

الجمهورية الجزائرية الديمقراطية الشعبية

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التعليم العالي والبحث العلمي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة الدكتور الطاهر مولاي سعيده

Université Saida Dr Tahar Moulay –

Faculté de TECHNOLOGIE



MEMOIRE

de fin d'étude pour l'obtention du Diplôme de MASTER

Spécialité : Electronique

Option : Instrumentation

Par : **ZIANI Mimoun**

BELARBI Omar

Sujet

Etude et réalisation d'un réseau de capteurs sans fil à base des modules XBee.

Soutenu publiquement, Septembre 2021, devant le jury composé de :

Mr. ARBOUCHE Omar	Professeur	Univ. Saida	Président
Mr. DAHANI Ameer	MCA	Univ. Saida	Rapporteur
Mr. DINE Khaled	MCA	Univ. Saida	Examineur
Mlle. SAHRAOUI Halima	Doctorante	Univ. Saida	Co-Encadreur

Année Universitaire 2020/2021

Remerciement

- ❖ Un grand remerciement à ALLAH de nous avoir donné le pouvoir, le courage et la volonté à fin de parvenir à finaliser ce travail.
- ❖ Nous adressons nos remerciements, en particulier à notre encadreur Mr. DAHANI AMEUR d'avoir accepté de nous encadrer ainsi que pour ses grands efforts, sa rigueur, sa disponibilité et ses remarques précieuses qui ont contribué efficacement à l'avancement de ce travail.
- ❖ Un immense remerciement à notre Co-encadreur Mlle. SAHRAOUI Halima paix et bénédiction sur lui.
- ❖ De même nous tenons à remercier le président et les membres du jury qui ont accepté d'évaluer ce modeste travail.
- ❖ Et nous profitons, à cette occasion d'exprimer notre gratitude à tous nos professeurs qui ont contribué sérieusement à faire réussir notre formation.

DEDICACE

A mes chers parents qui m'ont soutenu durant mon existence et ma scolarité.

A mes chères sœurs et mes chers frères.

A tous mes amis.

Résumé

Un réseau de capteurs sans fil est un réseau ad hoc d'un grand nombre de nœuds, qui sont des micro-capteurs capables de recueillir et de transmettre des données d'une manière autonome. La position de ces nœuds n'est pas obligatoirement prédéterminée. Ils peuvent être aléatoirement répartis dans une zone géographique, intitulée « champ de captage » correspondant au terrain concerné pour le phénomène capté.

Les données captées par les nœuds sont acheminées grâce à un routage multi-saut à un nœud, considéré être un « point de collecte », appelé nœud-puits (ou sink). Ce dernier peut être connecté à l'utilisateur du réseau (via Internet, un satellite ou un autre système). L'utilisateur peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises et récolter les données requises captées par le biais du nœud puits.

Dans ce projet de fin d'étude, on a fait une étude et une réalisation d'un réseau de capteurs sans fil en utilisant des modules XBee avec le protocole ZigBee. Le nœud End Device 1 (capteur LM35+Arduino+XBee) et le nœud End Device 2 (capteur DHT11 + Arduino + XBee) émettent leurs données au module XBee du nœud collecteur de réception. Le nœud de réception comporte également une carte raspberry pi pour le traitement du signal à la réception et identification de la trame de réception.

La comparaison des résultats obtenus au niveau de l'émission et de la réception ont confirmé la faisabilité, le fonctionnement et la fiabilité de ce réseau de capteurs sans fil pour la transmission des données produites en temps réel et à distance par ce système.

Abstract

A wireless sensor network is an ad hoc network of a large number of nodes, which are micro-sensors capable of collecting and transmitting data autonomously. The position of these nodes is not necessarily predetermined. They can be randomly distributed in a geographical area, called "well field" corresponding to the land concerned for the phenomenon captured.

The data captured by the nodes is routed through multi-hop routing to a node, considered to be a "collection point", called a sink node. The latter can be connected to the network user (via Internet, satellite or other system). The user can send requests to the other nodes of the network, specifying the type of data required and collect the required data captured through the sink node.

In this end-of-study project, we made a study and realization of a wireless sensor network using XBee modules using ZigBee protocol. End Device 1 (LM35 + Arduino + XBee sensor) and End Device 2 (DHT11 + Arduino + XBee sensor) send their data to the XBee module of the receiving collector node. The reception node also includes a raspberry pi card for processing the signal on reception and identification of the reception frame.

The comparison of the results obtained at the transmission and the reception confirmed the feasibility, operation and reliability of this network of wireless sensors for the transmission of data produced in real time and remotely by this system.

TABLE DES MATIERES

Remerciements

Dédicace

Résumé

Abstract

Table des matières.....5

Liste des figures.....9

Introduction générales.....9

Chapitre I : Généralités sur les réseaux (LAN et WAN) et les réseaux IP

I.1 Introduction.....12

I.1.1 Qu'est-ce qu'un réseau informatique et quel est son intérêt.....12

I.1.2 Intérêts d'un réseau.....12

I.2 Les différents types de réseaux.....13

I.2.1 Les LAN (Local Area Network en français Réseau Local).....13

I.2.2 Les MAN (Métropolitain Area Network).....13

I.2.3 Les WAN (Wide Area Network ou réseau étendu).....14

Les LAN.....14

I.2.4.1 Les catégories des réseaux.....14

I.2.4.2 Le réseau (peer to peer ou pair à pair).....14

I.2.4.3 Le réseau Server/Client.....15

I.2.3 Les Topologies des réseaux.....15

I.2.3.1 les topologies physiques simples.....15

I.2.3.2 Les topologies logiques.....16

I.2.4 L'architecture de réseaux.....27

I.2.4.1 Le modèle de référence OSI d'ISO.....17

I.2.4.2 Le modèle TCP/IP.....19

I.2.5 Les équipements réseau.....	20
I.2.6 Techniques de commutations.....	22
I.2.6.1 La commutation de circuits.....	22
I.2.6.2 La commutation de messages.....	23
I.2.6.3 La commutation de paquets.....	23
I.2.6.4 La commutation de cellules.....	23
I.2.6.5 La commutation de trames.....	24
I.3 Les réseau entendu WAN.....	24
I.3.1 Réseau numérique à intégration de services (RNIS).....	24
I.3.2 Les technologies xDSL (Digital Subscriber Line).....	27
I.4 Les réseaux IP.....	28
I.4.1 L'adressage IP et la structure d'adresses IP.....	28
I.4.2 Les classes d'adresses.....	29
I.4.3 Notions importantes.....	30
I.5 Conclusion.....	31

Chapitre II : Les Outils du réseau de transmission sans fil

II.1 Définition.....	32
II. 2 Les différentes cartes.....	32
II. 2.1 Définition de la carte Arduino Uno.....	33
II. 2.2 Utilisation de la carte Arduino Uno.....	33
II. 2.3 Partie physique de l'Arduino UNO.....	34
II. 2.3.1 Dimension et organisation.....	35
II. 2.3.2 Alimentation.....	36
II. 2.3.3 Les mémoires.....	36
II. 2.3.4 Reset.....	36
II. 2.3.5 Les Entrées / Sorties.....	37

II.2.4 Partie logiciel de l'Arduino.....	38
II. 2.4.1 L'environnement de développement de L'Arduino.....	38
II. 2.4.2 Structure d'un programme.....	40
II. 3 Presentation de module Xbee.....	40
II. 3. 1 Definition.....	40
II. 3. 2 Les principales caractéristiques du Xbee.....	41
II. 3. 3 Application.....	41
II. 3. 4 Les topologies des réseaux pour les Xbee.....	44
II. 3. 5 Les antennes.....	45
II. 3. 6 Les modes de communication.....	45
II. 3. 7 Alimentation.....	48
II. 3. 8 Montages et brochages des Xbee.....	48
II. 3. 9 La configuration des modules Xbee.....	48
II. 4 La carte FTDI.....	48
II. 5 Conclusion.....	49

Chapitre III : Réalisation du réseau de transmission sans fil

III. 1 Introduction.....	50
III. 2 Organigramme.....	50
III. 3 Matériel utilisé.....	50
III.3.1 Capteur LM35.....	51
III.3.2 LM35 et Arduino - Schéma du circuit.....	52
III.3.3 Le programme - Interfaçage LM35 et Arduino.....	52
III.3.4 Montages et branchages des XBee.....	53
III.3.5 Capteur de température et d'humidité DHT11.....	55
III.3.6 Le programme - Interfaçage DHT11 et Arduino.....	56
III.3.7 Montages et branchages DHT11, Arduino et Xbee.....	56

III.3.8 Réalisation du réseau de transmission sans fil.....	58
III.8.1 Partie Matériel et logiciel.....	58
III.9 Format de la trame en mode API.....	61
III.10 Réalisation pratique des nœuds.....	63
III.10.1. Nœud final (End Device).....	63
III.10.2. Collecteur.....	63
III.10.3. Configuration des modules XBee.....	64
III.11 Conclusion.....	69
Conclusion générale.....	64
Bibliographie.....	65

LISTE DES TABLEAUX

Tableau I.1 : Les principales fonctions dans l'environnement IDE (Barre des boutons).....	10
Tableau I.2 : Les caractéristiques des modules XBee S1 et S2.....	15

LISTE DES FIGURES

Figure I.1: Carte Arduino Uno.....	03
Figure I.2 : Carte Arduino Duemilanove.....	03
Figure I.3: Carte Arduino Mega.....	04
Figure I.4 : ArduinoUno.....	04
Figure I.5: Les différentes parties de la carte Arduino UNO «partie matérielle.....	06
Figure I.6: Microcontrôleur Atmel AT mega328.....	06
Figure I.7: Les principales fonctions dans l'environnement Arduino.....	10
Figure I.8 : Module Xbee.....	12
Figure I.9 : Les différentes applications des modules Xbee.....	14
Figure I.10 : XbeeSérie1.....	14
Figure I.11 : XbeeSérie2.....	15
Figure I.12 : Les topologies des réseaux pour les Xbee.....	16
Figure I.13 : Les différentes antennes.....	17
Figure I.14 : Circuit de l'émission.....	18
Figure I.15 : Le circuit de réception.....	19
Figure I.16: Logiciel de configuration des Xbee(XCTU).....	20
Figure I.17: La carte FT232R.....	20
Figure II.1: Les différents types des réseaux.....	24
Figure II.2: Topologie en bus.....	26
Figure II.3: Topologie en étoile.....	26

Figure II.4 : La topologie en anneau.....	26
Figure II.5 : Le modèle OSI.....	28
Figure II.6: Le modèle OSI en détail.....	28
Figure II.7: Le modèle TCP/IP.....	30
Figure II.8: La couche application.....	30
Figure II.9: La couche transport.....	30
Figure II.10: La couche Internet.....	31
Figure II.11: La couche réseau.....	31
Figure II.12: Différentes classes d'adresses.....	41
Figure II.13: Classes d'adresses.....	41
Figure III.1: Schéma blocs de l'émission.....	44
Figure III.2: LM35.....	45
Figure III.3: Programme interfaçage Arduino-LM35.....	46
Figure III.4: Branchement Arduinoavec Xbee.....	47
Figure III.5: Branchement Arduino avec XBee et LM35.....	48
Figure III.6: Capteur de température et d'humidité DHT11.....	49
Figure III.7: Programme interfaçage Arduino-DHT11.....	50
Figure III.8: Branchement Arduino et DHT11.....	51
Figure III.9: Branchement Arduino, XBee et DHT11.....	51
Figure III.10: Schéma synoptique du réseau sans fil.....	52
Figure III.11: Arduino Uno.....	53
Figure III.12: Raspberry pi.....	53
Figure III.13: module XBee.....	54
Figure III.14 : Format de la trame API à l'envoi.....	56
Figure III.15 : Format de la trame API à la réception.....	56
Figure III.16 : Photo du schéma de réalisation du réseau de communication sans fil.....	58

Figure III.17a : Photo de la configuration des XBee par XCTU.....	59
Figure III.17b : Photo de la configuration des XBee (End Device 1 et End Device 2) par XCTU...	60
Figure III.17c : Photo de la configuration des XBee (End Device 1 et 2) par XCTU (suite).....	60
Figure III.18a : Photo des résultats émis par End Device 1 et End Device 2.....	61
Figure III.18b : Photo des résultats reçus par le collecteur (raspberry pi) pour la trame API.....	62
Figure III.18c: Photo des résultats reçus par le collecteur (raspberry pi) pour la trame API.....	62

Introduction Générale

Introduction générale

Les progrès technologiques dans les domaines de la microélectronique, des communications sans fil, avec les efforts de réduction des coûts de production des composants électroniques, ont permis le développement de nouvelles générations de petits appareils électroniques, autonomes, équipés de capteurs microcontrôleur et antenne et capables de détecter, de calculer, et de communiquer entre eux sans fil.

L'utilisation de logiciel informatique dans le cadre du contrôle des procédés est de plus en plus courante. Elle permet d'implémenter des lois de contrôle utilisant les données délivrées par les capteurs pour produire des commandes au niveau des actionneurs. Actuellement ces systèmes sont de plus en plus complexes et la réutilisation des composants logiciels doit permettre de réduire les coûts de développement.

L'objectif de notre travail est de réaliser un réseau de communication sans fil à base des modules Xbee pour la transmission de données physiologiques collectés en utilisant des biocapteurs. Par exemple, la transmission de la donnée température à distance en temps réel. Les informations collectées par cette application peuvent aider à la surveillance de machines à distance, exploitées ou stockées dans une base de données pour faire des statistiques...etc.

Ce mémoire est organisé en trois chapitres. Dont le premier donne un aperçu sur Les supports du réseau de transmission sans fil, sur ces différents types et ses domaines d'utilisation. Le deuxième chapitre donne un aperçu général sur les réseaux de télécommunication. Le troisième chapitre constitue le cœur de notre travail, il présente une architecture complète permettant la réalisation d'un réseau de communication sans fil orienté mesurande.

Enfin, on donne une conclusion générale des résultats de notre travail de réalisation.

Chapitre I

Généralités sur les réseaux
(LAN et WAN) et les réseaux IP

Chapitre I

Généralités sur les réseaux (LAN et WAN) et les réseaux IP

I.1 Introduction

Dans l'univers des télécommunications, nous allons nous occuper d'un espace bien défini, celui des communications numériques, c'est à dire des échanges d'informations déjà numérisées, soit d'origine digitale (données informatiques), soit échantillonnées et quantifiées préalablement (par exemple un fichier d'une séquence vidéo compressée avant stockage). Dans cette catégorie d'échange se situent tous les transferts de données existant sous forme binaire ou octet (généralement exprimé en base hexadécimale).

Les réseaux informatiques sont nés du besoin de relier des terminaux distants à un site central puis des ordinateurs entre eux et en fin des machines terminales, telles que des stations de travail ou des serveurs. Dans un premier temps, ces communications étaient destinées au transport des données informatiques. Aujourd'hui, l'intégration de la parole téléphonique et de la vidéo sur ces réseaux informatiques devient naturelle même si cela ne va pas sans difficulté.

I.1.1 Qu'est-ce qu'un réseau informatique et quel est son intérêt?

Un réseau est un ensemble d'objets interconnectés les uns avec les autres. Il permet de faire circuler des éléments entre chacun de ces objets selon des règles bien définies. Dans le cas où les objets sont des ordinateurs on parle d'un réseau informatique. [5]

Les réseaux informatiques qui permettaient à leur origine de relier des terminaux passifs à de gros ordinateurs centraux autorisent à l'heure actuelle l'interconnexion de tous types, d'ordinateurs que ce soit de gros serveurs, des stations de travail, des ordinateurs personnels ou de simples terminaux graphiques. Les services qu'ils offrent font partie de la vie courante des entreprises et administrations (banques, gestion, commerce, bases de données, recherche,...) et des particuliers (messagerie, loisirs, services d'informations par minitel et Internet ...).

I.1.2 Intérêts d'un réseau

Un ordinateur est une machine permettant de manipuler des données. L'homme, un être de communication, a vite compris l'intérêt qu'il pouvait y avoir à relier ces ordinateurs entre eux

afin de pouvoir échanger des informations. Voici un certain nombre de raisons pour lesquelles un réseau est utile, un réseau permet:

- ✓ Le partage de fichiers, d'applications et de ressources.
- ✓ La communication entre personnes (grâce au courrier électronique, la discussion endirect, ...).
- ✓ La communication entre processus (entre des machines industrielles).
- ✓ La garantie de l'unicité de l'information (bases de données).
- ✓ Le jeu à plusieurs, ...
- ✓ Le transfert de la parole, de la vidéo et des données (réseaux à intégration de services ou multimédia).
- ✓ Les réseaux permettent aussi de standardiser les applications, on parle généralement de group ware. Par exemple la messagerie électronique et les agendas de groupe qui permettent de communiquer plus efficacement et plus rapidement. [4]

I.2 Les différents types de réseaux

On peut distinguer différents types de réseaux selon plusieurs critères tel que (la taille de réseau, la vitesse de transfert des données et aussi leur étendue) :

I.2.1 Les LAN (*Local Area Network* en français *Réseau Local*)

Il s'agit d'un ensemble d'ordinateurs appartenant à une même organisation et reliés entre eux dans une petite aire géographique par un réseau, souvent à l'aide d'une même technologie (la plus répandue étant Ethernet). La vitesse de transfert de données d'un réseau local peut s'échelonner entre 10 Mbit/s (pour un réseau Ethernet par exemple) et 1 Gbit/s (en FDDI ou Gigabit Ethernet par exemple). La taille d'un réseau local peut atteindre jusqu'à 100 voire 1000 utilisateurs. [4]

I.2.2 Les MAN (Métropolitain Area Network)

Interconnectent plusieurs LAN géographiquement proches (au maximum quelques dizaines de km) à des débits importants. Ainsi un MAN permet à deux nœuds distants de communiquer comme si ils faisaient partie d'un même réseau local. Un MAN est formée de commutateurs ou de routeurs interconnectés par des liens hauts débits (en général en fibre optique). [4]

I.2.3 Les WAN (Wide Area Network ou réseau étendu)

Interconnecte plusieurs LANs à travers de grandes distances géographiques.

Les débits disponibles sur un WAN résultent d'un arbitrage avec le coût des liaisons (qui augmente avec la distance) et peuvent être faibles. Les WAN fonctionnent grâce à des routeurs qui permettent de "choisir" le trajet le plus approprié pour atteindre un nœud du réseau.

Le plus connu des WAN est Internet. [4]

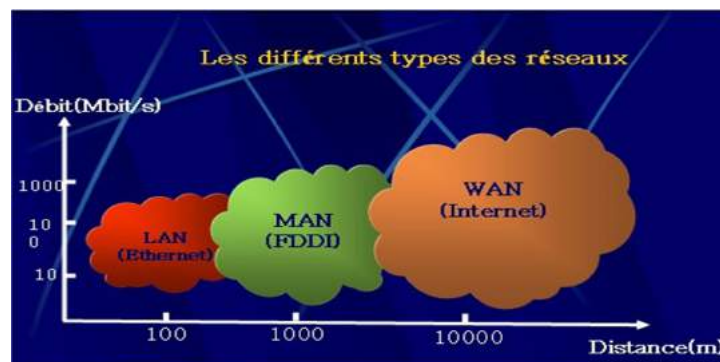


Figure 1.1 : les différents types des réseaux

I.2.4 LAN

I.2.4.1 Les catégories des réseaux

On distingue également deux catégories de réseaux :

- Réseaux poste à poste (peer to peer= P2P).
- Réseaux avec serveur dédié (Server/client)

I.2.4.1.1 Le réseau (peer to peer ou pair à pair)

Chaque poste ou station fait office de serveur et les données ne sont pas centralisées, l'avantage majeur d'une telle installation est son faible coût en matériel (les postes de travail et une carte réseau par poste). En revanche, si le réseau commence à comporter plusieurs machines (>10 postes) il devient impossible à gérer. [1]

Par exemple : Si on a 4 postes et 10 utilisateurs, chaque poste doit contenir les 10 mots de passe afin que les utilisateurs puissent travailler sur n'importe lequel des postes. Mais si maintenant il y a 60 postes et 300 utilisateurs, la gestion des mots de passe devient périlleuse.

I.2.4.1.2 Le réseau Server/Client

Il ressemble un peu au réseau poste à poste mais cette fois-ci, on y rajoute un poste plus puissant, dédié à des tâches bien précises. Cette nouvelle station s'appelle serveur. Le serveur Centralise les données relatives au bon fonctionnement du réseau.

Dans l'exemple précédant, C'est lui qui contient tous les mots de passe. Ainsi ils ne se trouvent plus qu'à un seul endroit. Il est donc plus facile pour l'administrateur du réseau de les modifier ou d'en créer d'autres. L'avantage de ce type de réseau est sa facilité de gestion des réseaux comportant beaucoup de postes. Son inconvénient majeur est son coût souvent très élevé en matériel. En effet, en plus des postes de travail il faut se procurer un serveur qui coûte cher car c'est une machine très puissante et perfectionnée. De plus la carte réseau que l'on y met est de meilleure qualité que Celle des postes de travail. [1]

I.2.3 Les Topologie des réseaux

I.2.3.1 les topologies physiques simples

Une topologie physique correspond à la disposition physique d'un réseau, mais ne spécifie pas les types de périphérique, les méthodes de connectivité ou les adresses d'un réseau. Les topologies physiques sont disposées selon trois principaux groupes de formes géométriques: le bus, l'anneau et l'étoile.

❖ La topologie en Bus

Dans cette topologie un même câble relie tous les nœuds d'un réseau sans périphérique de connectivité intermédiaire. Les deux extrémités des réseaux en bus sont équipées de résistances de 50 ohms (terminateurs) qui arrêtent les signaux une fois arrivés destination. Sans terminateurs, les signaux d'un réseau en bus continueraient à circuler sans fin ; ce qu'on appelle le rebond de signal.



Figure 1.2: Topologie en bus.

❖ La topologie en étoile

Dans cette topologie, chaque nœud du réseau est relié à un périphérique central, tel qu'un concentrateur (hub). Un même câble de réseau en étoile ne peut relier que deux périphériques, donc un problème de câblage ne touchera jamais plus de deux nœuds. Les nœuds transmettent des données au concentrateur, qui à son tour retransmet les informations au segment de réseau où le nœud de destination pourra les ramasser.



Figure 1.3: Topologie en étoile

Dans une topologie de l'anneau, chaque nœud est relié aux deux nœuds les plus proches, et l'ensemble du réseau forme un cercle. Les données sont transmises autour de l'anneau dans une seule direction. Chaque station de travail accepte et répond aux paquets qui lui sont adressés, puis les fait suivre à la prochaine station de l'anneau. [4]



Figure 1.4 : La topologie en anneau

I.2.3.2 Les topologies logiques

Le terme topologie logique désigne la façon par laquelle les données sont transmises entre les nœuds, plutôt que la disposition des voies ou chemins qu'empruntent les données. Une topologie logique s'appelle aussi un système de transport réseau. La topologie logique d'un réseau décrit la manière par laquelle les données sont mises en trames et comment les impulsions électriques sont envoyées sur le support physique du réseau. Les éléments d'une topologie logique appartiennent à la

fois aux couche liaison du modèle OSI.

Chaque topologie logique possède son propre ensemble de principe de signalisation de données, mais impose aussi des exigences particulières au niveau du média de transmission et de la topologie physique.

Ethernet et Token Ring sont les deux systèmes de transport réseau (topologie logique) les plus courants .mais il y a également d'autre topologie logique tel que FDDI et LocalTk...etc. [4]

I.2.4 L'architecture de réseaux

I.2.4.1 Le modèle de référence OSI d'ISO

Au début des années 70, chaque constructeur a développé sa propre solution réseau autour d'architecture et de protocole privés et il s'est vite avéré qu'il serait impossible d'interconnecter ces différents réseaux si une norme internationale n'était pas établie.

Cette norme établie par l'internationale standard organisation (ISO) est la norme open system interconnexion (OSI, interconnexion de systèmes ouverts).

Un système ouvert est un ordinateur, un terminal, un réseau, n'importe quel équipement respectant cette norme et donc apte à échanger des informations avec d'autres équipement hétérogènes et issus de constructeurs différents.

La première objectif de la norme OSI a été de définir un modèle de toute architecture de réseau base sur découpage en sept couches chacun de ces couches correspondant à une fonctionnalité particulière d'un réseau.

Les couches 1, 2,3 et 4 sont dites basses et les couches 5,6 et 7 sont dites hautes. [1]

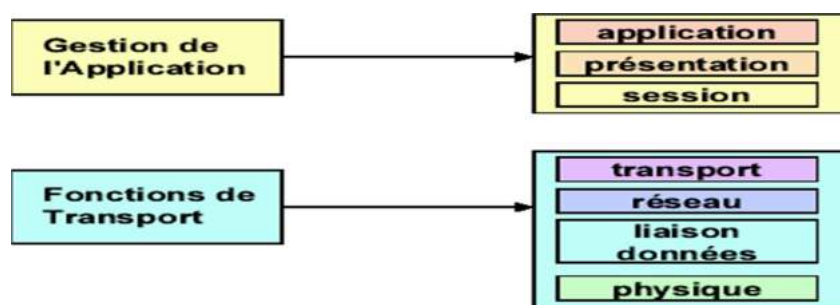


Figure 1.5 : Le modèle OSI

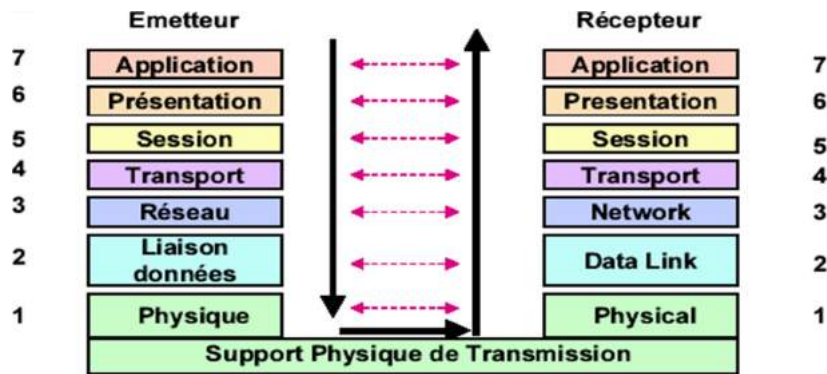


Figure1.6: Le modèle OSI en détail.

Cette couche définit les caractéristiques techniques, électriques, fonctionnelles et procédure les nécessaires à l'activation et à la désactivation des connexions physiques destinées à la transmission de bits entre deux entités de la couche liaisons de données.

❖ La couche liaison

Cette couche définit les moyens fonctionnels et procéduraux nécessaires à l'activation et à l'établissement ainsi qu'au maintien et à la libération des connexions de liaisons de données entre les entités du réseau. Cette couche détecte et corrige, quand cela est possible, les erreurs de la couche physique et signale à la couche réseau les erreurs irrécupérables.

❖ La couche réseau

Cette couche assure toutes les fonctionnalités de services entre les entités du réseau, c'est à dire : l'adressage, le routage, le contrôle de flux, la détection et la correction d'erreurs non résolues par la couche liaison pour préparer le travail de la couche transport.

❖ La couche transport

Cette couche définit un transfert de données entre les entités en les déchargeant des détails d'exécution (contrôle entre l'OSI et le support de transmission). Son rôle est d'optimiser l'utilisation des services de réseau disponibles afin d'assurer à moindre coût les performances requise par la couche session.

❖ La couche session

Cette couche fournit aux entités de la couche présentation les moyens d'organiser et de synchroniser les dialogues et les échanges de données. Il s'agit de la gestion d'accès, de sécurité et

d'identification des services.

❖ La couche présentation

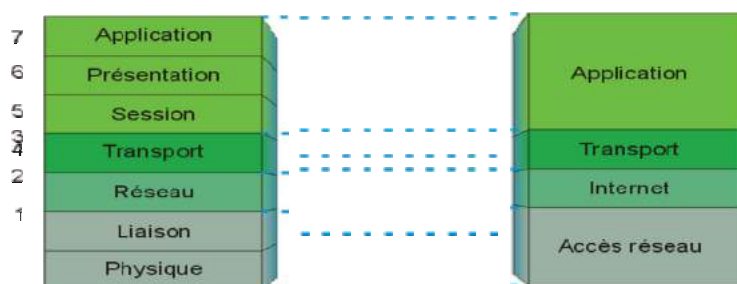
Cette couche assure la transparence du format des données à la couche application.

❖ La couche application

Cette couche assure aux processus d'application le moyen d'accès à l'environnement OSI et fournit tous les services directement utilisables par l'application (transfert de données, allocation de ressources, intégrité et cohérence des informations, synchronisation des applications). [1]

I.2.4.2 Le modèle TCP/IP

Le modèle TCP/IP peut en effet être décrit comme une architecture réseau à 4 couches



Le modèle OSI a été mis à côté pour faciliter la comparaison entre les deux modèles. Il y a 4 couches principales dans l'environnement TCP/IP :

❖ **La couche application** : les applications interagissent avec les protocoles de la couche Transport pour envoyer ou recevoir des données.



Figure 1.8 : La couche application

❖ **La couche transport** : chargé de fournir un moyen de communication de bout en bout entre 2 programmes d'application. Agit en mode connecté et en mode non connecté. Elle divise le flux de données venant des applications en paquets, transmis avec l'adresse destination IP au niveau IP.



Figure 1.9 : La couche transport

- ❖ **La couche Internet** : encapsule les paquets reçus de la couche Transport dans des datagrammes IP. Mode non connecté et non fiable.



Figure 1.10 : La couche Internet

- ❖ **La couche Hôte Réseau** : assure la transmission d'un datagramme venant de la couche IP en l'encapsulant dans une trame physique et en transmettant cette dernière sur un réseau physique. [2]



Figure 1.11 : La couche réseau

I.2.5 Les équipements réseau

L'interconnexion de réseaux peut être locale: les réseaux sont sur le même site géographique. Dans ce cas, un équipement standard (Répéteur, routeur ...etc.) sert à réaliser physiquement la liaison.

L'interconnexion peut aussi concerner des réseaux distants. Il est alors nécessaire de relier ces réseaux par une liaison téléphonique (modems, etc.).

❖ Les multiplexeurs

Les formes de transmission qui permet à plusieurs signaux de voyager simultanément sur un même media s'appelle transmission multiplex ou multiplexage. Pour accommoder plusieurs signaux sur le même support est logiquement séparé en plusieurs canaux donc un multiplexeur sert à transiter sur une seule et même ligne de liaison, dite voie haute vitesse, des communications appartenant à plusieurs paires d'équipements émetteurs et récepteurs. Chaque émetteur (respectivement récepteur) est raccordé à un multiplexeur (respectivement démultiplexeur) par une liaison dite voie basse vitesse. Plusieurs techniques de multiplexage sont possibles [5] :

- ✓ **Multiplexage temporel** : Ensemble de voies «basses vitesses» (VBi). Débit utile inférieur au débit théorique de la ligne de transmission (divisé par le nombre de transmission en parallèle), cas des voies muettes.
- ✓ **Multiplexage statistique** : Optimisation multiplexage temporel. Utilise un codage spécial type Huffman en vue d'améliorer la transmission, notamment prise en charge des voies muettes.
- ✓ **Multiplexage fréquentiel** : Partage de la bande passante disponible sur un système de transmission en canaux.

❖ Les concentrateurs (Hubs)

Servent à relier entre elles toutes les parties d'un même réseau physique, généralement tous les ordinateurs sont reliés à un Hub, sauf dans le cas d'un câblage coaxial où le Hub est inutile. Lorsqu'une information arrive sur un Hub, elle est rediffusée vers toutes les destinations possibles à partir de celui-ci, c'est à dire vers toutes ses prises. [4]

❖ Les commutateurs (Switches)

Le commutateur (ou Switch) est un système assurant l'interconnexion de stations ou de segments d'un LAN en leur attribuant l'intégralité de la bande passante, à l'inverse du concentrateur qui la partage.

Les commutateurs ont donc été introduits pour augmenter la bande passante globale d'un réseau d'entreprise et sont une évolution des concentrateurs Ethernet (ou HUB). [4]

❖ Les ponts (Bridges)

Ils servent à relier entre eux deux réseaux différents d'un point de vue physique. De plus ils

filtrent les informations et ne laissent passer que celles qui doivent effectivement aller d'un réseau vers l'autre. Ils peuvent être utilisés pour augmenter les distances de câblage en cas d'affaiblissement prématuré du signal. [4]

❖ **Les routeurs (Routers)**

Ils relient des réseaux physiques et/ou logiques différents, généralement distants. Comme les ponts ils filtrent les informations mais à un niveau beaucoup plus fin (le niveau logique), et l'on peut même s'en servir pour protéger un réseau de l'extérieur tout en laissant des réseaux "amis" accéder au réseau local. [4]

❖ **Les répéteurs (Repeater)**

Sont des dispositifs permettant d'étendre la distance de câblage d'un réseau local. Leur rôle consiste à amplifier et à répéter les signaux qui leur parviennent. Il existe également des répéteurs qui en plus régénèrent les signaux. Ceci réduit le bruit et la distorsion. Le répéteur intervient au niveau 1 du modèle OSI. [4]

❖ **Les passerelles (Gateway)**

Sont des dispositifs permettant d'interconnecter des architectures de réseaux différentes. Elles offrent donc la conversion de tous les protocoles, au travers des 7 couches du modèle OSI.

L'objectif étant de disposer d'une architecture de réseau évolutive, la tendance actuelle est d'interconnecter les réseaux par des routeurs, d'autant plus que le prix de ceux-ci est en baisse. [4]

I.2.6 Techniques de commutations

I.2.6.1 La commutation de circuits

C'est historiquement la première à avoir été utilisée, elle consiste à créer dans le réseau un circuit particulier entre l'émetteur et le récepteur avant que ceux-ci ne commencent à échanger des informations. Ce circuit sera propre aux deux entités communiquant et il sera libéré lorsque l'un des deux coupera sa communication.

Par contre, si pendant un certain temps les deux entités ne s'échangent rien le circuit leur reste quand même attribué. C'est pourquoi, un même circuit (ou portion de circuit) pourra être attribué à plusieurs communications en même temps. Cela améliore le fonctionnement global du réseau mais pose des problèmes de gestion (files d'attente, mémorisation,...). [4]

I.2.6.2 La commutation de messages

Elle consiste à envoyer un message de l'émetteur jusqu'au récepteur en passant de nœud de commutation en nœud de commutation. Chaque nœud attend d'avoir reçu complètement le message avant de le réexpédier au nœud suivant.

Cette technique nécessite de prévoir de grandes zones tampon dans chaque nœud du réseau, mais comme ces zones ne sont pas illimitées il faut aussi prévoir un contrôle de flux des messages pour éviter la saturation du réseau. Dans cette approche il devient très difficile de transmettre de longs messages.

En effet, comme un message doit être reçu entièrement à chaque étape si la ligne a un taux d'erreur de 10^{-5} par bit (1 bit sur 10^5 est erroné) alors un message de 100000 octets n'a qu'une probabilité de 0,0003 d'être transmis sans erreur. [4]

I.2.6.3 La commutation de paquets

Elle est apparue au début des années 70 pour résoudre les problèmes d'erreur de la commutation de messages. Un message émis est découpé en paquets et par la suite chaque paquet est commuté à travers le réseau comme dans le cas des messages. Les paquets sont envoyés indépendamment les uns des autres et sur une même liaison on pourra trouver les uns derrière les autres des paquets appartenant à différents messages.

Chaque nœud redirige chaque paquet vers la bonne liaison grâce à une table de routage. La reprise sur erreur est donc ici plus simple que dans la commutation de messages, par contre le récepteur final doit être capable de reconstituer le message émis en rassemblant les paquets. Ceci nécessitera un protocole particulier car les paquets peuvent ne pas arriver dans l'ordre initial, soit parce qu'ils ont emprunté des routes différentes, soit parce que l'un d'eux a dû être réémis suite à une erreur de transmission. [5]

I.2.6.4 La commutation de cellules

Une cellule est un paquet particulier dont la taille est toujours fixée à 53 octets (5 octets d'en-tête et 48 octets de données). C'est la technique de base des réseaux hauts débits ATM (Asynchronous Transfer Mode) qui opèrent en mode connecté où avant toute émission de cellules, un chemin virtuel est établi par lequel passeront toutes les cellules. Cette technique mixe donc la commutation de circuits et la commutation de paquets de taille fixe permettant ainsi de simplifier le travail des commutateurs pour atteindre des débits plus élevés. [5]

I.2.6.5 La commutation de trames

La commutation de trames est une extension de la commutation de paquets. Dans la commutation de paquets, les commutateurs récupèrent les entités de niveau 3 alors que la commutation de trames traite des entités de niveau 2. L'avantage de cette approche est de ne remonter qu'au niveau trame au lieu du niveau paquet. [4]

Plusieurs catégories de commutation de trames ont été développées suivant le protocole de niveau trame choisi. Les deux principales concernent le relais de trame et la commutation Ethernet. [4]

I.3 Les réseau étendu WAN

Bien que la plus part des entreprises nécessite des réseaux étendus, elles ne font pas nécessairement appel au même type de réseau étendu.

La méthode de transmission de réseau étendu diffèrent selon leur vitesse, leur fiabilité, leur coût, la distance parcourue et la sécurité fournie. En fait, il n'existe parfois qu'un seul ou quelque méthode de transmission de réseau étendu adéquat pour répondre à un besoin d'affaire particulier. Il est également possible de se servir de plusieurs technologies de réseau étendu dans un même réseau.

Parmi ces méthodes et ces technologies: (PSTN, X.25, RNIS, T1, DSL...). On va prendre comme un exemple la méthode RNIS et la technologie DSL. [4]

I.3.1 Réseau numérique à intégration de services (RNIS)

Un réseau numérique à intégration de services (RNIS, en anglais *ISDN* pour Integrated Services Digital Network) est une liaison autorisant une meilleure qualité et des vitesses pouvant atteindre 2 Mbit/s (accès S2) contre 56 kbit/s pour un modem classique.

On peut voir l'architecture RNIS comme une évolution entièrement numérique des réseaux téléphoniques existants, conçue pour associer la voix, les données, la vidéo et toute autre application ou service. RNIS s'oppose donc au réseau téléphonique commuté (RTC) traditionnel. [19]

❖ Présentation

Une connexion RNIS donne accès à plusieurs canaux numériques : les canaux de type B (64 kbit/s en Europe, 56 kbit/s en Amérique du Nord) et les canaux de type D (16 kbit/s). Les canaux B servent au transport de données et peuvent être agglomérés pour augmenter la bande passante. Les canaux D servent à la signalisation des communications.

Les réseaux RNIS bande de base fournissent des services à faible débit : de 64 kbit/s à 2 Mbit/s. L'actuelle technologie ATM dédiée aux réseaux grands distances (WAN) faisait à l'origine parties des définitions RNIS sous la dénomination RNIS large bande pour les services à haut débit : de 10 Mbit/s à 622 Mbit/s.

Avec RNIS, les sites régionaux et internationaux de petite taille peuvent se connecter aux réseaux d'entreprises à un coût mieux adapté à la consommation réelle qu'avec des lignes spécialisées. Les liaisons à la demande RNIS peuvent être utilisées soit pour remplacer les lignes spécialisées, soit en complément pour augmenter la bande passante ou assurer une redondance. Avec ces mêmes liaisons, les sites ou les utilisateurs distants peuvent accéder efficacement aux ressources critiques à travers l'Internet en toute sécurité.

❖ Le développement des réseaux RNIS

L'Union internationale des télécommunications (UIT) a défini la technologie RNIS comme un réseau fournissant une connectivité numérique de bout en bout avec une grande variété de services. Deux caractéristiques importantes des réseaux RNIS les distinguent des réseaux téléphoniques traditionnels :

- ✓ les connexions sont numériques d'une extrémité à l'autre ;
- ✓ RNIS définit un jeu de protocoles d'interface utilisateur/réseau standard. De cette façon, tous les équipements RNIS utilisent les mêmes connexions physiques et les mêmes protocoles de signalisation pour accéder aux services.

RNIS combine la large couverture géographique d'un réseau téléphonique avec la capacité de transport d'un réseau de données supportant simultanément la voix, les données et la vidéo.

En France et en Belgique, le réseau national de télécommunications a été entièrement numérisé et les protocoles d'accès implantés sont conformes au standard Euro-ISDN publié par l'ETSI et l'UIT.

❖ Fonctionnement

Dans un réseau téléphonique analogique, une boucle sur une paire torsadée de fils de cuivre entre le commutateur central de la compagnie de télécommunications et l'abonné supporte un canal de transmission unique. Ce canal ne traite qu'un seul service simultanément : la voix ou les données. Avec un Réseau Numérique à Intégration de Services, la même paire torsadée est divisée en plusieurs canaux logiques.

❖ Les canaux logiques RNIS

RNIS définit deux types de canaux logiques que l'on distingue par leurs fonctions et leurs débits. Les canaux B transmettent à un débit de 64 kbit/s en commutation de circuit ou de paquet les informations utilisateur : voix, données, fax. Tous les services réseau sont accessibles à partir des canaux B. Les canaux D transmettent à un débit de 16 kbit/s en accès de base et 64 kbit/s en accès primaire. Ils supportent les informations de signalisation : appels, établissement des connexions, demandes de services, routage des données sur les canaux B et enfin libération des connexions. Ces informations de signalisation ont été conçues pour cheminer sur un réseau totalement distinct des canaux B. C'est cette signalisation hors bande qui donne aux réseaux RNIS des temps d'établissement de connexion rapides (environ 4 secondes) relativement aux réseaux analogiques (environ 40 secondes). Il est aussi possible de transmettre des données utilisateur à travers les canaux D (protocole X.31b), mais comme le débit de ces canaux est limité ce type d'utilisation est rare.

❖ Les interfaces standard RNIS

Une interface d'accès à un réseau RNIS est une association de canaux B et D. Il existe deux interfaces standard. Elles correspondent à deux catégories d'utilisation distinctes :

Résidentielle : utilisation simultanée des services téléphoniques et d'une connexion Internet.

Professionnelle : utilisation d'un commutateur téléphonique (PABX) et/ou d'un routeur d'agence. Dans les deux cas, le nombre de canaux utilisés peut varier suivant les besoins, le débit maximum étant fixé par le type d'interface. [19]

✓ Accès de base

L'accès de base ou Basic Rate Interface (BRI) comprend 2 canaux B et un canal D pour la signalisation : 2B+D. [4]

✓ Accès primaire

L'accès primaire ou Primary Rate Interface (PRI) comprend 30 canaux B et un canal D à 64 kbit/s en Europe, en Afrique, en Amérique du Sud, au Moyen-Orient, en Asie (hors Japon) : 30B+D. Aux États-Unis, au Canada et au Japon la définition est différente : 23B+D. Seule la protection des marchés explique les différences de définition entre l'Europe, les États-Unis, le Canada et le Japon. Cet accès est l'équivalent RNIS des liaisons T1/E1 à 1,544 Mbit/s et 2,048 Mbit/s. [4]

❖ L'adaptation des débits

Les équipements non-RNIS n'ont pas nécessairement des débits compatibles avec la définition

du canal B : 64 kbit/s. Dans ce cas, les adaptateurs de terminal (TA) réalisent une adaptation en réduisant le débit effectif du canal B jusqu'à une valeur compatible avec le dispositif non-RNIS. Il existe 2 protocoles de gestion d'adaptation : V.110 très utilisé en Europe et V.120 aux États-Unis. Ces 2 protocoles gèrent les transmissions synchrones et asynchrones. Le protocole V.110 peut fonctionner avec le sous-système RNIS Linux et un téléphone cellulaire GSM par exemple. C'est au prestataire de téléphonie cellulaire de fournir la passerelle RNIS/V.110...

❖ L'allocation dynamique de bande passante

La bande passante dynamique ou l'allocation de canaux est obtenue par l'agrégation des canaux B. On obtient ainsi une bande passante maximale de 128 kbit/s pour l'accès de base (BRI) et de 1,875 Mbit/s pour l'accès primaire en Europe.

Cette fonctionnalité permet d'adapter le débit et donc le coût de communication aux besoins effectifs pour les flux entrants et sortants. Suivant les heures de la journée ou les jours de la semaine, les besoins de connectivité varient fortement. Il est possible que le coût forfaitaire d'utilisation d'une ligne spécialisée soit supérieur au coût en temps de communication d'une liaison RNIS, lorsque cette dernière utilise correctement la bande passante à la demande en ouvrant/fermant les connexions aux heures choisies.

Il existe deux techniques pour agréger les canaux B appelées *bonding* et *bundling*.

Le bonding travaille au niveau 1 (couche physique) du modèle OSI. Il assure une synchronisation au niveau bit. Cette technique nécessite donc un matériel spécifique. Elle est surtout utilisée dans les équipements dédiés de visioconférence et très peu dans les équipements de réseaux de données.

Le bundling est une technique générique qui travaille au niveau 2 (couche de liaison) du modèle OSI. Dans le cas d'une connexion RNIS, elle permet d'ouvrir simultanément plusieurs canaux B entre 2 systèmes. Le standard Multilink-PPP (ML-PPP) décrit comment séparer, recombinaison et séquencer des datagrammes sur plusieurs canaux B pour créer une connexion logique unique. Ce standard est dédié au protocole PPP, le standard de niveau liaison du modèle TCP/IP pour les accès téléphoniques aux réseaux locaux (LAN) et à Internet. [19]

I.3.2 Les technologies xDSL (Digital Subscriber Line)

Le but de la technologie DSL est de doper les paires téléphoniques de cuivre existantes en mixant le trafic de données, de voix et de vidéo en point à point sur le réseau téléphonique traditionnel.

La barrière théorique des 300 – 3400 Hz de bande passante utilisée sur les lignes téléphoniques peut être repoussée sous certaines conditions.

Il existe plusieurs types de DSL [7] :

- ❖ **ADSL**, Asymmetric DSL : Elle est basée sur un débit asymétrique, le flux descendant (du réseau vers l'utilisateur) étant plus important que le flux montant. ADSL préserve le canal de voix et convient bien aux applications interactives du type « accès à Internet » ou « vidéo à la demande ».
- ❖ **HDSL**, High data rate DSL : Elle permet un canal T1 ou E1 sur une boucle locale sans répéteur. HDSL peut être utilisé par les opérateurs pour l'interconnexion de PABX, par exemple.
- ❖ **SDSL**, Single line DSL : C'est une version monoligne de HDSL (qui utilise les deux paires téléphoniques).
- ❖ **VDSL**, Very high data rate DSL : Elle est en cours de développement et devrait permettre des débits de l'ordre de 50 Mb/s pour le flux descendant.
- ❖ **RADSL**, Rate Adaptive DSL : C'est une technique asymétrique qui a la particularité d'adapter le débit en fonction des capacités de la ligne.

I.4 Les réseaux IP

Les réseaux IP (Internet) devient non seulement un moyen de communication mais aussi un moyen de commerce global de développement et distribution.

TCP/IP est très connu dans le domaine des réseaux, il correspond à toute une architecture.

Il ne correspond pas à un seul protocole mais bien à un ensemble de petits protocoles spécialisés appelés sous protocoles (TCP, IP, UDP, ARP ICMP.....).

La plupart des administrateurs réseaux désignent ce groupe par TCP/IP.

- TCP (Transmission Control Protocol) qui est un protocole de niveau message.
- IP (Internet Protocol) qui est un protocole de niveau paquet.

I.4.1 L'adressage IP et la structure d'adresses IP

Comme l'Internet est un réseau, l'adressage est particulièrement important. Les adresses IP ont été définies pour être traitées rapidement. Les routeurs qui effectuent le routage en se basant sur le numéro de réseau sont dépendants de cette structure.

Les adresses IP peuvent donc être représentées sur 32 bits, Regroupée en quatre octets de 8 bits séparés par des points décimaux. [4]

Ces 32 bit sont séparés en deux zones de bits contiguës :

- ✓ **Network ID** : une partie décrit le numéro du réseau local auquel est rattachée la station.
- ✓ **Host ID** : une partie correspond au numéro de la station dans le réseau local lui-même, appelée numéro d'hôte.

Selon l'adresse IP on définit différentes classes d'adresses. Il existe cinq classes d'adresses avec la version 4 (IPv4 version courante) des protocoles TCP/IP, car les parties réseau et hôte n'ont pas toujours la même taille.

Il faut connaître une chose importantes dans l'adressage IP c'est que il y a des adresses réservées par exemple :

- ✓ 10.0.0.1 à 10.255.255.254
- ✓ 172.16.0.1 à 172.31.255.254
- ✓ 192.168.0.1 à 192.168.255.254

I.4.2 Les classes d'adresses

On distingue 5 classes d'adresses:

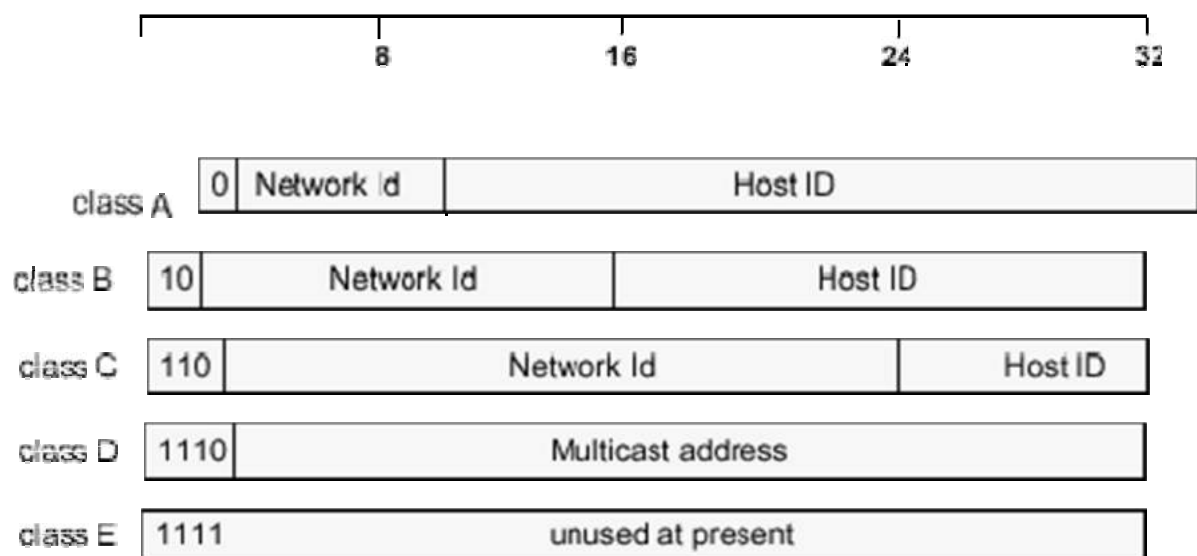


Figure 1.12 : Différentes classes d'adresses

	Nombre de réseaux possibles	Nombre maximum d'ordinateurs
e		
A	126	16777214
B	16384	65534
C	2097153	254

Figure 1.13 : Classes d'adresses

I.4.3 Notions importantes

Il se trouve plusieurs notions importantes dans les réseaux IP qui sont :

❖ Masque sous réseau

On fabrique un masque contenant des 1 aux emplacements des bits que l'on désire conserver, et des 0 pour ceux que l'on veut rendre égaux à zéro. Une fois ce masque créé, il suffit de faire un « ET » entre la valeur que l'on désire masquer et le masque afin de garder intacte la partie que l'on désire et annuler le reste. Ainsi, un masque réseau se présente sous la forme de 4 octets séparés par des points, un des intérêts de se masquer est de pouvoir connaître le réseau associé à une adresse IP.

[1]

❖ Le routage

Un Processus qui permettant d'acheminer un datagramme IP de son hôte émetteur Jusqu'à son hôte destinataire. Chaque datagramme est routé Indépendamment des autres.

Il existe 2 types de routage (Routage direct (intérieur) et Routage indirect (extérieur)).

❖ Le protocole NAT (Network Address Translation)

Il s'agit d'un procédé permettant de transcrire des adresses IP en d'autres, sans références directes avec les adresses MAC, traitées quant à elles par le protocole ARP.

NAT utilise l'adresse IP et le numéro de port d'une station et les transformer en une

adresse IP et un numéro de port qui n'est pas attribué à une application standard.

❖ Le DNS (Domain Name System)

Le DNS permet la mise en correspondance des adresses physiques dans le réseau et des adresses logiques. La structure logique est hiérarchique et utilise au plus haut niveau des domaines caractérisant principalement les pays, qui sont indiqués par deux lettres, comme dz pour l'Algérie, sn pour le Sénégal, ml pour le Mali et des domaines fonctionnels comme [4]:

- ✓ Com organisations commerciales
- ✓ Edu institutions académiques
- ✓ Net opérateurs de réseaux

❖ Le DHCP (Dynamic Host Configuration Protocol)

DHCP utilise un modèle client/serveur dans lequel le serveur DHCP assure la gestion centralisée des adresses IP utilisées sur le réseau. Les clients qui prennent en charge DHCP peuvent ensuite demander et obtenir la location d'une adresse IP auprès d'un serveur DHCP dans le cadre de leur procédure d'amorçage réseau.

❖ IPv6

A cause de la limite de l'adresse IPv4 la IPv6 a été définie pour décrire les adresses de 16 octets, elle comprend 8 groupes de 4 chiffres hexadécimaux séparés avec le symbole deux-points, IPv6 reconnaît trois types d'adresses : unicast, multicast et anycast. [1]

I.5 Conclusion

Pour satisfaire nos besoins dans un réseau il faut tout d'abord bien choisir: une topologie adéquate, un bon système de câblage, les meilleures techniques de transmission et de commutation et son oublier choisir une architecture conforme aux organismes de normalisation pour les réseaux.

L'ingénierie des réseaux IP reste un domaine complexe encore peu maîtrisé, mais à l'apparition de la norme IPv6 qui remplacera petit à petit la norme IPv4. En effet l'arrivée massive des terminaux mobiles et l'adressage direct de toutes les stations terminales impliquent l'adoption de l'adresse de nouvelles générations. D'autre part, la reconnaissance des flux aide à l'introduction de nouvelles fonctions, on remarque que les besoins sont croissants en matière de communication qui converge de plus en plus vers le multimédia (voix, son, image, vidéo) ne seront supportés par les réseaux que si ces derniers évoluent vers les hauts débits alliant performance, qualité de service, rapidité.

Chapitre II

Les outils du réseau de
transmission sans fil

Chapitre II

Les Outils du réseau de transmission sans fil

II.1 Définition

L'Arduino est une carte électronique en matériel libre pour la création artistique interactive. Elle peut servir :

- ✓ Pour des dispositifs interactifs autonomes simples
- ✓ Comme interface entre capteurs/actionneurs et ordinateur.
- ✓ Comme programmeur de certains microcontrôleurs.

II. 2 Les différentes cartes

Des cartes Arduino, il en existe beaucoup. Nous nous intéressons à la carte Uno et Duemilanove. Nous choisirons d'utiliser la carte portant le nom de «Uno» ou «Duemilanove». Ces deux versions sont presque identiques (Figure II.1) (Figure II. 2) (Figure II.3).



Figure II.1 : Carte Arduino Uno



Figure II.2 : Carte Arduino Duemilanove



Figure II.3 : Carte Arduino Mega

Il existe encore beaucoup d'autres cartes. Pour débiter, nous présentons la carte Arduino Uno: Avantages, inconvénients, utilisation et fonctionnement.

II.2.1 Définition de la carte Arduino Uno

Arduino Uno est un microcontrôleur programmable qui permet, comme son nom l'indique, de contrôler des éléments mécaniques: systèmes de lumières, moteurs, etc. Cette carte électronique permet donc à son utilisateur de programmer facilement des tâches et de créer des mécanismes automatisés, sans avoir de connaissances particulières en programmation. Il est un outil pensé et destiné aux inventeurs, artistes ou amateurs qui souhaitent créer leur propre système automatique en le codant de toute pièce (figure II.4).



Figure II.4 : Arduino Uno

II.2.2 Utilisation de la carte Arduino Uno

Pour commencer à utiliser l'Arduino Uno, vous allez dans un premier temps devoir télécharger la dernière version du logiciel Arduino. IL est compatible Windows, Mac ou Linux. Une fois cela fait, lancez le logiciel. Vous pourrez constater la simplicité de l'interface

du logiciel qui présente très sobrement une page vierge destinée au code, ainsi qu'une barre de boutons et un menu.

Plusieurs facteurs favorisent l'utilisation du module Arduino par rapport aux autres cartes de microcontrôleur, on peut citer quelques-uns :

- Le coût : les cartes Arduino sont peu coûteuses par rapport aux autres cartes de microcontrôleur.
- La multi plate-forme : Le logiciel Arduino (IDE) fonctionne sous les systèmes d'exploitation Windows, Macintosh OSX et Linux, par contre la plupart des systèmes à microcontrôleur sont limités à Windows.
- Un logiciel open source et extensibles : Le logiciel Arduino est publié en tant qu'outils open source c'est-à-dire il est disponible et gratuit. De plus il est extensible par les programmeurs expérimentés. Pour mieux l'utiliser il suffit d'apprendre ou de maîtriser les langages C et C++ qui présentent la base de ce logiciel.
- Un environnement de programmation simple et clair : le logiciel Arduino (IDE) est facile à être utilisé pour les débutants, pour se familiariser il suffit de suivre les instructions et les guides disponibles par les fondateurs de ce logiciel, ainsi il est assez flexible pour que les utilisateurs avancés puissent en tirer profit également.

II.2.3 Partie physique de l'Arduino UNO

Dans cette partie nous allons présenter les différents éléments constituant cette carte (figure II.5).

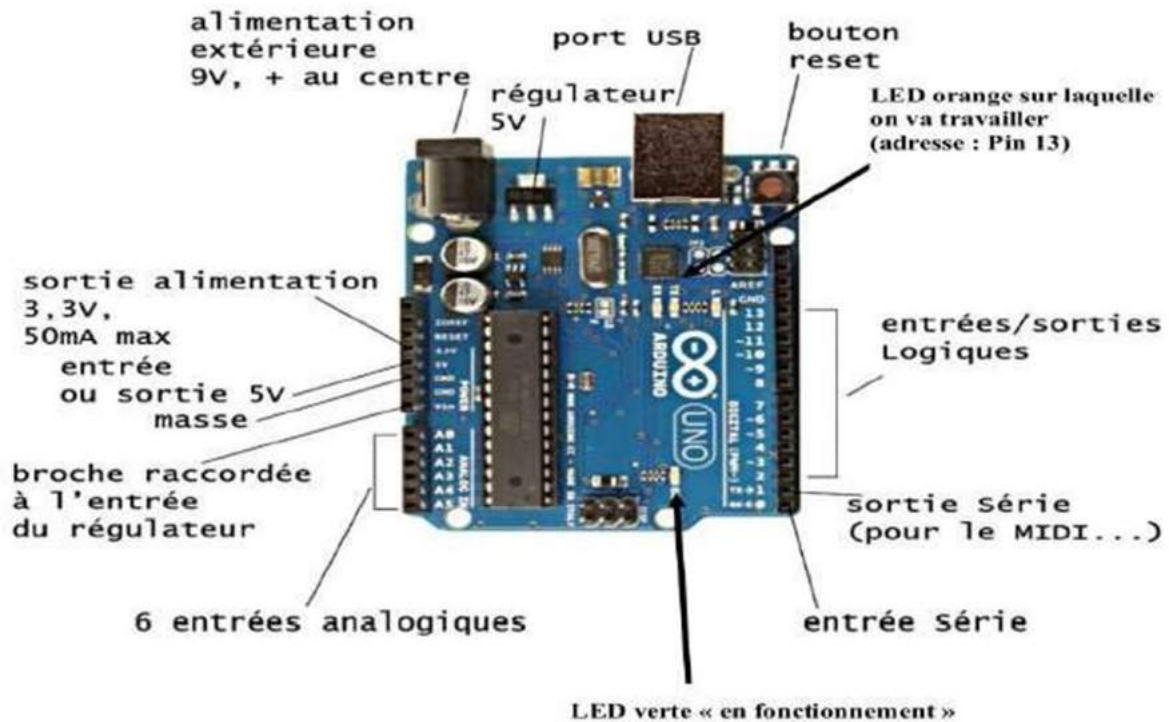


Figure II.5 : Les différentes parties de la carte Arduino UNO « partie matérielle ».

II.2.3.1 Dimension et organisation

Cette carte mesure environ 52 mm sur 65 mm, elle est équipée d'un microcontrôleur Atmel ATmega 328 (figure I.6). De plus elle est équipée de deux rangées de connecteurs situées de part et d'autre de la carte, permettent sa connexion au monde extérieur par l'utilisation des shields, cette connexion est faite directement sans recours à la soudure ou à l'utilisation des fils de connexion.



Figure II.6 : Microcontrôleur Atmel ATmega 328.

II.2.3.2 Alimentation

L'alimentation de fonctionnement du microcontrôleur (Vin) est fournie soit par la prise USB dans le cas où la carte Arduino est liée à un ordinateur, ou bien elle est fournie par un bloc extérieur via le jack standard présent sur la carte Arduino.

La carte Arduino peut alimenter l'extérieur par des tensions stabilisées de valeurs 5 volts ou 3.3 volts, la stabilisation de ces tension est assurée par des régulateurs implantés dans la carte. Ces tension peuvent alimenter des montages à faible consommation de courant environ 50 mA pour la sortie 3.3 volts.

II.2.3.3 Les mémoires

Le microcontrôleur ATmega utilise trois types de mémoires :

- **La mémoire Flash** : elle est de 32 kilo-octets de capacité, elle se programme électriquement de manière transparente lorsque l'Arduino est utilisé avec son logiciel de développement à travers le câble USB. Elle contient le programme qu'on veut l'exécuter, une fois cette mémoire est programmée elle conserve le contenu de ce programme même dans le cas où la carte est débranchée ou n'est pas alimentée.
- **La mémoire vive (RAM)** : elle est de 2 kilo-octets de capacité, elle est utilisée pour les variables employées dans les programmes, aussi pour stocker les résultats temporaires lors de calcul, elle est caractérisée par le pouvoir d'écriture et de lecture sur elle dans n'importe quel instant par le microcontrôleur, elle est dite oubliée car son contenu est effacé dès qu'elle est débranchée de l'alimentation.
- **La mémoire EEPROM**: elle est de 1 kilo-octet de capacité, c'est une mémoire programmable et effaçable électriquement, elle permet au microcontrôleur à tout instant de lire, d'écrire ou même d'effacer son contenu avec plus de facilité par rapport à la mémoire Flash. Son contenu est conservé même lorsqu'elle n'est plus alimentée.

II.2.3.4 Reset

Toutes les cartes Arduino sont équipées d'un bouton poussoir dit Reset manuel, un appui sur lui permet à l'utilisateur de relancer l'exécution d'un programme soit parce qu'il s'est planté ou bien de le refaire à partir de son début.

Il existe deux autres sources de Reset, la première est un Reset automatique permet le démarrage automatique du programme contenu en mémoire du microcontrôleur dès la mise

sous tension de la carte, la deuxième source de Reset est provoquée directement à l'aide de logiciel de programmation l'utilisateur lance le programme vers le microcontrôleur sans besoin d'appui sur le bouton poussoir de Reset.

II.2.3.5 Les Entrées / Sorties

Les Entrées / Sorties présentent le moyen à travers lequel le microcontrôleur communique avec le monde extérieur ou bien avec les Shields. L'Arduino Uno possède deux type entrées / Sorties:

- **Entrées / Sorties numériques**

La carte Arduino Uno contient quatorze lignes (broches) d'Entrées / Sorties, elles peuvent être utilisé soient en Entrée numérique ou en Sortie numérique, elles sont repérées par les chiffres 0 à 13, leurs fonctionnement est sous le contrôle du programme exécuté par le microcontrôleur, elles peuvent changer de l'état de manière dynamique, ces entrées / sorties admettent et délivrent des signaux logiques compatibles TTL compris entre 0 et 5 volts.

Ces lignes (broches) partagent leur rôle avec certaines interfaces spécialisées contenues dans le microcontrôleur, c'est-à-dire elles peuvent être utilisé comme des Entrées / sorties numériques ou bien elles peuvent être utilisé comme suit :

- ✓ L'environnement de développement est disponible gratuitement (open source) pour les ordinateurs fonctionnant sous Linus, Windows et Mac OSX. Il contient un éditeur de texte pour écrire les codes de programme, une zone de message, une console de texte, une barre d'outils avec des boutons pour les fonctions communes et une série de menus voir la figure II.5. Son utilisation ne demande pas assez d'effort mais des simples manipulations, il suffit d'écrire le programme dans la zone d'édition de programme, puis un simple clic sur l'icône appropriée de la barre d'outils pour l'envoyer à la mémoire de microcontrôleur pour être exécuter par ce dernier les lignes 0 et 1 utilisées respectivement comme entrée série (Rx) et sortie série asynchrone (Tx). Dans notre projet on a utilisé ces deux lignes dans la communication avec le module Xbee.

- ✓ Les lignes 2 et 3 utilisées comme des entrées d'interruptions externes.

- ✓ Les lignes 3, 5, 6, 9,10 et 11 utilisées comme des sorties PWM (Pulse Width Modulation en français modulation par largeur d'impulsion).

- ✓ Quatre entrées / sorties d'interface avec le bus série normalisé de type SPI :

SS sur la ligne 10, MOSI sur la ligne 11, MISO sur la ligne 12 et SCK sur la ligne 13.

- **Entrées analogiques**

La carte Arduino Uno dispose six entrées analogiques, repérées A0 à A5, ces entrées peuvent admettre une tension analogique comprise entre 0 et 5 volts ces entrées sont gérées par un convertisseur Analogique / Numérique de dix (10) bits de résolution c'est-à-dire la sortie varie entre 0 et 1023.

II.2.4 Partie logiciel de l'Arduino

L'Arduino est une carte à base d'un microcontrôleur donc il est possible de la programmer par le langage machine des microcontrôleurs de famille AVR. Mais les fondateurs de projet Arduino et dans le but d'avoir un maximum d'utilisateurs de leur produit, ils ont développé un langage de programmation et un environnement de développement propre à ce produit appelé : logiciel Arduino (IDE).

II.2.4.1 L'environnement de développement de L'Arduino

Cet environnement de développement est disponible gratuitement (open source) pour les ordinateurs fonctionnant sous Linux, Windows et Mac OSX. Il contient un éditeur de texte pour écrire les codes de programme, une zone de message, une console de texte, une barre d'outils avec des boutons pour les fonctions communes et une série de menus voir la figure I.7. Son utilisation ne demande pas assez d'effort mais des simples manipulations, il suffit d'écrire le programme dans la zone d'édition de programme, puis un simple clic sur l'icône appropriée de la barre d'outils pour l'envoyer à la mémoire de microcontrôleur pour être exécuter par ce dernier (Figure II.7) .

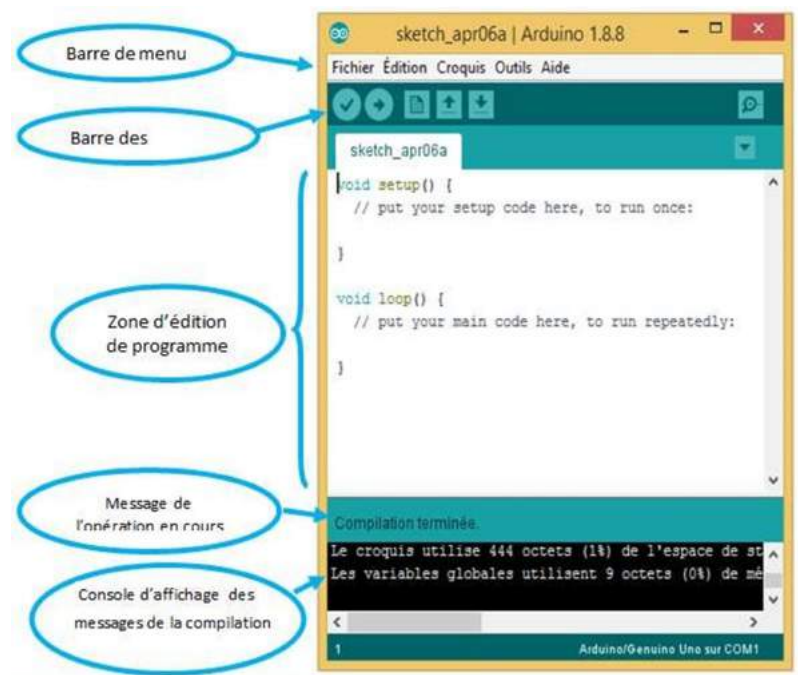


Figure II.7 : Les principales fonctions dans l'environnement Arduino.

Elles sont présentées par une barre de boutons situé en haut de la fenêtre de travail de l'environnement, ces boutons assurent 90% des besoins lors de l'écriture et du test d'un programme, on peut décrire leurs fonctions par le tableau II.1 suivant :

Tableau I.1 : Les principales fonctions dans l'environnement IDE (Barre des boutons).







Bouton	Fonction
	Vérifier : compile le sketch (programme) c'est à dire vérifier sa syntaxe
	Téléverser : compile le programme et le télécharger en mémoire de l'Arduino.
	Nouveau : ouvre une fenêtre vierge pour écrire un nouveau programme.
	Ouvrir : ouvre la liste de tous les sketches contenus dans votre <u>sketchbook</u> .
	Enregistrer : sauvegarde le sketch en cours d'édition.
	Moniteur Série : ouvre l'écran du moniteur série intégré à l'environnement

Tableau II.1 : La barre des boutons

II.2.4.2 Structure d'un programme

Un programme destiné à l'Arduino est composé par deux fonctions principales :

- **La fonction setup**

Elle contient les instructions d'initialisation de certaines ressources de la carte Arduino à titre d'exemple la déclaration des entrées sorties, la vitesse de fonctionnement du port série ...etc. la fonction setup s'exécute seulement une fois juste après le lancement de programme. Elle s'écrit de la façon suivante:

```
Void setup ()  
{  
// put your setup code here, to run once:  
}
```

- **La fonction loop**

Elle contient les instructions qui seront indéfiniment répétées tant que l'Arduino est sous tension. Elle s'écrit de cette façon :

```
void loop() {  
// put your main code here, to run repeatedly :
```

Dans un programme on peut trouver aussi une troisième partie, elle ne contient pas des instructions exécutables mais elle s'agit d'une zone de définition de constantes par l'utilisation des directives **define** et **const**, aussi elle contient l'inclusion de bibliothèques utilisées par ce programme. Cette partie est située avant la fonction setup.

II.3 Présentation de module Xbee [¹]

II.3.1 Définition

Les produits XBee™ sont des modules de communication sans fil très populaires fabriqués par l'entreprise Digi International (Figure II.8). Ils ont été certifiés par la communauté industrielle ZigBee alliance en 2006. La certification **Zigbee** se base sur le standard **IEEE 802.15.4** qui définit les fonctionnalités et spécifications des réseaux sans fil à dimension personnelle (Wireless Personal Area Networks: WPANs).

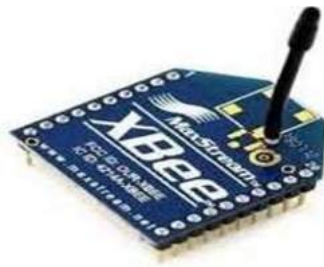


Figure II.8 : Module Xbee

Il faut distinguer entre Zigbee et Xbee, le premier est le protocole de communication et le second est une marque, un produit qui utilise le protocole ZigBee.

II.3.2 Les principales caractéristiques du Xbee

- Fréquence porteuse : 2.4Ghz.
- Portées variées (la distance de transmission) :
 - Assez faible pour les Xbee 1 et 2 (**10 - 100m**)
 - Grande pour le Xbee Pro (**1000m**)
- faible débit de transmission et réception (250 kbps)
- faible consommation d'énergie (Alimentation) :
 - Voltage 3.3 volts, ampérage 50 mA
- Entrées / Sorties :
 - 6 broches d'Entrées, 8 broches E/S numériques.
- Faible cout : 25 euros.
- Utilisation simple à travers le port série.

II.3.3 Application

L'utilisation de la communication sans fil est devenue très pratique dans la vie quotidienne, surtout par le protocole zigbee c'est-à-dire l'utilisation des module Xbee qui est moins couteux par rapport ou autre modules (bluetooth et wifi), parmi les applications on peut citer (Figure II.9) :

- Système de commande pour les portes et les fenêtres.
- Système de sécurité.
- Commande des lumières
- Télévision
- La climatisation et la température.

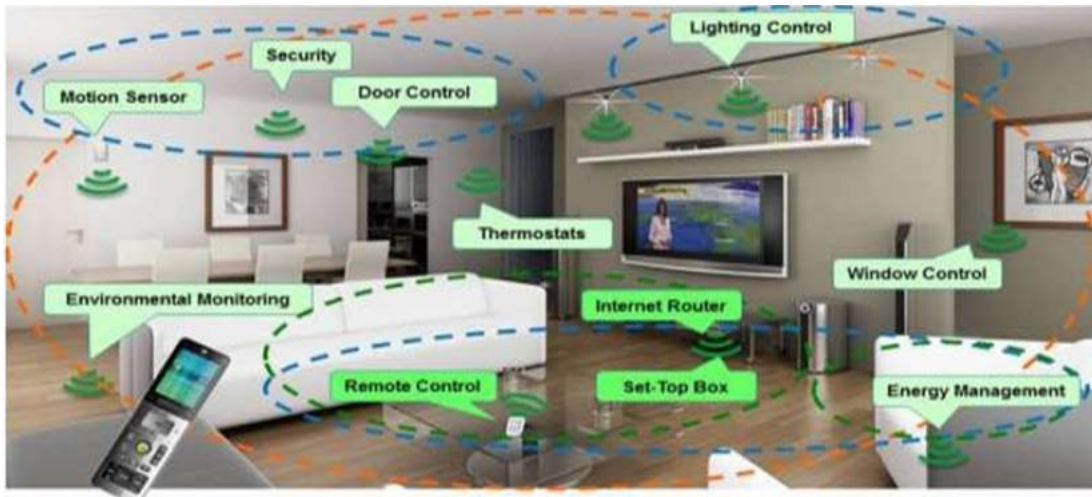


Figure II.9 : Les différentes applications des modules Xbee.

Xbee Série1 et Série 2

Il y a deux catégories des modules Xbee :

Les Xbee Série1 : Ces modules sont souvent liés au protocole "802.15.4" qui s'ajoutent leurs noms, voir la figure suivante (Figure II.10)



Figure II.10 : Xbee Série1

Les Xbee Série2 : ces modules utilisent le protocole **Zigbee**, ils sont disponibles en plusieurs versions : **Xbee ZN et 2.5** (obsolète), le **ZB** (l'actuel) et le **2B** (le plus récent). Vous avez aussi des **Xbee Pro**, qui font la même chose, mais avec de plus grandes capacités, notamment la portée qui semble pouvoir aller jusqu'à 1000 mètres, voir la figure II.11 suivante.



Figure II.11 : Xbee Série2.

Le tableau I.2 suivant présente les principales différences entre les deux modules Xbee S1 et Xbee S2.

	Series 1	Series 2
Typical (indoor/urban) range	30 meters	40 meters
Best (line of sight) range	100 meters	120 meters
Transmit/Receive current	45/50 mA	40/40 mA
Firmware (typical)	802.15.4 point-to-point	ZB ZigBee mesh
Digital input/output pins	8 (plus 1 input-only)	11
Analog input pins	7	4
Analog (PWM) output pins	2	None
Mesh, cluster tree topologies	No	Yes
Single firmware for all modes	Yes	No
Requires coordinator node	No	Yes
Point-to-point configuration	Simple	More involved

Tableau II.2 : Les caractéristiques des modules Xbee S1 et S2.

Nous remarquons :

- La portée et la consommation sont sensiblement les mêmes.
- Le nombre d'entrées et sorties est différent et surtout la série 2 ne possède pas de sorties analogiques PWM.
- Les modules Xbee S1 utilisent le protocole 802.15.4 et les modules Xbee utilisent le protocole Zigbee.
- Les topologies de réseaux possibles ne sont pas les mêmes. Avec la série 1, l'architecture est simple : point à point (*pair*) ou multipoint (*star*). La **série 2** permet en plus de créer des réseaux plus complexes: maillés (*mesh*) ou en "arbre" (*cluster tree*). Voir la figure suivante.

Ces différences rendent les deux modules Xbee S1 et Xbee S2 pas compatibles entre eux c'est-à-dire on doit utiliser des modules de la même catégorie pour avoir une communication.

II.3.4 Les topologies des réseaux pour les Xbee

On distingue quatre types de topologie voir la figure II.12 suivante:

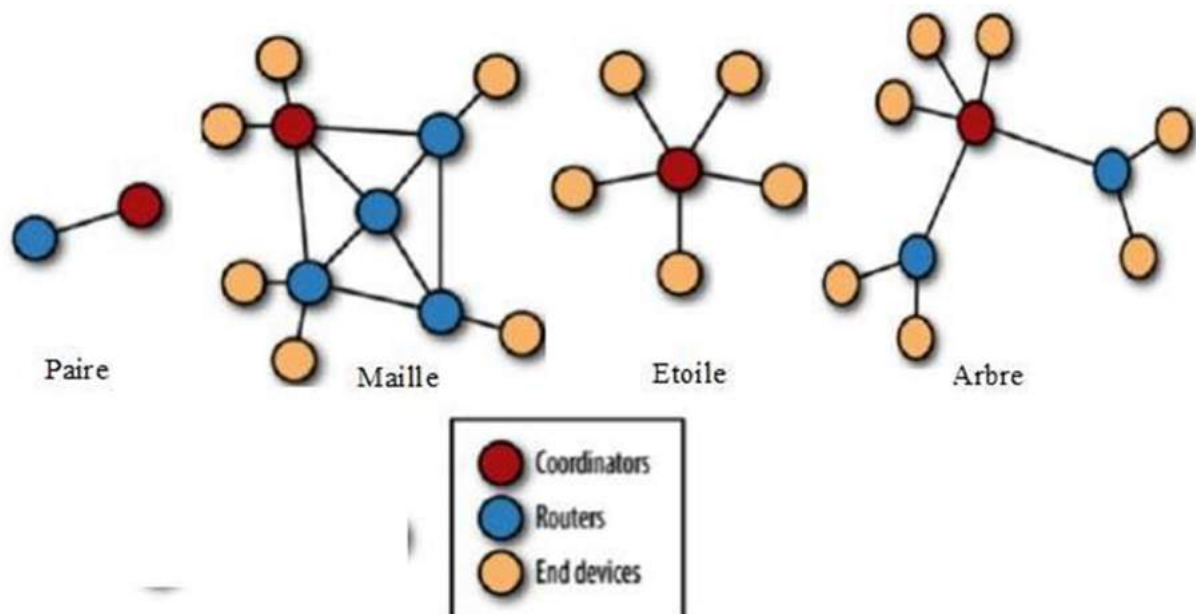


Figure II.12 : Les topologies des réseaux pour les Xbee.

II.3.5 Les antennes

Les modules Xbee sont équipés par des antennes pour assurer l'émission et la réception des ondes radio. On distingue quatre types d'antennes (figure II.13):

- **Wire** : simple, radiations omnidirectionnelles;
- **Chip** : puce plate en céramique, petite, transportable (pas de risques de casser l'antenne), radiations cardioïdes (le signal est atténué dans certaines directions).
- **U.F.L** : une antenne externe n'est pas toujours nécessaire.
- **RPSMA** : plus gros que le connecteur *U.F.L* permet de placer son antenne à l'extérieur d'un boîtier.

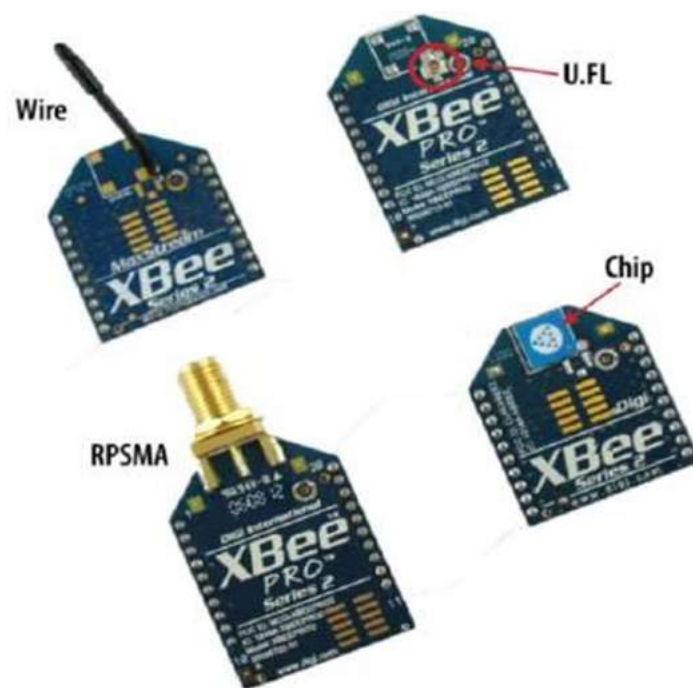


Figure II.13 : Les différentes antennes.

II.3.6 Les modes de communication

Les modules Xbee fonctionnent selon trois modes de communication :

- **Le mode transparent** : ce mode est le mode par défaut dès la mise en marche des modules la communication est assurée sans aucune condition.

- **Le mode command** : dans ce mode on doit configurer les entrées sorties des modules, l'adresse de destinataire, la vitesse de l'émission ...etc.
- **Le mode API (Application programming interface)** : Il désigne une interface fournie par un programme informatique, c'est-à-dire un ensemble de fonctions qui facilitent la programmation d'un côté et qui de l'autre communique en langage binaire pour le XBee sous forme de paquets.

II.3.7 L'alimentation

Les modules Xbee utilisent une alimentation de valeur 3.3 volts durant l'émission et la réception des données, donc pour brancher un Arduino qui utilise 5 volts comme alimentation avec un Xbee, il est nécessaire d'utiliser un limiteur ou bien régulateur de tension pour conserver ce dernier. Cette opération est faite soit par un diviseur de tension ou bien soit un régulateur dans notre cas on a utilisé un régulateur (le 74 HC 4050).

II.3.8 Montages et brochages des Xbee

Dans notre circuit, nous avons utilisé deux Xbee pour la transmission et la réception du signal PCG, le premier Xbee reçoit l'information détectée par Arduino pour la transmettre vers un deuxième Xbee. Le premier module Xbee est lié à une carte FTDI.

a- Le circuit de l'émission

La carte Arduino détecte le signal PCG qui sera envoyé à travers la liaison série de l'Arduino (Rx, Tx) vers les entrées Rx et Tx de module Xbee. Le brochage est donné par la figure II.14 suivante.

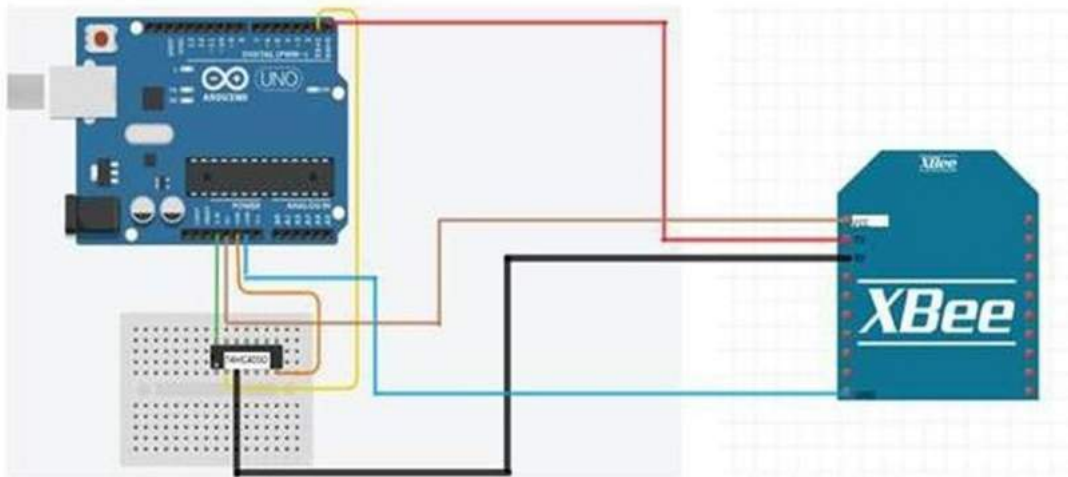


Figure II.14 : Circuit de l'émission.

On relie le Vcc de module Xbee avec la tension de sortie 3.3 volts de l'Arduino, la broche GND de l'Arduino avec celle de Xbee, le Tx de module Xbee avec Rx de l'Arduino, concernant le Tx de l'Arduino elle ne peut être reliée directement au Rx de module Xbee car le module Xbee ne supporte pas la tension 5 volts donc elle est reliée au régulateur (74HC4050) puis sa sortie au Rx de module Xbee.

b- Le circuit de réception

Le deuxième Xbee reçoit l'information à l'aide de son antenne, ce module est connecté directement sur une carte FTDI. Voir la figure II.15.



Figure II.15 : Le circuit de réception.

II.3.9 La configuration des modules Xbee

Avant de réaliser les montages précédents et pour assurer la communication entre les deux Xbee, on doit les configurer. Cette configuration est faite à l'aide de logiciel XCTU, elle permet à l'utilisateur de choisir le mode de communication (Command ou API) sinon ces deux Xbee se communiquent directement (mode transparent) sans aucune configuration (Figure II.16).

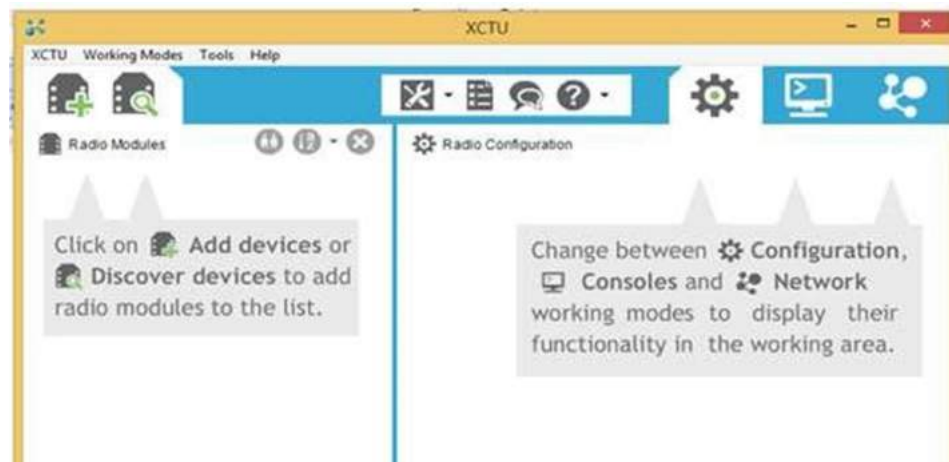


Figure II.16: Logiciel de configuration des Xbee (XCTU).

II.4 La carte FTDI

La carte FT232R est le dernier appareil à être ajouté à la gamme de circuits intégrés à interface USB UART de FTDI. Le FT232R est une interface USB / série UART avec une sortie de générateur d'horloge en option et la nouvelle fonction de clé électronique de sécurité FTDI Chip-ID™. Voir la figure II.17.

Dans notre projet elle est connectée avec le deuxième module Xbee, son rôle est d'assurer la liaison avec le micro-ordinateur pour transmettre le signal PCG reçu par le deuxième module Xbee.

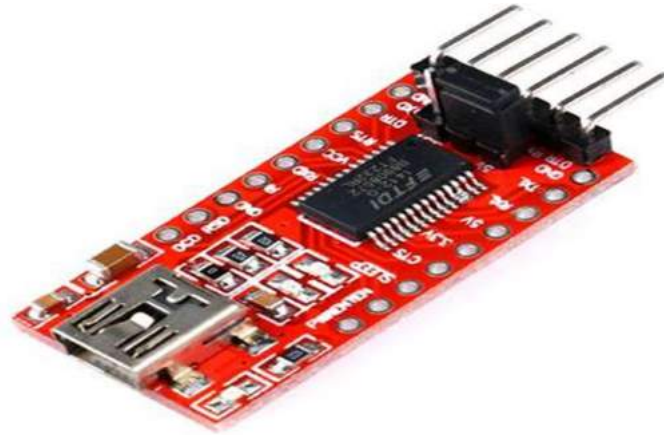


Figure II.17: La carte FT232R

II.5 Conclusion

Le rôle d'une acquisition de données est de transcrire l'information contenue dans un signal donné sous forme de données informatiques. Dans ce chapitre, nous avons présenté un système embarqué d'acquisition de données basé sur les fonctionnalités de la carte Arduino Uno.

Ce système d'acquisition de données informatisé est constitué à la fois d'une partie hardware constitué de la carte multifonctions Arduino Uno; et d'une partie software qui consiste en deux parties élémentaires.

Chapitre III

Réalisation du réseau de
transmission sans fil

Chapitre III

Réalisation du réseau de transmission sans fil

III.1 Introduction

Dans ce dernier chapitre nous allons présenter la partie de réalisation pratique.

Tous d'abord, on présentera le matériel utilisé ainsi que les programmes utilisés pour arduino et XCTU, puis on passe à la réalisation pratique.

III.2. Organigramme

Le schéma bloc des nœuds 1 et 2 de l'émission sont représenté dans la figure Fig. III.1.

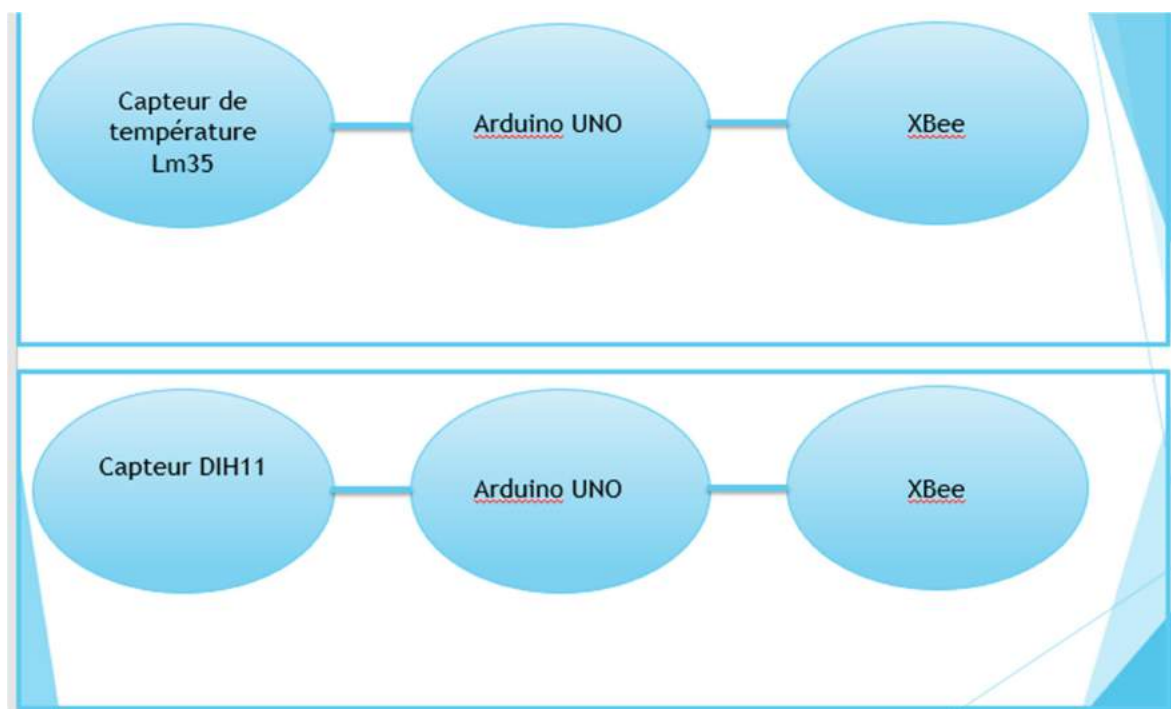


Figure III.1:Schéma blocs de l'émission.

III.3 Matériel utilisé

- Microprocesseur Raspberry
- Microcontrôleur Arduino uno

- modules Xbee
- Capteur lm35
- Capteur Dht11
- Plaque d'essai
- câbles de connexion
- Pc
- régulateur

III.3.1 Capteur LM35

Ce capteur est représenté à la figure III.2 ci-dessous.

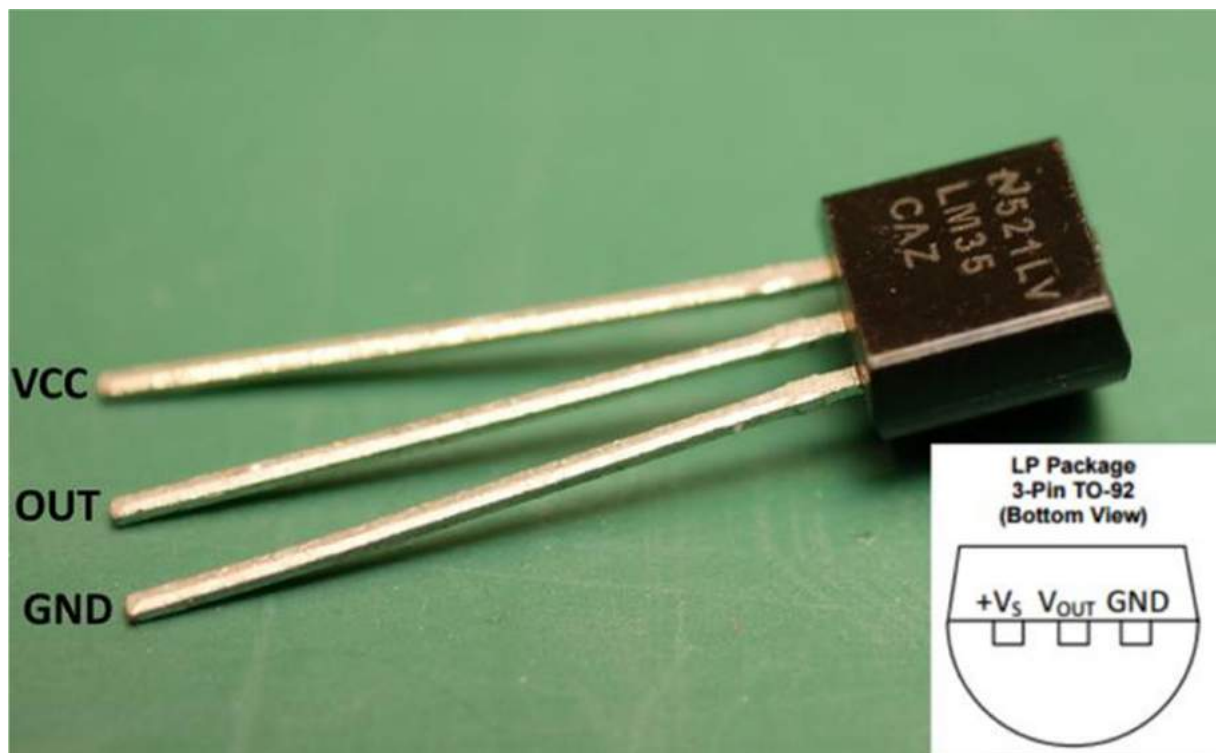


Figure III.2: LM35

Le LM35 est un capteur de température analogique. Cela signifie que la sortie du LM35 est un signal analogique. Les microcontrôleurs n'acceptent pas directement les signaux analogiques

comme entrée. Nous devons convertir ce signal de sortie analogique en signal numérique avant de pouvoir l'envoyer à l'entrée d'un microcontrôleur.

III.3.2 LM35 et Arduino - Schéma du circuit

Le LM35 est connecté à l'Arduino uno comme indiqué dans le schéma du circuit. Le +5v pour le LM35 peut être extrait de la broche +5v de arduino uno. La broche de terre du LM35 peut également être connectée à la broche GND de l'arduino uno. On connecte Vout (la sortie analogique du LM35) à n'importe quelle broche d'entrée analogique de arduino uno. Dans ce schéma de circuit, nous avons connecté Vout de LM35 à la broche A₀ de l'Arduino.

III.3.3 Le programme - Interfaçage LM35 et Arduino

A la figure III.3 ci-dessous on représente le programme d'interfaçage Arduino avec le capteur de température LM35.



```
lm35 | Arduino 1.8.15
Fichier Édition Croquis Outils Aide

lm35

/* LM35 analog temperature sensor with Arduino example code. More info: https://www.makerguides.com/ */
// Define to which pin of the Arduino the output of the LM35 is connected:
#define sensorPin A0

void setup() {
  // Begin serial communication at a baud rate of 9600:
  Serial.begin(115200);
}

void loop() {
  // Get a reading from the temperature sensor:
  int reading = analogRead(sensorPin);
  // Convert the reading into voltage:
  float voltage = reading * (5000 / 1024.0);
  // Convert the voltage into the temperature in degree Celsius:
  float temperature = voltage / 10;
  // Print the temperature in the Serial Monitor:
  Serial.print(temperature);

  Serial.println("C");
  delay(5000); // wait a second between readings
}
```

Figure III.3: Programme interfaçage Arduino-LM35.

III.3.4 Montages et branchages des Xbee

Dans notre circuit, nous avons utilisé deux circuits, un circuit d'émission et un circuit de réception. Le premier Xbee reçoit l'information détectée par Arduino pour la transmettre vers un deuxième Xbee. Le premier module Xbee est lié à une carte FTDI.

a- Le circuit de l'émission

La carte Arduino détecte le signal qui sera envoyé à travers la liaison série de l'Arduino (Rx, Tx) vers les entrées Rx et Tx de module Xbee (Fig. III.4 et Fig. III.5)).

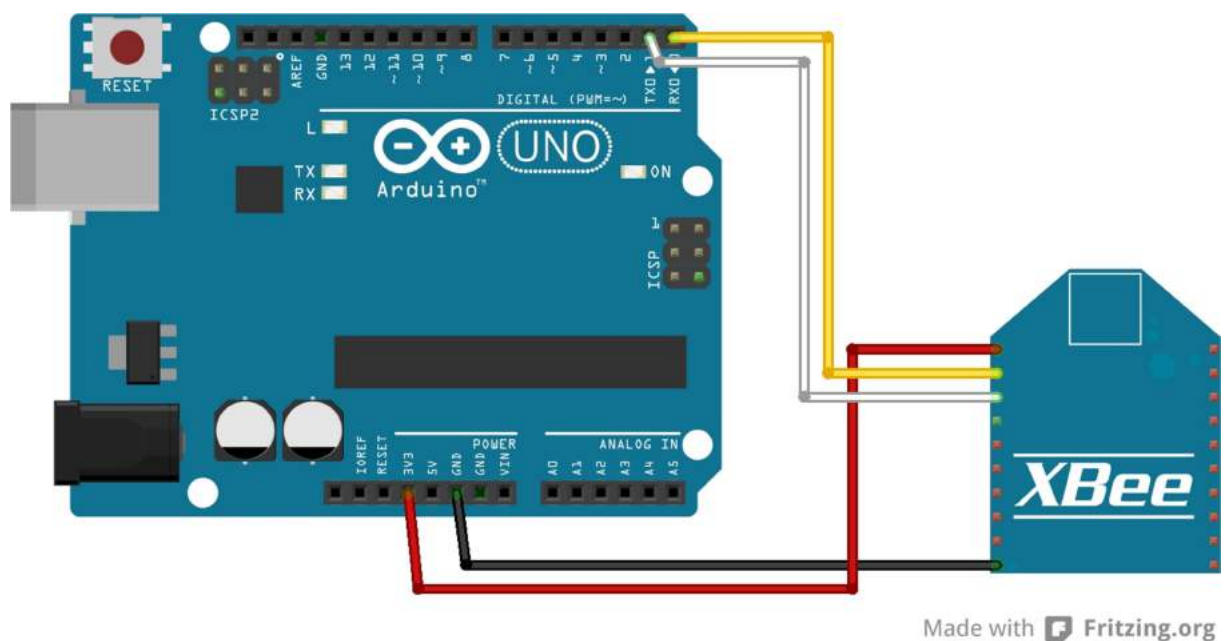


Figure III.4: Branchement Arduino avec Xbee

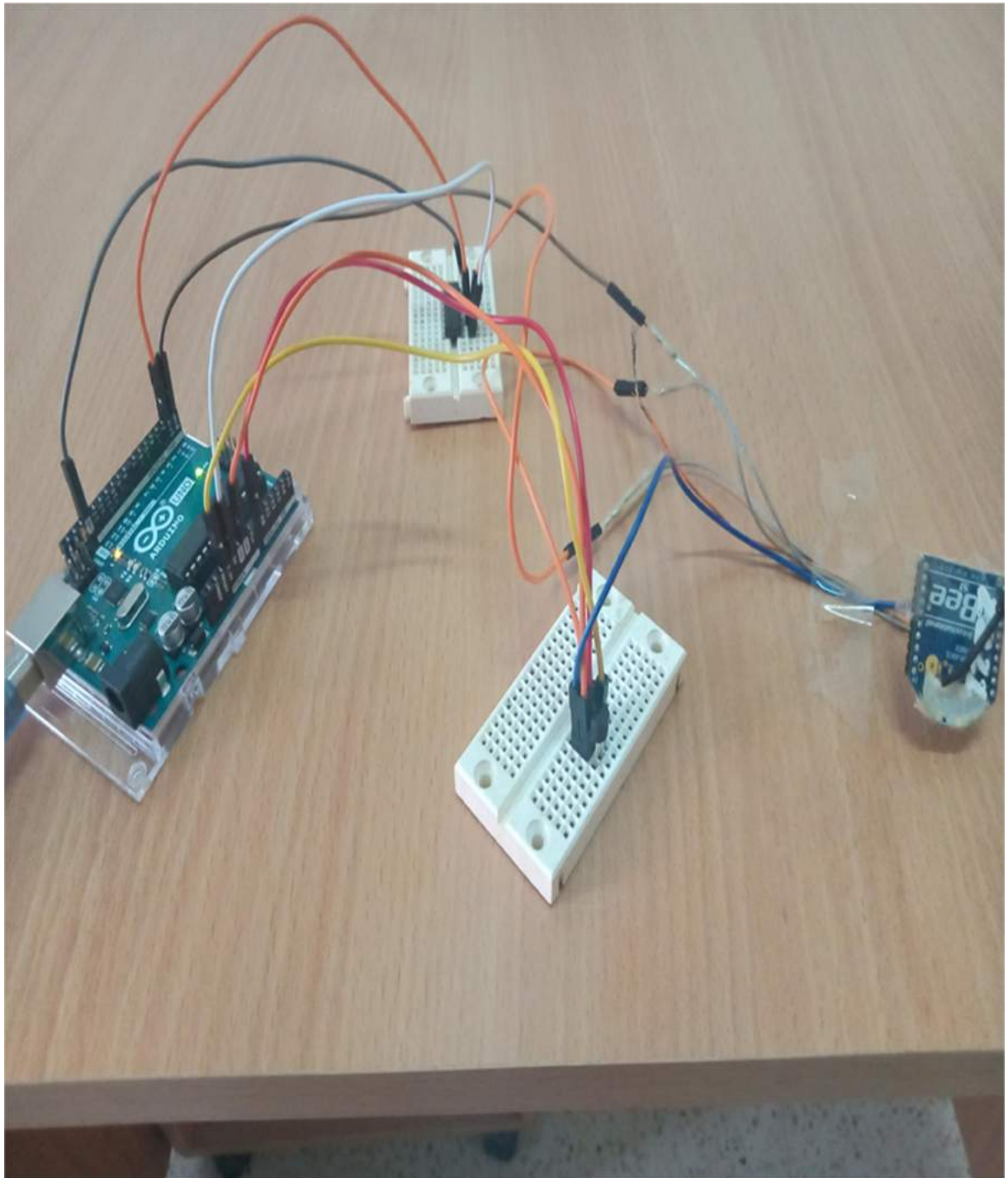


Figure III.5: Branchement Arduino avec Xbee et LM35.

III.3.5 Capteur de température et d'humidité DHT11

C'est un capteur de Température et d'Humidité très apprécié pour sa simplicité de mise en œuvre et son coût peu élevé (Fig. III.6). Il ne requiert qu'une résistance de tirage et une alimentation 3V ou 5V pour fonctionner. Sa programmation est facile à l'aide des bibliothèques Arduino,

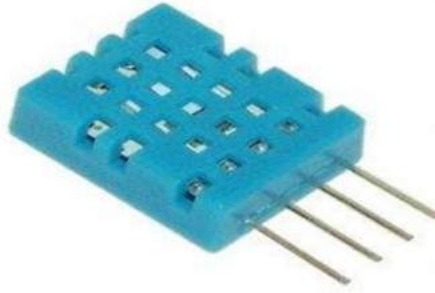


Figure III.6: Capteur de température et d'humidité DHT11.

Le module numérique DHT11 délivre et reçoit un signal digital sur une entrée/sortie série unique. Ses deux capteurs analogiques sont une résistance qui détermine le taux d'humidité et une thermorésistance de type NTC (Negative Temperature Coefficient) afin de mesurer la température.

La calibration du module est faite en usine et les paramètres sont sauvegardés dans une mémoire OTP (One Time Programming). Le lien entre l'Arduino et les capteurs est assuré par un microcontrôleur 8 bits. Une communication avec le module s'effectue sur 40 bits et dure typiquement 4ms. Le DHT11 est caractérisé par :

Hysteresis: $\pm 1\%$ RH, Stabilité : $\pm 0.5\%$ RH/an, Période de mesure: 2s, Dimensions: 12x15.5x5.5, Alimentation: 3-5,5 V DC, Signal de Sortie: Signal Numérique via single-bus, Capteur: Résistance Polymère, Plage de Mesure: Humidité: 20-90%RH; Température: 0-50°C.

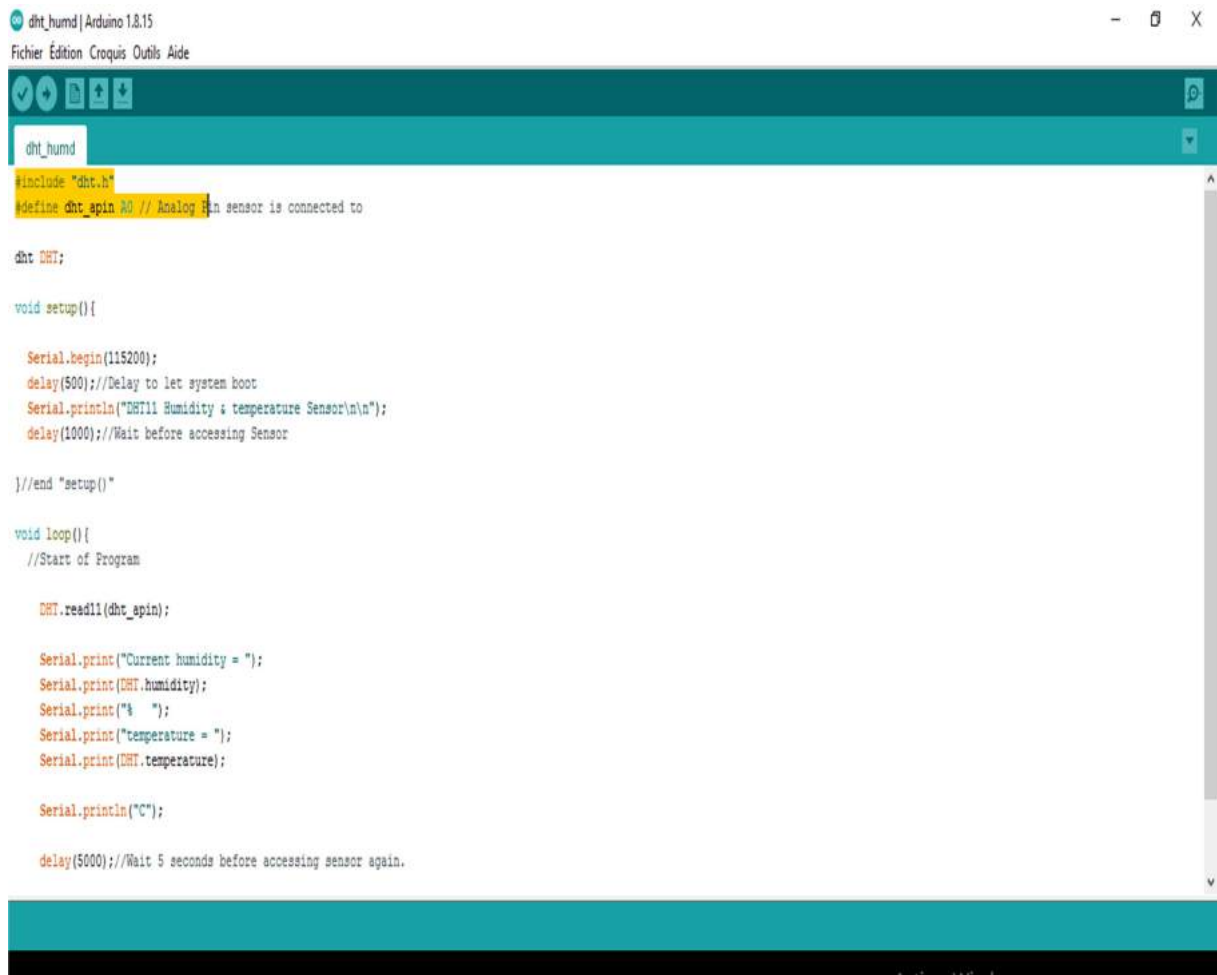
Précision:

Humidité : $\pm 4\%$ RH (Max $\pm 5\%$ RH); Température $\pm 2.0\text{ }^{\circ}\text{C}$.

Résolution: Humidité 1% RH; Température 0.1°C

III.3.6 Le programme - Interfaçage DHT11 et Arduino

A la figure III.7 ci-dessous on représente le programme d'interfaçage Arduino avec le capteur DHT11.



```
dht_humd | Arduino 1.8.15
Fichier Édition Croquis Outils Aide

dht_humd

#include "dht.h"
#define dht_apin A0 // Analog pin sensor is connected to

dht DHT;

void setup() {

  Serial.begin(115200);
  delay(500); // Delay to let system boot
  Serial.println("DHT11 Humidity & temperature Sensor\r\n\r\n");
  delay(1000); // Wait before accessing Sensor

} // end "setup()"

void loop() {
  // Start of Program

  DHT.read11(dht_apin);

  Serial.print("Current humidity = ");
  Serial.print(DHT.humidity);
  Serial.print("  ");
  Serial.print("temperature = ");
  Serial.print(DHT.temperature);

  Serial.println("C");

  delay(5000); // Wait 5 seconds before accessing sensor again.
```

Figure III.7: Programme interfaçage Arduino-DHT11

III.3.7 Montages et branchages DHT11, Arduino et Xbee

Comme le précédent montage de LM35, On branche le capteur DHT11 à l'Arduino à l'entrée analogique A₀ (Fig. III. 8), puis on relie l'arduino au module Xbee (Fig. III. 9).

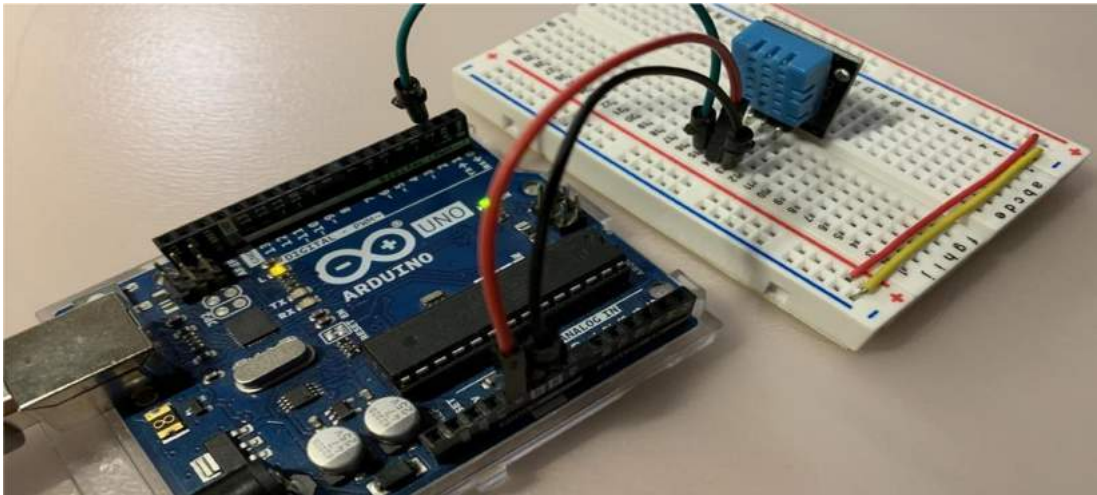


Figure III.8: Branchement Arduino et DHT11.

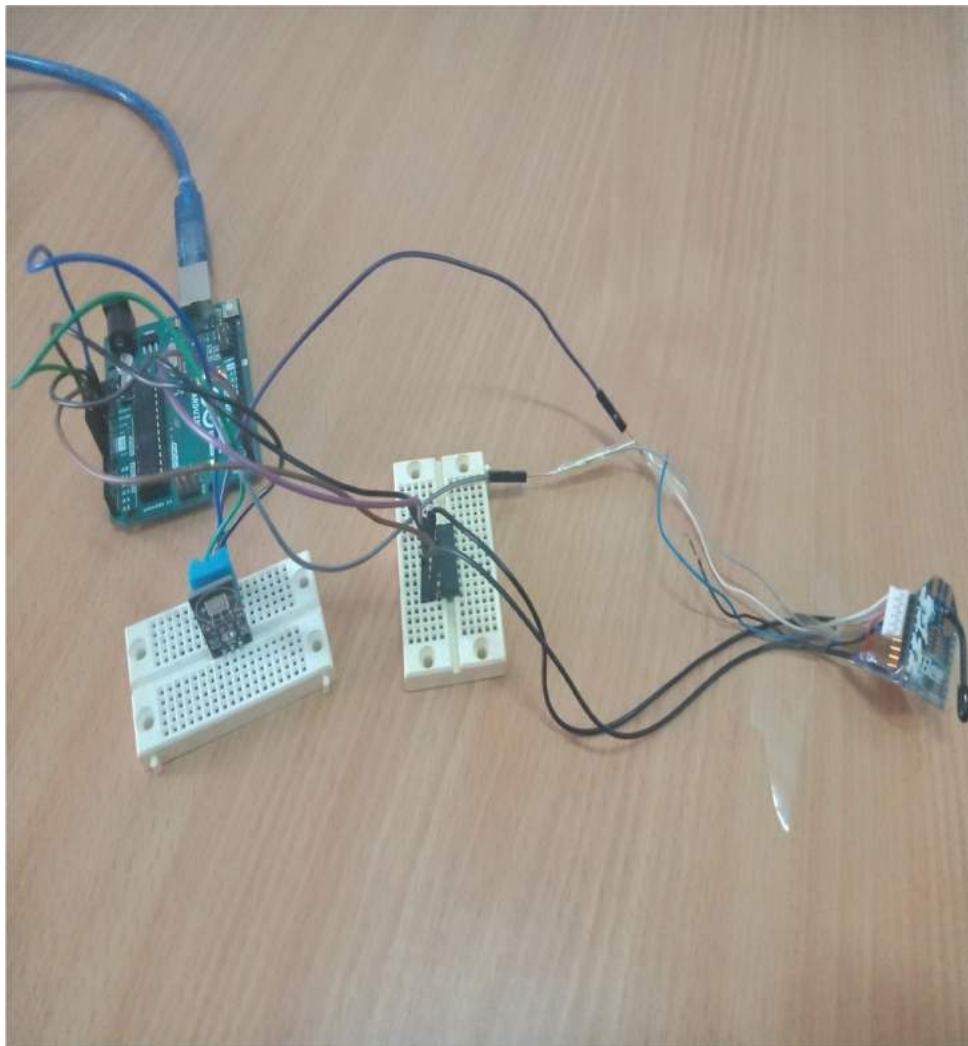


Figure III.9: Branchement Arduino, Xbee et DHT11.

III.8 Réalisation du réseau de transmission sans fil

La Figure III. 10 illustre le schéma synoptique du réseau sans fil à réaliser.

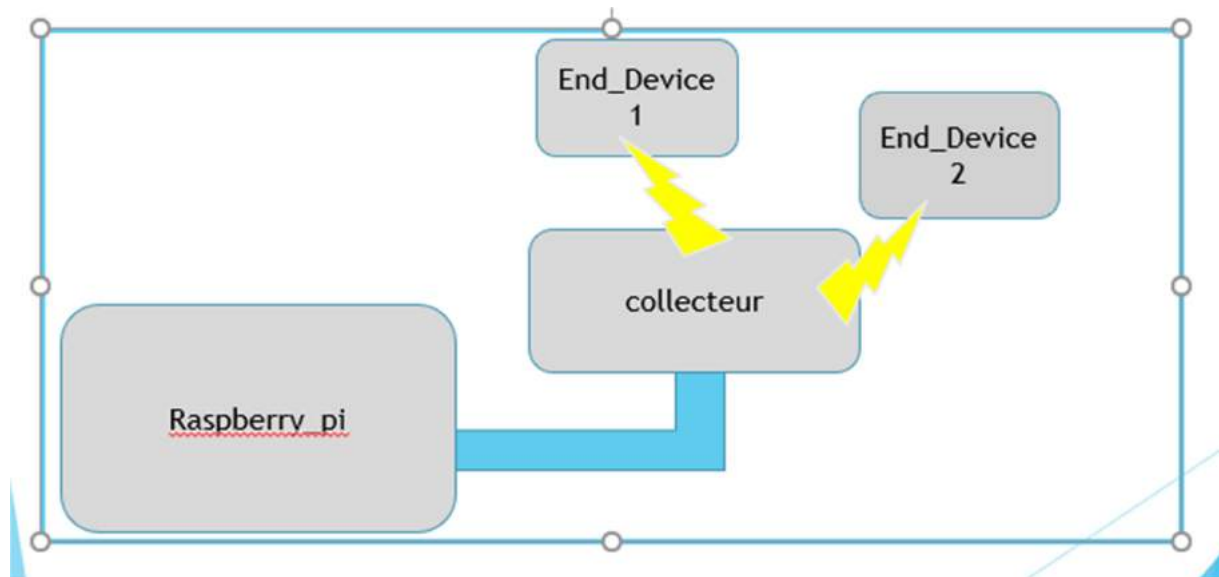


Figure III.10: Schéma synoptique du réseau sans fil.

III.8.1 Partie Matériel et logiciel

On présente dans cette partie le matériel et les logiciels nécessaires à la réalisation du réseau de transmission sans fil :

❖ **Arduino:**

- ✓ Arduino est une plateforme open source, basée sur un microcontrôleur ARM AT méga; la carte peut lire les entrées analogiques et numériques et les transformer en sortie (Fig. III. 11).
- ✓ Au cours de nos tests nous avons utilisé les Arduino comme étant des convertisseurs analogiques-digital.
- ✓ Nous avons utilisé un Arduino du type Arduino Uno, et nous avons choisi l'utilisation de l'Arduino car il nous permet de le programmer facilement et à moindre coût.



Figure III.11: Arduino Uno.

❖ Raspberry pi

Une photo du raspberry pi est donnée à la figure ci-dessous (Fig. III. 12).

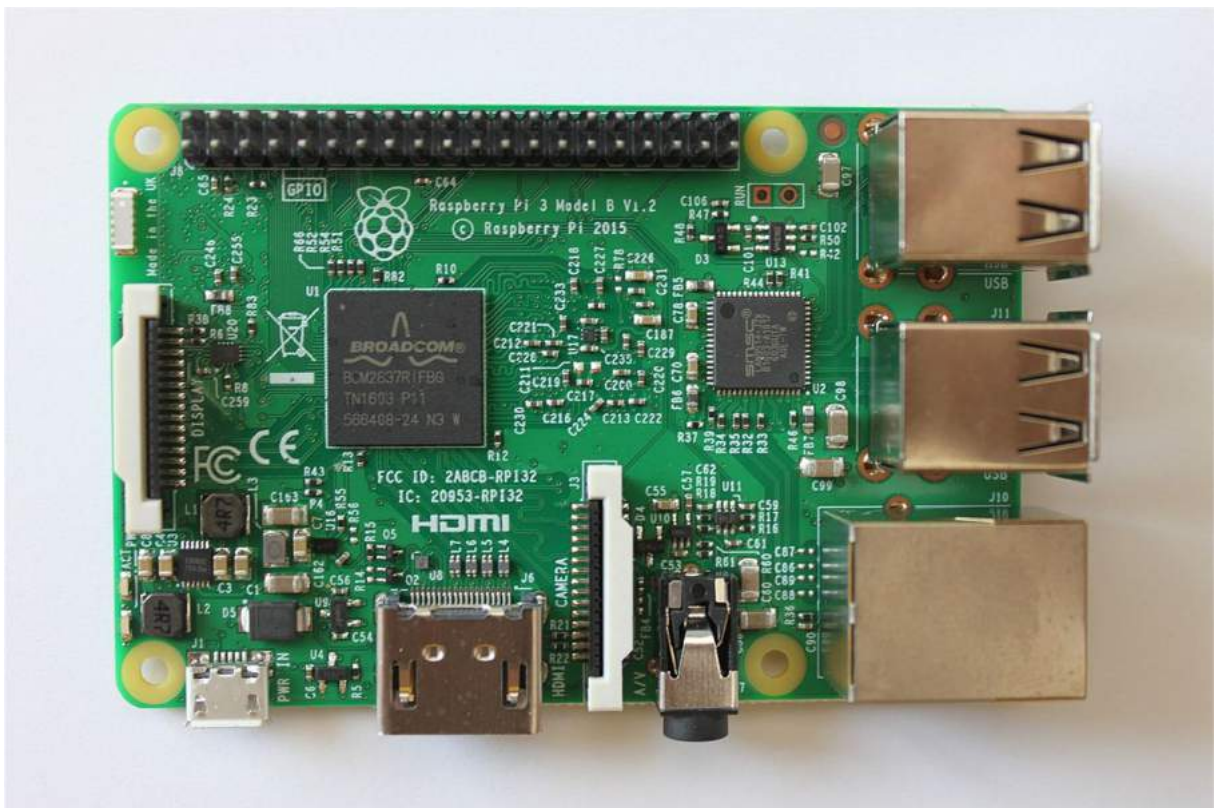


Figure III.12: Raspberry pi.

❖ XBee:

- ✓ Les modules radio XBee ont été conçus pour la transmission de données sans fil par la marque Digi (Fig. III. 13).
- ✓ Ces modules on peut les classer selon les critères les plus important à savoir : selon le protocole de transmission implémenté comme par exemple le wifi (802.11b/g/n), la norme 802.15.4 et la norme ZigBee. Selon la forme de montage.

Les types de Xbee :

A- Xbee Série 1 : portée maximale 100m.

B- Xbee Série 2 : portée maximale jusqu'à 1000m.



Figure III.13: module XBee.

- ✓ Les modules XBee contiennent un microcontrôleur programmé, prêt à l'emploi et configurable grâce au logiciel XCTU.
- ✓ Les modules XBee peuvent fonctionner avec un nombre minimum de connexions: une alimentation (3,3 V), mise à la terre, entrée et sortie de données (UART), les autres lignes recommandées étant Réinitialisation et Veille.
- ✓ De plus, la plupart des familles XBee intègrent d'autres lignes de contrôle de flux, d'entrée / sortie (I/O), de convertisseur analogique numérique (ADC) et d'indicateurs.

Nous avons utilisé ce module pour différentes raisons qui sont citées comme suit:

- 1- Son débit élevé.
- 2- Sa faible consommation en énergie.
- 3- Implémente la norme 802.15.4.

- 4- Simple à utiliser et à configurer.
- 5- S'adapte parfaitement aux autres modules de notre plateforme.
- ✓ Les périphériques XBee peuvent utiliser le mode de fonctionnement transparent AT ou API pour transmettre des données via l'interface série.
- ✓ On peut utiliser un mélange de périphériques exécutant le mode API et le mode transparent dans un réseau.
- ✓ Dans, un mode transparent (le mode AT) ou les XBee prennent en charge la communication, la construction de la trame, la vérification d'erreur, tout se fait d'une façon automatique et transparente.
- ✓ Tandis que dans le mode API, l'utilisateur s'occupe de la trame de transmission, et interprète cette trame en réception, cela peut être intéressant dans le cas du routage, pour le changement d'adresse dynamique, pour la localisation par la lecture de la puissance du signal grâce au RSSI (Received Signal Strength Indication) ou même pour la configuration à distance des XBee grâce à des commandes AT.

Dans notre cas on a utilisé le mode API dans la transmission des données car il convient à notre étude et il nous donne la possibilité de faire les tâches suivantes:

- 1- Former librement la trame d'envoi et par conséquent déterminer et connaître la taille adéquate de la charge utile à manipuler.
- 2- Réaliser le routage adéquat au niveau des routeurs.
- 3- Implémenter un algorithme de mobilité en exploitant la valeur du RSSI insérée dans la trame de réception.

III.9. Format de la trame en mode API

La trame API qu'utilise l'XBee lors d'une transmission comporte plusieurs champs de bits lors de l'envoi et de la réception des données (Fig. III. 14 et Fig. III. 15):

Start Délimiteur : 1 octet	Length (taille) de la trame : 2 octets	Type de trame : 1 octet	Identificateur de trame : 1 octet	Adresse du module : dynamique=16bits MAC=64bits	Option : 1 octet	Donnée ?	Checksum : 1 octet
----------------------------------	--	-------------------------------	--------------------------------------	---	---------------------	----------	-----------------------

Figure III.14 : Format de la trame API à l'envoi.

Start Délimiteur : 1 octet	Taille de la trame : 2 octets	Type de trame : 1 octet	Adresse du module	La puissance du signal reçu(RSSI)	Option : 1 octet	Donnée ?	Checksum : 1 octet
----------------------------------	--	-------------------------------	----------------------	--------------------------------------	---------------------	----------	-----------------------

Figure III.15 : Format de la trame API à la réception.

Les éléments de la trame sont décrits dans ce qui suit:

Start Délimiteur : c'est un octet indiquant le début de la trame égale à 0x7E en binaire c'est 0111 1110, une longue série de 1 qui réveille l'XBee s'il était en mode sleep.

Length : indiquant la taille de la trame en 2 octets.

Frame Type: type de trame (1 octet) pour spécifier son type si c'est une trame de transmission, de réception ou une des commandes AT.

Frame ID 'Identificateur de trame': un octet pour le numéro de trame afin de différencier entre les trames envoyés pour éviter une possible confusion en cas où il y a deux ou plusieurs trames identiques (retransmission).

Adresse: il y a deux types d'adresse, l'adresse fixe pour chaque XBee qui est l'adresse MAC ou l'adresse dynamique qui est mise dans (MY) lors de la configuration de l'XBee.

Option: c'est pour spécifier si l'XBee nous renvoi un (ACK) accusé de réception pour confirmer la bonne réception de la trame (sans erreur).

RSSI: indiquant la puissance du signal reçu en décibel, plus sa valeur augmente plus la portée est longue.

Donnée: peut aller jusqu'à 100 octets par trame dans la série 1 de l'XBee.

Checksum: c'est une valeur de détection d'erreur, c'est la somme de tous les octets de la trame, soustraite de 0xFF.

Les outils logiciels utilisés :

Dans cette partie nous décrivons tous les logiciels utilisés pour faire fonctionner la partie matérielle décrite précédemment : Python, Arduino IDE et XCTU.

III.10 Réalisation pratique des nœuds**III.10.1. Nœud final (End Device):**

- l'End Device représente le composant final dans un RCSF, il est responsable de l'acquisition des données analogiques, le traitement, et de l'envoi des données dans le réseau. On a conçu ce circuit autour d'un Arduino.
- Le capteur température LM35 communique avec la carte Arduino grâce à la pin analogique ADC1 et l'XBee communique avec la carte Arduino grâce aux entrées sorties Tx/Rx qui s'inverse lors de la connexion.
- Et la même chose pour le capteur DHT11.

III.10.2. Collecteur

- Le rôle du collecteur est de collecter toutes les données qui parviennent de tous ces nœuds dans le réseau. Nous avons réalisé le montage électronique à base du raspberry Pi (Fig. III. 16).

Cette étape consiste à recevoir les données envoyées par le nœud final. Puisque le montage de l'End device ressemble au montage du Sink (raspberry pi+xbec).

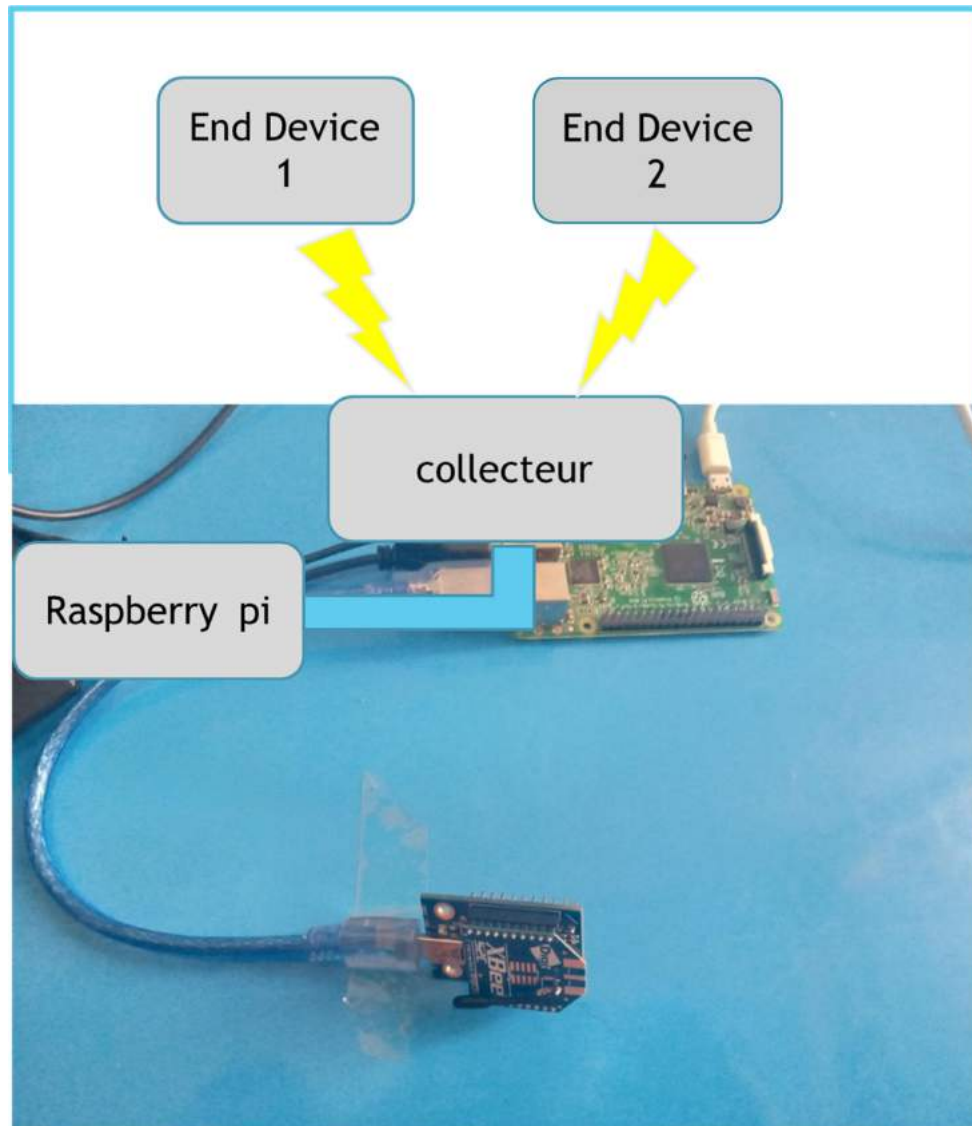


Figure III.16 : Photo du schéma de réalisation du réseau de communication sans fil.

III.10.3. Configuration des modules Xbee

Pour la communication des nœuds entre eux, on doit les configurer le même identificateur de réseau ID et le même

Pour que les appareils communiquent, on doit les configurer avec le même identifiant réseau ID et le même canal d'exploitation CH. Pour que les appareils communiquent, les commandes CH et ID doivent être identiques sur tous les appareils du réseau.

Dans notre réseau nous avons défini (Fig. III. 17a, b et c):

CH = channel C.

ID = Network PAN ID 2020.

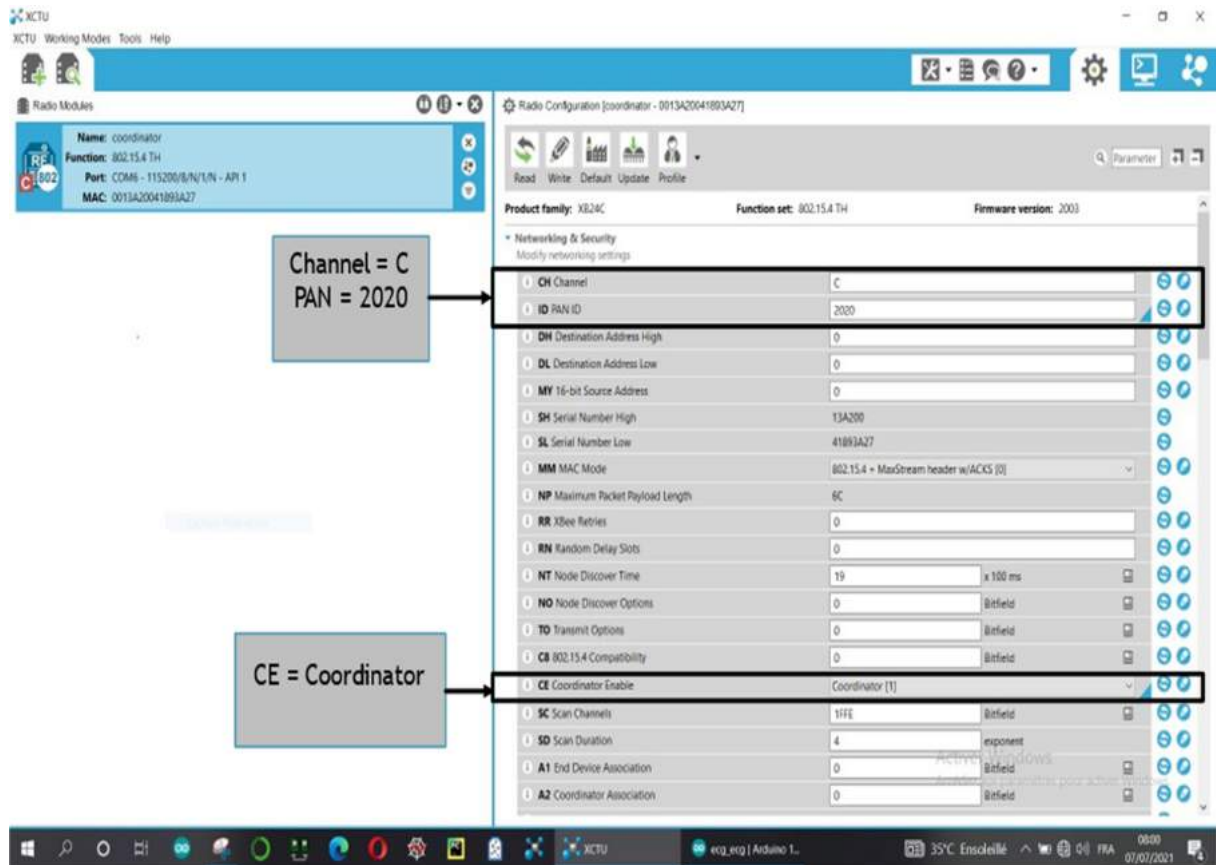


Figure III.17a : Photo de la configuration des XBee par XCTU.

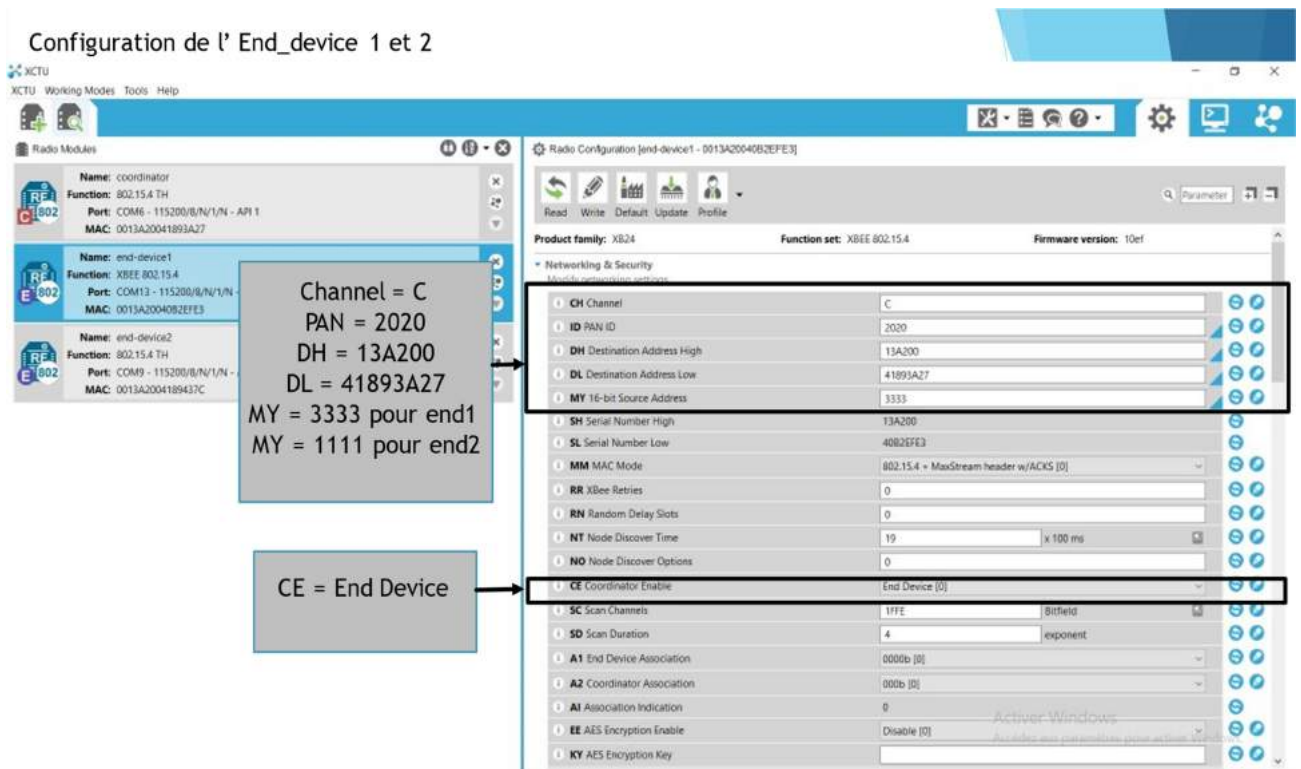


Figure III.17b : Photo de la configuration des XBee (End Device 1 et End Device 2) par XCTU.

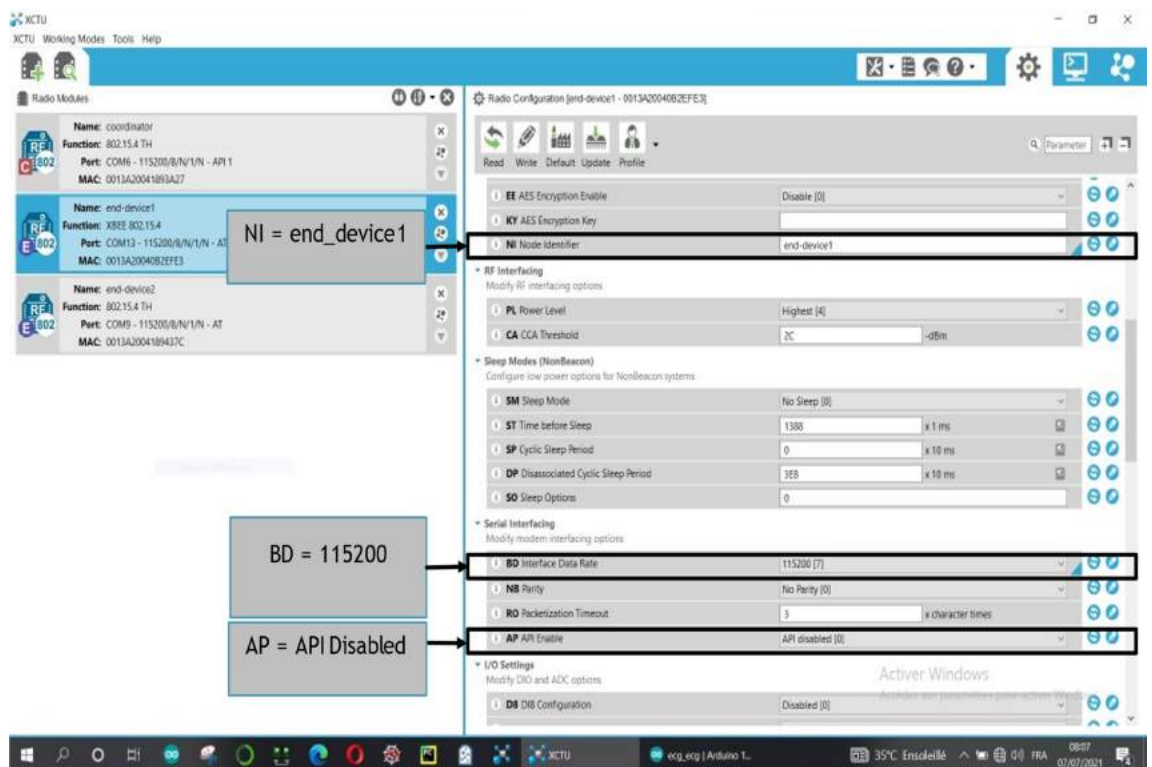


Figure III.17c : Photo de la configuration des XBee (End Device 1 et End Device 2) par XCTU (suite).

Les résultats obtenus sont donnés les photos des figures ci-dessous. Les résultats de l'émission sont donnés à la figure III.18a.

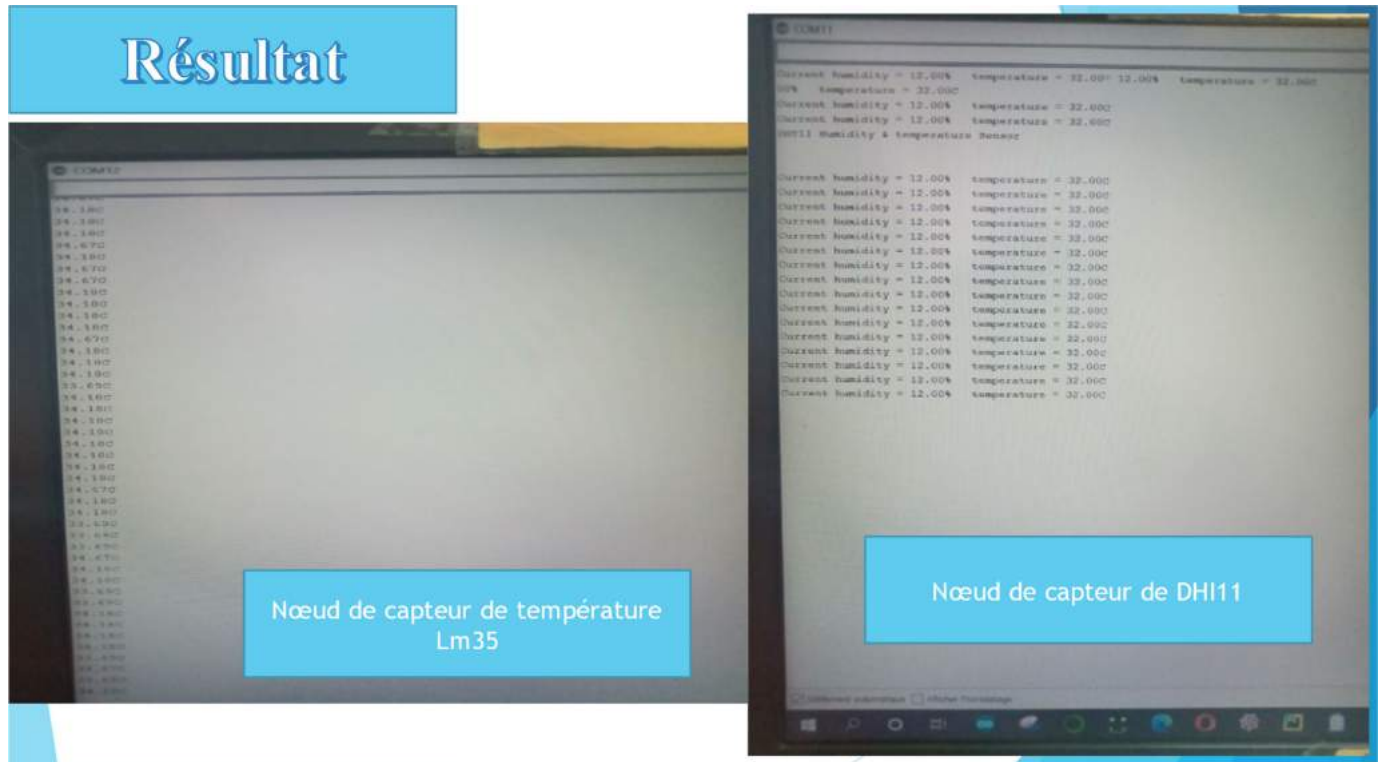
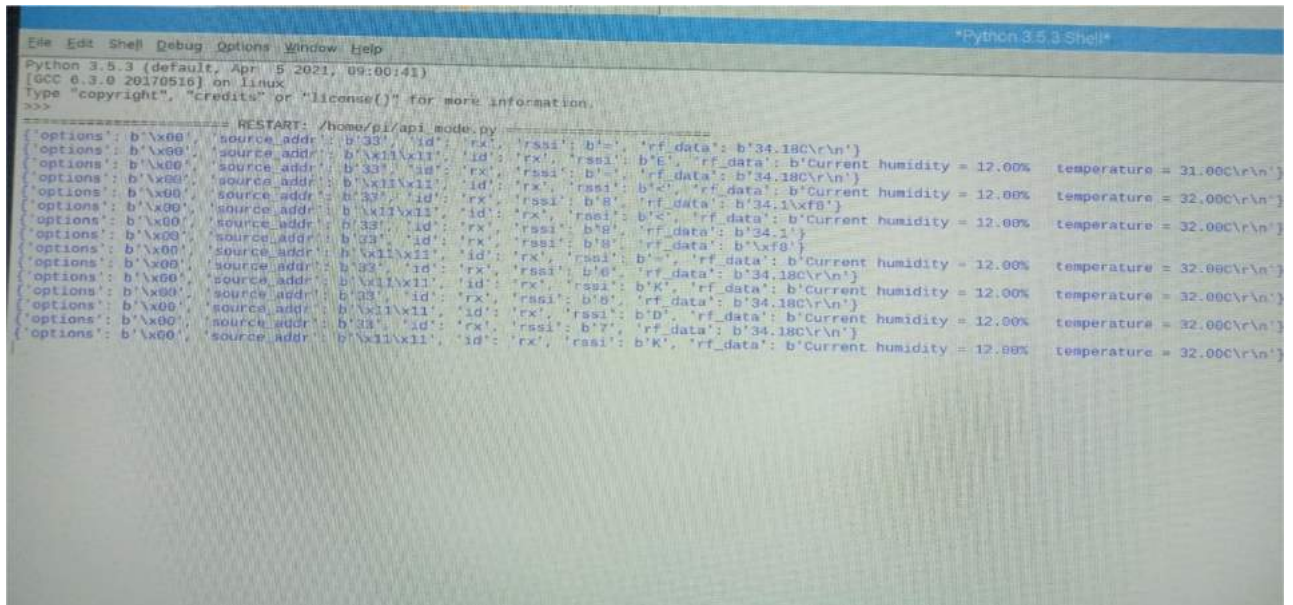


Figure III.18a : Photo des résultats émis par End Device 1 et End Device 2.

Au niveau de collecteur de réception (raspberry pi) on ajoute la bibliothèque (**XBeelib**) et la fonction **Xbee()** pour configurer la communication série (le numéro du port série, débit de transmission série). Cependant pour la réception des données on fait appel à la fonction **xbee.wait_read_frame()**, cette fonction n'accepte aucun paramètre, où elle est à l'écoute pour attendre d'éventuelle trames envoyées par les nœuds (Fig. III.18b).

Dans le cas où le Sink reçoit une trame, la fonction retourne chaque valeur des champs composant cette trame API, comme l'adresse source, la données utile, la valeur du RSSI. Ces valeurs peuvent être traitées séparément (Fig. III.18c).

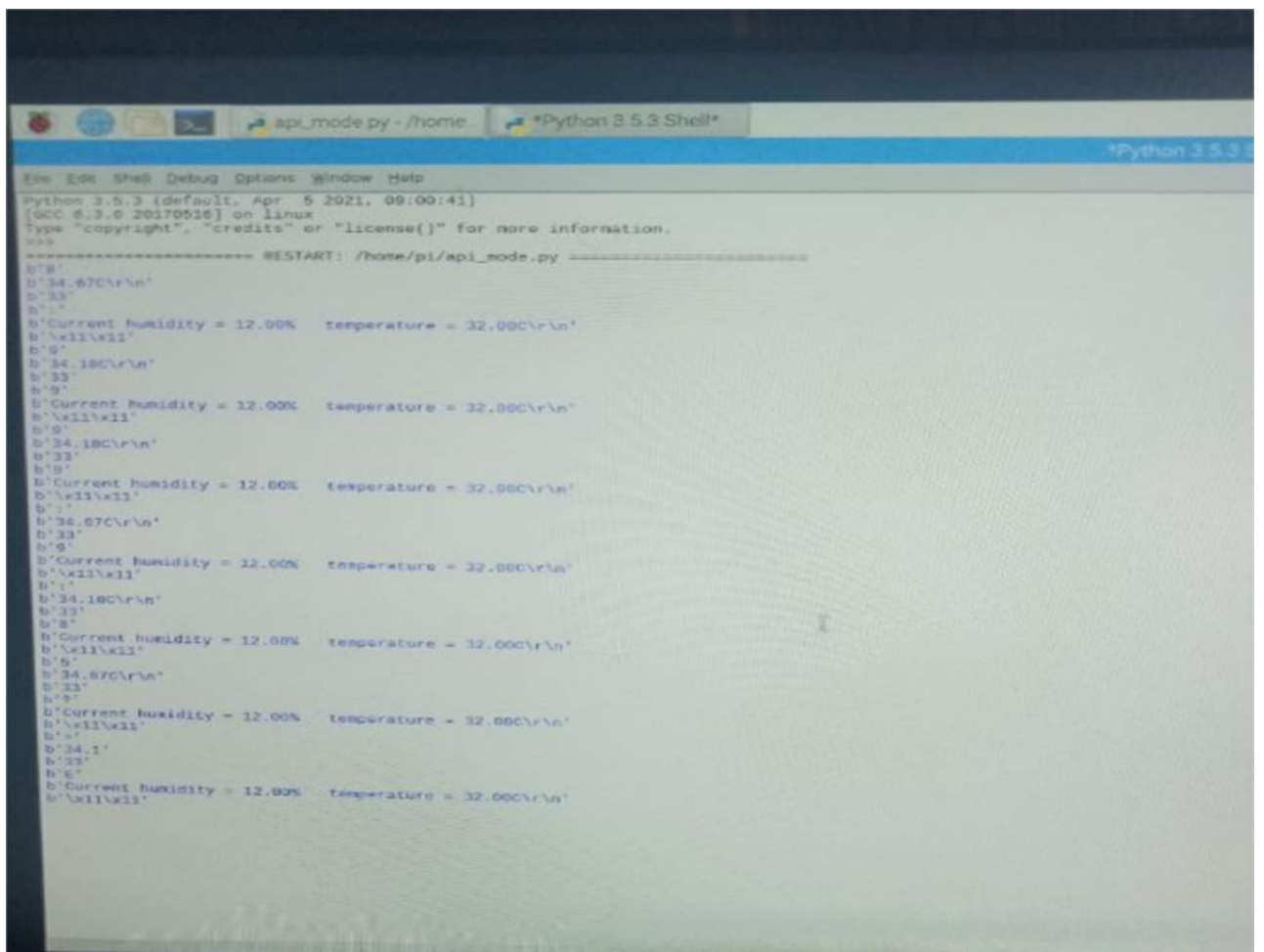


```

Python 3.5.3 (default, Apr 5 2021, 09:00:41)
[GCC 6.3.0 20170516] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/pi/api_mode.py =====
{"options": b'\x00', "source_addr": b'33', "id": "rx", "rssi": b'-', "rf_data": b'34.18C\r\n'}
{"options": b'\x00', "source_addr": b'\x11\x11', "id": "rx", "rssi": b'E', "rf_data": b'Current humidity = 12.00% temperature = 31.00C\r\n'}
{"options": b'\x00', "source_addr": b'33', "id": "rx", "rssi": b'E', "rf_data": b'34.18C\r\n'}
{"options": b'\x00', "source_addr": b'\x11\x11', "id": "rx", "rssi": b'E', "rf_data": b'Current humidity = 12.00% temperature = 32.00C\r\n'}
{"options": b'\x00', "source_addr": b'33', "id": "rx", "rssi": b'E', "rf_data": b'34.18C\r\n'}
{"options": b'\x00', "source_addr": b'\x11\x11', "id": "rx", "rssi": b'E', "rf_data": b'Current humidity = 12.00% temperature = 32.00C\r\n'}
{"options": b'\x00', "source_addr": b'33', "id": "rx", "rssi": b'E', "rf_data": b'34.18C\r\n'}
{"options": b'\x00', "source_addr": b'\x11\x11', "id": "rx", "rssi": b'E', "rf_data": b'Current humidity = 12.00% temperature = 32.00C\r\n'}
{"options": b'\x00', "source_addr": b'33', "id": "rx", "rssi": b'E', "rf_data": b'34.18C\r\n'}
{"options": b'\x00', "source_addr": b'\x11\x11', "id": "rx", "rssi": b'E', "rf_data": b'Current humidity = 12.00% temperature = 32.00C\r\n'}
{"options": b'\x00', "source_addr": b'33', "id": "rx", "rssi": b'E', "rf_data": b'34.18C\r\n'}
{"options": b'\x00', "source_addr": b'\x11\x11', "id": "rx", "rssi": b'E', "rf_data": b'Current humidity = 12.00% temperature = 32.00C\r\n'}
{"options": b'\x00', "source_addr": b'33', "id": "rx", "rssi": b'E', "rf_data": b'34.18C\r\n'}
{"options": b'\x00', "source_addr": b'\x11\x11', "id": "rx", "rssi": b'E', "rf_data": b'Current humidity = 12.00% temperature = 32.00C\r\n'}

```

Figure III.18b : Photo des résultats reçus par le collecteur (raspberry pi) pour la trame API.



```

Python 3.5.3 (default, Apr 5 2021, 09:00:41)
[GCC 6.3.0 20170516] on linux
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: /home/pi/api_mode.py =====
b'8'
b'34.67C\r\n'
b'33'
b'8'
b'Current humidity = 12.00% temperature = 32.00C\r\n'
b'\x11\x11'
b'8'
b'34.18C\r\n'
b'33'
b'8'
b'Current humidity = 12.00% temperature = 32.00C\r\n'
b'\x11\x11'
b'8'
b'34.18C\r\n'
b'33'
b'8'
b'Current humidity = 12.00% temperature = 32.00C\r\n'
b'\x11\x11'
b'8'
b'34.67C\r\n'
b'33'
b'8'
b'Current humidity = 12.00% temperature = 32.00C\r\n'
b'\x11\x11'
b'8'
b'34.18C\r\n'
b'33'
b'8'
b'Current humidity = 12.00% temperature = 32.00C\r\n'
b'\x11\x11'
b'8'
b'34.67C\r\n'
b'33'
b'8'
b'Current humidity = 12.00% temperature = 32.00C\r\n'
b'\x11\x11'
b'8'
b'34.18C\r\n'
b'33'
b'8'
b'Current humidity = 12.00% temperature = 32.00C\r\n'
b'\x11\x11'

```

Figure III.18c: Photo des résultats reçus par le collecteur (raspberry pi) pour la trame API.

III. 11 Conclusion

Les expérimentations effectuées au niveau du laboratoire ont confirmé la faisabilité, le fonctionnement et la fiabilité des données produites par ce système.

Conclusion générale

Conclusion générale

Ce projet nous a permis de réaliser un réseau de communication sans fil à base des modules Xbee pour transmettre des informations de plusieurs nœuds à distance et en temps réel. Le réseau ainsi réalisé comporte un point de réception et deux nœuds d'émissions. Le premier nœud d'émission est chargé de transmettre la donnée température en utilisant un capteur de température LM35, tandis que le second nœud transmette la température et l'humidité en utilisant le capteur DHT11. Les données analogiques issues des capteurs (LM35 et DHT11) subissent une conversion analogique-numérique en utilisant des microcontrôleurs du type Arduino Uno. Les données numériques obtenues sont transmises vers le point de réception via les modules Xbee utilisant le protocole ZigBee. Le point réception reçoit les données qui arrivent via le module Xbee de réception, qui transmette lui aussi les signaux numériques au microprocesseur Raspberry pi. Ce dernier permet la lecture et l'affichage de ces données analogiques (température et température + humidité) sur un écran de visualisation

Cette technologie, facile à mettre en place grâce à des modules comme le XBee que nous avons utilisé se montre très adaptée à des petits systèmes embarqués ne nécessitant pas de haut débit de transfert, tels que les appareils de contrôle et d'acquisition. Ce qui la rend encore plus adaptée à ce type de systèmes est sa faible consommation et la petite taille des modules. Elle est aussi bien adaptée à une utilisation en milieu industriel grâce à sa portée et son bon fonctionnement dans des environnements bruités.

On a noté une très bonne concordance entre les données de l'émission et celles collectées à la réception justifiant un bon fonctionnement de notre réseau de transmission sans fil.

Bibliographie

- [1] Walid Benlahcene : Un éclairage redondant, mémoire de fin d'étude pour d'ingénieur d'état en instrumentation, Université Batna, (2007).
- [2] Mr. DJAFRI MENAD, Mr. CHELOUCHE Djalal, Etude et Réalisation d'une Carte Arduino; mémoire de fin d'études pour l'obtention du diplôme de MASTER Université A. MIRA DE BEJAIA, 2016.
- [3] «Réseau de capteurs sans fil»
<http://www.technoscience.net/?onglet=glossaire&definition=11711>, Consulte le 10/02/2018
- [4] D. E. BOUBICHE "Protocole de routage pour les réseaux de capteurs sans fil", Pour l'obtention du Diplôme de Magister en Informatique, Université de l'Hadj Lakhdar-Batna, 2008.
- [5] C. Lamorinière, D. Ligot "Réseaux de capteur", EPS, European Project Semester, 2008.
- [6] Mlle Chaouche Hanane "Réalisation d'un prototype d'un réseau de capteurs sans fil dédié à l'agriculture de précision "Juin 2015" Université Abou Bakr Belkaid –Tlemcen
- [7] YACINE CHALLAL, "Réseaux de Capteurs Sans Fils" 04/05/2016,
Paternité - Pas d'Utilisation Commerciale :
<http://creativecommons.org/licenses/by-nc/2.0/fr/>
- [8] D. Martins, sécurité dans les réseaux de capteurs sans fil Sténographie et réseaux de confiance, Thèse de doctorat, Université de France-Compté, 2010.