x78...	0x00...	3	0x78...	0x1808301000230000	37
-	-	-	-	-	-	-	-	-	-	-	-	-	-	18
DAT	LBP...	0	-	-	-	0x6B	1	3	0x78...	0x18...	2	0x78...	0x0000	63
-	-	-	-	-	-	-	-	-	-	-	-	-	-	9
DAT	LBP...	0	-	-	-	0x58	1	2	0x78...	0x00...	3	0x78...	0x1808301000230000	118
-	-	-	-	-	-	-	-	-	-	-	-	-	-	63
DAT	LBP...	0	-	-	-	0x59	1	2	0x78...	0x00...	3	0x78...	0x1808301000230000	37
-	-	-	-	-	-	-	-	-	-	-	-	-	-	63
DAT	LBP...	0	-	-	-	0x6C	1	3	0x78...	0x18...	2	0x78...	0x0000	63
-	-	-	-	-	-	-	-	-	-	-	-	-	-	9
DAT	LBP...	0	-	-	-	0x5A	1	2	0x78...	0x00...	3	0x78...	0x1808301000230000	118
-	-	-	-	-	-	-	-	-	-	-	-	-	-	63
DAT	LBP...	0	-	-	-	0x5B	1	2	0x78...	0x00...	3	0x78...	0x1808301000230000	37
-	-	-	-	-	-	-	-	-	-	-	-	-	-	63
DAT	LBP...	0	-	-	-	0x6D	1	3	0x78...	0x18...	2	0x78...	0x0000	63
-	-	-	-	-	-	-	-	-	-	-	-	-	-	9
DAT	LBP...	0	-	-	-	0x5C	1	2	0x78...	0x00...	3	0x78...	0x1808301000230000	118
-	-	-	-	-	-	-	-	-	-	-	-	-	-	63
CMD	BREQ	0	-	-	-	0x5D	0	2	0xFF...	0xFF...	0	-	-	13
BCN	-	0	-	-	-	0xEB	0	0	-	-	2	0x78...	0x0000	18

FIGURE 4.21: PLC Sniffer

De la figure précédente on peut remarquer la détection des paquets générés par la passerelle, et ceci en examinant par exemple la colonne **SrcAddress** qui indique l'adresse de la source de la trame, à laquelle est associé **PduType** qui est une donnée. On peut donc conclure du bon fonctionnement de la passerelle, et par conséquent, du bon fonctionnement de toute la pile protocolaire ainsi que de l'interface du PL360.

Une autre approche de test plus simple mais aussi efficace étant la vérification des signaux sur le réseau électrique avec filtrage de la composante 50Hz.

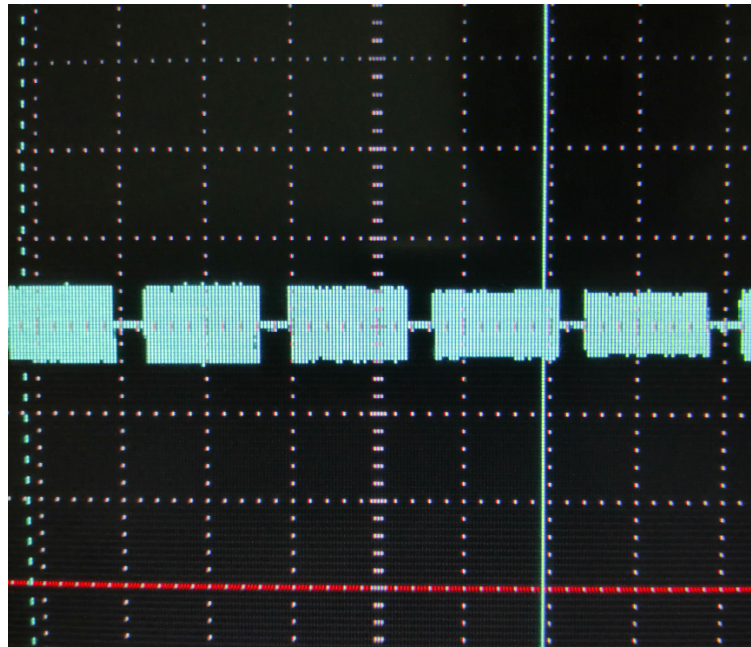


FIGURE 4.22: Détection frame G3-PLC

De la figure précédente on distigue clairement une trame G3-PLC, ce qui valide le fonctionnement.

## 4.9 Etude de la portabilité sur Raspberry Pi

### 4.9.1 Intérêt

La possibilité de mise en place d'un OS sur la carte Raspberry forme un intérêt majeur, tant sur le temps de développement que sur la multitude des services offerts, en ajoutant à ça un prix concurrentiel relativement a toutes les possibilités offertes, et a sa puissance de calcul permettant un traitement parallèle optimal, dans notre cas on pourrait imaginer, grâce au Wifi qui est natif sur le Raspberry, récupérer les données émise par les véhicules connectés, et les transmettre via une liaison CPL vers des centres dédiés, et ainsi permettre une gestion complète des réseaux routiers avec un coût qui en serait presque dérisoire.

### 4.9.2 Approche d'adaptation

Comme pour le contrôle du PL360 avec la carte SAMA5D4, la préparation du Raspberry se fait de la même manière avec :

- Préparation d'un OS adapté.
- Choix de 'Uclibc' comme bibliothèque C standard.
- Modification du fichier DT.

```

&spi0 {
    cs-gpios = <&gpio 8 0>;
    status = "okay";
    pinctrl-0 = <&spi0_pins &spi0_cs_pins>;
    pinctrl-names = "default";

    plcboard_pl360amb {
        compatible = "microchip,pl360";
        spi-max-frequency = <8000000>;
        reg = <0>;
        #address-cells = <1>;
        #size-cells = <0>;
        interrupt-parent = <&gpio>;
        interrupts = <5 0>;
        microchip,gpio-ldo_en= <&gpio 6 GPIO_ACTIVE_HIGH>;
        microchip,gpio-reset = <&gpio 26 GPIO_ACTIVE_HIGH>;
        microchip,gpio-carrier-detect = <&gpio 13 GPIO_ACTIVE_HIGH>;
    };
};

```

FIGURE 4.23: Déclaration signaux de commande du PL360 sur Raspberry

- Ajout des pilotes nécessaires au noyau Linux pour l'interfaçage du circuit PL360.
- Installation d'un client et/ou Broker MQTT.

### 4.9.3 Contraintes

Pour ce faire et comme indiqué précédemment toutes les étapes qui ont été faite sur la SAMA5D4 ont été adaptées pour un Raspberry, la seule différence étant que le Raspberry ne possède pas de manière native de Securam et ne supporte pas SPI.bits\_per\_word=16 qui permettrait au Bootloader de charger un programme sécurisé d'un hôte externe vers la mémoire interne du PL360.

### 4.9.4 Solutions retenues

Dans le but de répondre aux contraintes précédemment exposées, deux solutions sont proposées :

- Ajouter à la carte Raspberry un chip qui fera office de Securam, et modifier les pilotes du PL360 pour qu'au lieu d'envoyer directement des mots codé sur 16 bits (SPI.bits\_per\_word=16), envoyé le même mot en deux fois codé sur 8 bits (SPI.bits\_per\_word=8).
- L'autre solution étant la modification des pilotes pour le chargement d'un programme non sécurisé, dans ce cas plus besoin de Securam.

## 4.10 Conclusion

Ce chapitre étant l'aboutissement de tout le travail mené dans le cadre de ce projet de fin d'étude, il a permis la mise en évidence :

- Du positionnement de toutes les parties exposées dans les chapitres précédents, MQTT, protocole IPv6, Pile G3-PLC . . .
- Liaison CPL pour la transmission de données entre passerelle et nœud périphérique,
- Passerelle, permettant la mise en place d'un réseau hybride, alliant ainsi les liaisons CPL et le réseau internet, ce qui permet un développement moins contraignant de réseau d'objets connectées, en se basant sur les installations électriques préexistante, et sur la technologie et algorithmes avancé déjà implémenter sur internet.
- Client permettant la transmission de message MQTT à l'aide du circuit PL360 de MICROCHIP

# Conclusion générale

L'évolution de l'internet des objets et la naissance des villes intelligentes nécessite une capacité de prise en charge de toute sorte de capteurs et des données qu'ils recueillent.

Pour fournir la base de cette prise en charge, nous avons présenté dans ce projet, une solution s'appuyant sur la technologie CPL comme moyen de transmission de données, cette technologie nous a permis d'utiliser les lignes électriques déjà en place sans avoir recours à de nouvelles installations beaucoup plus coûteuses. Ce travail est spécifique au smart lighting pour une meilleure automatisation de l'éclairage dans les villes du futur.

En première lieu, nous avons fait une présentation des architectures générales de l'IoT et de la technologie CPL ainsi qu'une comparaison entre les protocoles les plus répandu répondant à notre problématique. Ce qui nous a permis de choisir le protocole G3-PLC qui est plus adéquat pour ce projet.

Ensuite, l'aperçu sur les couches de la pile protocolaire G3-PLC que nous avons donnée a permis de sélectionner le circuit PL360 en vue de sa satisfaction au cahier des charges qu'impose ce protocole.

La dernière partie concerne l'implémentation du système complet dédié au smart lighting basé sur la technologie CPL. Dans cette partie, nous avons exposé l'architecture générale pour laquelle nous avons opté en mettant en valeur l'implémentation de toute la chaîne, partant du nœud périphérique jusqu'à la passerelle.

Au final, nous proposons une solution alternative pour pouvoir porter le travail qui

a été fait sur d'autres nano ordinateurs, pour cela, les différents paramètres à prendre en compte pour y parvenir ont été présentés.

Grâce à ce travail, nous avons pu acquérir une bonne expérience dans le domaine de l'embarqué, en commençant par la maîtrise de différents outils, la familiarisation avec la multitude de protocoles pouvant être utilisés et allant jusqu'à la conception d'un système complet qui relie toutes les connaissances apprises

### **Perspectives**

Le travail effectué étant dédié au smart lighting, néanmoins, il peut être enrichi ou étendu pour d'autres utilisations, notamment :

- Réalisation d'un circuit embarquable comportant le PL360, un hôte de contrôle et tous les périphériques nécessaires à la transmission sur le réseau CPL,
- Développement d'une solution sur cloud pour le smart lighting,
- Adaptation de cette solution pour des applications de smart lighting,
- Ajout au nœud périphérique la capacité de communiquer avec les véhicules connectés et permettre ainsi une gestion complète des réseaux routiers.

# Bibliographie

- [1] F. Moulin M. Tlich, A. Zeddami and F. Gauthier. Indoor powerline communications channel characterization up to 100 mhz ; part ii : Time-frequency analysis. 2008.
- [2] T. Banwell and S. Galli. On the symmetry of the power line channel. 2001.
- [3] H. Philipps. Modelling of powerline communication channels.
- [4] M.Zimmermann and K.Dostert. A multi path model for the powerline channel.
- [5] Y. Mortazavi A. Dabak I. H. Kim M. Nassar, J. Lin and B. Evans. Local utility power line communications in the 3-500 khz band : Channel impairments, noise, and standards. 2012.
- [6] O.G.Hooijen. On the relation between network-topology and power line signal attenuation.
- [7] J. Newbury H. Ferreira, L. Lampe and T. Swart. Power line communications : Theory and applications for narrowband and broadband communications over power lines. 2011.
- [8] S. Galli and T. Banwell. A novel approach to the modeling of the indoor power line channel—part i : Transfer function and its properties.
- [9] A. M. Tonello. Wideband impulse modulation and receiver algorithms for multiuser power line communications.
- [10] F. Moulin M. Tlich, A. Zeddami and F. Gauthier. Indoor power line communications channel characterization up to 100 mhz—part ii : one-parameter deterministic model.

- 
- [11] IEEE. Ieee standard for low-frequency (less than 500 khz) narrowband power line communications for smart grid applications. 2013.
- [12] Zbigniew Sadowsk. Comparison of plc-prime and plc-g3 protocols,xii international school on nonsinusoidal currents and compensation.
- [13] ERDF. G3-plc physical layer specification.[en ligne].
- [14] Aes.[en ligne].[consulté le 15 février 2019] disponible sur [https://fr.wikipedia.org/wiki/advanced\\_encryption\\_standard](https://fr.wikipedia.org/wiki/advanced_encryption_standard).
- [15] Ipv6.[en ligne].[consulté le 05 mai 2019]. disponible sur <https://fr.wikipedia.org/wiki/ipv6>.
- [16] 6lowpan.[en ligne].[consulté le 05 février 2019] disponible sur <https://fr.wikipedia.org/wiki/6lowpan>.
- [17] R. Hinden Nokia S. Deering Cisco. Rfc2460.[en ligne].[consulté le 15 mars 2019].disponible sur <https://tools.ietf.org/html/rfc2460>.
- [18] Microsoft Corporation C. Schumacher Danfoss. N. Kushalnagar Intel Corp, G. Montenegro. Rfc4919[en ligne].[consulté le 18 avril 2019]. disponible sur <https://tools.ietf.org/html/rfc4919>.
- [19] J. Hui D. Culler Arch Rock G. Montenegro Microsoft Corporation, N. Kushalnagar Intel Corp. Rfc4944.[en ligne].[consulté le 13 mars 2019]. <https://tools.ietf.org/html/rfc4944>.
- [20] Microchip. Pl360-data-sheet-ds70005364b.pdf.[en ligne].[consulté le 23 décembre 2018].
- [21] Microchip. G3-plc firmware stack user guide.pdf.[en ligne].[consulté le 23 février 2019].
- [22] Wikipedia. .[en ligne].[consulté le 1 avril 2019].disponible sur <https://fr.wikipedia.org/wiki/mqtt>.
- [23] Arlen Nipper (EuroTech) Andy Stanford-Clark (IBM). .[en ligne].[consulté le 30 avril 2019]. disponible sur <http://mqtt.org/>.



- 
- [24] Microchip. Pl360ek datasheet.[en ligne].[consulté le 30 décembre 2019]. disponible sur <http://ww1.microchip.com/downloads/en/devicedoc/pl360-ek-user-guide-ds50002730a.pdf>.
- [25] Microchip. Sama5d4 datasheet.[en ligne].[consulté le 29 mars 2019]. disponible sur <http://ww1.microchip.com/downloads/en/devicedoc/60001525a.pdf>.
- [26] Buildroot. .[en ligne].[consulté le 12 février 2019].disponible sur <https://buildroot.org>.
- [27] Microchip. Sam4cms16c datasheet.[en ligne].[consulté le 29 avril 2019]. disponible sur <http://ww1.microchip.com/downloads/en/devicedoc/atmel-11203-32-bit-cortex-m4-microcontroller-sam4cm32-sam4cm16-sam4cm8-sam4cm4.pdf>.
- [28] Cyclone TCP. .[en ligne].[consulté le 12 avril 2019].disponible sur [https://www.oryx-embedded.com/cyclone\\_tcp.html](https://www.oryx-embedded.com/cyclone_tcp.html).
- [29] Linux. .[en ligne].[consulté le 20 avril 2019].disponible sur [https://elinux.org/device\\_tree\\_reference](https://elinux.org/device_tree_reference).