الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR
ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA

FACULTE DE TECHNOLOGIE

Département du Génie Électrique



Mémoire de Fin d'Études

Présenté pour l'obtention du Diplôme
De Master en Génie Électrique

Option : *Systèmes de Télécommunications*

THEME

Conception et mise au point d'une plateforme pour réseau de capteurs sans fil Multiniveaux

Proposé et dirigé par :

Mr. CHADLI Samir

Dr.Bengharbia Billel

Préparé et présenté par :

Mr. MERADI Tarek

M^{elle}. MAMI Kaouther

Dédicace

Je dédie ce modeste travail avec un grand plaisir, à tous ceux qui ont crus en moi, spécialement à ceux qui ont été mes guides dans la vie :

A la plus belle femme du monde ma mère, qui a veillé sur moi jour et nuit depuis mon enfance qui ma donne l'amour la gentillesse le respect a moi-même c'est la reine de mon cœur.

A mon cher frère Racim et ma sœur Rawya

A ma chère binôme Kaouther

A tous mes enseignants depuis mon premier pas a l'école

jusqu'aujourd'hui, qui mon enrichie de leur savoir.

A mes copines et copines de ma promotion, Pour tous les bons moments partagés, pour tant de gentillesse et de disponibilité.

Grace aux quels ses années ont été ponctuées de moments d'évasions

Enfin, a tous ceux sont encrés dans mon cœur

ceux que je ne peux pas citer et que le manque de place ne permet pas les nommer ici.

Merci pour les moments grandioses passés et a venir.

MERADI Tarek

Dédicace

Je dédie ce modeste travail avec un grand plaisir, à tous ceux qui ont crus en moi, spécialement à ceux qui ont été mes guides dans la vie :

Mes chers parents qui m'ont entouré de leur amour, leurs sacrifices, leurs sacrifices, leur protection et de leur générosité durant toute la durée de mes études.

A mon binôme Tarek qui m'a toujours accepté tel que je suis, et m'a si souvent remonté le moral …. Ensemble pour toujours.

A tous mes enseignants depuis mon premier pas a l'école jusqu'aujourd'hui, qui mon enrichie de leur savoir.

A mes copines et copines de ma promotion, Pour tous les bons moments partagés, pour tant de gentillesse et de disponibilité.

Grace aux quels ses années ont été ponctuées de moments d'évasions

Enfin, a tous ceux sont encrés dans mon cœur ceux que je ne peux pas citer et que le manque de place ne permet pas les nommer ici.

Merci pour les moments grandioses passés et a venir.

MAMI kaouther

Remerciement

Tout d'abord, nous tenons à remercier Allah, le tout puissant qui nous avoir donné la force et le courage de mener à bien ce modeste travail.

En second lieu, nous tenons à remercier notre encadreur Dr : S.CHADLI de ses précieux conseils et son aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail Et de l'enrichir par leurs propositions.

Nous tenons à remercier nos chers amis pour les précieux conseils et ses aides.

Sans oublier nos parents pour leurs soutiens. Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Résumé

L'objectif de ce projet est de réaliser un réseau de capteur sans fil (RCSF) multi-niveaux à base des cartes Arduino et des modules radio nRF24L01+, permettant de transmettre les données vers un nœud de base (SINK) par voie RF en utilisant des nœuds intermédiaires (routeurs) qui assurent l'acheminement des données.

Les mots clés : WSN, Réseau Multi-niveaux, Topologie-Mesh.

Abstract

The objective of this project is to realize a multi-level wireless sensor network (WSN) based on Arduino boards and nRF24L01+ modules radio, to transmit data to a base node (SINK) over the RF channel using intermediaries nodes (routers) that ensure the routing of data.

Key words: WSN, Network Multi-level, Topology-Mesh.

ملخص

الهدف من هذا المشروع هو تحقيق شبكة مستشعرات لاسلكية متعددة المستويات قائمة على نظام

بطاقات أردوينو ووحدات راديو + nRF24L01 ، مما يسمح بنقل البيانات إلى عقدة أساسية عبر قنوات التردد الراديو باستخدام عقد الوسطاء (الهوجه) التي تضمن توجيه البيانات.

الكلمات المفتاحية

WSN , شبكة متعددة المستويات, بنية شبكية.

Sommaire

Introduction générale 1
I.1 Introduction 3
I.2 Présentation Réseaux de capteurs sans fil (RCSF) 3
I.2.1 Définition d'un RCSF ou WSN (Wireless Sensor Network) 3
I.2.2 Définition d'un capteur 4
I.2.2.1 Définition d'un nœud de capteurs sans fil : 5
I.2.2.2 Architecture d'un nœud de capteur:[4] 5
I.2.3 Caractéristiques des RCSF :[3] 6
I.2.4 Fonctionnement d'un réseau de capteurs :[1] 7
I.2.5 Domaines d'application des réseaux de capteurs sans fil :[5] 7
I.2.5.1 Les applications militaires 8
I.2.5.2 Les applications de surveillance 8
I.2.5.3 Les services de santé 9
I.2.5.4 Les applications environnementales 9
I.2.6 Catégories des réseaux sans fils :[3] 10
I.2.7 Topologies des réseaux de capteurs sans fil :[1] 10
I.2.7.1 Topologie en étoile 11
I.2.7.2 Topologie en toile : (Mesh Network) : 11
I.2.7.3 Topologie hybride : 11
I.3 La topologie retenue pour cette étude 11
I.4 Conclusion 12
II.1 Introduction 14
II.2 La partie Hardware 14
II.3 La partie de traitement 15
II.4 Conclusion 29
III.1 Introduction 31
III.2 La partie software 31
III.3 mise en marche de l'application 33
III.4 conclusion 53
Conclusion général 55

LISTE DES FIGURES

CHAPITRE I

Figure I.1 *Architecture d'un RCSF*.....3

Figure I.2 *Schéma de principe d'un capteur* 4

Figure I.3 *L'architecture d'un nœud de capteurs*.....5

Figure I.4 *Robot militaire commandé sans fil* 8

Figure I.5 *Exemple d'une application de surveillance d'une résidence*.....8

Figure I.6 *Exemple d'application médical*.....9

Figure I.7 *SENSORSCOPE pour la surveillance du réchauffement climatique*.....9

Figure I.8 *Topologies des réseaux de capteurs sans fil*.....10

CHAPITRE II

Figure II.1 *Schéma bloc du RCSF multi niveaux.*.....14

Figure II.2 *La carte Arduino nano*.....18

Figure II.3 *La carte Arduino Uno.*.....19

Figure II.4 *Le module Radio nRF24L01+.* 20

Figure II.5 *Schéma synoptique du mode de transmission nRF24L01+.*.....21

Figure II.6 *Mode de fonctionnement de bus SPI.*.....24

Figure II.7 *Transmission des données dans le bus SPI.* 24

Figure II.8 *L'interfaçage du nRF24L01+ avec Arduino Nano.* 25

Figure II.9 *La Shield I/O, Expansion Board, Wireless V3 pour l'Arduino Nano*.....25

Figure II.10 *L'interfaçage du nRF24 avec la carte Arduino Uno.*.....26

Figure II.11 *Le schéma électrique de la shield Wireless pour la carte Uno* 27

Figure II.12 *Le modèle en couches du réseau de capteurs sans fil.* 27

CHAPITRE III

Figure III.1 *L'IDE Arduino*.....31

Figure III.2 *L'installation et la mise à jour des bibliothèques relatives au nRF24*.....32

Figure III.3 *Téléchargement du soft de programmation de nRF24L01+* 32

Figure III.4 *Choix du bootloader du processeur* 33

Figure III.5 *Choix du bootloader du processeur* 33

Figure III.6 *La mise en marche du mode multi cievers* 34

Figure III.8 *La fenêtre de débogage relative au nœud de base.* 35

Figure III.7 *La fenêtre de débogage du nœud de capteur n°5* 36

Figure III.9 *Représentation des données reçus.*.....36

Figure III.10 *L'organigramme du programme du nœud de base* 38
Figure III.11 *L'organigramme du nœud d'émetteur* 40
Figure III.12 *La partie de déclaration au niveau de nœud end_device.* 41
Figure III.13 *Communication RCSF multiniveaux.* 42
Figure III.14 *La topologie RF24Network.* 44
Figure III.15 *La mise en marche du mode multiniveaux* 45
Figure III.16 *La fenêtre de débogage du nœud d'end_device.* 46
Figure III.17 *La fenêtre de débogage relative au nœud de base.* 47
Figure III.18 *La fenêtre de débogage relative au nœud de routeur.* 47
Figure III.19 *L'organigramme du nœud de sink* 48
Figure III.20 *Déclaration des différentes bibliothèques* 49
Figure III.21 *Code d'arduino utiliser au niveau du sink* 49
Figure III.22 *L'organigramme du nœud de routeur* 50
Figure III.23 *Partie de retransmission.* 51
Figure III.24 *L'organigramme du nœud d'end_device* 52
Figure III.25 *Code d'arduino utilisé au niveau d'end_device* 53

LISTE DES TABLEAUX

CHAPITRE I

Tableau I.1 Comparaison entre réseau de capteurs et réseau Ad-Hoc[1]……..……5

CHAPITRE II

Tableau II.1 Tableau comparatif des différents modèles de cartes Arduino. 17

Tableau II.2 Caractéristiques de la carte Arduino nano.[8].....18

Tableau II.3 Caractéristiques de la carte Arduino Uno.[10] 19

Tableau II.4 Comparaison des standards.[4] 20

Tableau II.5 Caractéristiques du module nRF24L01.[11].....21

Tableau II.6 La structure de la Trame du protocole ShockBurst.[4].....21

Tableau II.7 Description des pins du nRF24L01.....22

Tableau II.8 Exemple d'échange de l'adresse entre deux module 22

CHAPITRE III

Tableau III.1 Comparaison entre le nombre théorique et pratique des échantillons reçus….37

Liste des acronymes

symboles	signification
ACK	Acknowledgement
CE	Chip Enable
CSN	Chip Select
CRC	Cyclic Redundancy Code
ESB	Enhanced shock Burst
FIFO	First in First out
GFSK	Gaussian frequency shift keying
ICSP	In-Circuit Serial Programming ou (Programmation Série Dans le circuit)
IDE	integrated development environment
IRQ	Interrupt Reques
LED	light emitting diode
MISO	Master Input, Slave Output
MOSI	Master Output, Slave Input
OSI	Open systems Interconnection
PWM	Pulse width modulation (Modulation de la largeur des pics)
RCSF	Réseau de Capteur Sans Fil
RF	Radio Frequency
RX	Receiver
SCK	Serial Clock
SPI	Serial Peripheral Interface
SSH	secured shell
TX	Transmitter
USB	Universal serial Bus
WSN	Wireless Sensor Network

INTRODUCTION GÉNÉRALE

Introduction générale

Dans le but de faciliter la vie et de faire éliminer les difficultés autour de l'être humain, plusieurs technologies sont apparues pour garantir ces besoins, l'une de ces technologies est les réseaux de capteurs sans fil.

Les réseaux de capteurs sans fil représentent une révolution technologique des instruments de mesure, issus de la convergence des systèmes, électroniques miniaturisés et des systèmes de communication sans fil.

Les réseaux de capteurs sans fil (RCSF) - Wireless Sensor Networks (WSN) - sont des réseaux généralement constitués d'entités autonomes miniaturisées appelés nœuds capteurs pouvant communiquer entre eux par liaison radio.

Les nœuds sont généralement de taille réduite, construits à partir des composants pas chers. Ce type de réseau est composé de centaines ou de milliers d'éléments (capteurs), a pour but la collecte de données de l'environnement, leur traitement et leur transmission vers le monde extérieur

L'objectif de ce mémoire est de concevoir et d'implémenté une plateforme pour un réseau de capteur sans fil multiniveaux. Ce travail retenu, assure en plus la possibilité d'étendre le réseau de capteurs à un nombre de nœuds plus important, assure une portée plus grande de la zone de captage qui dépasse la portée du module radio utilisé à travers l'emploi des nœuds routeurs. Notre contribution s'articule autour de trois parties, la première consiste à transmettre les données (data) à l'aide d'un module de communication nRF24L01+ vers un destinataire nœud de sink lors de sa porté chaque nœud a une adresse entre 1-255 , mais pour qu'il fait la transmission il utilise des intermédiaires plusieurs nœuds routeurs (passerelles) qui assurent l'acheminement des données des nœuds de capteurs vers le nœud de base sink. Ce dernier reçoit les data et les affiche sur un pc.

Ce travail est organisé autour de trois chapitres :

- Dans le premier chapitre, nous présenterons quelques généralités sur les réseaux de capteurs sans fil.
- le deuxième chapitre traite la partie matérielle et les différents composants utilisés dans cette étude.
- Le troisième chapitre est concernera le détaille et la mise en marche de notre application, nous présenterons en premier la configuration multicievers et en seconde la configuration multi niveaux, ainsi que leurs résultats obtenu.

CHAPITRE I

GÉNÉRALITÉS SUR LES RÉSEAUX DE CAPTEURS SANS FIL

I.1 Introduction

Durant la dernière décennie, l'utilisation des réseaux de capteurs sans fil (RCSF) a connu une évolution considérable dans diverses applications de la vie courante. Ainsi on les trouve de plus en plus dans les systèmes de supervision, de détection, de mesure et de sécurité.

En conséquence, les réseaux de capteurs sans fil peuvent renseigner l'utilisateur sur l'état d'un système, d'un lieu ou d'un environnement à distance à travers un ensemble de capteurs reliés entre eux formant ainsi un réseau sans fil de mesure ou de détection en se basant sur des techniques et des protocoles de communication.

I.2 Présentation Réseaux de capteurs sans fil (RCSF)

I.2.1 Définition d'un RCSF ou WSN (Wireless Sensor Network)

Un réseau de capteurs sans fil est un type spécial des réseaux Ad Hoc avec un grand nombre de nœuds qui sont le plus souvent des nœuds de capteurs (End_device) capables d'effectuer des mesures à travers des capteurs, et de recevoir et de transmettre ces données environnementales d'une manière autonome sans intervention humaine[1].

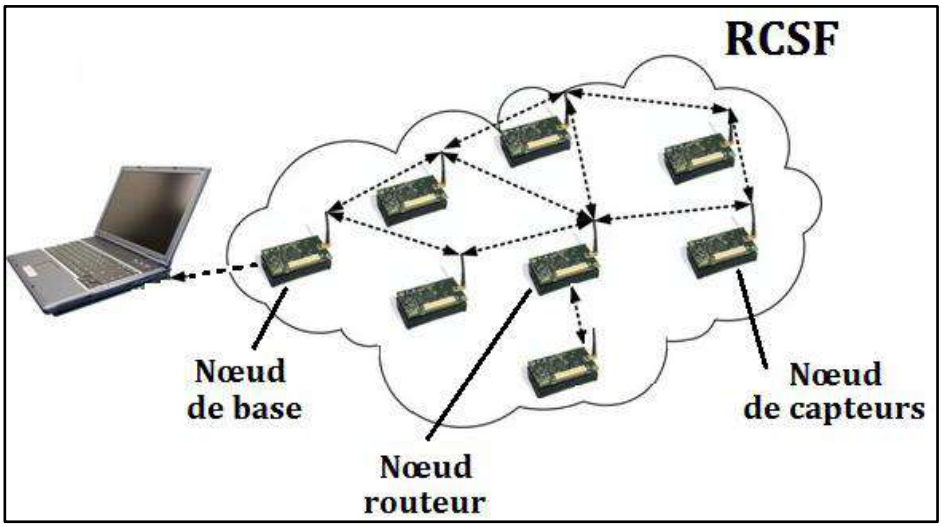


Figure I.1 *Architecture d'un RCSF*

En général, le réseau RCSF possède un nœud particulier qui est le nœud de base ou puits (sink), dont le rôle est la collecte des données des divers nœuds de capteurs du réseau. En trouve souvent un ou plusieurs nœuds routeurs (passerelles) qui assurent l'acheminement des données des nœuds de capteurs vers le nœud de base (Fig. I.1).

La position des nœuds d'un RCSF n'est pas obligatoirement prédéterminée, ils peuvent être aléatoirement dispersés dans une zone géographique appelée « champ de captage » correspondant au terrain d'intérêt pour le phénomène capté (par exemple : pour étudier les phénomènes vulcanologiques et leurs évolutions, un ensemble de nœuds de capteurs sont lâchés par un avion autour d'un volcan).

I.2.2 Définition d'un capteur

Un capteur « Sensor en anglais » est un dispositif qui soumis à l'action d'un mesurande non électrique (grandeur physique) présente en sortie une caractéristique de nature électrique (charge, tension, courant, …) qui est fonction de la grandeur physique d'entrée (Fig. I.2) [2]. Ainsi les capteurs sont capables d'effectués des mesures sur des environnements divers, pour quantifier des grandeurs de nature varié comme la température, la vibration, la pression, …etc.

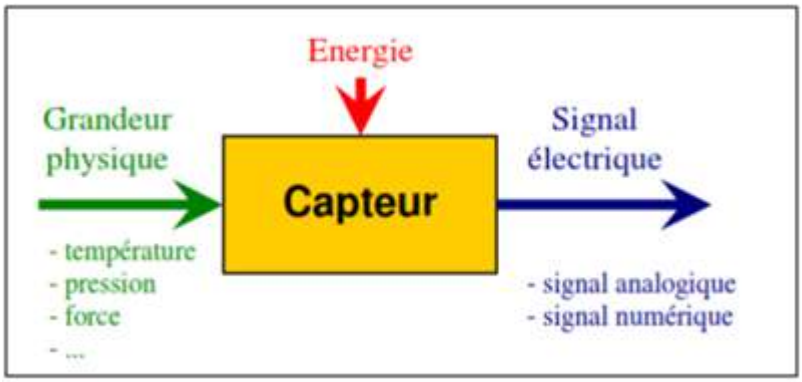


Figure I.2 *Schéma de principe d'un capteur*

Les capteurs modernes (intelligents) assurent en plus de la quantification de la grandeur physique, le traitement et la communication de l'information vers un autre système électronique [2].

Le tableau I.1 présente une petite comparaison entre les réseaux de capteurs et les réseaux Ad-Hoc :

Tableau I.1 *Comparaison entre réseau de capteurs et réseau Ad-Hoc*[1]

Réseau de capteurs	Réseau Ad-Hoc
Objectif spécifique	Générique (communication)
Nœuds en collaboration pour remplir un objectif	Chaque nœud à son propre objectif
Flot de données (plusieurs vers un nœud)	Flot (any to any)
Un grand nombre de nœuds n'ayant pas tous un ID	Notion d'ID
L'énergie (facteur déterminant)	Le débit (facteur déterminant)
Utilisation du broadcast	Communication point à point

I.2.2.1 Définition d'un nœud de capteurs sans fil :

Les nœuds de capteurs sans fil sont des dispositifs à base de capteurs associés à un circuit électronique de taille extrêmement réduite avec des ressources très limitées, autonomes, capable de traiter des informations et de les transmettre, via les ondes radio, à une autre entité (nœud de capteurs, routeur, coordinateur……) sur une distance limitée[3].

I.2.2.2 Architecture d'un nœud de capteur:[4]

Un nœud capteur est composé de plusieurs éléments ou modules correspondant chacun à une tâche particulière d'acquisition, de traitement, ou de transmission de données. Il comprend également une source d'énergie.

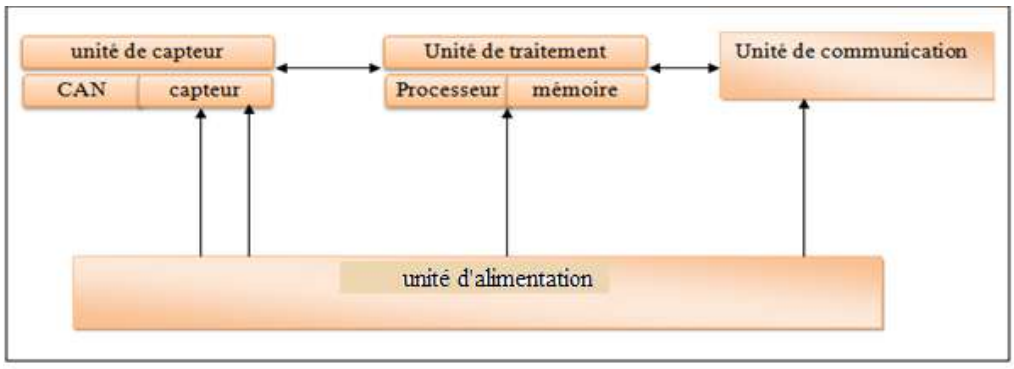


Figure I.3 *L'architecture d'un nœud de capteurs*

- **Unité de capture ou d'acquisition (Sensing unit)**

Est généralement composée de deux sous-unités : les capteurs et les convertisseurs analogique-numérique (CANs). Les capteurs obtiennent des mesures numériques sur les paramètres environnementaux et les transforment en signaux analogiques. Les CANs convertissent ces signaux analogiques en signaux numériques.

- **Unité de traitement (Processing unit)**

Les données captées sont communiquées au processeur où elles sont stockées dans la mémoire. Cette unité est composée de deux interfaces, une interface avec l'unité d'acquisition et une autre avec le module de transmission. Elle contrôle les procédures permettant au nœud de collaborer avec les autres nœuds pour réaliser les tâches d'acquisition, et stocker les données collectées.

- **Unité de communication (Transceiver unit)**

Elle est composée d'un émetteur/récepteur (module radio), elle est responsable de toutes les communications via un support de communication radio qui relie le nœud au réseau.

- **Unité d'énergie (Power unit)**

En générale elle est base d'une batterie qui est souvent ni rechargeable ni remplaçable, elle permet d'alimentée les trois autres unités.

I.2.3 Caractéristiques des RCSF :[3]

- **Absence d'infrastructure :** les réseaux Ad Hoc en général, se distinguent des autres réseaux sans fil par la propriété d'absence d'infrastructure préexistant et de tout genre d'administration centralisée. Les nœuds du réseau sont responsables d'établir et de maintenir la connectivité du réseau d'une manière continue.

- **Un grand nombre de nœuds :** des réseaux de 10000 nœuds peuvent être envisagés dans certaines applications.

- **Contrainte d'énergie :** dans plusieurs applications, les nœuds de capteurs sont placés dans des surfaces distantes, dans ce cas la durée de vie du nœud est déterminée par la durée de vie de la batterie, ce qui exige la minimisation des dépenses énergétiques.

- **Topologie dynamique :** les capteurs peuvent être attachés à des objets mobiles qui se déplacent d'une façon libre et arbitraire rendant ainsi, la topologie du réseau fréquemment changeante.

- **Auto organisation du réseau :** ceci peut être nécessaire dans plusieurs cas. Par exemple, un réseau comportant un grand nombre de nœuds placés dans des endroits hostiles ou la configuration manuelle n'est pas faisable, doit être capable de s'auto organiser. Un autre cas est celui ou un nœud est inséré ou retiré (à cause d'un manque d'énergie ou de destruction physique), ainsi le réseau doit être capable de se reconfigurer pour continuer sa fonction.

- **Sécurité physique limitée :** les RCSF sont le plus souvent touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes hardware du nœud et les contraintes physiques de transfert des données par voie RF.

I.2.4 Fonctionnement d'un réseau de capteurs :[1]

Les données captées par les nœuds sont acheminées directement ou grâce à un routage à un nœud au nœud-puits (sink) qui est considéré comme un "point de collecte". Ce dernier peut être connecté à l'utilisateur du réseau via Internet, un satellite ou un autre système. L'usager peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises pour récolter les données environnementales captées par le biais du nœud de capteurs.

Il faut noter que les progrès conjoints de la microélectronique, du micro-usinage, des technologies de transmission sans fil et des applications logicielles ont permis de produire à coût raisonnable des micro-capteurs de quelques millimètres cubes de volume, susceptibles de fonctionner en réseaux. Ces derniers intègrent : une unité de captage chargée de capter des grandeurs physiques (température, humidité, pression, vibrations, rayonnement...), un dispositif transformant ces grandeurs analogiques en grandeurs numériques, une unité de traitement informatique et de stockage des données et un module de transmission sans fil.

I.2.5 Domaines d'application des réseaux de capteurs sans fil :[5]

Les RCSF offrent une facilité de déploiement qui présente un avantage considérable pour utiliser ce type de technologie dans différents domaines. En plus, les réseaux de capteurs sans fil peuvent être constitués de différents types de capteurs tels que thermique, infrarouge, acoustique, magnétique, optique et sismiques, pour surveiller une large variété de paramètres physiques**.**

I.2.5.1 Les applications militaires

Les RCSF sont susceptibles d'être une partie intégrante des équipements du soldat de future, du dispositif de commandement militaire, des outils de la surveillance du champ de bataille (Fig. I.4), des techniques de la reconnaissance et des systèmes de localisation**.**



Figure I.4 *Robot militaire commandé sans fil*

I.2.5.2 Les applications de surveillance

Dans la surveillance des lieux ou des résidences (Fig. I.5), les nœuds de capteurs sont déployés sur pour détecter divers types de phénomènes. Lorsque les capteurs détectent l'événement surveillé (incendie, fuite de gaz, intrusion, inondation, …etc.), une information sera envoyée à travers le réseau de capteurs vers le propriétaire du lieu**.**



Figure I.5 *Exemple d'une application de surveillance d'une résidence*

I.2.5.3 Les services de santé

Parmi les applications dans le domaine de santé, les réseaux de capteurs soutiennent des interfaces pour les personnes handicapées, la surveillance intégrée des patients, les diagnostics et l'administration de médicaments dans les hôpitaux, la télésurveillance des données physiologiques de l'homme, le suivi et la surveillance des médecins ou des patients à l'intérieur d'un hôpital.

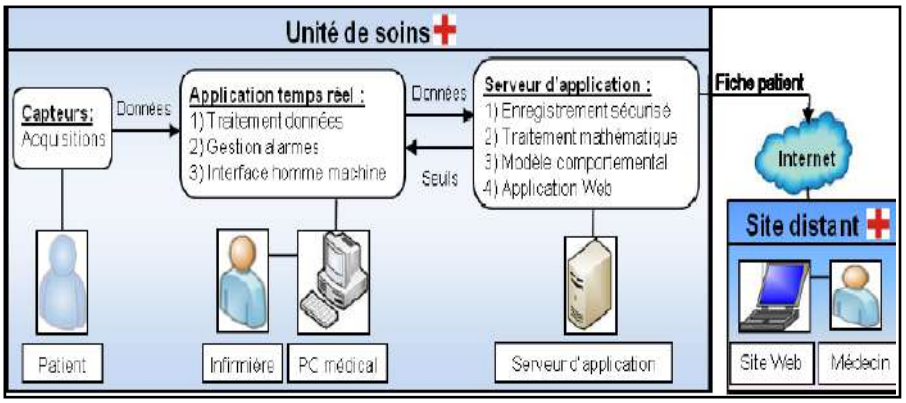


Figure I.6 *Exemple d'application médical*

I.2.5.4 Les applications environnementales

Ces types de RCSF peuvent servir à l'observation d'un site susceptible de subir les effets d'une pollution, ou bien d'éventuel incendie, inondation, volcan ou tsunami. La construction en temps réel d'une cartographie grâce à des capteurs, disséminés sur le site capables de relever des informations sur la pollution, la température, le niveau d'eau et d'oxygène…, etc. Ces informations doivent être relayées vers les services spécialisés.



Figure I.7 *SENSORSCOPE pour la surveillance du réchauffement climatique*

I.2.6 Catégories des réseaux sans fils :[3]

On distingue actuellement plusieurs catégories de réseaux sans fils, selon le périmètre géographique offrant une connectivité (appelé zone de couverture) :

a) Réseaux personnels sans fils (WPAN).

b) Réseaux locaux sans fils (WLAN).

c) Réseaux métropolitains sans fils (WMAN).

d) Réseaux étendus sans fils (WWAN).

I.2.7 Topologies des réseaux de capteurs sans fil :[1]

Un RCSF est composé d'un ensemble de nœuds capteurs et de nœuds routeurs(Gateway) qui s'occupent de collecter les données des capteurs et de les transmettre à l'utilisateur par le bais du nœud de base via internet ou via les réseaux de communications. Ces réseaux de capteurs sans fil peuvent avoir plusieurs topologies, la Fig. 1.8 présente l'aspect des trois topologies les plus utilisées :

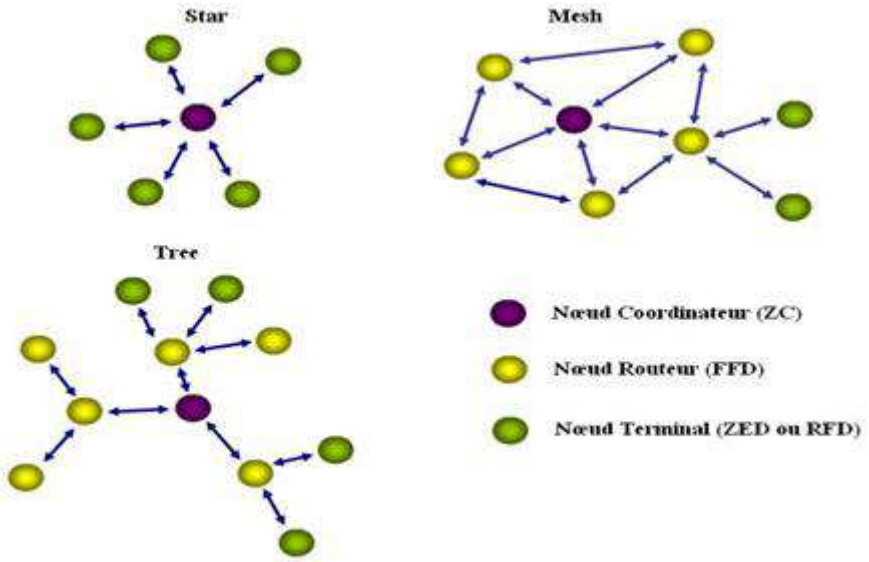


Figure I.8 *Topologies des réseaux de capteurs sans fil*

I.2.7.1 Topologie en étoile

La topologie en étoile est un système uni-saut. Tous les nœuds envoient et reçoivent seulement des données avec la station de base. Cette topologie est simple et elle demande une faible consommation d'énergie, mais la station de base est souvent altérable et la distance entre les nœuds et la station est limitée.

Avantage : simplicité et faible consommation d'énergie des nœuds, faible latence de communication entre les nœuds et la station de base.

Inconvénient : la station de base est vulnérable, car tout le réseau est géré par un seul nœud.

I.2.7.2 Topologie en toile : (Mesh Network) :

La topologie en toile (maillé) est un système multi-saut. La communication entre les nœuds et la station de base est possible. Chaque nœud a plusieurs chemins pour envoyer les données. Cette topologie a plus de possibilités de passer à l'échelle du réseau, avec redondance et tolérance aux fautes, mais elle demande une consommation d'énergie plus importante.

Avantage : Possibilité de passer à l'échelle du réseau, avec redondance et tolérance aux fautes.

Inconvénient: Une consommation d'énergie plus importante induite par la communication multi-sauts. Une latence est créée par le passage des messages des nœuds par plusieurs autres nœuds avant d'arriver à la station de base.

I.2.7.3 Topologie hybride :

La topologie hybride est un mélange des deux topologies précédentes. Les stations de base forment une topologie en toile et les nœuds autour d'elles sont en topologie étoile. Elle assure la minimisation de la consommation de l'énergie dans les réseaux de capteurs.

I.3 La topologie retenue pour cette étude

Dans ce travail, nous nous proposons de concevoir et d'implémenté une plateforme hardware et software faible cout d'un RCSF à topologie hybride (en tree). La structure multi niveaux retenue, assure en plus la possibilité d'étendre le réseau de capteurs à un nombre de nœuds plus important, assure une portée plus grande de la zone de captage qui dépasse la portée du module radio utilisé à travers l'emploi des nœuds routeurs et garantis la possibilité

d'organiser les différents nœuds du RCSF en sous-ensembles (sous-réseaux) selon les choix désirés (chemins des données, types de données, zone de déploiement, …etc).

Cette plateforme qui peut être destinée à divers types d'applications dans les domaines : du sport, de la santé, de la mesure, de la surveillance, …etc, est architecturée autour d'un microcontrôleur 8 bits et d'un module radio de faibles couts.

I.4 Conclusion

Dans ce chapitre, nous avons présenté des généralités sur les réseaux de capteurs sans fil(RCSF), la constitution du nœud de capteurs et son architecture. Nous avons également cité les caractéristiques d'un réseau de capteurs sans filet nous avons présenté quelques applications des RCSF dans différents domaines de la vie, ainsi que leurs topologies.

CHAPITRE II

CONCEPTION

DU RÉSEAU DE CAPTEUR SANS FIL

II.1 Introduction

Dans ce chapitre nous présentons la partie hardware du réseau de capteur sans fil implémenté, ainsi que les divers choix retenus pour la conception des différents nœuds et principalement la partie de traitement à base de microcontrôleur et le module radio.

II.2 La partie Hardware

Schéma bloc du système :

La Fig. II.1 présente le schéma bloc du réseau de capteur sans fil multiniveaux qu'on amis au point, les blocs qui sont au-dessous servent à collecter les informations (température, humidité, pression, présence des intrus…etc.).

Les blocs qui sont au milieu sert à recevoir les data (données) via le module radio et les renvoyer vers un autre point en réception afin de les stockées ou les traitées.

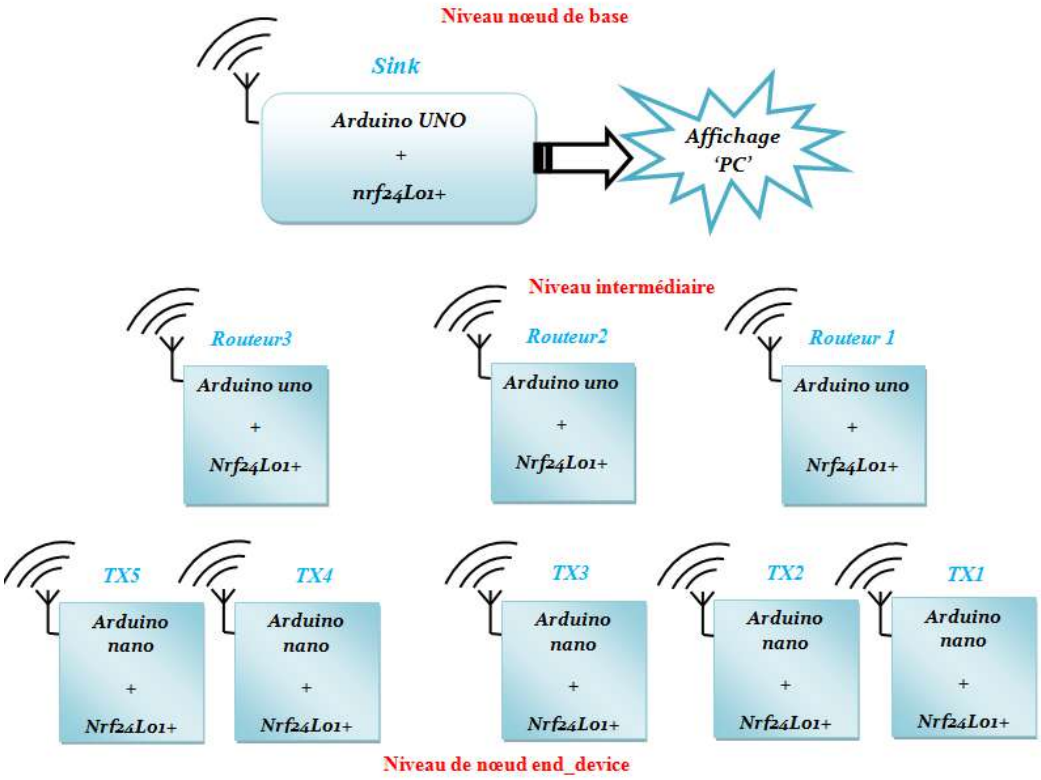


Figure II.1 *Schéma bloc du RCSF multiniveaux.*

II.3 La partie de traitement

Pour notre RCSF, nous avons choisi l'une des plateformes de développent à microcontrôleur 8 bits les plus populaires qui est Arduino. Cette dernière présente un outil open source, destinés pour les amateurs et les professionnels pour créer des systèmes électroniques plus ou moins complexes.

C'est une plateforme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation.

a. Applications : [6]

La plateforme de prototypage Arduino permet aux utilisateurs de réaliser une grande variété d'applications telles que :

- Le contrôle des appareils domestiques,
- L'acquisition des signaux à partir de capteurs,
- La commande de moteurs et des robots,
- La réalisation des systèmes d'affichages,
- La communication avec l'ordinateur, GSM, … ;
- Le contrôle d'un appareil mobile (modélisme), etc…

Dans le commerce, il existe une multitude de plateformes de développent à microcontrôleurs qui permettent de faire la même chose. Notamment celles à base des microcontrôleurs « PIC » du fabricant Micro chip ou celles à base des microcontrôleurs MSP430 du fabricant Texas Instrument. Le choix de la plateforme Arduino a été réalisé pour plusieurs raisons, parmi lesquelles nous citons:

- Multiplateforme : le logiciel Arduino, écrit en Java, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. Certains systèmes à microcontrôleurs sont limités uniquement à Windows.
- Un environnement de programmation clair et simple facile à utiliser pour les débutants.
- Logiciel Open Source et extensible : l'IDE Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmateurs expérimentés.

- Matériel Open source et extensible: les cartes Arduino sont basées sur des microcontrôleurs 8 et 32 bits de plusieurs constructeurs (Atmel, Intel, ….).
- Une communauté très active : la communauté Arduino est impressionnante et le nombre de ressources à son sujet est en constante évolution sur internet.

b. Les outils de développement Arduino :[6]

Cette plateforme est composée de deux types d'outils, qui sont le matériel et le logiciel.

b.1 Le matériel :

La carte Arduino est architecturée autour d'un circuit de traitement et de commande de type microcontrôleur associé à des entrées et des sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes : en entrées des *capteurs* qui collectent des informations sur leur environnement et en sorties : des *actionneurs* qui agissent sur le monde physique.

La carte Arduino est généralement constituée des éléments suivants :

- **Le microcontrôleur :**

C'est l'élément de base de la carte de prototypage. C'est l'organe qui va recevoir et exécute le programme développé par l'utilisateur et qui va être stocké dans sa mémoire.

- **L'alimentation :**

C'est le circuit électronique qui assure la mise en marche de la carte de prototypage et fourni l'énergie électrique nécessaire au fonctionnement du microcontrôleur et ses périphériques.

- **L'interface de communication :**

C'est la partie qui assure la communication entre la carte de prototypage et l'outil de programmation (le plus souvent un PC), elle peut être de nature filaire (via un câble USB ou RJ45) ou de nature radio (via une Interface Wifi ou Bluetooth).Cette interface est utilisée pour la programmation, le débogage, …etc.

- **Le bloc de visualisation à Leds:**

Un ensemble de trois à quatre LEDs qu'on trouve souvent sur la carte, ces LEDs servent :

1. La LED (L) : elle est connectée à une broche du microcontrôleur (le plus souvent la ligne numérique 13) et qui va servir pour tester le matériel.

2. Les deux LED Tx et Rx: servent à visualiser l'activité sur la voie série (la première pour l'émission et l'autre pour la réception).
3. La LED Pwr / ON : pour visualiser l'état du circuit d'alimentation de la carte.

b.2 Les différents modèles de cartes de développements Arduino :[3]

Actuellement, il existe une grande variété de cartes de prototypage Arduino, le tableau suivant présente les caractéristiques de quelques modèles de cartes :

Tableau II.1 *Tableau comparatif des différents modèles de cartes Arduino.*

Modèle	Processeur	Fréq CPU Mhz	Mémoire SRAM Kb	Mémoire FLASH Kb	Mémoire EEPROM Kb	Entrées Analog	E/S mé et PWM	Photos
UNO	ATmega 328	16	2	32	1	6	14/6	
LEONARDO	ATmega 32u4	16	2.5	32	1	12	20/7	
DUEMILA NOVE	ATmega 328	16	2	32	1	6	14/6	
NANO	ATmega 328	16	2	32	1	8	14/6	
MEGA2560	ATmega 2560	16	8	256	4	16	54/14	
DUE	AT91SA M3X8E	84	96	512	-	12	54/12	
MEGA ADK	ATmega 2560	16	8	256	4	16	54/14	
YUN	ATmega 32u4	16	2.5	32	1	12	20/7	

Pour notre cas, nous avons choisi les cartes Arduino Nano et Uno afin de les utilisées dans notre projet car elles sont adaptées à nos besoins, notamment pour : leurs prix, leurs dimensions, leurs nombre d'entrées/sorties et leurs tailles de mémoire programme et RAM.

Nous avons utilisé le modèle Uno comme nœud de base (sink) afin de recevoir et afficher les données collectées ainsi que pour les nœuds routeurs qui servent comme passerelle à l'intérieur du réseau.

Pour les nœuds de capteurs, nous avons utilisé le modèle Nano pour réaliser l'acquisition des données via les capteurs ou détecteurs.

b.3 La carte Arduino Nano :[3]

L'Arduino Nano (Fig. II.2) est une version qui ressemble à la carte Arduino mini (qui est la carte la plus petite de la famille Arduino) avec en plus une connexion USB intégré, la carte Nano a plus de broches d'entrées analogiques comparé à la carte Arduino mini.



Figure II.2 *La carte Arduino nano*

Le tableau ci-dessous résume les caractéristiques principales de ce modèle de carte :

Tableau II.2 *Caractéristiques de la carte Arduino nano.*[7]

Microcontrôleur	ATmega328
Vitesse de l'horloge	16MHz
Tension de fonctionnement	5V
Tension d'entrée (recommandé)	7-12V
Digital I/O Pins	22 (dont 6 fournissent une sortie PWM)
Broches d'entrées analogiques	8
Courant max par pin d'E/S	40 mA
Courant max pour la pin 3,3 V	50 mA
Mémoire Flash	32 Ko (ATmega328) dont 2 Kb utilisé par le Bootloader
Mémoire SRAM	2 Kb
Mémoire EEPROM	1 Kb

b.4 La carte Arduino Uno :[8]

La carte Uno est la version de base des cartes Arduino. Elle est conçue autour du microcontrôleur ATmega328P. Elle est dotée :

- De 14 entrées/sortie (dont 6 fournissent une sortie PWM)
- De 6 entrées analogiques
- D'un cristal à 16 MHz
- D'une connexion USB
- D'une prise jack d'alimentation
- D'une interface de programmation ICSP ;

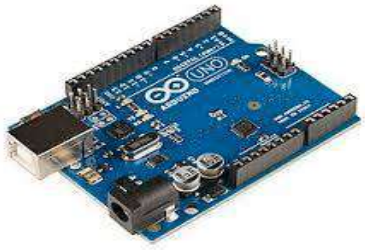


Figure II.3 *La carte Arduino Uno.*

Le tableau ci-dessous présente les caractéristiques de la carte Arduino Uno :

Tableau II.3 *Caractéristiques de la carte Arduino Uno.*[9]

Microcontrôleur	ATmega328P
Tension d'alimentation interne	5 V
Tension d'alimentation (recommandée)	7 à 12 V
Entrées/sorties numériques	14 dont 6 sorties PWM
Entrées analogiques	6
Courant max par broches E/S	40 mA
Courant max sur la sortie 3,3V	50 mA
Mémoire Flash	32 kb dont 0.5 kb utilisée par le bootloader
Mémoire SRAM	2 kb
Mémoire EEPROM	1kb
Fréquence d'horloge	16 MHz
Dimensions	68.6x 53.3mm

c. le module Radio :

Pour notre RCSF, nous avons opté pour l'un des modules Radio les plus populaires à l'heure actuelle, qui est le nRF24L01+. Ce module radio est d'un cout très compétitif et de caractéristiques très intéressantes par rapport à d'autres modules radio du commerce.

Une comparaison entre les différents standards est présentée dans le tableau ci-dessous:

Tableau II.4 *Comparaison des standards.*[5]

Protocole	Zigbee	Bluetooth	Wifi	nRF24l01+
IEEE	802.15.4	802.15.1	802.11a/b/g	-
Autonomie avec pile	Années	Jours	Heures	Années
Nombre de nœuds	65 000	7	32	127 (6 VPN)
Vitesse de transfert	250 Kbps	1 Mbps	11 - 54 - 108 Mbps	2 Mbps
Portée	100 m	10-100 m	300 m	100 m
Prix (Emetteur/Récepteur) en DA	3500 - 9000	1000 - 3000	1000 - 4500	500 – 1200

c.1 Description du module nRF24L01:[5]

Le module nRF24L01+ vient du constructeur norvégien Nordic Semiconductor, il regroupe des puces de communications utilisant la bande des 2.4 GHz. Dans cette famille, le chipset nRF24L01+ a connaît un grand succès auprès des utilisateurs. Depuis 2010 ce composant est embarqué dans la plupart des claviers et souris sans fils qui inondent le marché grand public.

Le nRF24L01+ offre une communication radio accessible via une interface SPI standard.

La même puce permet de recevoir et d'émettre (d'où le nom transmetteur), et peut fonctionner sur plus d'une centaine de canaux, et consomme très peu d'énergie en modes : émission, réception ou standby.

Ce module Radio utilise une modulation GFSK avec 127 canaux, des sauts de fréquences, couplés à un dispositif CR permettent un antiparasitage très efficace. Il possède aussi une fonction d'auto ACKNOWLEDGE (l'auto renvoie) et support jusqu'à 6 connexions simultanées (pipeline) possédant chacune son propre buffer de réception de 32 bytes.

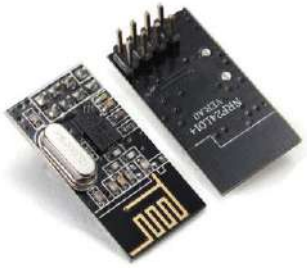


Figure II.4 *Le module Radio nRF24L01+.*

Le tableau ci-dessous présente les caractéristiques principales du module nRF24L01+ :

Tableau II.5 *Caractéristiques du module nRF24L01.*[10]

Fonctionne dans la bande ISM (Industriel, scientifique, médical)	2,4 GHz
Protocole de communication	Enhanced ShockBurst (ESB).
Débit de transmission des données	256 Kbps,1 Mbps et 2 Mbps
Nombre de pipeline	6
La portée d'émission (module avec antenne gravée)	70 ~100 m
Dimensions	15 x 29 x 0.8 mm
Les broches utilisables	8
Alimentation	1.9 ~ 3.6V (Non 5V !)
Courant de travail nécessaire	13.5mA
Mode de travail	mode RX / mode TX

Les puces nRF24L01+ utilisent des canaux démarrant à 2.4 GHZ+ RF_CH (avec RF_CH est le numéro du canal allons de 0 à 126). Ce module intègre un protocole spécial appelé ESB: « Enhanced ShockBurst », ce protocole permet de garantir une transmission fiable de manière transparente sans s'occuper des détails.

c.2 Le protocole Enhanced ShockBurst :[11]

ShockBurst est une couche de liaison de données à base de paquets qui est responsable de l'assemblage automatique, la synchronisation, la reconnaissance automatique et la retransmissions de paquets. ShockBurst permet la mise en œuvre d'une communication ultra haute performance à faible puissance avec des microcontrôleurs à faible coût. Les principales caractéristiques de ShockBurst sont :

- ✓ La charge utile est d'une longueur dynamique de1 à 32 octets,
- ✓ Le traitement automatique de paquets,
- ✓ La gestion automatique et transparente des transactions de paquets.

Le tableau suivant présente la structure de la trame utilisée par le nRF24L01 :

Tableau II.6 *La structure de la Trame du protocole ShockBurst.*[5]

Préambule	Adresse	En-tête	Charge utile	CRC
1 octet	3-5 octets	9 bits	0 – 32 octet(s)	0 – 2 octet(s)

Les trames du protocole ShockBurst contiennent un préambule radio, l'adresse de destination, un en-tête, la charge utile (data), et un champ CRC. La charge utile peut être d'une langueur fixe ou variable, sa taille est renseignée dans 6 des 9 bits de l'en-tête.

Le tableau suivant montre la description des broches du module Radio nRF24L01+:

Tableau II.7 *Description des pins du nRF24L01+*.[5]

Broche	NOM	FONCTION	Description
1	GND	Alimentation	Masse
2	VCC	Alimentation	1.9 à 3.6V DC
3	CE	Entrée	Chip enable permet de sélectionner le mode RX ou TX.
4	CSN	Entrée	SPI chip Select (Chip select Not) ce pin est normalement en état bas pour activer le bus SPI du module
5	SCK	Entrée	SPI Clock
6	MOSI	Entrée	Master Data output Slave Data Input
7	MISO	Sortie	Master Data intput Slave Data Output
8	IRQ	Sortie	interruption pin. Active en état bas

c.3 Mode de fonctionnement :[11]

Le module nRF24 est doté de 6 canaux logiques dans une seule bande de fréquence : un seul canal est réservé pour l'émission tandis que les 6 canaux peuvent être utilisés en réception où chaque canal est associé à un « pipe » c'est-à-dire une adresse physique, en effet le module peut écouter 6 autres modules simultanément comme indique-la Figure suivante.

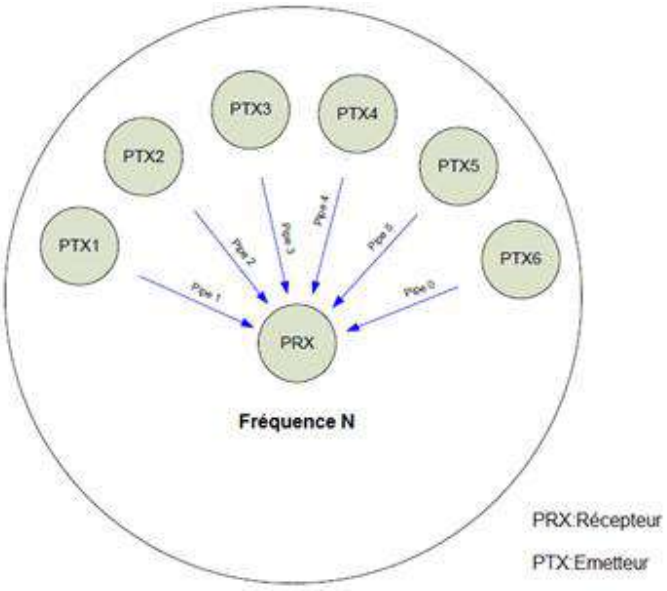


Figure II.5 *Schéma synoptique du mode de transmission nRF24L01+.*

En cas de réception d'un paquet, la puce gère le protocole ShockBurst en envoyant un ACK en retour puis en désassemblant le paquet pour ne garder que son contenu utile (les données transmises) dans une file d'attente (FIFO) spécifique pour chaque canal d'écoute.

c.4 Principe d'utilisation :[12]

- Premier point : Ce composant dispose de 6 "pipe" qui servent à communiquer entre modules. Le pipe 0 sert pour envoyer les messages, et peut éventuellement servir en réception. Les autres pipes (2 à 5) ne peuvent être utilisées que pour la réception.
- Second point : Ces pipes contiennent une adresse sur 40 bits, qui permet de savoir avec qui le module communique.

Tableau II.8 *Exemple d'échange de l'adresse entre deux module*

Module 1	Module 2
pipe 0(écriture) : Adresse 0	pipe 0(écriture) : Adresse 1
pipe 1(lecture) : Adresse 1	pipe 1(lecture) : Adresse 0

Les données récupérées par le module nRF24L01+ peuvent être lus par le microcontrôleur associé à l'aide d'une liaison série SPI.

c.5 Protocole de communication SPI :[11]

La liaison SPI est un bus de données séries synchrone baptisé par Motorola, et qui opère en Full Duplex. Les circuits communiquent selon un schéma maitre esclaves, ou le maitre s'occupe totalement de la communication. Plusieurs esclaves peuvent coexister sur un bus, la sélection du destinataire se fait par une ligne dédiée entre le maitre et l'esclave appelée "chip select". Le bus SPI contient 4 signaux logiques:

- SCLK : horloge (généré par le maitre).
- MOSI : Master Output, Slave Input (généré par le maitre).
- MISO : Master Input, Slave Output (généré par l'esclave).
- SS : Slave Select, Actif à l'état bas (généré par le maitre).

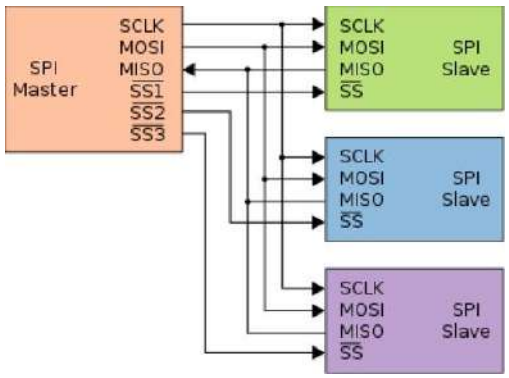


Figure II.6 *Mode de fonctionnement de bus SPI.*

Pour commencer une communication, premièrement, le maitre configure l'horloge, ensuite il sélectionne la seule puce esclave désirée avec qui il veut communiquer via la ligne SS.

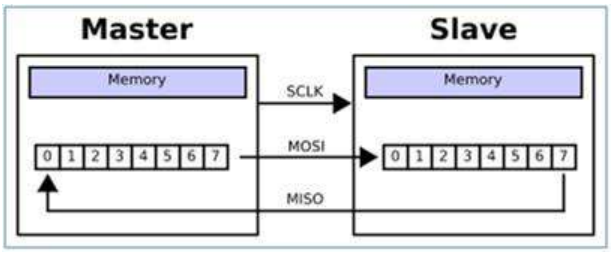


Figure II.7 *Transmission des données dans le bus SPI.*

Durant chaque cycle d'horloge SPI, une transmission de donnée full duplex se produit, à chaque coup d'horloge le maître et l'esclave s'échangent un bit, après huit coups d'horloges le maître a transmis un octet à l'esclave et vice versa. La vitesse de l'horloge est réglée selon des caractéristiques propres aux périphériques. Lorsqu'il n'y a plus de données à transmettre le maitre arrête son horloge, et cela désactive l'esclave.

d. Conception du réseau de capteurs sans fil :

La Conception des nœuds de notre réseau de capteurs est effectuée par l'interfaçage du module nRF24 avec la carte Arduino choisie, la communication est effectuée en respectant l'affectation des pins. Après l'interfaçage de l'Arduino avec le module nRF24, afin de créer les différents nœuds du réseau de capteurs, il est indispensable de programmer leurs comportements.

d.1 L'interfaçage du module nRF24 et la carte Arduino :

Pour interconnecté le module Radio avec la carte Arduino Nano des nœuds de capteurs on a réalisé le câblage suivant :

nrf24L01+ **Arduino Nano**

- VCC ----------→ 3V3
- GND ----------→GND
- SCK ----------→D13
- MISO----------→D12
- MOSI----------→D11
- CSN ----------→D10
- CE ----------→D9
- IRQ ----------→ Non connecté.

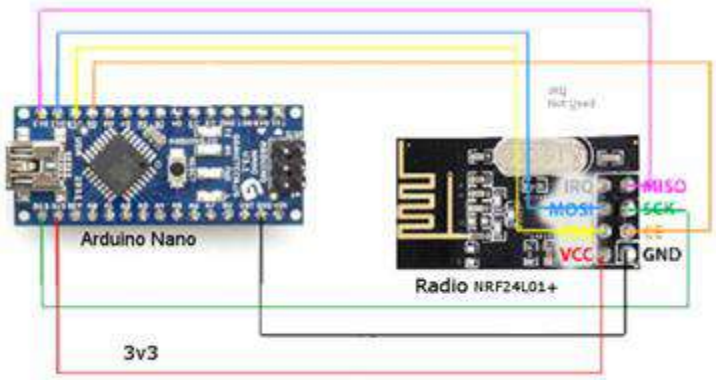


Figure II.8 *L'interfaçage du nRF24L01+ avec Arduino Nano.*

Afin de simplifier la mise au point du nœud de capteurs nous avons utilisé la shield I/O & Wireless Nano du constructeur chinois RobotDyn (Fig. II-9).

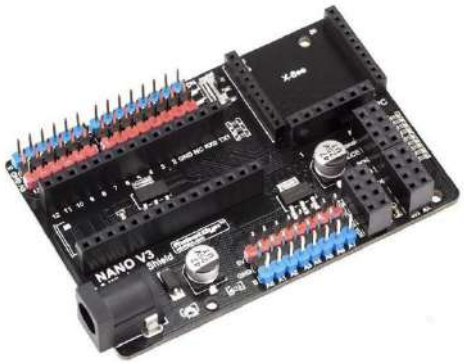


Figure II.9 *La Shield I/O, Expansion Board, Wireless V3 pour l'Arduino Nano*

Pour interconnecter le module Radio avec la carte Arduino Uno des nœuds routeurs et du nœud de base on a réalisé le câblage suivant :

nrf24L01+ Arduino UNO

- VCC ----------→ 3V3
- GND ----------→GND
- SCK ----------→D13
- MISO----------→D12
- MOSI----------→D11
- CSN ----------→D10
- CE ----------→D9
- IRQ ----------→Non connecté.

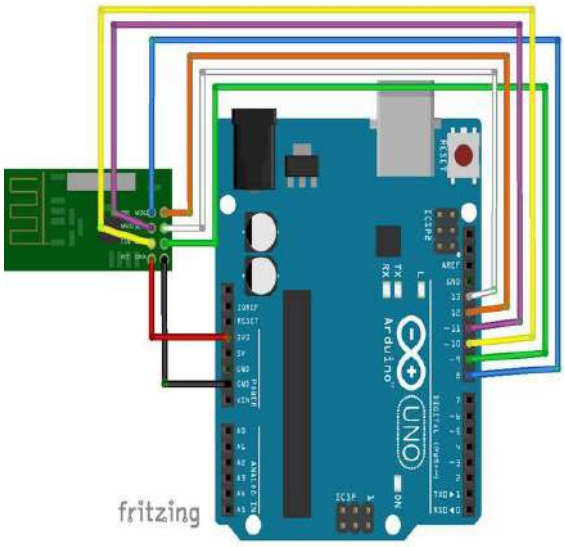


Figure II.10 *L'interfaçage du nRF24 avec la carte Arduino Uno***.**

Comme nous ne disposons pas d'une shield similaire à celle utilisée avec la carte Nano pour les nœuds de capteurs et pour simplifier la mise au point du nœud de base et des nœuds passerelles nous avons réalisé nos propres shield qui nous ont permis d'interfacer les modules radio avec nos cartes Uno.

Le circuit électrique de la shield Wireless utilisée avec notre nœud de base et nos nœuds routeurs est très simple, il est conçu autour du module nRF24 associé à deux condensateurs de 100nF et de 10uF et un ensemble de connecteurs et jumpers.

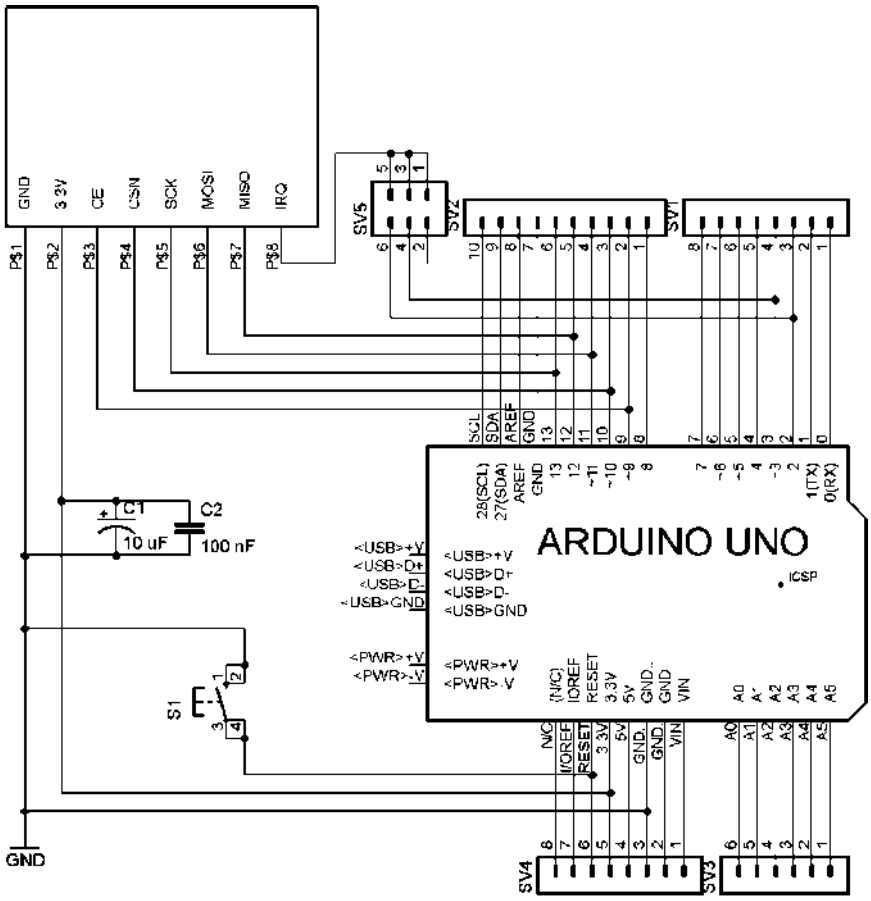


Figure II.9 *Le schéma électrique de la shield Wireless pour la carte Uno*

d.2 Architecture de communication dans les réseaux de capteurs :[11]

Le modèle de communication des réseaux de capteurs comprend cinq couches qui ont les mêmes fonctions que celles du modèle OSI.

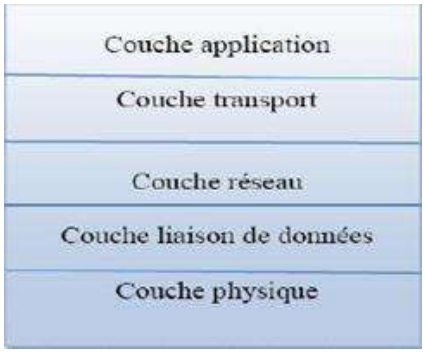


Figure II.12 *Le modèle en couches du réseau de capteurs sans fil.*

Pour le module nRF24 il existe sur le web plusieurs bibliothèques de développement qui possèdent chacune des caractéristiques bien spécifiques. Pour notre implémentation et après de longs tests nous avons retenu les bibliothèques suivantes :

- **La bibliothèque RF24 pour la couche liaison de donnée :**

Pour exploiter les fonctionnalités du module nRF24 nous avons fait appel à la librairie RF24, cette dernière est destinée à gérer la couche physique, elle est responsable, de la bande fréquentielle utilisée, le débit de transmission ainsi que la bonne configuration des différents registres du module nRF24.

- **La bibliothèque RF24Network pour la couche réseau :**

La classe RF24Network implémente la normalisation de la couche réseau en utilisant les modulesnRF24, elle est pilotée par la librairie RF24, cette librairie permet l'échange des données entre les différents nœuds à condition qu'on attribue manuellement dans le programme à chaque nœuds une adresse physique entre 1 à255. Les adresses du réseau sont assignées sous forme d'un arbre. L'adresse d'un nœud est un nombre octal qui décrit exactement la position de ce dernier dans l'arbre. La manière d'affecter des adresses pour les différents nœuds par le programmeur est la suivante :

- Le nœud 00 c'est l'adresse du parent principal.
- Les nœuds 01-05 sont des nœuds fils du parent principal.
- Les nœuds 011-021-031 sont des nœuds fils du parent secondaire 01.
- Les nœuds 012-022-032 sont des nœuds fils du parent secondaire 02.

- **La librairie RF24Mesh pour la couche transport :**

La bibliothèque RF24Network fournit un système d'adressage et de routage pour les modules de radio nRF24, qui permet aux grands réseaux de capteurs sans fil d'être construits. Cette bibliothèque exploite la capacité du module nRF24 de communiquer simultanément avec 6 de ces semblables. La bibliothèque RF24Mesh offre des fonctionnalités supplémentaires, à l'image de la configuration automatique et dynamique des adresses associées aux capteurs sans fil.

Remarque : Le module nRF24 comme le ZigBee permet d'avoir les trois topologies réseau: Arbre, étoile et maillée.

Le choix d'une technologie dépend des services proposés, ainsi que des besoins du concepteur du réseau. Certains paramètres comme la puissance, le débit, la portée, le coût, la sécurité, et le nombre de nœuds supportés doivent être pris en compte.

II.4 Conclusion

Dans ce chapitre, on a présenté le schéma bloc du RCSF réalisé et le choix des composants retenus lors de la conception.

CHAPITRE III

RÉALISATION ET MISE EN MARCHE DU SYSTÈME

III.1 Introduction

Dans ce chapitre, nous allons présenter les outils de développement software de notre RCSF, ensuite nous décrirons la méthode que nous avons adoptée pour faire communiquer la carte arduino avec le module nRF24 et finalement nous présenterons les organigrammes des programmes de transmission multiciever et le réseau multiniveaux.

III.2 La partie software

a. L'environnement Intégré de développement d'Arduino :

L'environnement de programmation Arduino (IDE en anglais) est une application écrite en Java inspirée du langage Processing, il permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte.

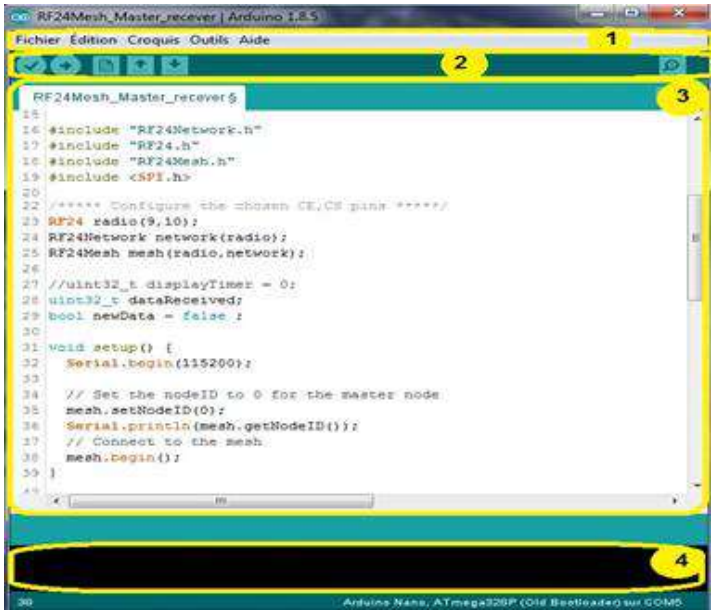


Figure III.1 *L'IDE Arduino*

1. La barre des menus de l'IDE ;
2. Les boutons respectivement de compilation, télé versement du programme, Nouveau, Ouvrir et Enregistrer.
3. L'espace d'édition du programme.
4. La zone d'affichage des résultats de compilation et de télé versement du programme.

b. L'utilisation du module nRF24 avec Arduino :

Pour la mise en marche de notre RCSF, il est nécessaire de programmer l'interface de communication entre les cartes Arduino et les modules nRF24. Pour cela et afin de simplifier

le développement nous avons utilisé un ensemble de bibliothèques spécifiques sous l'IDE Arduino. En effet, une bibliothèque est un ensemble de fonctions et de classes qui s'ajoutent aux fonctions de base de l'environnement Arduino et qui s'adressent à gérer les différentes couches du protocole de communication de notre réseau de capteur sans fil. Dans ce qui suit nous décrivons les étapes nécessaires afin d'ajouter ces bibliothèques à la version de base de l'IDE pour qu'il puisse exploiter les capacités du module nRF24.

- Après l'installation de l'IDE Arduino, dans le menu *croquis* on choisis : *inclure une bibliothèque* puis *Gérer les bibliothèques*:

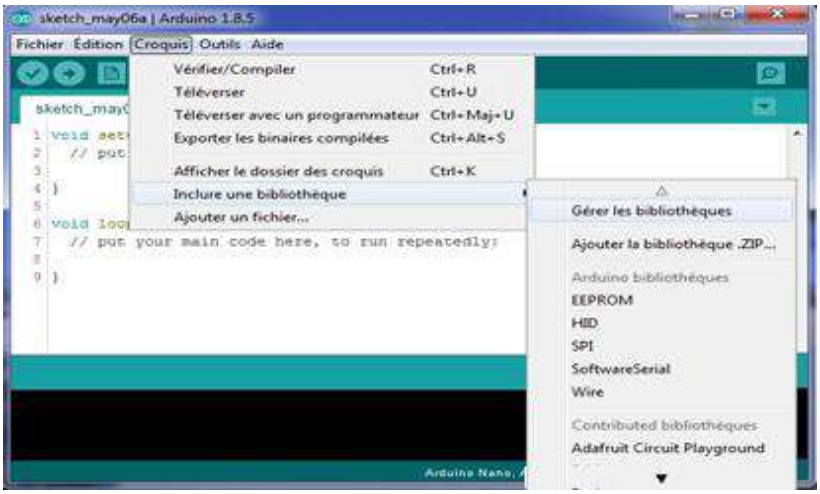


Figure III.2 *L'installation et la mise à jour des bibliothèques relatives au nRF24*

- Ensuite sur la fenêtre de la Fig. III.3 et dans le champ recherche, on tape "rf24" et on choisit tous les bibliothèques créé par "TMRh20".

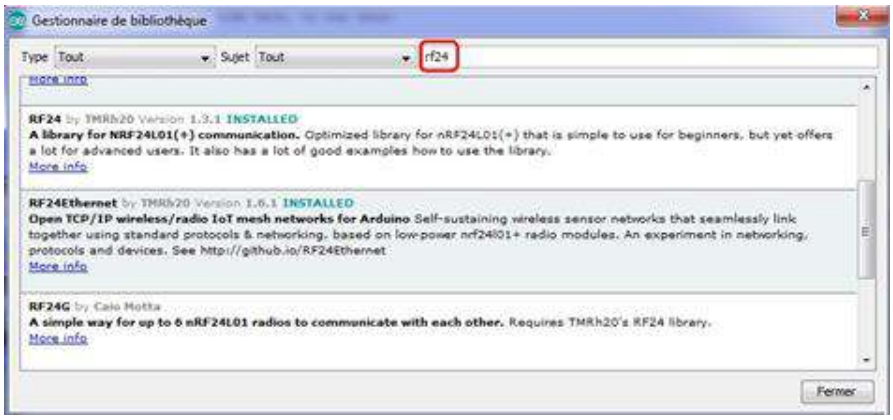


Figure III.3 *Téléchargement du soft de programmation de nRF24L01+*

- Par la suite on va installer les bibliothèques suivantes : *RF24*, *RF24Mesh* et *RF24Network*.
- Durant les premières phases de développement, nous avons constaté que les cartes Nano n'arrivent pas à ce programmé correctement. Ce problème vient du fait que la nouvelle version de l'IDE Arduino exploite une nouvelle version du bootloader qui communique

avec baud rate de 115200, tandis que l'ancien bootloader communique avec baud rate de 57600. Donc pour résoudre le problème il fallait configurer l'IDE à utiliser les réglages de l'ancien version de bootloader, ceci a été réalisé par la procédure suivante : on choisit le menus *Outils* puis Processeur:'ATmega328P'ensuite*'ATmega328P (Old bootloader)'*

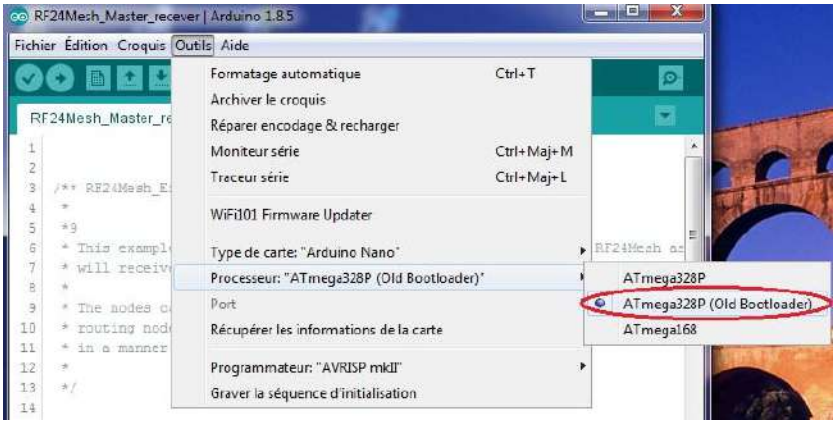


Figure III.4 *Choix du bootloader du processeur*

III.3 mise en marche de l'application

a. Fonctionnement multicievers :

En premier lieu, nous avons commencé par l'implémentation du mode multicievers des modules nRF24 à base d'un seul récepteur (nœud de base) et cinq émetteurs (nœuds de capteurs), Le nRF24 du nœud de base a été configuré pour recevoir les données à travers 5 pipelines différents. L'organigramme relatif à cette implémentation ainsi que les classes et bibliothèques utilisées sont détaillés dans les paragraphes suivants.

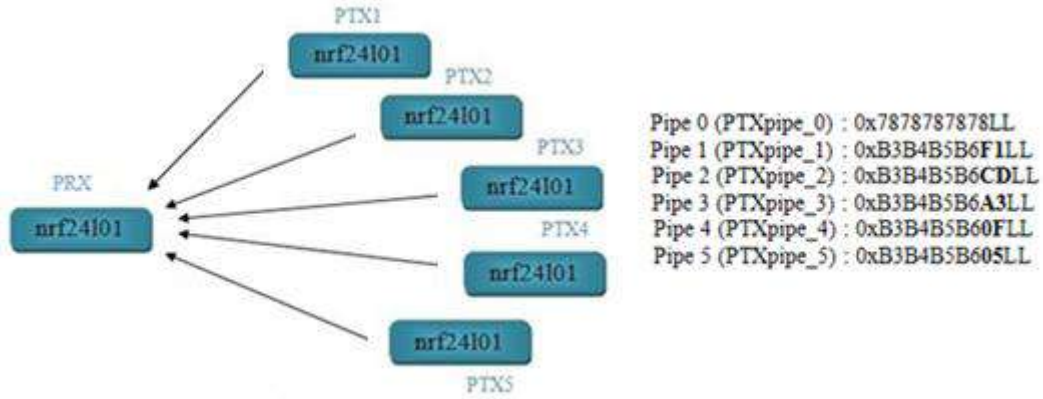


Figure III.10 *Communication RCSF multicievers*

La capacité de définition de plusieurs destinataires est d'avoir jusqu'à 6 canaux (pipes) de communication radio ouverts dans un mode de réception RX ("lecture") simultanément. Cela permet au transmetteur nRF24 d'avoir une structure en étoile formé par un nœud

récepteur PRX (récepteur primaire) et de six nœuds émetteurs au maximum (émetteurs primaires PTX1 - PTX6).

La Fig. III.5, montre un réseau à cinq pipes d'écriture (données) qui sont ouverts dans le nœud récepteur primaire (PRX). Chaque nœud PTX est relié à l'un de ces pipes pour l'utiliser dans l'émission. Chaque pipe possède une adresse, les adresses / pipes doivent avoir un motif d'octets distinct: seul le cinquième octet est entièrement unique parmi tous les pipes est connu comme l'octet le moins significatif.

La réception des données est en mode RX. Cela signifie que tout nRF24 peut réellement agir avec 6 récepteurs différents en même temps. Toutes les pipes utilisent le même canal RF (quel que soit le canal défini dans le registre). Chaque fois que nous voulons envoyer des données à une adresse différente, nous devons changer deux adresses, mais avec la bibliothèque nRF24L01 la gestion se fait d'une manière transparente à condition de bien fixer les adresses des pipes au départ.

La séquence TX -> RX suit un modèle simple dans une esquisse. Les données utiles peuvent être de longueur statique ou dynamique et peuvent également être attachées à des paquets ACK (acquittement) qui sont régulièrement renvoyés du récepteur à l'émetteur pour confirmer la bonne transmission des paquets

La Fig. III.6 montre la configuration du test multicievers que nous avons réalisé :



Figure III.11 *La mise en marche du mode multicievers*

Pour simuler la transmission entre un ensemble de cinq nœuds de capteur et le nœud de base en mode multicievers, nous avons programmé un compteur sur chaque nœud de capteur

avec un indice différent pour pouvoir identifier les données au niveau du nœud de base. Et afin de simuler des fréquences d'échantillonnage différentes pour les nœuds de capteurs, nous avons définie pour chacun des nœuds une périodicité d'émission différente des autres nœuds :

1. Le compteur du nœud de capteur n°1 commence à partir de la valeur 1000 avec une période d'émission de 50 ms ;
2. Le compteur du nœud de capteur n°2 commence à partir de la valeur 2000 avec une période d'émission de 60 ms ;
3. Le compteur du nœud de capteur n°3 commence à partir de la valeur 3000 avec une période d'émission de 80 ms ;
4. Le compteur du nœud de capteur n°4 commence à partir de la valeur 4000 avec une période d'émission de 100 ms ;
5. Le compteur du nœud de capteur n°5 commence à partir de la valeur 5000 avec une période d'émission de 250 ms.

La figure suivante montre les messages de débogage relatifs au nœud de capteurs n° 5 qui montre l'achèvement de l'émission de neuf paquets vers le nœud de base :



Figure III.12 *La fenêtre de débogage du nœud de capteur n°5*

Du côté du nœud de base et pour une fenêtre de réception de 2 s**,** un ensemble de data des différents nœuds de capteurs a été reçu comme le montre la Fig.III.8, dans laquelle on peut clairement identifier et dénombrer les échantillons reçus :

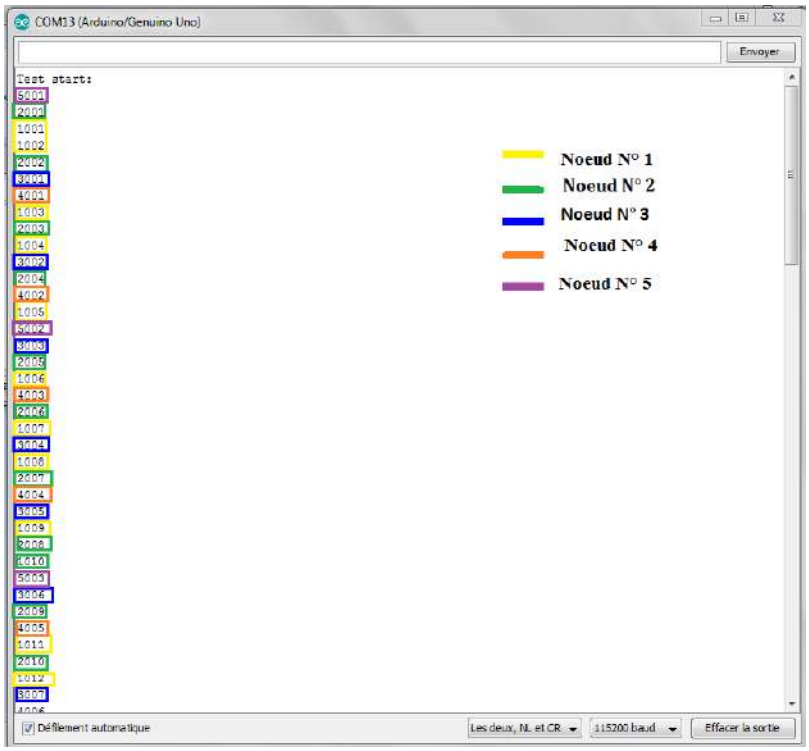


Figure III.13 *La fenêtre de débogage relative au nœud de base*

La Fig.III.9 montre les échantillons reçus de chaque nœud de capteurs que nous avons organisé dans un fichier Excel :

Nœud_1	Nœud_2	Nœud_3	Nœud_4	Nœud_5
1001	2001	3001	4001	5001
1002	2002	3002	4002	5002
1003	2003	3003	4003	5003
1004	2004	3004	4004	5004
1005	2005	3005	4005	5005
1006	2006	3006	4006	5006
1007	2007	3007	4007	5007
1008	2008	3008	4008	5008
1009	2009	3009	4009	5009
1010	2010	3010	4010	
1011	2011	3011	4011	
1012	2012	3012	4012	
1013	2013	3013	4013	
1014	2014	3014	4014	
1015	2015	3015	4015	
1016	2016	3016	4016	
1017	2017	3017	4017	
1018	2018	3018	4018	
1019	2019	3019	4019	
1020	2020	3020	4020	
1021	2021	3021		
1022	2022	3022		
1023	2023	3023		
1024	2024	3024		
1025	2025	3025		
1026	2026			
1027	2027			
1028	2028			
1029	2029			
1030	2030			
1031	2031			
1032	2032			
1033	2033			
1034	2034			
1035				
1036				
1037				
1038				
1039				
1040				
40	34	25	20	9

Figure III.14 *Représentation des données reçus.*

Le tableau III.1 présente un résumé n des résultats du test réalisé :

Tableau III.1 *Comparaison entre le nombre théorique et pratique des échantillons reçus*

Nœud n°	Nbr théorique d'échantillons	Nbr d'échantillons reçus
1	40	40
2	34	34
3	25	25
4	20	20
5	8	9

Nous constatons qu'il n y'a pas de pertes de données, sauf que nous avons eu une donnée en plus de la part du nœud de capteur n° 5 ceci est due probablement à la temporisation que nous avons utilisée avec l'instruction '*delay*', car cette dernière n'est pas aussi bien précise que les temporisations réalisées à l'aide des timers et des interruptions. Ceci induit donc une petite marge d'erreur sur la durée de la temporisation.

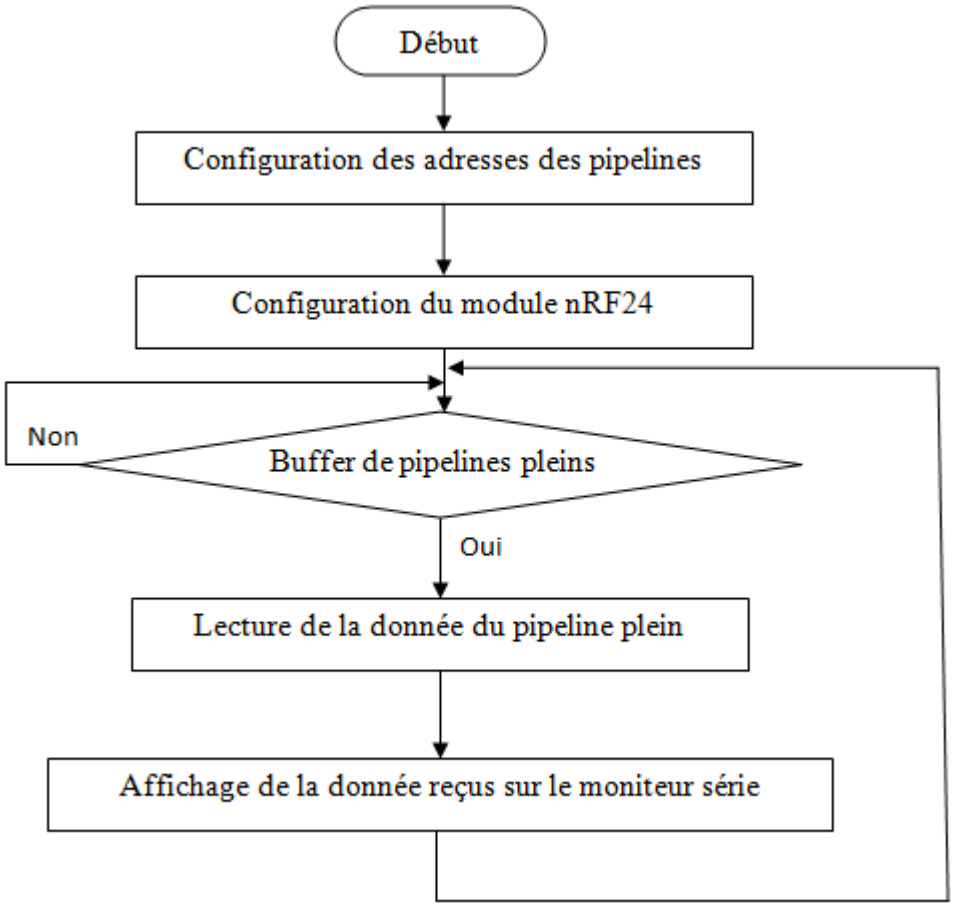


Figure III.10 *L'organigramme du programme du nœud de base*

Dans ce programme nous avons utilisé les bibliothèques suivantes :

- **SPI.h** : pour gérer l'interface de communication avec le module nRF24 ;
- **nRF24L01.h** : pour gérer les registres internes du module Radio ;
- **RF24.h** : pour piloter le modem radio.

Dans la partie déclarative du programme nous avons commencé par la création de l'entité ***radio*** qui réfère le module nRF24 associé à notre Arduino, ensuite on a configuré les pins de l'Arduino pour piloter le module nRF24, ensuite nous avons défini les adresses de différentes pipes, ces étapes ont été achevées en utilisant les fonctions des bibliothèques suivantes :

- **RF24 radio (pinCE, pinCSN) :** cette fonction permet de crée et de configuré le module ''nRF24'' connecté à l'Arduino. Les arguments **pinCE** et **pinCSN** sont des numéros de broche numériques auxquels les signaux CE et CSN sont connectés dans notre cas sur les lignes 9 et 10 respectivement
- **rAddress [ ] :** représente le vecteurs des adresses des différents pipelines.

Dans la partie de configuration nous avons utilisé les fonctions suivantes:

- **Serial.begin ( ) :** définit la vitesse de transmission avec le PC via le moniteur série.
- **radio.begin ( ) :** cette fonction active le modem ‘’nRF24’’.
- **radio.setChannel ( ) :** définit le canal de communication RF.
- **radio.setPALevel (**) **:** permet le réglage du niveau de puissance de l’émission.
- **radio.setDataRate ( ) :** permet le réglage du débit de transmission des données
- **radio.setAutoAck ( ) :** activer/désactiver les paquets d'accusé de réception automatique.
- **radio.setRetires ( ) :** Il indique combien de fois le module va réémettre les données en cas de nonaquisition d’un paquet ACK.
- **radio.setCRCLength ( ) :** définisse la longueur du champ CRC sur 16 bits pour s’assurer des data reçus.
- **radio.openReadinPipe ( ) :** détermine l'adresse du pipeline qui reçoit les données.
- **radio.startListening ( ) :** À partir de ce moment, le modem attend les données envoyées à l'adresse spécifiée.

Dans le programme principal nous avons effectué les opérations suivantes si les données sont arrivées à l'adresse du modem en utilisant la fonction **"radio.available ( )"**. Cette fonction renvoie une valeur "oui" si nous avons reçu des données, ou "non" si aucune donnée ni reçus.

- **radio.read(&dataReceived, sizeof(dataReceived)) :** permet la lecture de la donnée reçue.

L'organigramme du nœud capteurs (émetteur**) :**

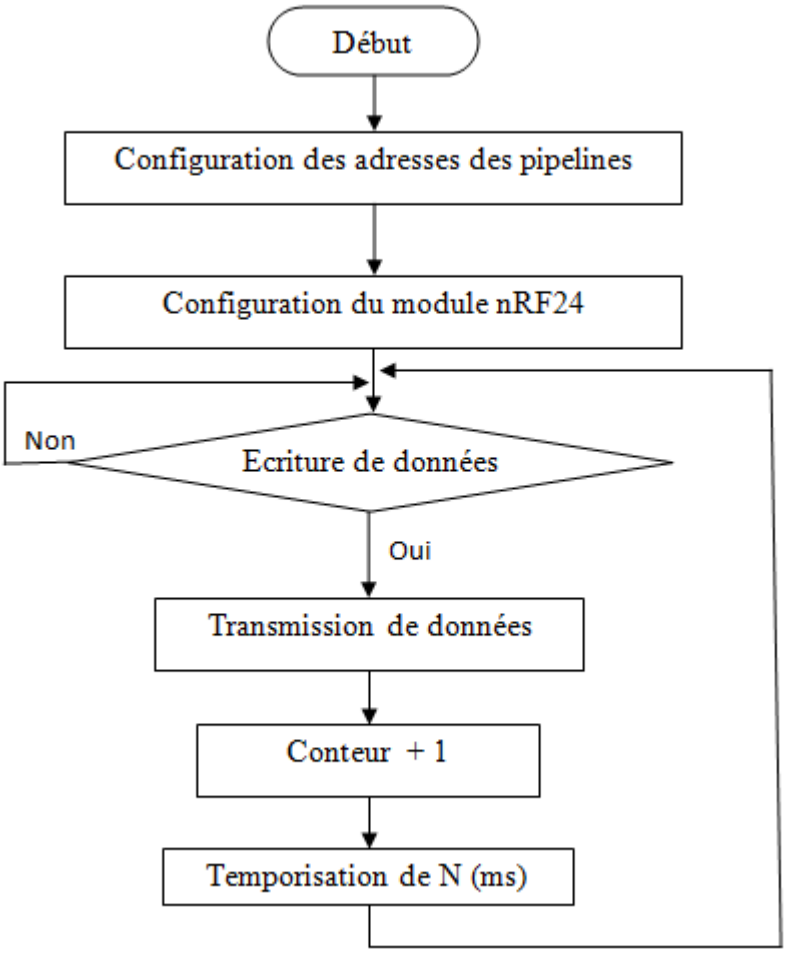


Figure III.15 *L'organigramme du nœud d'émetteur*

Dans cet organigramme nous avons utilisé les mêmes bibliothèques que le récepteur.

Dans la partie de déclaration nous avons déclaré les pins, ensuite nous devons créer un objet appelé « radio » ou nous posons les pipes, ainsi que nous avons déclaré la donnée à transmettre

```
const int pinCE = 9;                                    // This pin is used to set the nRF24 to standby (0) or active mode (1)
const int pinCSN = 10;                                  // This pin is used to tell the nRF24 whether the SPI communication is a

RF24 radio(pinCE, pinCSN);                              // Create your nRF24 object or wireless SPI connection

#define NODE 1                                          // must be a number from 1 - 6 identifying the PTX node

const uint64_t wAddress[] = {0x7878787878LL, 0xB3B4B5B6F1LL, 0xB3B4B5B6CDLL, 0xB3B4B5B6A3LL, 0xB3B4B5B60FLL, 0xB3B4B5B605LL};
const uint64_t PTXpipe = wAddress[ NODE ];              // Pulls the address from the above array for this node's pipe

int counter_1 = 1001;                                   // used to count the packets sent
bool test_done = true;
unsigned long start_timer = 0;
```

Figure III.16 *La partie de déclaration au niveau de nœud end_device***.**

- **define NODE 1 :** on identifiant les numéros de nœuds d émission.
- **const uint64_t PTXpipe = wAddress[NODE]** : chaque node prend une adresse.
- Puis on a déclaré la donnée de cet émetteur **int counter_1 = 1001.**

Dans la partie de configuration nous avons utilisé presque les mêmes classes sauf que nous remplaçons **startListening** par **stopListening** pour passer en mode émission.

Dans le programme principal 'loop ()' nous avons effectuons les opérations suivantes :

- **radio.openWritingPipe ( ) :** ouvre le pipe pour faire la transmission.
- **Radio.write (&counter, sizeof (counter)) :** nous envoyons la donnée via la radio au modem (l'adresse du destination a été configurée plus tôt en utilisant "openWritingPipe").
- **radio.write** Cette fonction renvoie une valeur si nous avons "oui" le module fait la transmission de données. Sinon il n y'a pas de transmission.

Une petite analyse :

Nous avons pu commander **que** six module nrf24l01+ parce que nous avons trouvé un problème au niveau des pipes; donc la bibliothèque que nous avons utilisé n'est pas suffisante pour faire avec elle un réseau multiniveaux, car il contient plusieurs pipes c'est pour ça nous avons cherché une autre option (autre bibliothèque) pour crier notre réseau.

Nous passons à la deuxième phase : réseau multiniveaux

b. Fonctionnement d'un réseau multiniveaux :

Dans notre expérience d'un RCSF multiniveaux, elle contient trois types de nœuds nœud de sink, nœud routeur et nœud end_device, par la suite nous allons décrive l'utilité et l'étude de chaque nœud aussi nous allons citer leurs organigrammes ensuite les bibliothèques utilisées ainsi que nous détaillerons les classes.

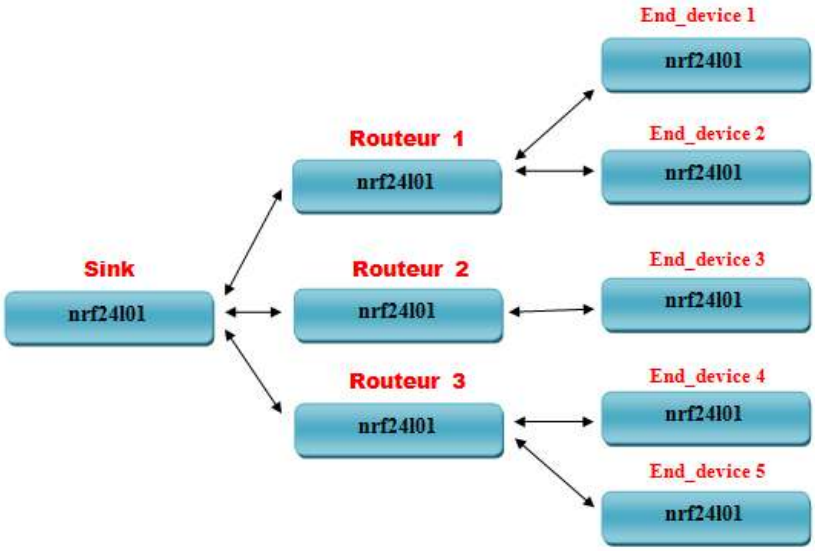


Figure III.17 *Communication RCSF multiniveaux***.**

- **la communication radio et la topologie :**

Lorsqu'une transmission a lieu d'un module radio nRF24L01+ à un autre, la radio réceptrice communiquera de nouveau avec l'expéditeur avec un accusé de réception (ACK) pour indiquer le succès. Si l'expéditeur ne reçoit pas d'accusé de réception, le module radio nRF24L01+ effectue automatiquement une série de tentatives chronométrées, à des intervalles définis. Les modules radio utilisent des techniques comme l'adressage et la numérotation des charges utiles pour gérer cela, mais tout se fait automatiquement par la puce nRF, à l'abri des regards de l'utilisateur.

Lorsque vous travaillez sur un réseau radio, certaines de ces techniques automatisées peuvent en fait entraver la transmission de données dans une certaine mesure. Réessayer à plusieurs reprises des charges utiles échouées sur un réseau radio peut gêner la communication pour les nœuds voisins ou réduire le débit et les erreurs sur les nœuds de routage.

Les modules radio de ce réseau sont liés par des adresses attribuées aux pipes. Chaque nRF24L01+ peut écouter 6 adresses sur 6 pipes, donc chaque module radio a un pipe parent et 5 pipes enfants, qui sont utilisés pour former une structure arborescente. Les nœuds communiquent directement avec leurs nœuds parents et enfants. Tout autre trafic vers ou depuis un nœud doit être acheminé via le réseau.

- **Topologie pour les réseaux maillés utilisant nRF24L01+ :**

Toute personne familiarisée avec les réseaux IP devrait être en mesure de comprendre facilement la topologie RF24Network. Le nœud maître peut être vu comme la passerelle, avec jusqu'à 4 nœuds directement connectés. Chacun de ces nœuds crée un sous-réseau en dessous, avec jusqu'à 4 nœuds enfants supplémentaires. Le schéma de numérotation peut également être lié à des adresses IP, afin de comprendre la topologie via un sous-réseau. Les nœuds peuvent avoir 5 enfants si la multidiffusion est désactivée.

Exprimer les adresses RF24Network au format IP:

A titre d'exemple, nous pourrions désigner le nœud maître en théorie, comme l'adresse 10.10.10.10

Les nœuds enfants du maître seraient 10.10.10.1, 10.10.10.2, 10.10.10.3, 10.10.10.4 et 10.10.10.5

Les nœuds enfants de 10.10.10.1 seraient 10.10.1.1, 10.10.2.1, 10.10.3.1, 10.10.4.1 et 10.10.5.1

Dans RF24Network, le maître est juste 00

Les enfants du maître sont 01, 02, 03, 04,05

Les enfants de 01 sont 011 021 031 041 051

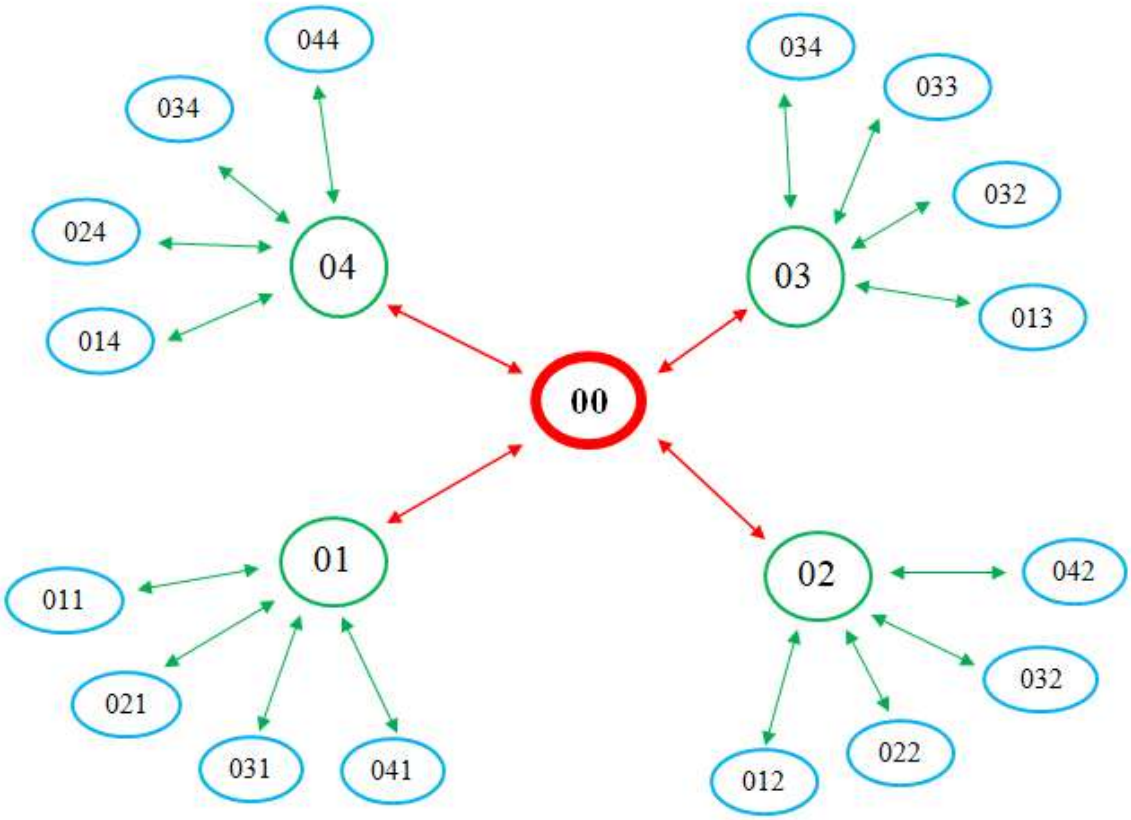


Figure III.18 *La topologie RF24Network.*

- **Routage :**

Le routage du trafic est géré de manière invisible pour l'utilisateur, par la couche réseau. Si les adresses réseau sont attribuées conformément à la disposition physique du réseau, les nœuds achemineront le trafic automatiquement selon les besoins.

- Exemple: Le noeud 00 envoie au noeud 01. Les nœuds utiliseront les fonctions intégrées d'auto-réessai et d'auto-ack.
- Exemple: Le nœud 00 envoie au nœud 011. Le nœud 00 envoie au nœud 01 comme précédemment. Le nœud 01 transmet le message à 011. Si la distribution a réussi, le nœud 01 transmet également un message au nœud 00, indiquant le succès.

Ancienne fonctionnalité: le nœud 00 envoie au nœud 011 en utilisant la fonction auto-ack. Le nœud 00 envoie d'abord à 01, 01 reconnaît. Le nœud 01 transmet la charge utile à 011 à l'aide de la fonction de réception automatique. Si la charge utile échoue entre 01 et 011, le nœud 00 n'a aucun moyen de le savoir.

b.1 Deuxième test : configuration multiniveaux

Notre application multiniveaux est composée de trois niveaux, le premier niveau représente le nœud sink qui est chargé de recevoir les informations et les afficher, le deuxième niveau c'est un niveau intermédiaire qui est chargé de retransmettre les informations provenant des nœuds end_device et le dernier niveau end_device qui est chargé de transmettre les informations au destinataire sink ou il utilise un intermédiaire par la suite nous décrivons l'étude de notre réalisation.

La Fig III.14 montre la configuration du test de réseau multiniveaux que nous avons réalisé :



Figure III.15 *La mise en marche du mode multiniveaux*

L'objectif de cette application est de concevoir et d'implémenter une plateforme pour un réseau de capteur sans fil multiniveaux, donc nous avons utilisé un arduino UNO comme nœud de sink et nœud de router, ainsi nous avons fait cinq End_device où nous avons utilisé des arduino NANO.

Et pour simuler la transmission des échantillons entre ce réseau nous avons utilisé un compteur pour chaque nœud End_device comme l'expérience multicievers.

Nous avons utilisé un temps de transmission des données '' millis () '' pour chaque end_device.

1. Le nœud d'end_device n°1 a un échantillon chaque 1000ms.

2. Le nœud d'end_device n°2 a un échantillon chaque 1500ms.
3. Le nœud d'end_device n°3 a un échantillon chaque 2500ms.
4. Le nœud d'end_device n°4 a un échantillon chaque 3500ms.
5. Le nœud d'end_device n°5 a un échantillon chaque 4500ms.

Dans ce qui suit, nous allons montrer ce que nous avons obtenir au niveau de End_device, de router et de sink.

La figure suivante montre les messages de débogage relatifs au nœud d'end_device qui montre l'achèvement de l'émission paquets vers le nœud de routeur :

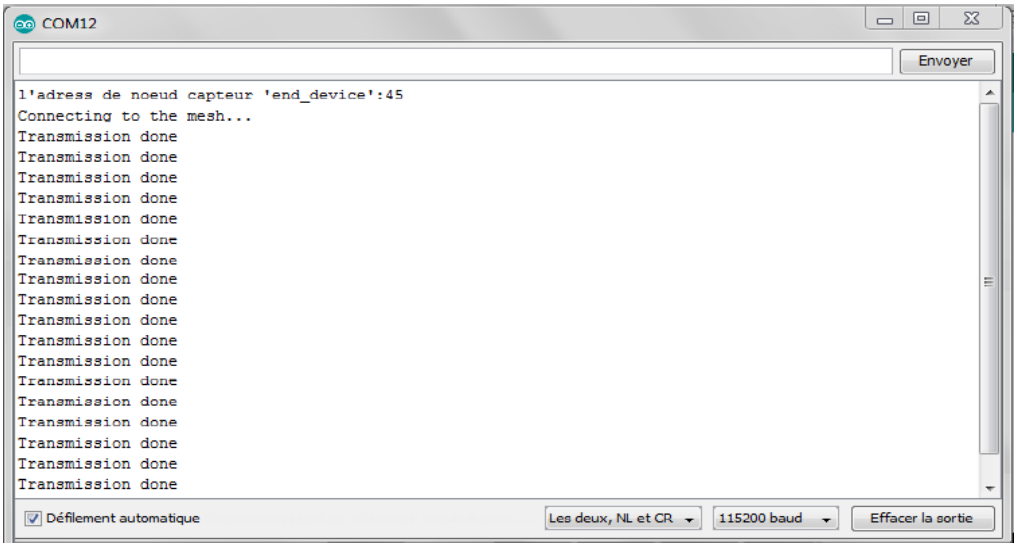


Figure III.16 *La fenêtre de débogage du nœud d'end_device***.**

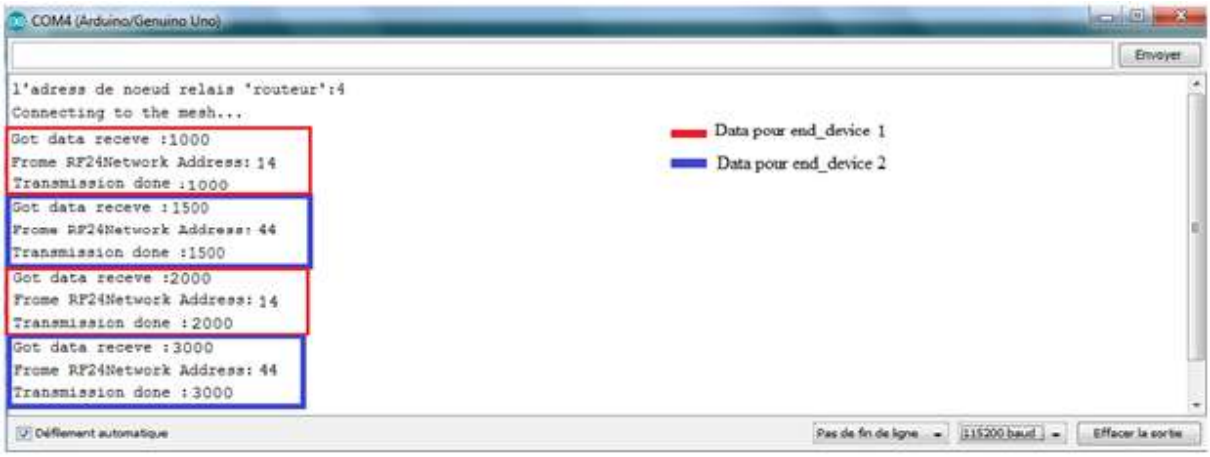


Figure III.17 *La fenêtre de débogage relative au nœud de routeur.*

Du coté du nœud routeur, un ensemble de data des déférents nœuds d'end_device a été reçu et transmis comme la Fig III.16 dans laquelle on peut clairement identifier et dénombrer les échantillons reçus et retransmis.

Du côté du nœud de base sink**,** un ensemble de data des différents nœuds de routeur a été reçu comme le montre la FigIII.17, dans laquelle on peut clairement identifier et dénombrer les échantillons reçus de la part d'End_device 1 qui a utilisé le routeur qui contient l'adresse 4:

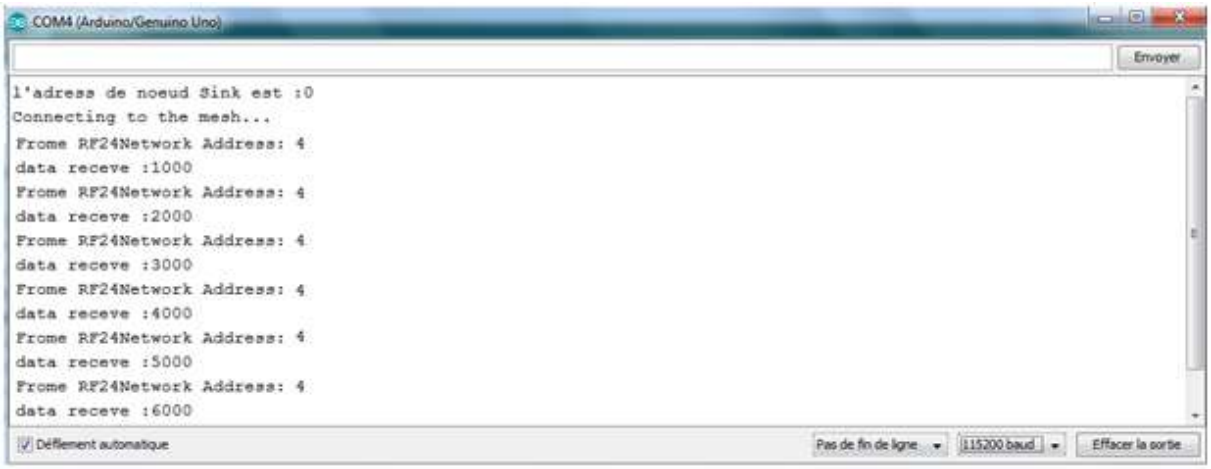


Figure III.18 *La fenêtre de débogage relative au nœud de base***.**

b.1.2 L'organigramme du nœud de sink (récepteur):

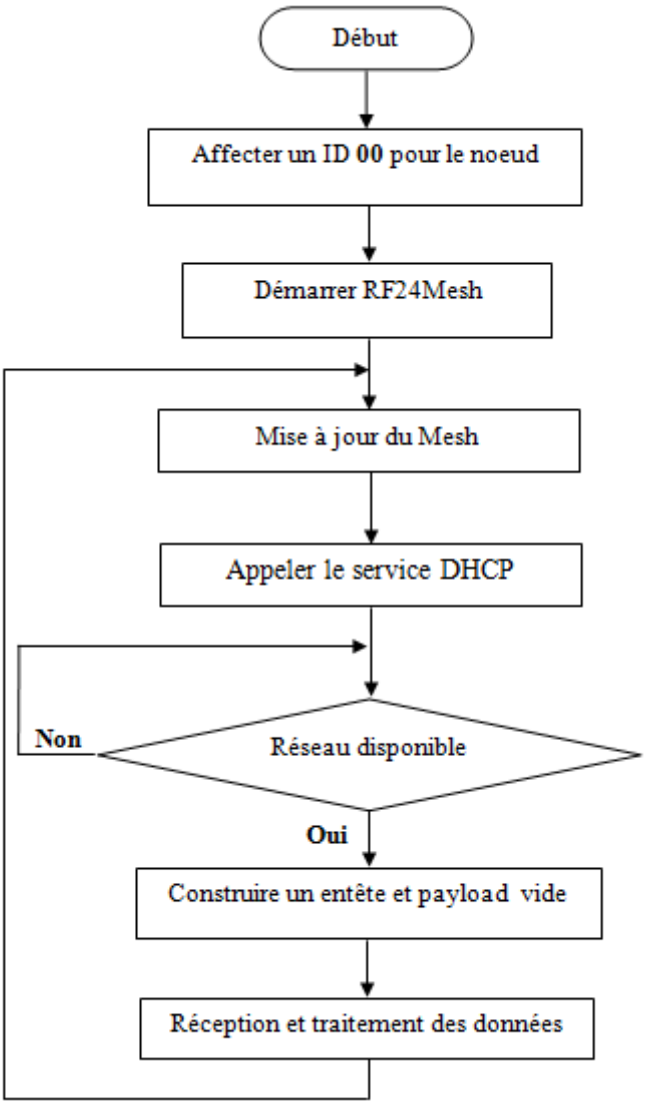


Figure III.19 *L'organigramme du nœud de sink*

Ce dernier a été traduit dans l'environnement arduino pour construire le programme du nœud reciever (Sink).

- Le code si dessous montre la façon avec laquelle nous avons utilisé les différentes bibliothèques avec une carte (uno et nano) :

```
#include "RF24Network.h"    //Bibliothèque Network
#include "RF24.h"
#include "RF24Mesh.h"       //Bibliothèque Mesh
#include <SPI.h>
//Include eeprom.h for AVR (Uno, Nano) etc. except ATTiny
#include <EEPROM.h>

/***** Configure the chosen CE,CS pins *****/
RF24 radio(9,10);
RF24Network network(radio);      //construction du reseau network ''notre reseau multiniveaux''
RF24Mesh mesh(radio,network);    //construction du mesh
```

Figure III.20 *Déclaration des différentes bibliothèques*

- Le code si dessous montre la façon avec laquelle nous avons appliqué l'organigramme sur le nœud Sink :

```
void setup() {
  Serial.begin(115200);
  mesh.setNodeID(0);
  Serial.println(mesh.getNodeID());
  Serial.println(F("Connecting to the mesh..."));
  mesh.begin();
}
void loop() {
  mesh.update();
  mesh.DHCP();
  if(network.available()){
    RF24NetworkHeader header;
    network.peek(header);
    uint32_t dataReceived=0;
    switch(header.type){
      case 'M': network.read(header,&dataReceived,sizeof(dataReceived));
      Serial.print("data receve :");
      Serial.println(dataReceived);
      Serial.print("Frome RF24Network Address: ");
      Serial.println(header.from_node, OCT); break;
      default: network.read(header,0,0); Serial.println(header.type);break;
    }
  }
}
```

Figure III.21 *Code d'arduino utilisé au niveau du sink*

Explication des classes :

- **mesh.setNodeID ( ) :** donner une adresse ou bien un ID (0 seulement pour le master nœud) pour le nœud puis Sink.
- **mesh.begin ( ) :** cette fonction permet de connecté au mesh
- **mesh.update ( ) :** Télécharger les derniers mise à jours pour la bibliothèque RF24Mesh
- **mesh.DHCP ( ) :** démarrer l service DHCP et gardez ce service en cours d'exécution sur le noeud maître afin que les adresses être affecté aux nœuds de capteurs.
- **network.available ( ) :** Testez s'il existe un message disponible pour ce noeud.les messages entrantes des capteurs.
- **network.read ( ) :** elle permet d'afficher les données reçue depuis les nœuds capteurs.

b.1.3 L'organigramme du nœud de routeur :

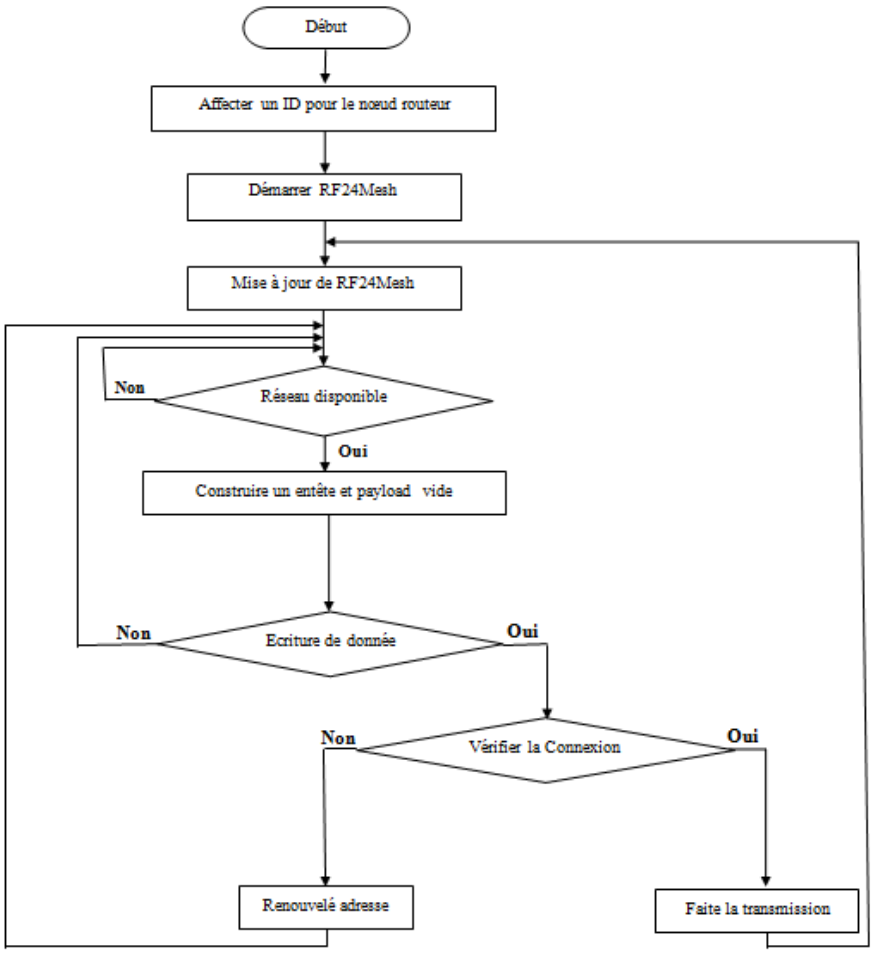


Figure III.22 *L'organigramme du nœud de routeur.*

```
void transmidata(){
  // Send to the master node every second
 if(!mesh.write(&dataReceived1,'M',sizeof(dataReceived1),otherNodeID)){
          Serial.println("Sending failed");
         if( ! mesh.checkConnection() ){
           Serial.println("Renewing Address");
           mesh.renewAddress();
         }else{
           Serial.println("Send fail, Test OK");
         }
          } else {
 if(millis() - dataReceived1 > 1000)
 {
     Serial.print("Transmission done :");
     Serial.println(dataReceived1);
     }
   }
     delay(1000);
 }
```

Figure III.23 *Partie de retransmission.*

Explication des classes :

- **mesh.write** : Construire automatiquement un en-tête et envoyer notre donnée à la fonction standard **network.write ()**, qui peut être utilisée directement. Nous avons inclue le paramètre nodeID, une recherche automatique d'adresse est effectuée.
- **checkConnection() :** Teste la connectivité de ce noeud au Mesh.

Si cette fonction échoue, la radio sera mise en veille et ne recevra aucune charge utile tant que l'adresse n'aura pas été renouvelée.

- **renewAddress()** : Reconnectez-vous au Mesh et renouvelez l'adresse RF24Network actuelle.

Utilisé pour rétablir une connexion au Mesh si l'emplacement physique a changé ou si un nœud de routage tombe en panne.

b.1.4 L'organigramme du nœud End_device

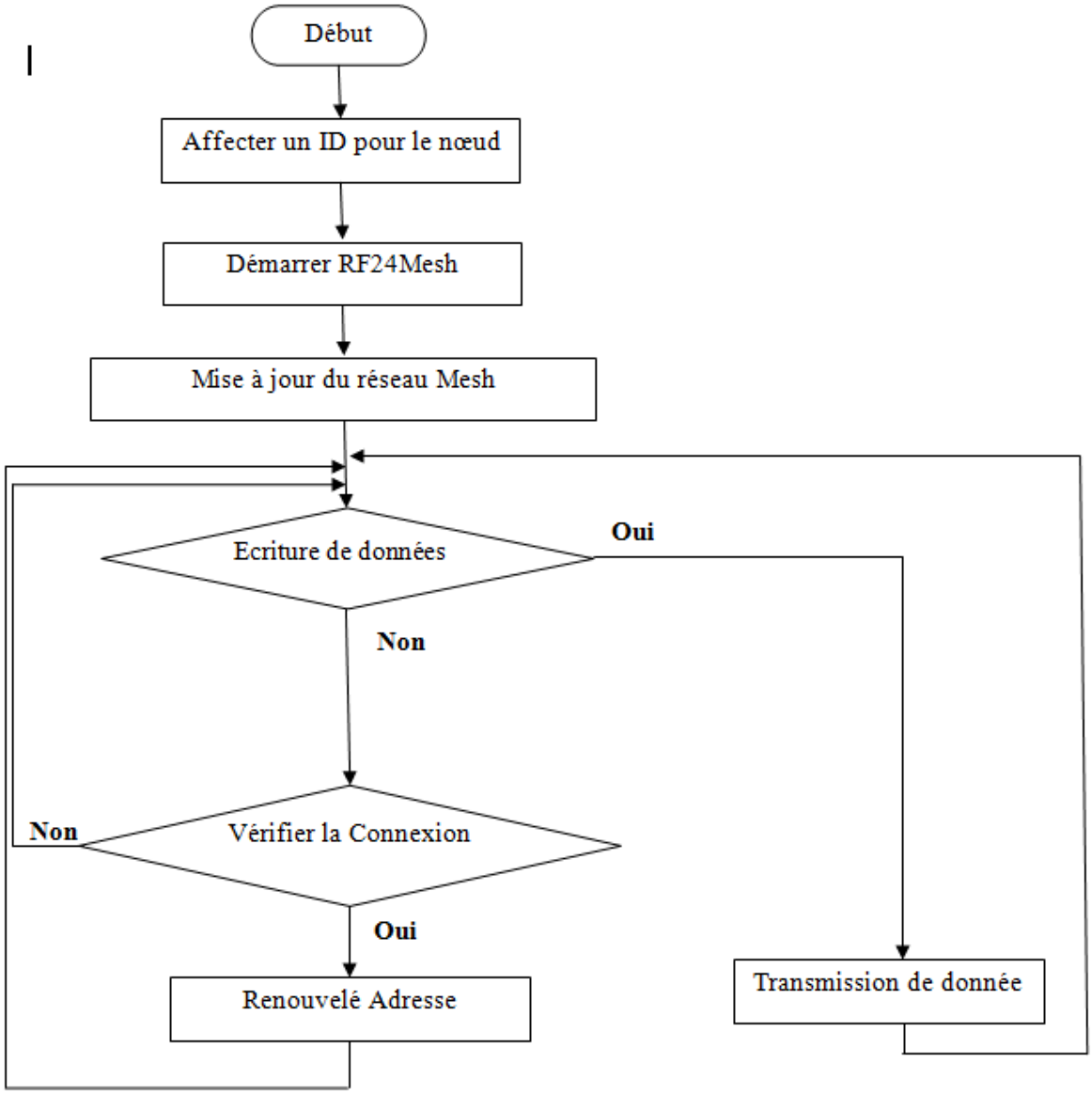


Figure III.24 *L'organigramme du nœud d'end_device*

```
void loop() {
  if (millis() - millisTimer >= 1000 ) {
    millisTimer = millis();
    if (!mesh.write(&millisTimer, 'M', sizeof(millisTimer), otherNodeID)) {
      Serial.println(F("Send fail"));
      if ( ! mesh.checkConnection() ) {
        Serial.println(F("Renewing Address"));
        mesh.renewAddress();
      } else {
        Serial.println(F("Send fail, Test OK"));
      }
    } else {
      Serial.print(F("Send OK: "));
      Serial.println(millisTimer);
       }
  }

}
```

Figure III.25 *Code d'arduino utilisé au niveau d'end_device*

III.4 conclusion

Dans ce travail nous avons développé une application multiceivers afin de crier un réseau de capteur sans fil multiniveaux, et la mise au point d'une transmission de données a travers une liaison sans fil, basé sur des modules arduino qui peuvent communiquer entre eux via des modules nrf24l01+, Il servira comme plateforme, et l'envoie des données, à la réception et les afficher sur un pc.

CONCLUSION GÉNÉRALE

Conclusion général

Dans ce projet nous avons réalisé et implémenté un réseau de capteur sans fil multi niveaux basé sur des composants simples comme l'arduino et le module nRF24L01+, la présence de la bibliothèque network et mesh sur la plateforme arduino facilite la tâche de programmation.

Lors des séances de l'étude et réalisation, et en menant notre projet, nous avons pu mettre en œuvre des informations et des connaissances reçues lors des cours d'électroniques et d'informatique ainsi on a pu traduire des études théoriques en études pratiques.

Ce projet s'est déroulé en trois étapes, qui sont brièvement :

- des nœuds d'end_device à base d'Arduino Nano et le module nrf24L01+ serrent a transmettre les données.
- des nœuds de routeur à base d'Arduino uno et le module nRF24L01+ assurent l'acheminement des données des nœuds de capteurs vers le nœud de sink.
- un nœud de sink à base d'Arduino uno et le module nRF24L01+ reçoit les data et les affiche sur un pc.

Nous avons pu réaliser un projet personnel de manière autonome, Ce projet nous a permis de rassembler des domaines différents car a été faite par transmission via une transmission RF. Nous avons beaucoup appris sur la programmation des modules.

BIBLIOGRAPHIE

Bibliographie

[1] Z. Bouzidi and A. Benameur, “Mise en place d’un réseau de capteurs sans fil pour l’irrigation intelligente,” Université Abou Bakr Belkaid– Tlemcen, 2012.

[2] Georges Asch et coll, *Les capteurs en Instrumentation Industrielle", Dunod, 7ème édition, 2010.* 1989.

[3] C. S. Pr and M. Yasmine, “Etude et réalisation d ’ un serveur WiFi d ’ acquisition de données,” Université Yahia Farés de MEDEA, 2017.

[4] M. H. DEHAL and M. l e. A. DILMI, “ETUDE ET IMPLEMENTATION D’UNE PLATEFORME DE COMMUNICATION ZIGBEE A BASE D’UN MICROCONTROLEUR ATMEGA,” UNIVERSITE YAHIA FARES DE MEDEA.

[5] K. Soufiane, “Conception d ’ un Réseau de Capteurs Sans Fil,” École normale supérieure d’enseignement technique - Rabat, 2015.

[6] M. B. Elmahdi and Melle. BENYAIHIA Dalal, “A s s e r v i s s e m e n t v i s u e l d ’ u n r o b o t m o b i l e,” UNIVERSITE YAHIA FARES DE MEDEA.

[7] A. Asalas and L. Lyes, “Conception et réalisation d’une application mobile cross-Platform pour l’Internet of things.,” Université A/Mira de Bejaia, Faculté des sciences exactes, département d’informatique, 2017.

[8] “La carte Arduino UNO Caractéristiques techniques : Détails techniques.”

[9] M. A. Baghdad and M. R. M. Lamine, “ETUDE ET REALISATION D’UN NOEUD DE CAPTEUR SANS FIL A BASE D’ARDUINO UNO,” Université Docteur Yahia Farès de MEDEA, 2014.

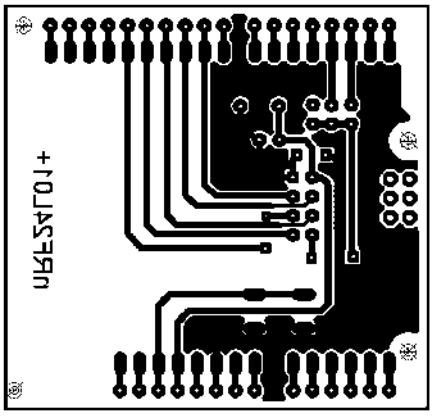
[10] O. A. Mohammed, “La Conception d’une prise connectée basée sur la technologie d’IoT (Internet of Things),” Université Sidi Mohamed Ben Abdellah, 2016.

[11] S. Cherifa and S. Boutheyna, “Surveillance des nourrissons par reseau de capteurs,” UNIVERSITÉ DE TLEMCEN, 2016.

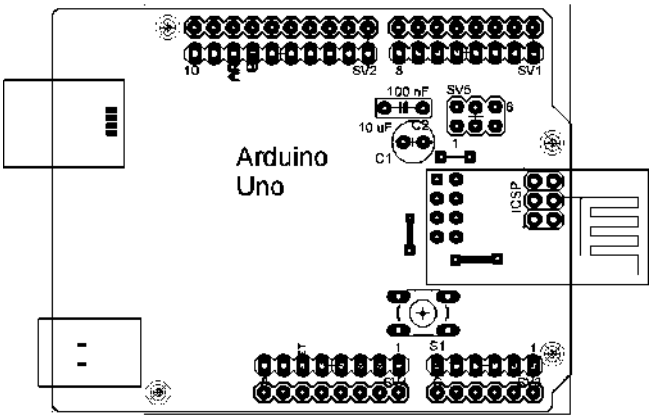
[12] “NRF24L01+ — Centre de Ressources Numériques - Labomedia.” .

Annexes

Annexe A :



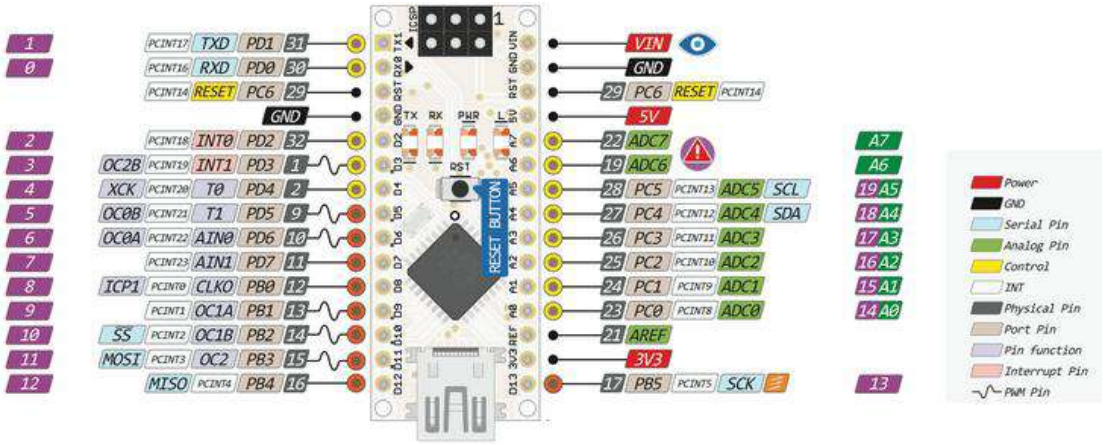
Le circuit imprimé du shield nrf24l01



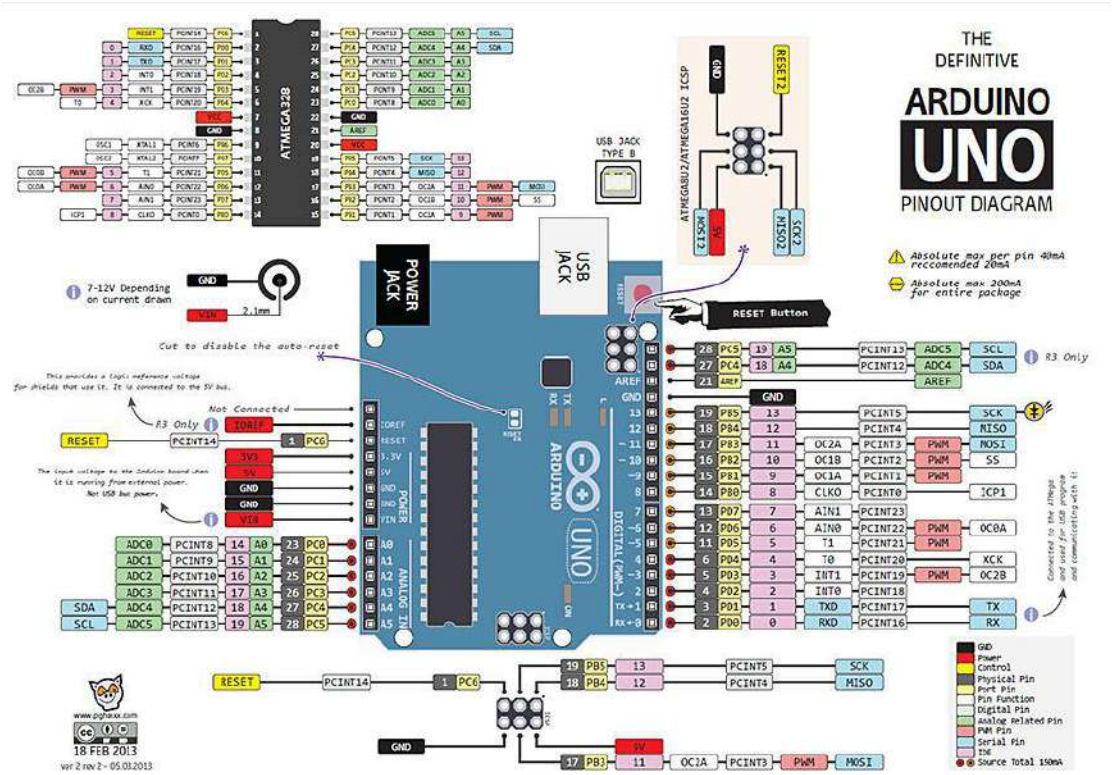
Le schéma d'implantation des composants

Annexe B :

- Le brochage de la carte Arduino Nano :



- Le brochage de la carte ArduinoUno :



Annexe C :

Datasheet de module nrf24l01+ :

1 Introduction

The nRF24L01+ is a single chip 2.4GHz transceiver with an embedded baseband protocol engine (Enhanced ShockBurst™), suitable for ultra low power wireless applications. The nRF24L01+ is designed for operation in the world wide ISM frequency band at 2.400 - 2.4835GHz.

To design a radio system with the nRF24L01+, you simply need an MCU (microcontroller) and a few external passive components.

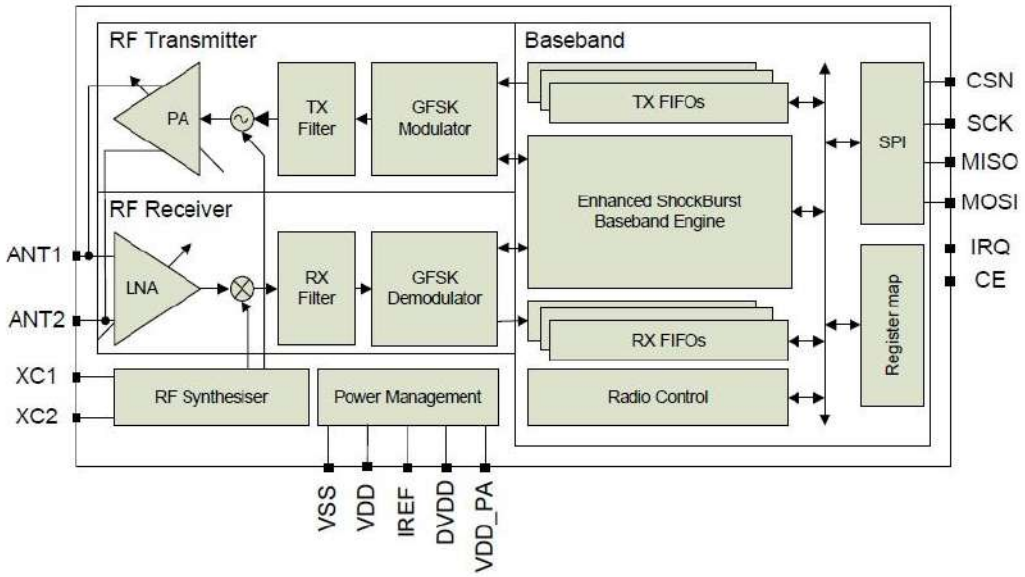
You can operate and configure the nRF24L01+ through a Serial Peripheral Interface (SPI). The register map, which is accessible through the SPI, contains all configuration registers in the nRF24L01+ and is accessible in all operation modes of the chip.

The embedded baseband protocol engine (Enhanced ShockBurst™) is based on packet communication and supports various modes from manual operation to advanced autonomous protocol operation. Internal FIFOs ensure a smooth data flow between the radio front end and the system's MCU. Enhanced ShockBurst™ reduces system cost by handling all the high speed link layer operations.

The radio front end uses GFSK modulation. It has user configurable parameters like frequency channel, output power and air data rate. nRF24L01+ supports an air data rate of 250 kbps, 1 Mbps and 2Mbps. The high air data rate combined with two power saving modes make the nRF24L01+ very suitable for ultra low power designs.

nRF24L01+ Preliminary Product Specification

1.2 Block diagram



1.1 Features

Features of the nRF24L01+ include:

- Radio
  - Worldwide 2.4GHz ISM band operation
  - 126 RF channels
  - Common RX and TX interface
  - GFSK modulation
  - 250kbps, 1 and 2Mbps air data rate
  - 1MHz non-overlapping channel spacing at 1Mbps
  - 2MHz non-overlapping channel spacing at 2Mbps
- Transmitter
  - Programmable output power: 0, -6, -12 or -18dBm
  - 11.3mA at 0dBm output power
- Receiver
  - Fast AGC for improved dynamic range
  - Integrated channel filters
  - 13.5mA at 2Mbps
  - -82dBm sensitivity at 2Mbps
  - -85dBm sensitivity at 1Mbps
  - -94dBm sensitivity at 250kbps
- RF Synthesizer
  - Fully integrated synthesizer
  - No external loop filer, VCO varactor diode or resonator
  - Accepts low cost ±60ppm 16MHz crystal
- Enhanced ShockBurst™
  - 1 to 32 bytes dynamic payload length
  - Automatic packet handling
  - Auto packet transaction handling
  - 6 data pipe MultiCeiver™ for 1:6 star networks
- Power Management
  - Integrated voltage regulator
  - 1.9 to 3.6V supply range
  - Idle modes with fast start-up times for advanced power management
  - 26µA Standby-I mode, 900nA power down mode
  - Max 1.5ms start-up from power down mode
  - Max 130us start-up from standby-I mode
- Host Interface
  - 4-pin hardware SPI
  - Max 10Mbps
  - 3 separate 32 bytes TX and RX FIFOs
  - 5V tolerant inputs
- Compact 20-pin 4x4mm QFN package

