الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne démocratique et populaire
Ministère de l'Enseignement Supérieur & de la Recherche Scientifique
Université Docteur Yahia Farès de MEDEA

Faculté des sciences et de la technologie
Département du Génie Electrique
Et de L'informatique

Projet de Fin d'études

Présenté pour l'obtention du diplôme

Master en Génie Électrique

Spécialité : *Systèmes de Télécommunications*

Thème :

ETUDE ET REALISATION D'UN NŒUD DE CAPTEUR SANS FIL MULTICANAL

Proposé et dirigé par :

M. BENGHERBIA Billel Et M. KARA Rida

Réalisé Par :

M. BABCHIKH Abderrahmane

M. LARBI Zakaria

Année universitaire2014-2015

Dédicace

C'est avec un très grand honneur que je dédie ce modeste travail aux personnes les plus chères au monde pour leurs sacrifices :

A Mes chères parents que dieu les gardes et préserve.

A Ma chère grande mère a qui je souhaite une longue vie.

A Mon frère : SMAIL et sans oublier mes amis.

A toute la famille « BABCHIKH» et « BENTERKIA».

A mon ami et mon binôme ZAKARIA qui a contribué à la réalisation de ce modeste travail.

Mes amis intimes: Hamza, Belkacem, Amine, youcef, Mehdi, Khaled, Mahfoud, Mohamed, Ayoub, Abdessalem, Moussa.

A Tous mes collègues…

ABDERRAHMANE

Dédicace

A mes parents,

A mes frères, et ma sœur,

Vous vous êtes dépensés pour moi sans compter.

En reconnaissance de tous les sacrifices consentis par tous et chacun pour me permettre d'attendre cette étape de ma vie.

…Avec toute ma tendresse.

A toute ma famille,

Vous avez de près ou de loin contribué à mon travail.

Affectueuse reconnaissance.

A mes camarades de L'Université.

Zakaria

Remerciements

Au terme de ce travail nous tenons à faire louange

A « ALLAH » qui nous a donné la force d'achever cette réalisation.

Nous tenons à exprimer notre profonde gratitude à notre promoteur : M. BENGHERBIA Billel pour ses conseils et son aide précieux et sans oubliez M. KARA Rida.

Nos sincères remerciements vont également à tous les enseignants du département de génie électrique & informatique qui nous ont offert tout ce savoir.

Nos gratitudes à tous les membres de jury qui font l'honneur de juger notre travail.

Un grand merci à tous ceux qui ont contribué de prés ou de loin pour l'élaboration de ce projet surtout : RAMLI Amine et DEHBINE Youcef et tous mes amis qui nous connaissent sans exception. On leur dit merci de tout notre cœur…

ABDERRAHMANE ET ZAKARIA.

ملخص:

الهدف من هذا المشروع هو تصميم و إنجاز عقدة استشعار لاسلكية متعددةالقنوات باستعمال عدة متحكمات أردوينو إينو في الإرسال ومستقبل واحد للتمكين من استقبال عدة معطيات(درجة الحرارة ، درجة الرطوبة، مستوى الماء ، التقاط الصوت ...)بواسطة أجهزة الاستشعار(لواقط) عادية. و نقل القيمة إلى محطة أساسية باستعمال الأداة+nRF24L01,وعرض النتائج على صفحة الانترنيت بواسطة درع الأردوينو .

كلمات مفتاحية: عقدة الاستشعار اللاسلكية , أرد وينو إينو, درع الأردوينو، لاقط.

Résumé :

L'objectif de ce projet est de concevoir et réaliser un nœud de capteur sans fil (RCSF) multi canal utilisant plusieurs contrôleurs d'Arduino Uno à l'émission avec un seul maitre à la réception, permettant l'acquisition de plusieurs paramètres la température, l'humidité, le niveau d'eau, capture de détection du son par des capteurs analogique (DHT11, capteur de niveau d'eau, capteur du son) et de transmettre les valeurs vers une station de base par voie RF en utilisant le nRF24L01+, et de les recevoir sur une page web à travers le shield d'Arduino.

Les mots clés : nœud de capteur sans fil, capteur, Arduino, nRF, shield.

Abstract:

The objective of this project is to design and implement a wireless sensor node using multiple Arduino Uno controllers, the issue with only one master at reception, allowing the acquisition of several parameters the temperature, humidity, water level, sound capture for analog sensors, and transmit the values to one RF path from base station using the nRF24L01+, and receive on a web page through the shield to Arduino.

Keywords: a wireless sensor node, sensor, Arduino, nRF.

TABLE DES MATIERES

LISTE DES FIGURES ..... **I**
LISTE DES TABLEAUX ..... **III**
LISTE DES ACRONYMES ..... **IV**
Introduction générale **1**
Chapitre I : Généralités **2**
I.1 Introduction 3
I.2 Les réseaux de capteurs sans fils 3
I.2.1 Définition 3
I.2.2 Le nœud de capteur sans fil 3
I.2.3 Architecture d'un nœud capteur [1] 3
I.2.4 Modèle d'un nœud [1] 5
I.2.5 Architecture de communication d'un RCSF 6
I.3 La transmission RF 6
I.3.1 Définition et principe générale [1] 6
I.3.2 L'importance de la transmission RF [1] 7
I.3.3 Domaines d'application des réseaux de capteurs sans fil 7
I.4 Conclusion 10
Chapitre II : Etude du système **12**
II.1 Introduction 13
II.2 Méthodes de transfert des données [2] 13
II.2.1 Transmission sérielle et parallèle 13
II.3 Protocole de communication série 16
II.3.1 Définition et principe générale 16
II.3.2 UART (Universal Asynchronous Receiver Transmitter). 16
II.3.3 Protocol I2C (Inter-Integrated Circuit) 17
II.3.4 Interface et le protocole de communication SPI 18
II.4 L'Arduino 22
II.4.1 Présentation de l'Arduino 22
II.4.2 Comparaison des modules microcontrôleurs ARDUINO [1] 23
II.4.3 La carte Arduino Uno [1] 24
II.4.4 La carte Arduino Ethernet [5] 28
II.4.5 Programmation de la carte Arduino Uno [1] 30
II.5 Le module RF [1] 31
II.5.1 Présentation du module nRF24L01 31

II.5.2 Caractéristiques techniques 32
II.6 Protocol ShockBurst [6] 32
II.6.1 Principe de ShockBurst [6] 33
II.6.2 Les organigrammes de ShockBurst d'émission et de réception 33
II.6.3 Description de paquet de données [6] 36
II.6.4 Le protocole Multi récepteur [7] 38
II.7 Les capteurs 40
II.7.1 Définition d'un capteur 40
II.7.2 Modèle d'un capteur 41
II.7.3 Le Capteur DHT 11 [8] 41
II.7.4 Le Capteur de niveau d'eau [9] 42
II.7.5 Le Capteur de détection de son [10] 43
II.8 Conclusion 43
Chapitre III : réalisation et mise en marche 44
III.1 Introduction 45
III.2 Schéma bloc du système 45
III.3 Réalisation Pratique 47
III.3.1 Le Mode émission 47
III.3.2 Le mode réception 50
III.3.3 Le mode d'actionneur 52
III.4 Résultats 55
III.5 Conclusion 55
Conclusion générale 56
Bibliographie 57

LISTE DES FIGURES

FIGURE I.1. LES COMPOSANTS D'UN NŒUD CAPTEUR.....4
FIGURE I.2. ARCHITECTURE DE COMMUNICATION D'UN RCSF. 6
FIGURE I.3. APPLICATIONS DES RCSF. 10
FIGURE II.1. LIAISON PARALLELE.....13
FIGURE II.2. LIAISON SERIE. 14
FIGURE II.3. LA TRANSFORMATION PARALLELE - SERIE.....15
FIGURE II.4. LA TRANSFORMATION SERIE – PARALLELE. 15
FIGURE II.5. LA COMMUNICATION D'UN UART. 17
FIGURE II.6. UNE TRAME SERIE D'UN UART. 17
FIGURE II.7. L'ARCHITECTURE D'UN BUS I2C. 17
FIGURE II.8. LA TRANSMISSION DES DONNEES SUR LE BUS I2C. 18
FIGURE II.9. EXEMPLE DE CONFIGURATION DE BUS SPI AVEC UN MAITRE ET PLUSIEURS ESCLAVES. 19
FIGURE II.10. TRANSMISSION DE DONNEES EN MAITRE ET ESCLAVE PAR REGISTRE A DECALAGE....20
FIGURE II.11. EXEMPLE D'UNE TRANSMISSION DE DONNEES SPI DE 1 OCTET POUR CPHA=0.....21
FIGURE II.12. EXEMPLE D'UNE TRANSMISSION DE DONNEES SPI DE 1 OCTET POUR CPHA=1.....22
FIGURE II.13. DIFFERENTS CARTE ARDUINO. 23
FIGURE II.14. DIFFERENTS ELEMENTS DE LA CARTE ARDUINO UNO. 25
FIGURE II.15. LE MICROCONTROLEUR ATMEGA328. 27
FIGURE II.16. LA CARTE ARDUINO ETHERNET. 29
FIGURE II.17. LOGICIEL ARDUINO. 31
FIGURE II.18. LA BARRE DE BOUTONS. 31
FIGURE II.19. LE MODULE NRF24L01.....32
FIGURE II.20. SYNCHRONISATION DES DONNEES AVEC MCU ET LA TRANSMISSION AVEC TECHNOLOGIE SHOCKBURST. 33
FIGURE II.21. LA CONSOMMATION DE COURANT AVEC ET SANS LA TECHNOLOGIE SHOCKBURST...33
FIGURE II.22. L'ORGANIGRAMME DE L'EMISSION. 34
FIGURE II.23. L'ORGANIGRAMME DE RECEPTION. 35
FIGURE II.24. TRAME DE PAQUET DE DONNEES. 36
FIGURE II.25. PRX UTILISANT MULTI RECEPTEUR. 38
FIGURE II.26. EXEMPLE DE L'ADRESSAGE DES PIPES DE DONNEE 0-5. 39
FIGURE II.27. EXEMPLE D'ADRESSAGE DES PIPES DE DONNEES DANS MULTI RECEPTEUR. 40
FIGURE II.28. CAPTEUR. 40
FIGURE II.29. LE CAPTEUR DHT11. 42
FIGURE II.30. LE CAPTEUR DE NIVEAU D'EAU. 42
FIGURE II.31. LE CAPTEUR DE DETECTION DU SON.....43
FIGURE III.1. SCHEMA BLOC DU RESEAU A REALISE. 46
FIGURE III.2. L'ORGANIGRAMME DE L'EMISSION. 47
FIGURE III.3. PARTIE EMISSION. 48
FIGURE III.4.A LE RESULTAT DE DHT11. 49
FIGURE III.4.B LE RESULTAT DU CAPTEUR DE NIVEAU D'EAU. 49
FIGURE III.4.C LE RESULTAT DU CAPTEUR DE DETECTION DU SON.....50

FIGURE III.5. *L'ORGANIGRAMME DE LA RECEPTION.* *51*
FIGURE III.6. *PARTIE RECEPTION.* *52*
FIGURE III.7. *L'ORGANIGRAMME DE L'ACTIONNEUR.* *53*
FIGURE III.8. *PARTIE D'ACTIONNEUR.* *54*
FIGURE III.9. *PARTIE EMISSION RECEPTION.* *54*
FIGURE III.10. *RESULTATS.* *55*

LISTE DES TABLEAUX

TABLEAU II.1. *MODES DE TRANSMISSION SPI*.....*20*
TABLEAU II.2. *TABLEAU COMPARATIFS DES CARACTERISTIQUES DES MODULES ARDUINO*. *23*
TABLEAU II.3. *DESCRIPTION DE PAQUET DE DONNEES*.....*37*

LISTE DES ACRONYMES



ACK	acknowledgement
ADC	Add Data Memory to ACC with Carry
ADC	Analog to Digital Converters
AVR	Advanced Virtual RISC
CRC	Cyclic Redundancy Check
CPHA	Clock Phase
CPOL	Clock Polarity
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile Communications
I2C	Inter-Integrated Circuit
I/O	Input Output
LED	light-emitting diode
LNA	Low Noise Amplifier.
MCU	Microcontrôleur Unit
MIC	Modulation par Impulsion
MISO	Master Input, Slave Output
MOSI	Master Output, Slave Input
PRX	récepteur primaire
PTX	émetteur primaire
PWM	Pulse Width Modulation
RCSF	Réseau de Capteurs Sans Fil
SCK	Serial Clock
SCL	Serial Clock Line
SDA	Serial Data Line
SPI	Serial Peripheral Interface

SS	Slave Select
TTL	Transistor–transistor logic
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
WLAN	Wireless Local Area Network

Introduction générale

Dans la vie courante, l'utilisation des nœuds de capteurs sans fil est de plus en plus demandée pour la supervision et l'observation des différents changements d'environnement. Un réseau de capteurs sans fil est composé d'un ensemble de terminaux ou ce qu'on appelle des nœuds capteurs qui peuvent communiquer via des liaisons radio, sans infrastructure fixe préalable. Le réseau devra fonctionner de façon autonome, sans intervention humaine.

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.

C'est dans ce cadre que s'insère notre projet, qui se base sur l'étude et la réalisation d'un nœud de capteur sans fil à base d'Arduino pour une transmission multicanal.

Arduino est une plate-forme de prototypage d'objets interactifs à usage créatif constituée d'une carte électronique et d'un environnement de programmation. Cet environnement matériel et logiciel permet à l'utilisateur de formuler ses projets par l'expérimentation directe à l'aide de nombreuses ressources disponibles en ligne. Pont tendu entre le monde réel et le monde numérique, Arduino permet d'étendre les capacités de relations humain/machine ou environnement/machine. Arduino est un projet en source ouverte : la communauté importante d'utilisateurs et de concepteurs permet à chacun de trouver les réponses à ses questions. La carte Arduino repose sur un circuit intégré (appelé également microcontrôleur) associée à des entrées et sorties qui permettent à l'utilisateur de brancher différents types d'éléments externes. [1]

Ce mémoire est structuré en trois chapitres ; le premier présente des concepts de bases de la transmission RF, des réseaux de capteurs sans fil et leurs applications, ainsi la description générale d'un nœud de capteur sans fil. Le deuxième traite la partie matérielle et les différents composants utilisés dans cette étude, aussi que la conception hardware du système. Tandis que le dernier chapitre décrit l'aspect software ou programmation associée, ainsi que la réalisation expérimentale du nœud de capteur.

Chapitre I : Généralités

Chapitre I : Généralité

I.1 Introduction

Les réseaux de capteurs sans fil (RCSF) c'est une nouvelle technologie qui a surgi après les grands progrès technologiques concernant le développement des capteurs intelligents. Les réseaux de capteurs sans fil sont constitués de nœuds disposant de peu d'énergie avec des capacités limitées de traitement et de transmission.

Dans ce chapitre nous allons présenter l'architecture générale d'un nœud de capteur sans fil, ainsi les différentes model de nœud de capteurs les plus rencontré en pratique.

I.2 Les réseaux de capteurs sans fils

I.2.1 Définition

Un Réseau de Capteurs Sans Fil (RCSF) est un ensemble de dispositifs très petits, nommés nœuds capteurs, variant de quelques dizaines d'éléments à plusieurs milliers. Dans ces réseaux, chaque nœud est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil. [1]

I.2.2 Le nœud de capteur sans fil

Un nœud de capteur sans fil est un petit dispositif électronique capable de mesurer une valeur physique environnementale (température, lumière, pression, humidité, vibration, etc.), et de la communiquer à un centre de contrôle via une station de base. Chaque capteur assure les trois principales fonctions de base qui sont : l'acquisition de données, les traitements sur ces données et leurs communications aux stations de bases [1].

I.2.3 Architecture d'un nœud capteur [1]

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.1)

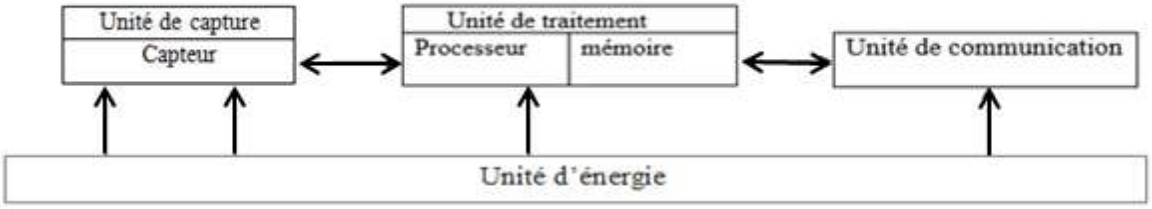


Figure I.1. *Les composants d'un nœud capteur.*

I.2.3.1 Unité de capture ou d'acquisition (Sensing unit)

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

I.2.3.2 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 stocke les données collectées.

I.2.3.3 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.

I.2.3.4 Unité d'énergie (Power unit)

C'est la batterie qui, n'est généralement ni rechargeable ni remplaçable. La capacité d'énergie limitée au niveau des capteurs représente la contrainte principale lors de conception de protocoles pour les réseaux de capteurs. Les unités d'énergie peuvent être supportées par des photopiles qui permettent de convertir l'énergie lumineuse en courant électrique. [1].

Il existe des capteurs qui sont dotés d'autres composants additionnels : les systèmes de localisation.

I.2.4 Types d'un nœud [1]

Selon l'application et la structure choisie, un RCSF peut contenir différents types de nœuds.

I.2.4.1 Un nœud puits

Est un nœud régulier doté d'un convertisseur série connecté à une seconde unité de communication (GPRS, Wi-Fi, WiMax, etc.). La seconde unité de communication fournit une retransmission transparente des données provenant de nœuds capteurs à un utilisateur final ou d'autres réseaux comme internet.

I.2.4.2 Un nœud passerelle (ou Gateway)

Est un nœud régulier permettant de relayer le trafic dans le réseau sur le même canal de communication.

Pour optimiser certains paramètres comme la durée de vie du réseau ou le délai de livraison des données, certains travaux se sont focalisés sur l'architecture (plat, hiérarchique, multi niveaux) des RCSF. Ces architectures définissent le plus souvent les rôles joués par les nœuds dans un RCSF. Nous distinguons principalement 3 rôles à savoir :

I.2.4.3 Nœud Source (NS)

Dont le rôle principal est de détecter les phénomènes physiques ou physiologiques se produisant dans son environnement immédiat afin de les transmettre, directement ou via multiples sauts, à un utilisateur final. C'est en fait un nœud capteur.

I.2.4.4 Nœud Relais (NR)

Ils ont pour rôle d'agréger et de retransmettre les mesures provenant des NS afin que celles-ci parviennent à un utilisateur final. Dans une architecture à plat, quelques travaux considèrent généralement un NS comme un NR.

Dans une architecture à 2 niveaux, un nœud passerelle joue le rôle de NR pour un ou plusieurs nœuds sources. Dans une telle configuration réseau, la capacité de transmission du NR est supposée généralement plus grande que celle du NS.

I.2.4.5 Nœud Collecteur (NC)

Ils ont pour rôle de collecter les mesures provenant des nœuds sources et éventuellement de les agréger. Généralement, un "Cluster-Head" ou chef de cluster, est utilisé comme NC dans une architecture hiérarchique où les NS sont partitionnés en plusieurs groupes.

I.2.5 Architecture de communication d'un RCSF

Les nœuds capteurs sont habituellement dispersés dans une zone de capture. Chacun de ces nœuds a la possibilité de collectées les données et de les router vers une ou plusieurs stations de base (sinknodes). Ce dernier est un point de collecte de données capturées. Il peut communiquer les données collectées à l'utilisateur final à travers un réseau de communication, éventuellement l'Internet. L'utilisateur peut à son tour utiliser la station de base comme passerelle, afin de transmettre ses requêtes au réseau. Cette architecture est illustrée dans la figure.[1]

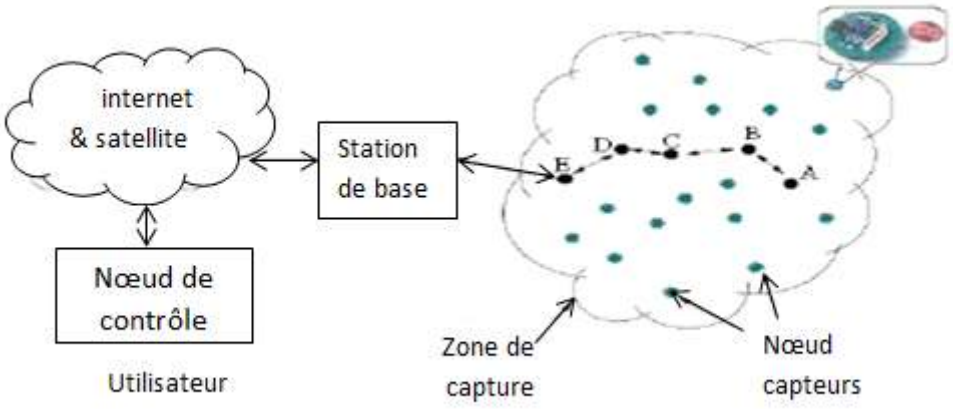


Figure I.2. *Architecture de communication d'un RCSF.*

I.3 La transmission RF

I.3.1 Définition et principe générale [1]

Le terme « RF » fait généralement référence à des communications de données par voie radio fréquence. Les Réseaux locaux sans fil (WLAN) sont aujourd'hui une forme très populaire de la transmission RF.

Le terminal mobile d'un système GSM est un autre exemple très courant des systèmes RF, ce dernier fait le lien (interface) entre l'utilisateur et le système de téléphonie mobile, il a pour rôle de recueillir les données à envoyer (la parole) et de recevoir les données transmis par le réseau par voie Radio .

Le terminal a également une fonction d'émetteur et de récepteur en même temps. Les technologies dites hertziennes ou bien radio, sont basées sur la radiodiffusion, ou des ondes électromagnétiques portant des signaux à transmettre sont émises dans l'air.

Physiquement, une onde électromagnétique correspond à une variation régulière de la valeur du champ électromagnétique en un point donné. Le nombre de variations par seconde est la fréquence de l'onde (exprimée en Hertz) .

Donc la RF est la transmission de données sans fil par des signaux radio numérique ou analogiques à une fréquence particulière .

I.3.2 L'importance de la transmission RF [1]

Les avantages d'un système de communication RF sont très nombreux, d'un côté, par le simple fait que la transmission de données est sans fil, ceci ne nécessite pas de poser des câbles pour disposer d'une installation ou d'un réseau de transmission.

Ces câbles sont dans la plus part des temps cher, moins flexibles (que la couverture RF) et sont vulnérables aux dommages. Pour les installations nouvelles, mettre en œuvre une infrastructure sans fil peut être plus économique que de faire dérouler des kilomètres de câbles à travers un environnement donnée surtout si la configuration spatiale est complexe.

L'accessibilité à l'information est un autre avantage, dans la mesure où la transmission RF permet aux utilisateurs d'être à la portée du système, ils sont toujours en contact avec leurs données d'une manière plus simple et plus libre. Cet avantage permet d'éviter la rupture de la productivité et les décisions cruciales peuvent être faites et des au point de l'activité. Moins de temps perdu signifie une gestion immédiate et efficace sans ajouter de personnel supplémentaire.

I.3.3 Domaines d'application des réseaux de capteurs sans fil

Les RCSF peuvent avoir beaucoup d'applications (figure I.3). Parmi elles, nous citons :

I.3.3.1 Applications militaires

Le déploiement rapide, l'auto-configuration et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui font de ce type de réseaux un outil appréciable dans un tel domaine. Déploiement sur un endroit stratégique ou difficile d'accès, afin de surveiller toutes les activités des forces ennemies ou d'analyser le terrain avant d'y envoyer des troupes (par la détection d'agents chimiques, biologiques ou de radiations, par exemple) [1].

I.3.3.2 Applications liées à la sécurité

Les altérations dans la structure d'un bâtiment, suite à un séisme ou au vieillissement, pourraient être détectées par des capteurs intégrés dans les murs ou dans le béton, sans alimentation électrique ou autres connexions filaires. Les capteurs doivent s'activer périodiquement et peuvent ainsi fonctionner durant des années, voire des décennies.

Un réseau de capteurs de mouvements peut constituer un système d'alarme distribué qui servira à détecter les intrusions sur un large secteur [1].

Déconnecter le système ne serait plus aussi simple, puisqu'il n'existe pas de point critique. La surveillance de voies ferrées pour prévenir des accidents avec des animaux et des êtres humains peut être une application intéressante des réseaux de capteurs. La protection des barrages pourrait être accomplie en y introduisant des capteurs. La détection prompte de fuites d'eau permettrait d'éviter des dégâts. Les êtres humains sont conscients des risques et attaques qui les menacent. Du coup, ils mettent à disposition toutes les ressources humaines et financières nécessaires pour leur sécurité.

Cependant, des failles sont toujours présentes dans les mécanismes de sécurisation appliqués aujourd'hui, sans oublier leur coût très élevé. L'application des réseaux de capteurs dans le domaine de la sécurité pourrait diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et à la protection des êtres humains tout en garantissant de meilleurs résultats [1].

I.3.3.3 Applications environnementales

Des thermo-capteurs dispersés à partir d'un avion sur une forêt peuvent signaler un éventuel début d'incendie dans le champ de captage; ce qui permettra une meilleure efficacité pour la lutte contre les feux de forêt. Dans les champs agricoles, les capteurs peuvent être semés avec les graines. Ainsi, les zones sèches seront facilement identifiées et l'irrigation sera donc plus efficace. Sur les sites industriels, les centrales nucléaires ou dans les pétroliers, des capteurs peuvent être déployés pour détecter des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.) et alerter les utilisateurs dans un délai suffisamment court pour permettre une intervention efficace. Une grande quantité de capteurs peut être déployée en forêt ou dans un environnement de conservation de la faune afin de recueillir des informations diverses sur l'état du milieu naturel et sur les comportements de déplacement.

Par exemple, l'université de Pise en Italie a réalisé des réseaux de capteurs pour le contrôle des parcs naturels (feux, animaux,..). Il est ainsi possible "d'observer", sans déranger, des espèces animales difficiles à étudier dans leur environnement naturel et de proposer des solutions plus efficaces pour la conservation de la faune. Les éventuelles conséquences de la dispersion en masse des micro-capteurs dans l'environnement ont soulevé plusieurs inquiétudes.

En effet, chaque micro-capteur est doté d'une batterie qui contient des métaux nocifs. Néanmoins, le déploiement d'un million de capteurs de 1 millimètre cube chacun ne représente qu'un volume total d'un litre. Même si tout ce volume était constitué de batteries, cela n'aurait pas des répercussions désastreuses sur l'environnement.

Dans les immeubles, le système de climatisation peut être conçu en intégrant plusieurs micro-capteurs dans les tuiles du plancher et les meubles. Ainsi, La climatisation pourra être déclenchée seulement aux endroits où il y a des personnes présentes et seulement si c'est nécessaire. Le système distribué pourra aussi maintenir une température homogène dans les pièces. Utilisée à grande échelle, une telle application permettrait de réduire la demande mondiale en énergie réduisant du même coup les gaz à effet de serre. Rien que pour les États-Unis, on estime cette économie à 55 milliards de dollars par an avec une diminution de 35 millions de tonnes des émissions de carbone dans l'air. Ainsi, dans un contexte mondial où le réchauffement de la planète devient une préoccupation grandissante, une telle conséquence environnementale serait un pas dans la bonne direction [1].

I.3.3.4 Applications médicales

Surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité facilitant ainsi le diagnostic de maladies grâce à des micro-capteurs qui pourront être ingères ou implantés sous la peau.

- les micros-cameras qui peuvent être ingérées et sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain,
- la création d'une rétine artificielle composée d'une centaine de micro-capteurs pour améliorer la vision de l'œil [1].

I.3.3.5 Applications écologiques

L'intégration de plusieurs micro-capteurs dans le système de climatisation et de chauffage des immeubles. Ainsi, la climatisation ou le chauffage ne sont déclenches qu'aux endroits où il y a des personnes présentes et seulement si c'est nécessaire. Le système distribué peut aussi maintenir une température homogène dans les pièces. Utilisée à grande échelle, une telle application permettrait probablement de réduire la demande mondiale en énergie [1].

I.3.3.6 Applications de traçabilité et de localisation

Suite à une avalanche il est nécessaire de localiser les victimes enterrées sous la neige en équipant les personnes susceptibles de se trouver dans des zones à risque par des capteurs. Ainsi, les ´équipes de sauvetage peuvent localiser plus facilement les victimes. Contrairement aux solutions de traçabilité et de localisation basées sur le système de GPS (Global Positionning System), les réseaux de capteurs peuvent être très utiles dans des endroits clos comme les mines par exemple. [1]

I.3.3.7 Applications commerciales

Il est possible d'intégrer des nœuds capteurs au processus de stockage et de livraison. Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la position actuelle du paquet. Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré [1].

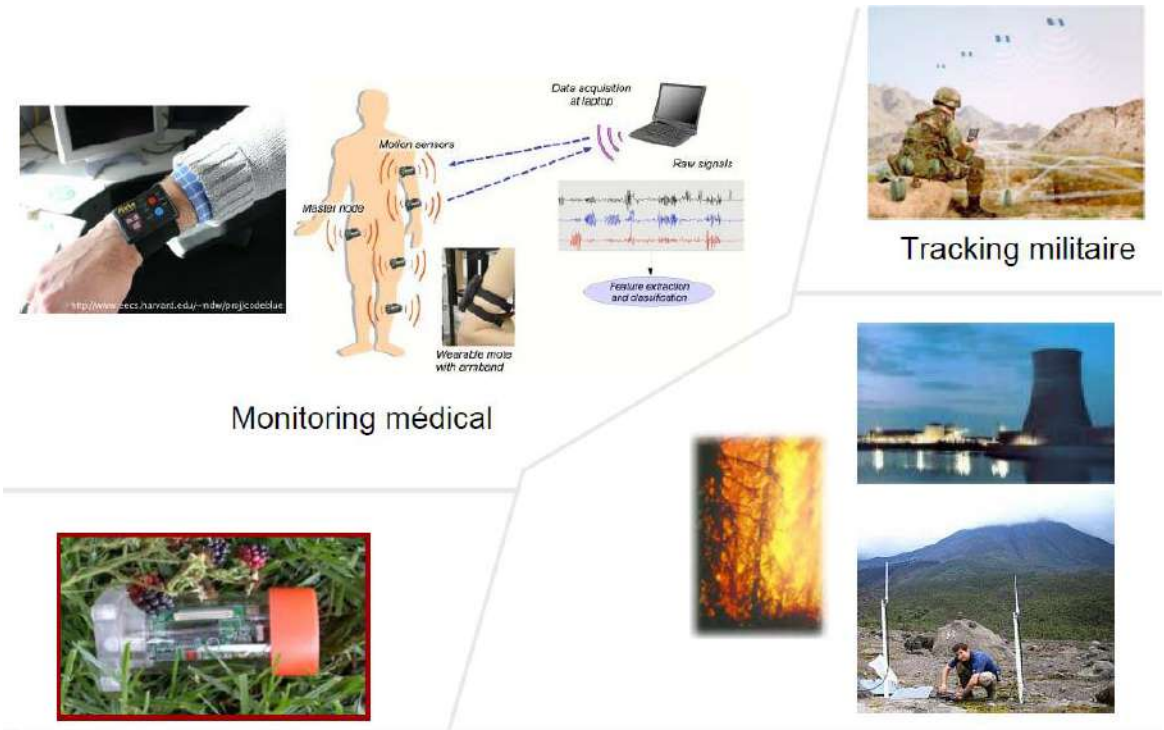


Figure I.3. *Applications des RCSF.*

I.4 Conclusion

Les réseaux de capteurs sans fil présentent un intérêt considérable et une nouvelle étape dans l'évolution des technologies de l'information et de la communication.

Cet intérêt vient d'un part, La diversité actuelle des solutions RF proposées dans le commerce, Et d'autre part de la diversité et la simplicité des systèmes de traitement (Microcontrôleurs simple et puissant).

Ceci , nécessite de la part du concepteur une étude approfondie lors de la mise au point d'un système de nœuds de capteur sans fil, afin de satisfaire les critères de fiabilité, de couts, de sécurité et autres.

Chapitre II : Etude Du Système

Chapitre II : Etude du système

II.1 Introduction

Dans ce chapitre nous allons présenter les méthodes de transfert de données, ainsi les protocoles de communication série, et la mise au point d'un réseau de transmission de données par voie RF. Il servira comme plateforme pour un petit réseau de capteur sans fil (RCSF), notre nœud est réalisé autour d'un module RF de type nRF24L01+ et des carte de type Arduino UNO, ces modules ont la particularité d'être simple à utiliser et le coût réduit de ce type de module.

II.2 Méthodes de transfert des données [2]

II.2.1 Transmission sérielle et parallèle

Le mode de transmission désigne le nombre d'unités élémentaires d'informations (bits) pouvant être simultanément transmises par le canal de communication. En effet, un processeur (donc l'ordinateur en général) ne traite jamais (dans le cas des processeurs récents).

Un seul bit à la fois, il permet généralement d'en traiter plusieurs (la plupart du temps 8, soit un octet), c'est la raison pour laquelle la liaison de base sur un ordinateur est une liaison parallèle.

II.2.1.1 Liaison parallèle

On désigne par liaison parallèle la transmission simultanée de N bits. Ces bits sont envoyés simultanément sur N voies différentes (une voie étant par exemple un fil, un câble ou tout autre support physique). La liaison parallèle des ordinateurs de type PC nécessite généralement 10 fils.

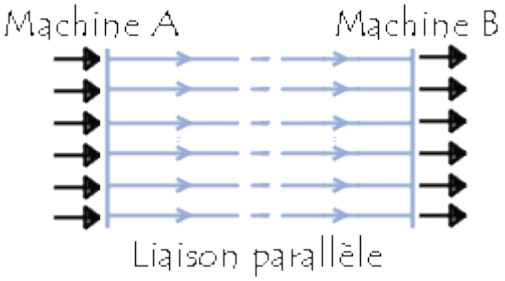


Figure II.1. *Liaison parallèle.*

Ces voies peuvent être :

- N lignes physiques: auquel cas chaque bit est envoyé sur une ligne physique (c'est la raison pour laquelle les câbles parallèles sont composés de plusieurs fils en nappe).
- Une ligne physique divisée en plusieurs sous-canaux par division de la bande passante.
- Ainsi chaque bit est transmis sur une fréquence différente...

Etant donné que les fils conducteurs sont proches sur une nappe, il existe des perturbations (notamment à haut débit) dégradant la qualité du signal...

II.2.1.1.1 Avantage :

- ❖ débit élevé de données (vitesse).

II.2.1.1.2 Inconvénients :

- ❖ pas fiables sur des longues distances.
- ❖ les données peuvent arriver à des moments différents.

II.2.1.2 Liaison série

Dans une liaison en série, les données sont envoyées bit par bit sur la voie de transmission. Toutefois, étant donné que la plupart des processeurs traitent les informations de façon parallèle, il s'agit de transformer des données arrivant de façon parallèle en données en série au niveau de l'émetteur, et inversement au niveau du récepteur.

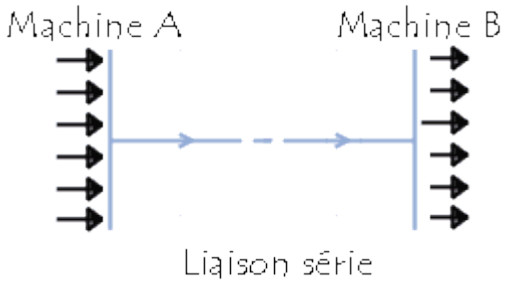


***Figure II.2**. Liaison série.*

II.2.1.2.1 Avantages :

- ❖ Utilisé sur des longues distances.
- ❖ L’exigence réduite de câblage.

II.2.1.2.2 Inconvénients :

- La transmission des données est plus lente.

Ces opérations sont réalisées grâce à un contrôleur de communication (la plupart du temps une puce UART, Universal Asynchronous Receiver Transmitter). Le contrôleur de communication fonctionne de la façon suivante :

- ***La transformation parallèle-série :***

Se fait grâce à un registre Le registre de décalage permet, grâce à une horloge, de décaler le registre (l'ensemble des données présentes en parallèle) d'une position à gauche, puis d'émettre le bit de poids fort (celui le plus à gauche) et ainsi de suite :

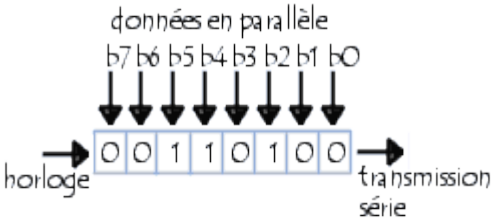


***Figure II.3**. La transformation parallèle - série*

- ***La transformation série-parallèle :***

Se fait quasiment de la même façon grâce au registre de décalage. Le registre de décalage permet de décaler le registre d'une position à gauche à chaque réception d'un bit, puis d'émettre la totalité du registre en parallèle lorsque celui-ci est plein et ainsi de suite.

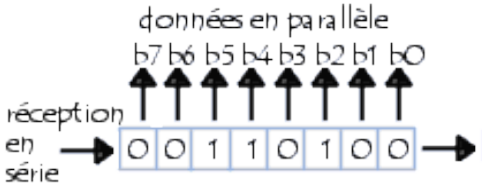


***Figure II.4**. La transformation série – parallèle.*

II.3 Protocole de communication série

II.3.1 Définition et principe générale

La communication série consiste à transmettre des informations après les avoir préalablement découpées en plusieurs morceaux de taille fixe (cette taille est le nombre de lignes de communication disponibles). Elle s'oppose à la communication parallèle où les informations à transmettre n'ont pas à être découpées avant d'être envoyées (car il y a au moins autant de lignes de communication disponibles que de bits pour transmettre l'information).

Les protocoles série sont :

- UART (Universal Asynchronous Receiver Transmitter).
- I2C (Inter-Integrated Circuit) ou TWI (Tow Wire Interface).
- SPI (serial peripheral interface).

II.3.2 UART (Universal Asynchronous Receiver Transmitter).

L'UART (Universal Asynchronous Receiver Transmitter) est un sous-système complet disponible en un seul circuit intégré LSI et conçu pour la transmission et la réception de données asynchrones en mode duplex ou semi-duplex. L'émetteur de l'UART accepte les caractères en parallèle et les convertit en une suite binaire sérielle. Le récepteur convertit les données sérielles en données parallèles. En plus de ses fonctions de conversion, l'UART remplit également des fonctions de contrôle et de commande.

L'UART ajoute aux données transmises le bit de départ (START) et les bits d'arrêt (STOP). En outre, c'est un circuit programmable qui permet de sélectionner :

la longueur du mot binaire (5, 6, 7 ou 8 bits).

le type de parité (paire, impaire ou sans parité).

le nombre de bits d'arrêt (1, 1,5 ou 2).

L'UART permet aussi de détecter les erreurs de parité, de format et de débordement.

Enfin, un circuit UART peut fonctionner avec deux horloges séparées : une pour l'émetteur et une pour le récepteur. Cela permettra de transmettre et de recevoir à deux vitesses différentes, en cas de besoin. [3]

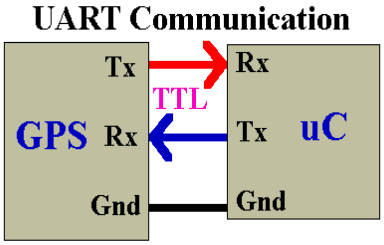


Figure II.5. *La communication d'un UART.*



Figure II.6. *Une trame série d'un UART.*

II.3.3 Protocol I2C (Inter-Integrated Circuit)

II.3.3.1 Présentation

Le bus I2c est caractérisé par une liaison en mode série réalisée à l'aide de 2 fils. C'est la société Philips qui en a créé le concept au début des années 80. Son succès est lié à sa simplicité. Voici l'architecture type d'un bus I2C est donné sur la figure II.7. les données transitent par les lignes :

- SDA : signal de donnée, généré par le Maître ou l'Esclave.
- SCL : signal d'horloge généré par le Maître.

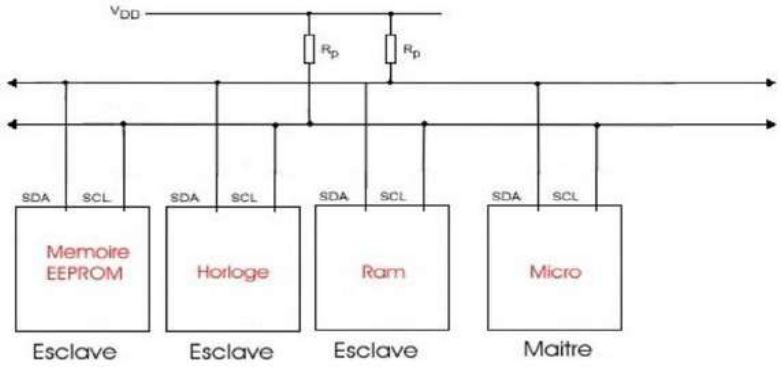


Figure II.7*. L'architecture d'un bus I2C.*

La communication sur le bus est orchestrée de manière suivante :

- Le Maître envoie sur le bus l'adresse du composant avec qu'il souhaite communiquer, chacun des esclaves ayant une adresse fixe.
- L'esclave qui reconnaît son adresse répond à son tour par un signal de confirmation, puis le Maître continue la procédure de communication... .(écriture/lecture).
- Dans tous les cas, les transactions seront confirmées par un ACK.

II.3.3.2 Prise de contrôle du bus

- Le bus doit être au repos ; à savoir SDA et SCL à 1.
- Pour transmettre des données il faut surveiller :

- la condition de départ : SDA à 0 et SCL reste à 1.

- la condition d'arrêt : SDA passe à 1 et SCL reste à 1.

- Après avoir vérifié que le bus est libre puis pris le contrôle de celui-ci, le circuit en devient le maître ; c'est lui qui génère le signal d'horloge. [4]

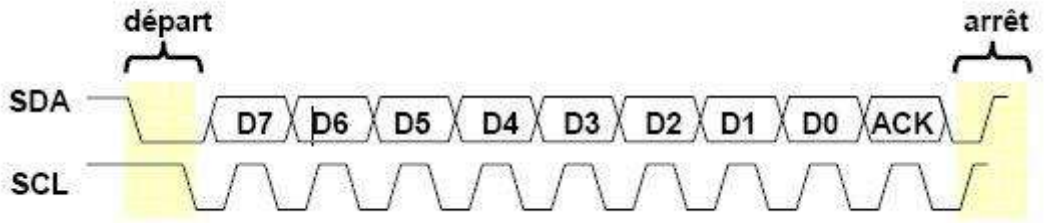


***Figure II.8**. La transmission des données sur le bus I2C.*

II.3.4 Interface et le protocole de communication SPI

II.3.4.1 La liaison SPI (Serial Peripheral Interface)[1]

La liaison SPI (Serial Peripheral Interface) est un bus pour la transmission synchrone de données « série » entre un maître et un ou plusieurs esclaves (en théorie, il n'y a pas de limitations dans le nombre d'esclaves). La transmission a lieu en Full Duplex, c'est-à-dire que les données circulent simultanément dans les deux sens. Le maître génère l'horloge et initialise une transmission de données en sélectionnant l'esclave avec qu'il veut communiquer.

Chaque esclave est adressé par le maître par une ligne individuelle SS (Slave Select) et n'est actif que lorsqu'il est sélectionné.

Le bus SPI est composé de deux lignes de signal et deux lignes de données:

- Slave Select (SS).
- Horloge (SCK).
- Master Output, Slave Input (MOSI).
- Master Input, Slave Output (MISO).

Toutes ces lignes sont unidirectionnelles.

II.3.4.2 Configuration du bus [1]

II.3.4.2.1 SCK (Horloge)

L'horloge est générée par le maître et sert à la synchronisation de tous les appareils connectés.

II.3.4.2.2 SS (Slave Select)

Chaque esclave est adressé par le maître par une ligne individuelle SS. Le maître sélectionne un esclave donné en mettant sa ligne SS au niveau BAS. Le nombre possible d'esclaves au sein d'un tel système de bus n'est limité que par le nombre possible de raccordements SS du maître.

II.3.4.2.3 MOSI (Master Output - Slave Input)

Sur la ligne MOSI le maître transmet des données à l'esclave.

II.3.4.2.4 MISO (Master Input - Slave Output)

Sur la ligne MISO l'esclave transmet des données au maître.

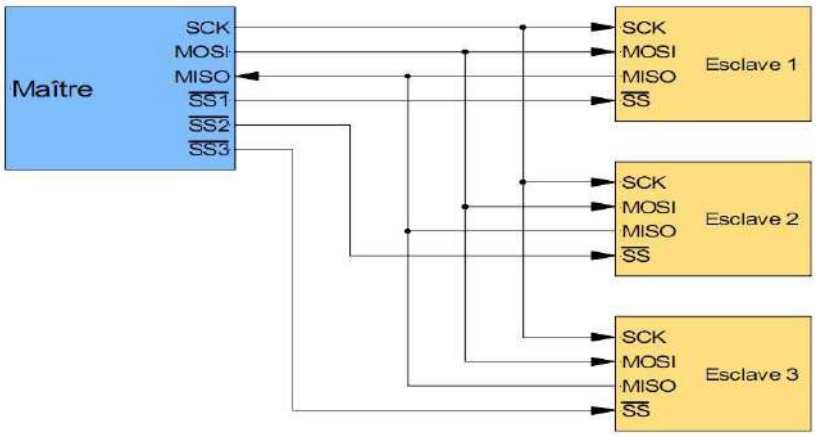


***Figure II.9**. Exemple de configuration de bus SPI avec un maitre et plusieurs esclaves.*

II.3.4.3 Protocole du bus SPI [1]

II.3.4.3.1 Transmission de données

Pour démarrer une transmission de données, le maître sélectionne un esclave en mettant sa ligne SS respective au niveau BAS. Ensuite, le maître transfert ses données dans le registre de sortie et rapidement après il commence à générer le signal d'horloge.

La transmission de données a lieu en Full Duplex en utilisant des registres à décalage aussi bien par le maître que par l'esclave. Avec chaque coup d'horloge le maître envoie sur la ligne MOSI une donnée à l'esclave pendant que l'esclave envoie simultanément une donnée au maître sur la ligne MISO. La transmission de données est terminée dès que la ligne SS est remise au niveau HAUT.

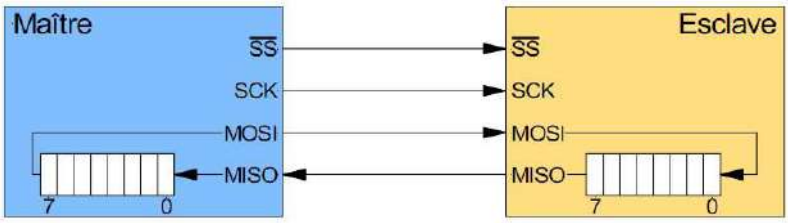


***Figure II.10**. Transmission de données en maitre et esclave par registre à décalage.*

II.3.4.3.2 Mode de service

Le protocole bus SPI détermine le type de signal d'horloge (SCK) grâce à deux paramètres: le bit de poids fort CPOL (Clock Polarity) et le bit de poids faible CPHA (Clock Phase). Ils en résultent quatre différents modes de transmission comme indiqués en Tab II .1

Ces paramètres de réglage déterminent sur quel front d'horloge les données sont transmises, et à quel moment ces données peuvent être modifiées. Pour une transmission de données correcte il est important que ces paramètres soient réglés de manière identique pour tous les appareils reliés au bus.

***Tableau II.1**. Modes de transmission SPI*

Mode SPI	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

II.3.4.3.3 CPOL (ClockPolarity)

Le CPOL détermine si au repos l'horloge est sur le niveau BAS (CPOL=0) ou HAUT (CPOL=1).

II.3.4.3.4 CPHA (Clock Phase)

Le CPHA détermine à quel front de l'horloge les données sont transmises. CPHA=0 signifie que les données sont valides au premier front d'horloge, CPHA=1 qu'elles sont valides au deuxième front.

II.3.4.4 Format du transfert de données [1]

II.3.4.4.1 CPHA = 0

Lorsque CPHA est mis à 0, les données sont valides au premier front du signal d'horloge. La polarité CPOL détermine s'il s'agit d'un front montant ou descendant. A CPOL=0, au repos, l'horloge est sur le niveau BAS; le premier front est donc un front montant. A CPOL=1, au repos, l'horloge est sur le niveau HAUT; le premier front est donc un front descendant. Cependant, la polarité de l'horloge n'a pas d'influence sur le moment où le premier bit de données est valide et par conséquent n'a pas d'effet sur le format du transfert de données, comme indiqué en figure II.11.

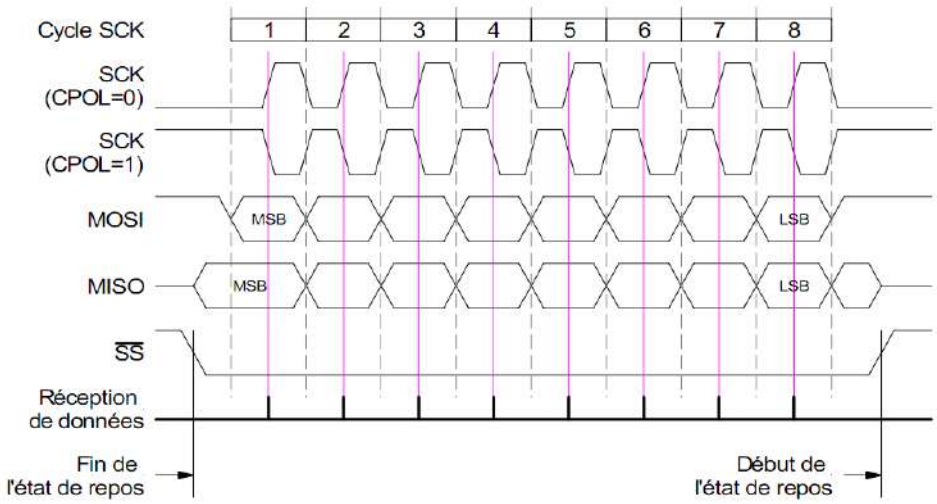


***Figure II.11**. Exemple d'une transmission de données SPI de 1 octet pour CPHA=0.*

II.3.4.4.2 CPHA = 1

Lorsque CPHA est mis à 1, les données sont réceptionnées avec le deuxième front du signal d'horloge. La polarité CPOL détermine s'il s'agit d'un front montant ou descendant. Avec CPOL=0, au repos, l'horloge est sur le niveau BAS et monte au niveau HAUT après le premier front; le deuxième front est donc un front descendant. Avec CPOL=1, au repos, l'horloge est au niveau HAUT et descend au niveau BAS après le premier front; le deuxième

front est donc un front montant.

Cependant, la polarité de l'horloge n'a pas d'influence sur le moment où le premier bit de données est valide et par conséquent n'a pas d'effet sur le format du transfert de données, comme indiqué en Figure II.12.

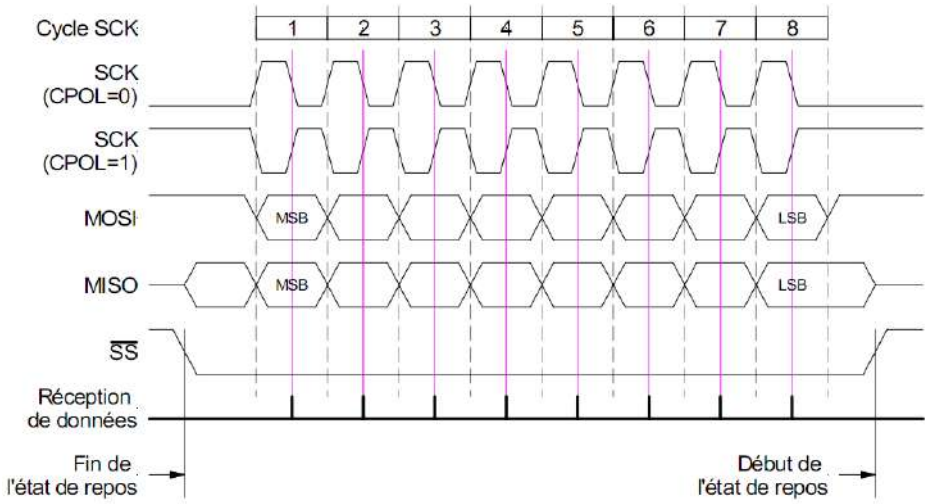


***Figure II.12**. Exemple d'une transmission de données SPI de 1 octet pour CPHA=1.*

II.4 L'Arduino

II.4.1 Présentation de l'Arduino

L'Arduino est une plateforme open source d'électronique programmée qui est basée sur une carte à microcontrôleur et un logiciel. Plus simplement, on peut dire que l'Arduino est un module électronique, doté d'un microcontrôleur programmable. Il peut être utilisé pour développer des objets interactifs, munis d'interrupteurs ou de capteurs, et peut contrôler un moteur ou toutes autres sorties matérielles.

Le principe de fonctionnement est simple:

1. On réalise le programme sur un ordinateur.
2. On connecte l'ordinateur à l'Arduino via une prise USB.
3. On envoie le programme sur l'Arduino.
4. L'Arduino exécute enfin le programme de manière autonome [1].

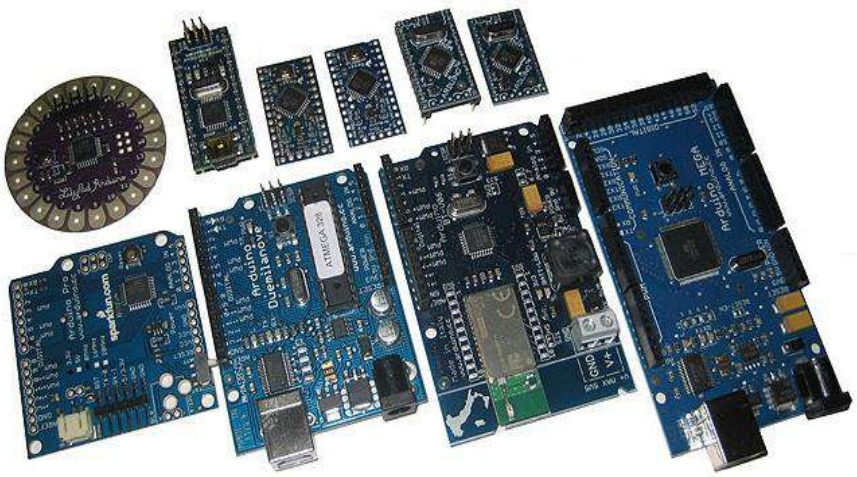


***Figure II.13**. Différents carte Arduino.*

II.4.2 Comparaison des modules microcontrôleurs ARDUINO [1]

Le tableau suivant représente la comparaison entre les différentes cartes Arduino.

***Tableau II.2**. Tableau comparatifs des caractéristiques des modules ARDUINO.*

MODULE	Processeur	CPU	SRAM	FLASH	EEPROM	Broches An. Entrée	Broches Num. ES/PWM
UNO	ATmega328	16 Mhz	2 kb	32 kb	1 kb	6	14/6
LEONARDO	ATmega32u4	16 Mhz	2.5 kb	32 kb	1 kb	12	20/7
DUEMILANOVE	ATmega328	16 Mhz	2 kb	32 kb	1 kb	6	14/6
NANO	ATmega328	16 Mhz	2 kb	32 kb	1 kb	8	14/6
MEGA2560	ATmega2560	16 Mhz	8 kb	256 kb	4 kb	16	54/14
DUE	AT91SAM3X8E	84 Mhz	96 kb	512 kb	-	12	54/12
MEGA ADK	ATmega2560	16 Mhz	8 kb	256 kb	4 kb	16	54/14

II.4.2.1 Avantages de l'Arduino

- Les principaux avantages du système Arduino sont :
  - Matériel peu onéreux : Les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes de développement. La moins chère des versions du module Arduino peut être assemblée à la main, et même les cartes Arduino pré-assemblées coûtent moins de 25 €.

- Matériel Open source et extensible : Les schémas des modules sont publiés sous une licence Créative Commons, et il est possible de réaliser nos propres versions des cartes Arduino, en les complétant et en les améliorant.
- Logiciel gratuit et Open Source: Le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés. Le langage peut être aussi étendu à l'aide de librairies C++.
- Logiciel multiplateforme : Le logiciel Arduino, écrit en Java, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux.

Modules « shield » : Cartes supplémentaires se connectant sur le module Arduino pour augmenter les possibilités : afficheur graphique couleur, interface Ethernet, GPS, etc...[1].

II.4.3 La carte Arduino Uno [1]

II.4.3.1 Introduction

Il existe plusieurs types de cartes Arduino. La carte Arduino Uno est une des versions majeures des cartes Arduino.

Le modèle UNO de la société ARDUINO est une carte électronique dont le cœur est un microcontrôleur ATMEL de référence ATMega328. Le microcontrôleur ATMega328 est un microcontrôleur 8 bits de la famille AVR dont la programmation peut être réalisée en langage C.

L'intérêt principal des cartes ARDUINO (d'autres modèles existent) est leur facilité de mise en œuvre. ARDUINO fournit un environnement de développement s'appuyant sur des outils open source. Le chargement du programme dans la mémoire du microcontrôleur se fait de façon très simple par port USB. En outre, des bibliothèques de fonctions "clé en main" sont également fournies pour l'exploitation d'entrées-sorties courantes : gestion des E/S TOR, gestion des convertisseurs ADC, génération de signaux PWM, exploitation de bus TWI/I2C.

II.4.3.2 Présentation

Il s’agit d’une carte au format « standard » Arduino c'est-à-dire environ 52 mm sur 65 mm.

Deux rangées de connecteurs, situées de part et d'autre de la carte, permettent la connexion des composants extérieurs ou des modules shields, au format Arduino, il est possible d'enficher directement ces derniers dans la carte de base, sur plusieurs niveaux si nécessaire.

➢ *Différents éléments de la carte Arduino Uno*

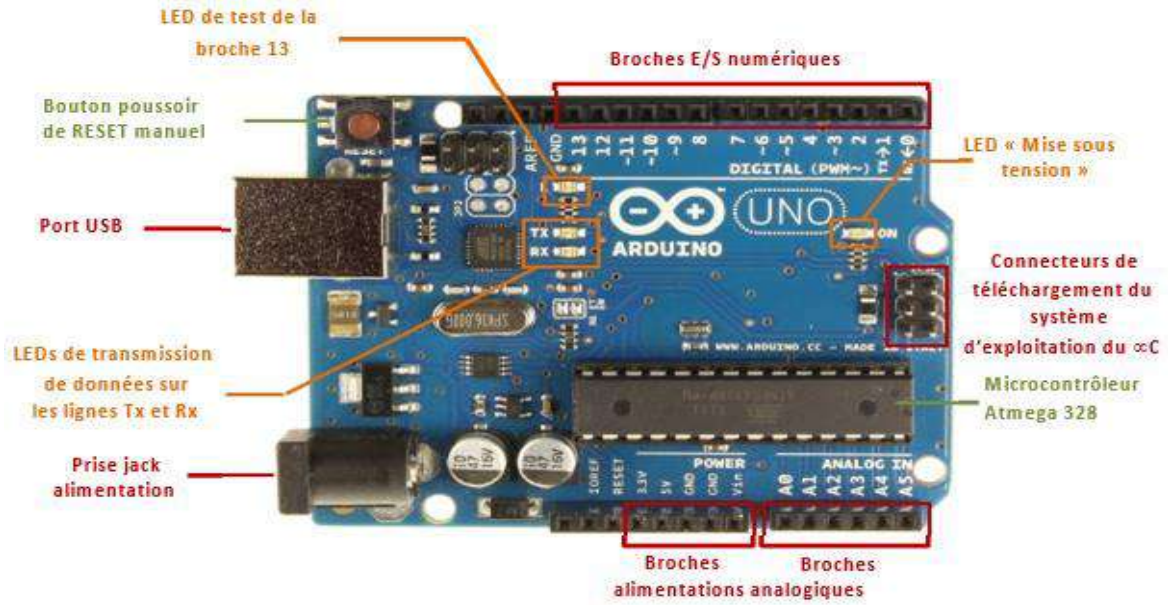


***Figure II.14**. Différents éléments de la carte Arduino Uno.*

❖ Caractéristiques de la carte Arduino Uno :

- Microcontrôleur : ATmega328
- Tension d'alimentation : 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 sortie 3,3V : 50mA
- Mémoire Flash 32 kb dont 0.5 kb utilisée par le bootloader
- Mémoire SRAM 2 kb
- mémoire EEPROM 1 kb
- Fréquence horloge : 16 MHz
- Dimensions : 68.6mm x 53.3mm

II.4.3.3 L'Alimentation

La carte Arduino UNO peut être alimentée par le câble USB, par un bloc secteur externe connecté grâce à une prise « jack » de 2,1mm ou bien par un bloc de piles dont le raccordement est réalisé par l'intermédiaire des « GND » et « VIN » du connecteur

d'alimentation (POWER). L'alimentation extérieure doit présenter une tension comprise entre 7 à 12 V.

La carte génère, par l'intermédiaire de régulateurs intégrés, deux tensions stabilisées : 5 V et 3,3 V. Ces deux tensions permettent l'alimentation des composants électroniques de la carte Arduino. Etant disponibles sur connecteurs placés sur le pourtour des cartes, elles permettent également l'alimentation des modules shields.

II.4.3.4 Les entrées/sorties

La carte « Arduino Uno » dispose de 14 E/S numériques et de 6 entrées analogiques.

- ***Entrées/sorties numériques :***

Chacune des 14 broches numériques (repérées 0 à 13) peut être utilisée en entrée (input) ou en sortie (output) sous le contrôle du programme. Le sens de fonctionnement pouvant même changer de manière dynamique pendant son exécution.

Elles fonctionnent en logique TTL (0V-5V) ; chacune pouvant fournir (source) ou recevoir un courant maximal de 40 mA et dispose si besoin est d'une résistance interne de 'pull-up'.

- ***Entrées analogiques :***

Les six entrées analogiques, repérées A0 à A5 (PC0 à PC5), peuvent admettre toute tension analogique comprise entre 0 et 5 V (par défaut mais cela peut être modifié). Ces entrées analogiques sont gérées par un convertisseur analogique/numérique de 10 bits, dont la sortie peut varier de 0 à 1023.

Les entrées A4 et A5 peuvent également être utilisées respectivement comme la ligne de donnée SDA et la ligne d'horloge SCL de l'interface série I2C. [1]

II.4.3.5 Le microcontrôleur ATMega 328 [1]

Le microcontrôleur utilisé sur la carte Arduino UNO est un microcontrôleurATMega328. C'est un microcontrôleur ATMEL de la famille AVR 8bits.

- Les principales caractéristiques sont :

FLASH = mémoire programme de 32Ko.

- SRAM = données (volatiles) 2Ko.
- EEPROM = données (non volatiles) 1Ko.
- Digital I/O (entrées-sorties Tout Ou Rien) =3 ports Port B, Port C, Port D (soit 23

broches en tout I/O).

- Timers/Counters : Timer0 et Timer2 (comptage 8 bits), Timer1 (comptage 16bits) Chaque timer peut être utilisé pour générer deux signaux PWM. (6 broches OCxA/OCxB).
- Plusieurs broches multifonctions *:* certaines broches peuvent avoir plusieurs fonctions différentes choisies par programmation.
- PWM = 6 broches OC0A (PD6), OC0B(PD5), 0C1A(PB1), OC1B(PB3), OC2A(PB3),OC2B (PD3).
- Analog to Digital Converter (résolution 10bits) = 6 entrées multiplexées ADC0(PC0) à ADC5(PC5).
- Gestion bus I2C (TWI Two Wire Interface) = le bus est exploité via les broches SDA(PC5)/SCL(PC4).
- Port série (USART) = émission/réception série via les broches TXD(PD1)/RXD(PD0).
- Comparateur Analogique = broches AIN0(PD6) et AIN1 (PD7) peut déclencher interruption.
- Gestion d'interruptions (24 sources possibles) : en résumé :
    - Interruptions liées aux entrées INT0 (PD2) et INT1 (PD3).
    - Interruptions sur changement d'état des broches PCINT0 à PCINT23.
    - Interruptions liées aux Timers 0, 1 et 2 (plusieurs causes configurables).
    - Interruption liée au comparateur analogique.
    - Interruption de fin de conversion ADC.
    - Interruptions du port série USART.

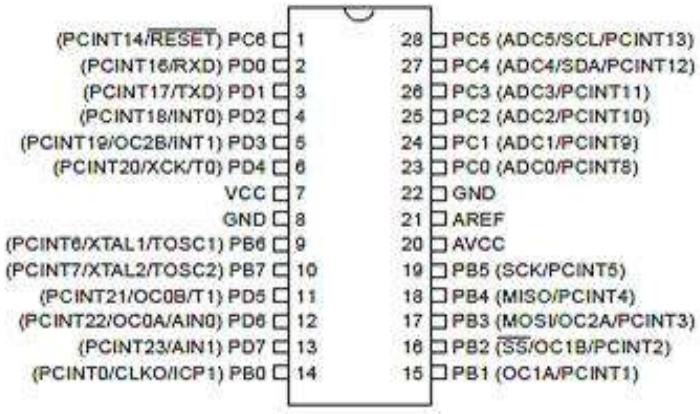


Figure II.15*. Le microcontrôleur ATMega328.*

II.4.3.6 L'interface de Communication

La carte « Arduino Uno » a de nombreuses possibilités de communications avec l'extérieur.

L'Atmega328 possède une interface de communication série UART accessible, grâce aux broches numériques 0 (Rx) et 1 (Tx). D'autre part elle supporte le bus I2C accessible, grâce aux broches analogiques 4 (SDA) et 5 (SCL) et la liaison série synchrone SPI grâce aux broches numériques 10 (SS) ,11 (MOSI), 12(MISO) et 13 (SCK).[1]

II.4.3.7 Reset

A la mise sous tension un reset automatique permet au programme contenu en mémoire du microcontrôleur de démarrer automatiquement dès que la carte Arduino est alimentée.

La carte « Arduino Uno » est également équipée d'un bouton poussoir de reset manuel. Un appui sur celui-ci permet de relancer l'exécution d'un programme si nécessaire, soit parce qu'il s'est « planté » soit tout simplement parce que l'on souhaite le faire repartir de son début. [1]

II.4.3.8 Protection de sur intensité USB

Par mesure de sécurité pour l'ordinateur auquel sera relié l'Arduino, un fusible réarmable ou « Polyfuse » est présent sur la connexion d'alimentation 5 V de la prise USB. Toute consommation supérieure à 500 mA, provoque le déclenchement de ce fusible, protégeant ainsi le port USB de l'ordinateur auquel la carte est reliée. Le fusible étant de type réarmable, il retrouvera son état normal quelques secondes après que la consommation excessive aura cessée. [1]

II.4.4 La carte Arduino Ethernet [5]

II.4.4.1 Présentation

L'Arduino Ethernet est une carte électronique basée sur le microcontrôleur Wiznet W5100. Il dispose de 14 entrées / sorties numériques, 6 entrées analogiques, d'un oscillateur à cristal 16 MHz, une connexion RJ45, une prise d'alimentation, un connecteur ICSP, et un bouton de réinitialisation. Cela réduit le nombre de broches disponibles à 9, avec 4 sorties PWM disponible.

REMARQUE : Les broches 10, 11, 12 et 13 sont réservées pour l'interfaçage avec le module Ethernet et ne doivent pas être utilisées autrement.

L'Arduino Ethernet diffère des autres cartes Arduino UNO du fait qu'elle ne dispose pas d'une puce USB-série, afin de laisser place à l'interface Ethernet Wiznet.

C'est la même que l'on retrouve sur le shield Ethernet. Un support carte micro SD est aussi présent, il peut être utilisé pour stocker des fichiers pour servir pour le réseau, est accessible par la bibliothèque SD.

La Pin 10 est réservée à l'interface Wiznet, le SS de la carte SD est sur la broche 4.

Un connecteur à six broches permet de connecté un câble FTDI USB/Série ou un module USB2SERIAL nécessaire pour la programmation.



Figure II.16*. La carte Arduino Ethernet.*

- Caractéristiques de la carte Arduino Ethernet :
  - Microcontrôleur : ATmega328
  - Tension d'alimentation nominale : 5V
  - Tension d'alimentation (recommandé) :7-12V
  - Tension d'alimentation (limites) :6-20V
  - Entrées/sorties digitales : 14 (dont 6 pouvant être utilisées comme sorties PWM)

- Entrées Analogiques : 6
- Courant max par broches E/S : 40 mA
- Courant max sur sortie 3,3V : 50mA
- Mémoire Flash : 32 kb dont 0.5 kb utilisée par le bootloader
- Mémoire SRAM : 2 kb
- mémoire EEPROM : 1 kb
- Fréquence horloge : 16 MHz
- W5100 TCP/IP contrôleur Ethernet intégré
- Power over Ethernet prêt Prise magnétique
- Carte Micro SD, avec des traducteurs de tension actifs.

II.4.5 Programmation de la carte Arduino Uno [1]

Le logiciel de programmation des modules Arduino est une application Java, libre et multi- plateformes, servant d'éditeur de code et de compilateur, et qui peut transférer le firmware et le programme au travers de la liaison USB. Le langage de programmation utilisé est un mélange de C et de C++, restreint et adapté aux possibilités de la carte.

Le logiciel Arduino a pour fonctions principales :

- de pouvoir écrire et compiler des programmes pour la carte Arduino ;
- de se connecter avec la carte Arduino pour y transférer les programmes ;
- de communiquer avec la carte Arduino.

Pour ouvrir l'IDE d'Arduino, il faut utiliser l'icône : 

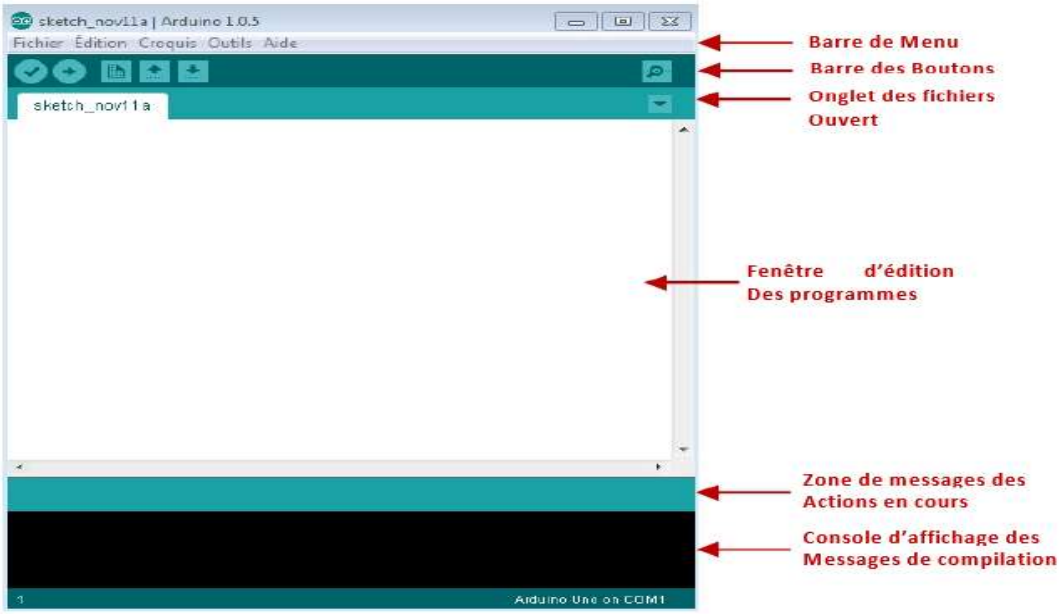


***Figure II.17**. Logiciel Arduino.*

La « Barre de Boutons » qui donne un accès direct aux fonctions essentielles du logiciel. Les différents boutons sont :

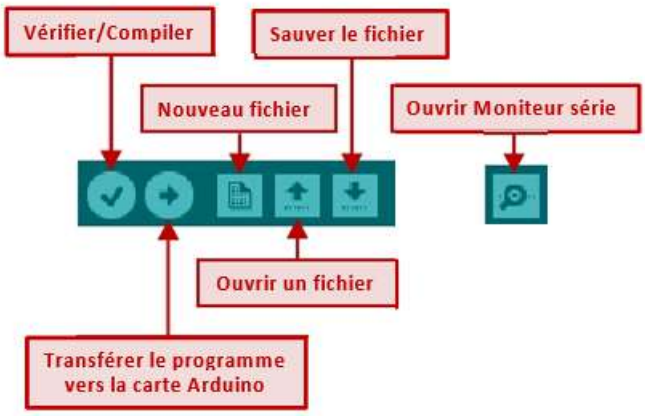


***Figure II.18**. La Barre de Boutons.*

II.5 Le module RF [1]

II.5.1 Présentation du module nRF24L01

Le nRF24L01 est un module radio transmetteur (à la fois récepteur et émetteur), sur une porteuse 2.4GHz avec un protocole intégré de Nordic Semi conducteur nommé "ShockBurst".

Il est conçu pour opérer sur la bande de fréquence de 2,4 à 2,4835 GHz. Le nRF24L01 est un module hautement intégré, ultra faible puissance (Ultra Law Power). Le taux de transfert de données sans fil que la nRF24L01 peut supporter et configurable jusqu'à 2Mb/s. Ce taux de fait de cette puce un composant tout désigné pour les projets à basse énergie.

Elle possède différents modes de fonctionnement dont un mode "relai automatique de données".



***Figure II.19**. Le module nRF24L01.*

II.5.2 Caractéristiques techniques

- jusqu'à 2Mb/s sur le débit de données
- Consommation en émission à 0 dBm de 11.3 mA.
- Consommation en réception à 2 Mb/s de 12.3 mA.
- 22µA en mode veille
- Le régulateur de tension intégré
- Gamme alimentation entre 1.9 to 3.6V.
- Manipulation automatique de paquets.

II.6 Protocol ShockBurst [6]

Le protocole ShockBurst utilise un module FIFO sur puce pour cadencé les données à un faible taux de transfert de données et le transmettre à un taux très élevé permettant ainsi une réduction importante de puissance.

Lors du fonctionnement de nRF24L01 dans mode ShockBurst, on peut atteindre des débits de données jusqu'à (1 Mbps) sans la nécessité des microcontrôleurs(MCU) coûteuse, et de haute vitesse du traitement des données.

Dans ce mode le nRF2401 offre les avantages suivants :

- Une consommation de courant très réduite.
- Un coût du système réduit.

- Le risque de collisions « sur l'air» est considérablement réduit en raison de la rapidité de transmission.

Le nRF2401 peut être configuré à l'aide d'une interface à 3 fils simple où le débit de données est déterminé par la vitesse du microcontrôleur.

En permettant à la partie numérique de l'application de s'exécuter à basse vitesse tout en optimisant le débit de données sur la liaison RF, la nRF dans le mode ShockBurst réduite considérablement la consommation moyenne du courant dans les applications.

II.6.1 Principe de ShockBurst [6]

Lorsque la nRF24L01 est configuré en ShockBurst, TX ou RX, l'opération est menée de la manière suivante (10 kbit/s pour l'exemple uniquement).

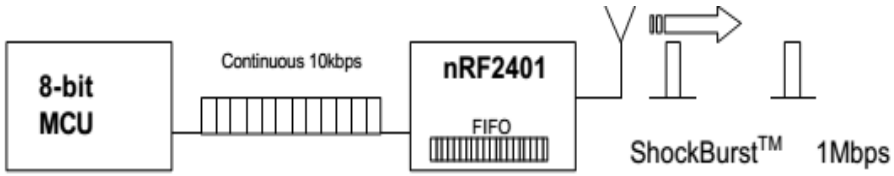


Figure II.20. *Synchronisation des données avec MCU et la transmission avec technologie ShockBurst.*

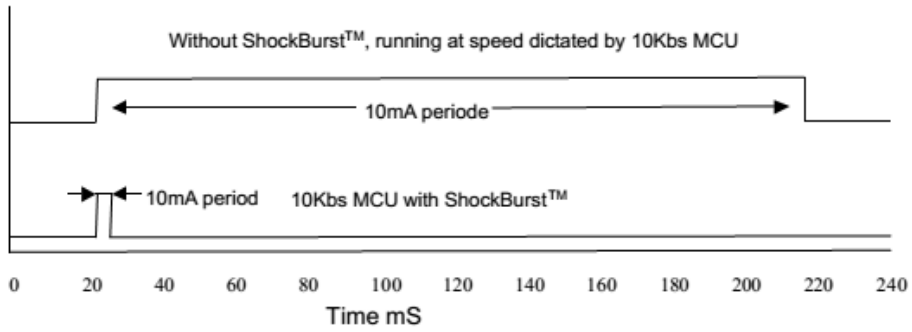


Figure II.21. *La consommation de courant avec et sans la technologie ShockBurst.*

II.6.2 Les organigrammes de ShockBurst d'émission et de réception

II.6.2.1 La transmission

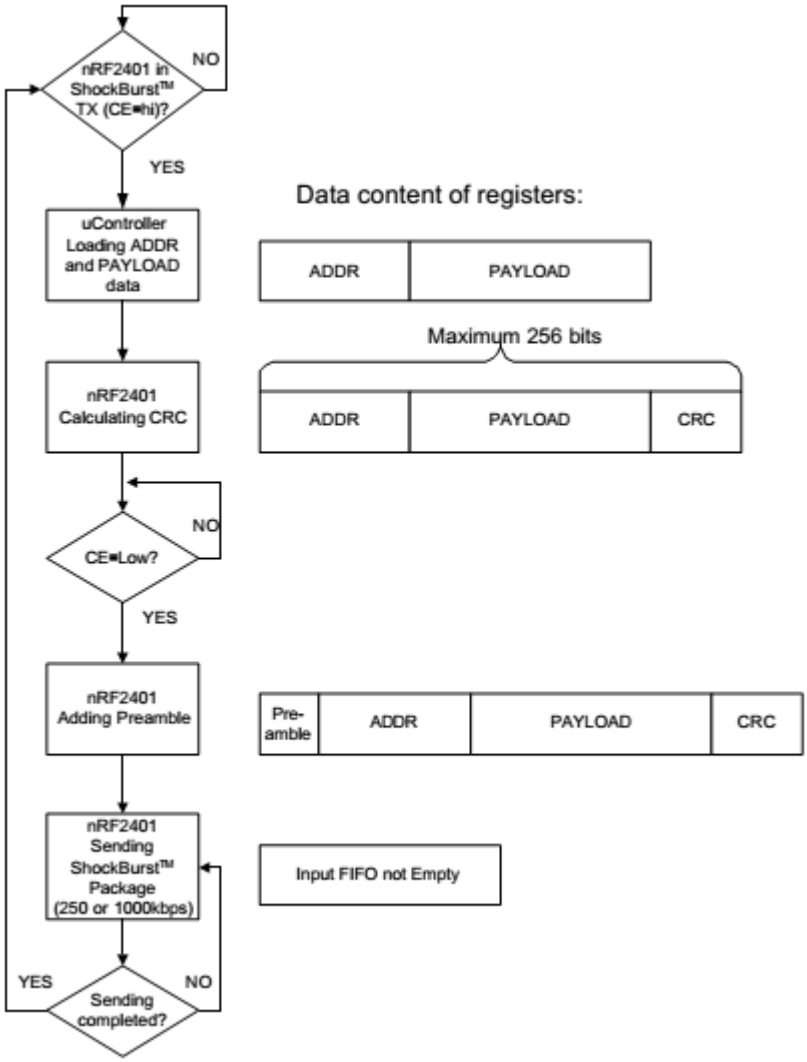


Figure II.22. *L'organigramme de l'émission.*

Broches d'interface MCU : CE, CLK1, DATA

- Lorsque l'application du microcontrôleur a une donnée à envoyer, la pine CE est configuré en état haut. Ceci active la nRF24L01 et le traitement des données à bord.
- L'adresse du nœud de réception (RX) et la charge utile de données est configuré dans la nRF24L01. Le protocole de l'application où le microcontrôleur définit la vitesse <1Mbps (ex : 10kbps).
- Le microcontrôleur configure ensuite la pine CE en état bas, ce qui active une transmission nRF24L01 ShockBurst.
- nRF24L01 ShockBurst :

  - La mise sous tension du module RF.

- Le Paquet RF est terminé (préambule ajouté, CRC calculée).
- Les données sont transmises à grande vitesse (250 kbps ou 1 Mbps configurée par l'utilisateur).
- nRF2401 revenir en mode stand-by lorsque la transmission est terminé.

II.6.2.2 La réception

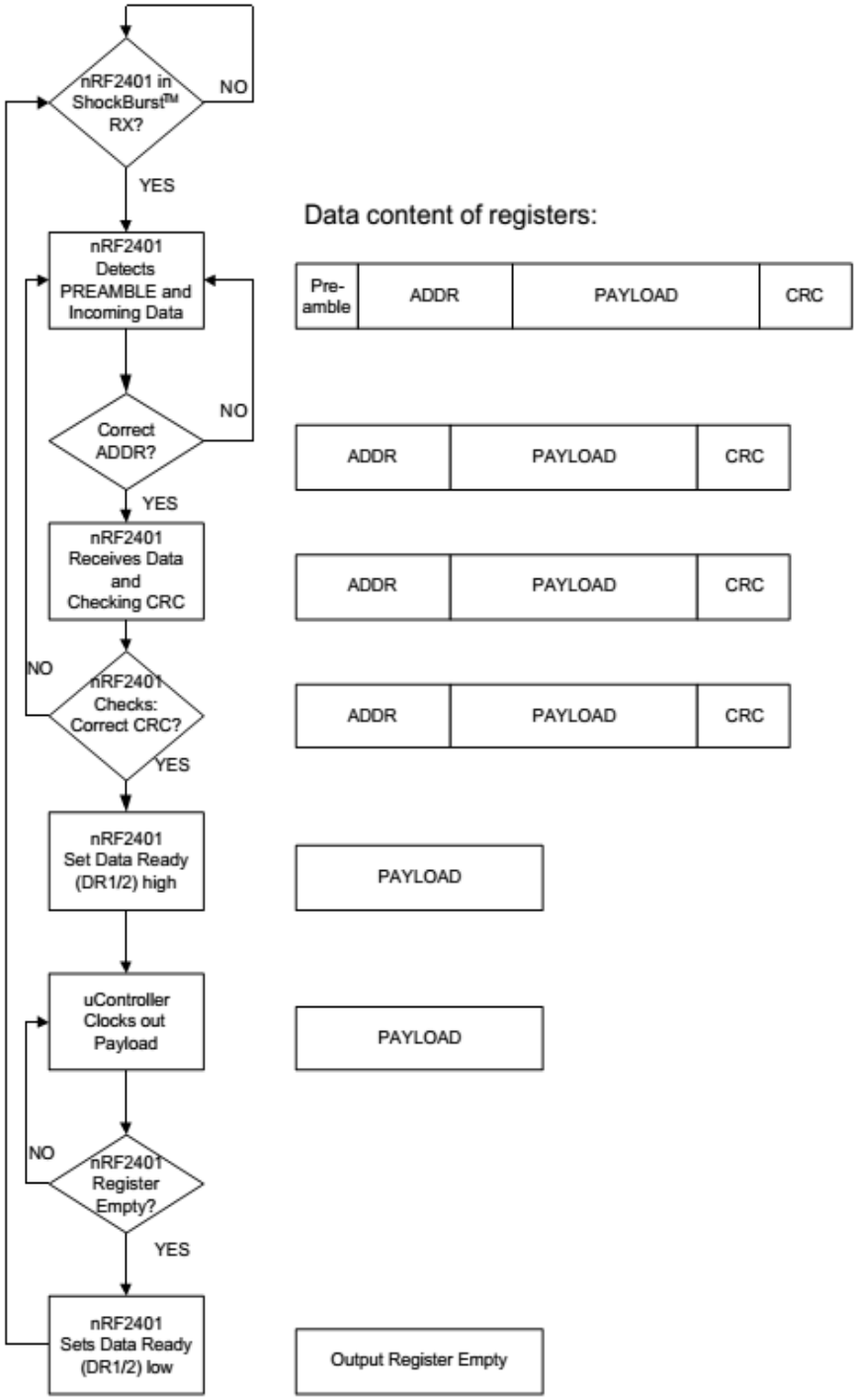


Figure II.23. *L'organigramme de réception.*

Les broches d'interface MCU sont: CE, DR1, CLK1 et Data.

- L'adresse correcte et taille de fichier (charge utile) du paquet RF transmit sont ensemble configuré lorsque la nRF2401 est configuré dans le mode ShockBurst RX.
- Le CE est configuré en état haut pour activer la réception RX.
- Après 200 µs la décantation, la nRF2401 est en surveillance de l'air pour des communications entrantes.
- Lorsqu'un paquet valide a été reçu (l'adresse correcte et le CRC trouvé), nRF2401 supprime le préambule, l'adresse et le bit CRC.
- nRF2401 notifie ensuite (interruptions) le MCU en configure la broches DR1 en état haute.
- MCU peuvent (ou pas) configure le CE en état bas pour désactiver le module RF (mode courant faible).
- Le MCU va récupérer la charge utile des données (payload) à un taux approprié (ex. 10kbits/s).
- Lorsque toutes les données sont récupérées, la nRF2401 remit le DR1 en état bas, et est prêt pour une nouvelle séquence de données entrant.
- Si le CE est de nouveau en état bas, Une nouvelle séquence de réception peut commencer.[6]

II.6.3 Description de paquet de données [6]

PREAMBULE (1 octet)	***ADRESSE (3-5 octet)***	***CHAREGE UTILE (0-32 octet)***	***CRC (1 ou octet2)***

Figure II.24. *Trame de paquet de données.*

Le paquet de données pour les deux modes : mode ShockBurst et mode communication Direct constitue et divisé en 4 sections selon la table suivante :

***Tableau II.3**. Description de paquet de données.*

1. **préambule**	• Le champ préambule est nécessaire dans ShockBurst et modes Direct. • Préambule est de longueur 8 bits et est tributaire du premier bit de données en mode direct. PREAMBULE 1er BIT d'ADDR 01010101 0 10101010 1 • Préambule est automatiquement ajouté au paquet de données dans ShockBurst et donne ainsi un espace supplémentaire pour la charge utile. En mode Direct le MCU doit gérer le préambule. • En mode ShockBurst RX, le préambule est éliminé des données de sortie reçues, En mode direct le préambule est transparent pour les données de sortie.
2. **Adresse**	• Le champ d'adresse est nécessaire en mode ShockBurst. • 8-40bits de longueur. • ADRESSE automatiquement retiré de paquet reçu en mode ShockBurst. En mode Direct MCU doit gérer l'adresse. En mode direct MCU doit traiter l'adresse.
3. **Charge utile**	• Les données à transmettre. • Dans la taille de la charge utile en mode ShockBurst est de 256 bits, moins ce qui suit :(Adresse: 8 to 40 bits. + CRC 8 ou 16 bits). • En mode direct la taille de paquet maximale (longueur) est à 1 Mbps, 4000 bits. (4ms).
4. **CRC**	• Le CRC est facultative dans le mode ShockBurst, et n'est pas utilisé en mode direct. • 8 ou 16 bits de longueur. • Le CRC est éliminé des données de sortie reçues dans ShockBurst RX.

II.6.4 Le protocole Multi récepteur [7]

Le protocole multi récepteur est une fonction utilisée en mode RX qui contient un ensemble de six pipes de données parallèles avec des adresses uniques. La pipe de données est un canal logique dans le canal RF physique. Chaque pipe de données dispose de sa propre adresse physique (adresse de la pipe de données) décodé dans la nRF24L01.

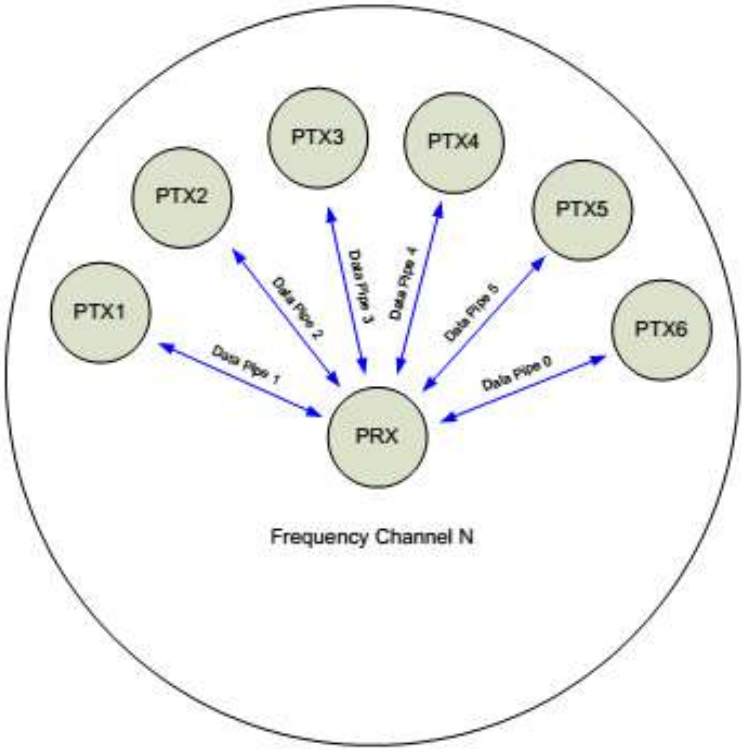


***Figure III.25**. PRX utilisant multi récepteur.*

La nRF24L01configuré comme PRX (récepteur primaire) peut recevoir des données adressées à six différentes pipes de données dans un canal de fréquence, comme illustré à la (Figure II.25). Chaque pipe de données dispose de sa propre unique adresse et peut être configuré pour des comportements individuels.

Jusqu'à 6 nRF24L01configuré comme PTX (émetteur primaire) peut communiquer avec nRF24L01 configuré comme PRX. Toutes les adresses des pipes de données sont recherchées simultanément. Une seule pipe de données peut recevoir un paquet à la fois.

Les paramètres suivants sont communs à tous les pipes de données :

- CRC activé/désactivé (CRC toujours activé lorsque ShockBurst est activé).

- RX largeur d'adresse.
- Fréquence canal.
- Débit de données d'air.
- Gain LNA (Löw Noise Amplifier).

REMARQUE : Toujours il faut assurer qu'aucun des pipes de données ont la même adresse. Chaque pipe peut avoir jusqu'à 5 octet d'adresse configurable. La pipe de données 0 a une unique adresse de 5 octet. Les pipes de données 1 à5 partage les même premier quatre octets d'adresse les plus significatif. L'octet le moins significatif doit être unique pour toutes les 5 pipes. est un exemple de l'adressage des pipes est donnée sur la Figure II.26 [7]

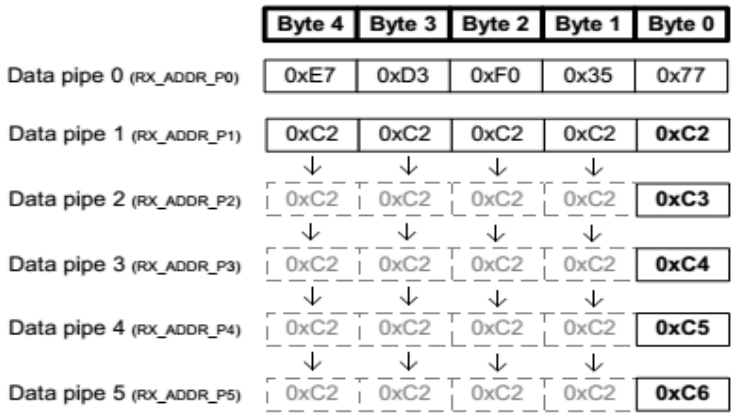


***Figure II.26**. Exemple de l'adressage des pipes de donnée 0-5.*

Le PRX, utilise les protocoles multi récepteur et le ShockBurst, reçoit des paquets de plus d'un PTX. Pour s'assurer que le paquet d'ACK du PRX est transmis à la bonne PTX.

Le PRX prend l'adresse de la pipe de données dont il a reçu le paquet et l'utilise comme adresse TX lorsque il retransmettre le paquet ACK. (Figure II.27).

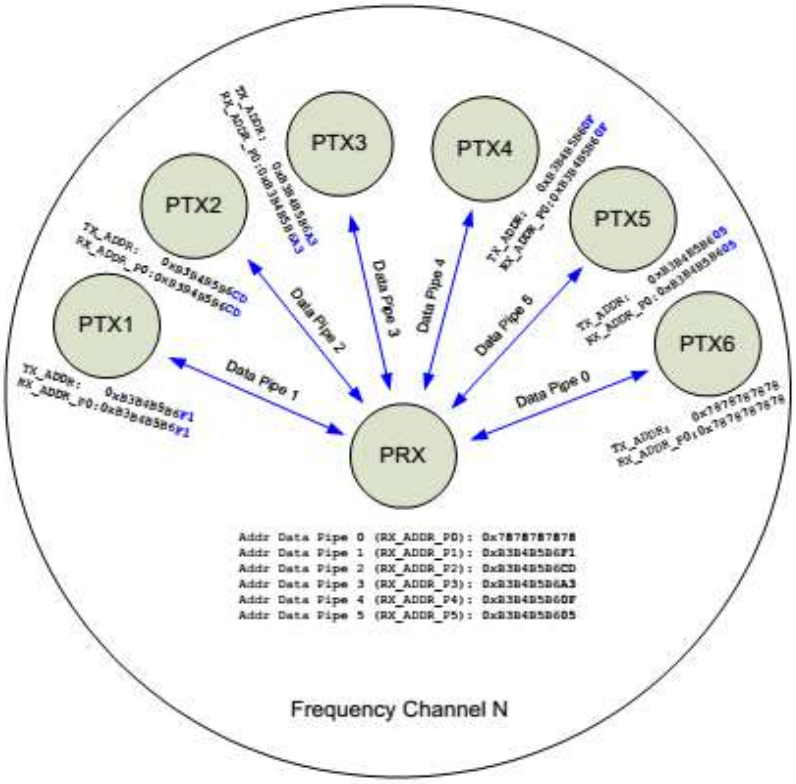


***Figure II.27**. Exemple d'adressage des pipes de données dans Multi récepteur.*

Uniquement lorsqu'une pipe de données reçoit un paquet complet, les autres pipes de données commencent à recevoir des données, Lorsque plusieurs PTXs sont transmettant à un PRX, Les ARD peuvent être utilisé pour déformer la retransmission automatique afin qu'elles seules se bloquer les uns des autres à la fois. [7]

II.7 Les capteurs

II.7.1 Définition d'un capteur

Un capteur est un dispositif qui transforme une grandeur physique d'entrée appelée mesurande, en une grandeur de nature électrique. [1]

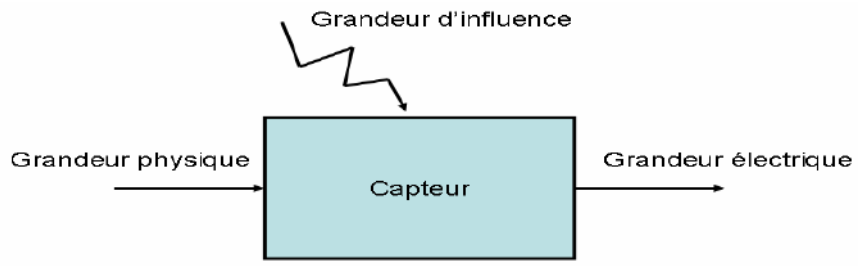


***Figure II.28**. Capteur.*

La grandeur physique à mesurer est désignée comme le mesurande. La grandeur électrique est soit un courant ou une tension. Le capteur peut être vu comme une boîte noire ayant comme entrée un mesurande et comme sortie une tension ou un courant électrique. [1]

II.7.2 Modèle d'un capteur

Du point de vue électrique, deux types de capteur existent. Le premier type est dit capteur passif. Il s'agit de capteur dont la grandeur de sortie est une variation résistance ou une impédance. Il constitue des éléments passifs d'un circuit électrique. Ce circuit a pour rôle de leur apporter une énergie pour convertir la variation d'impédance en une variation de tension ou de courant. Le modèle électrique équivalent est celui d'un dipôle dont l'impédance.

Equivalente varie en fonction du mesurande. Le circuit porte le nom de conditionneur.

A l'inverse les capteurs actifs fournissent en sortie une tension ou un courant. Il prélève une partie de l'énergie utilisée par la grandeur physique à mesurer pour la transformée en énergie électrique.

Le modèle électrique de ce type de capteur que l'on nomme aussi transducteur est un générateur équivalent soit de Thévenin pour les tensions, soit de Norton pour les courants. [1]

II.7.3 Le Capteur DHT 11 [8]

Ce capteur d'humidité et de température est très rependu dans le contrôle de climatisation, il est constitué d'un capteur de température à base de NTC (Coefficient de température négatif), et d'un capteur d'humidité résistif, un microcontrôleur s'occupe de faire les mesures, les convertir et de les transmettre.

II.7.3.1 Caractéristiques

Alimentation +5V.

Température : de 0 à 50°C, précision : +/- 2°C.

Humidité : de 20 à 90% RH, précision +/- 5% RH.



***Figure II.29**. Le capteur DHT11.*

II.7.4 Le Capteur de niveau d'eau [9]

Capteur de niveau d'eau facile à utiliser, bon rapport qualité / prix. L'utilisation en parallèle de plusieurs capteurs ouvre la porte à de grandes installations automatisées. L'utilisation des pistes parallèles sur la board permettent de détecter le niveau d'eau de manière analogique, facile à convertir en numérique pour l'Arduino.

II.7.4.1 Caractéristiques

- Tension de fonctionnement: 3-5Vdc.
- Courant en fonctionnement: moins de 20mA.
- Type de capteur: analogique.
- Dimension de la zone de détection: 40 x 16mm.
- Température de fonctionnement: 10-30°.
- Humidité ambient: 10-90% (Sans condensation).
- Taille de la carte: 62mmx20mmx8mm.
- Connexions:
  - Vcc
  - GND
  - Signal de sortie



***Figure II.30.** Le capteur de niveau d'eau.*

II.7.5 Le Capteur de détection de son [10]

Capteur de Détection de Son de haute qualité Module Arduino capteur détection de son du microphone Haute sensibilité Deux sorties:

- AO, sortie analogique, sortie en temps réel du signal de tension de microphone.
- DO, lorsque l'intensité sonore atteint un seuil, sortie du seuil de signal haut et bas - Réglage de la sensibilité du potentiomètre.
- Alimentation: DC 5V Avec indicateur de puissance convient pour la détection de son du microphone.

Taille: 4,5 x 1,5 x 1,3 CM



Figure II.31. *Le capteur de détection du son.*

II.8 Conclusion

Nous avons détaillé dans ce chapitre les composants principale de notre nœud de capteur, les différents aspect théorique de la conception, et la mise au point d'un réseau de transmission de données par voie RF. Nous avons parlés aussi du Protocol de communication shockburst qui est nécessaire dans ce travail.

Notre choix des modules RF a été très restreint, vu la non-disponibilité de ce type de module sur le marché local. Cependant, le matériel disponible (ARDUINO UNO et quelque type du capteur) offre les outils nécessaires pour mettre au point de notre réseau de capteur sans fil qui répond à nos besoins (taille réduite, fiable et à faible coût).

Chapitre III : Réalisation et mise en marche

Chapitre III : réalisation et mise en marche

III.1 Introduction

Ce chapitre traite la partie du développement Software et hardware du nœud à réaliser, nous présentons en premier le schéma bloc du système a réalisé. Ensuite nous présenterons l'organigramme général du programme de transmission, réception et nœud en mode actionneur, ainsi que la configuration et le brochage des différents modules. Nous terminerons par la représentation du module réalisé.

III.2 Schéma bloc du système

Nous avons essayé d'expliquer en détail la structure du système ainsi la mise en œuvre en utilisant un simple module programmable (Arduino) et une communication par radio fréquence et sur un serveur web. La figure si dessous (figure III.1) décrit le schéma bloc du réseau que nous nous proposons de réaliser.

Le réseau de capteurs est organisé comme suite :

- Un nœud de réception central (Nœud Collecteur) qui collecte les données de tous les autres nœuds de capteur (Nœud puits) et l'envoyer sur Internet pour l'affichage.
- Plusieurs nœuds de capteur qui transmettent à intervalle régulier les donnes issu des différents capteurs. (3 capteurs de différents type sont utilisé : DHT11, capteur de niveau d'eau et capteur de détection du son).

La partie de transmission/réception ainsi que la partie de traitements des données de chaque capteur se faite par l'Arduino Uno.

Les données récupérer sont envoyés vers le Nœud centrale par liaison RF, ces données son t'envoyé de nouveau vers un PC à travers une liaison Ethernet, pour cela on utiliser un Shield Ethernet connecté à l'Arduino Centrale.

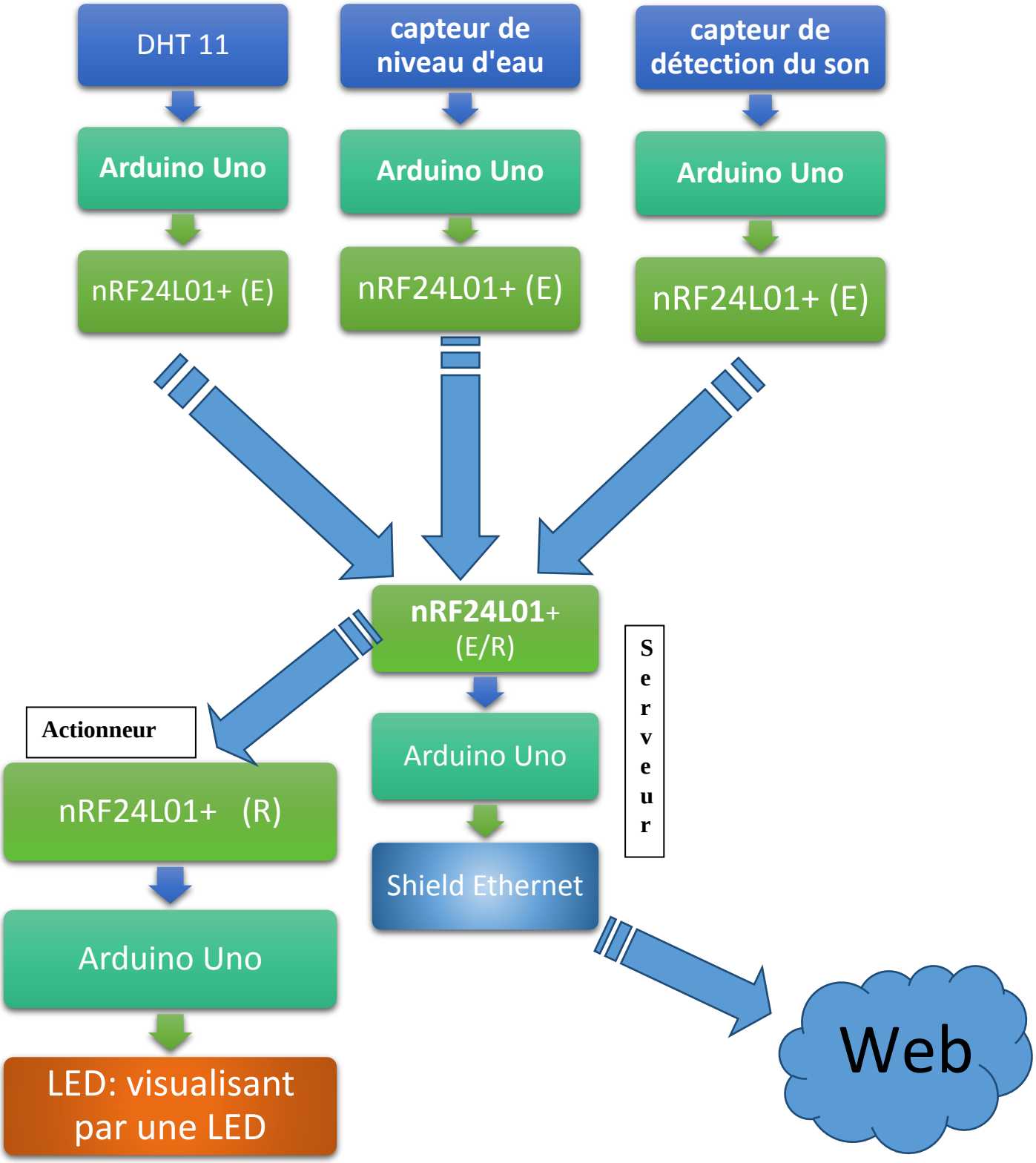


Figure III.1. *Schéma bloc du réseau a réalisé.*

III.3 Réalisation Pratique

III.3.1 Le Mode émission

Pour cette partie les modules utilisés sont :

- 3 modules Arduino Uno.
- Capteur de température DHT11, capteur de niveau d'eau et capteur de détection du son.
- 3 modules RF : nRF24L01+.

III.3.1.1 L'organigramme de l'émission

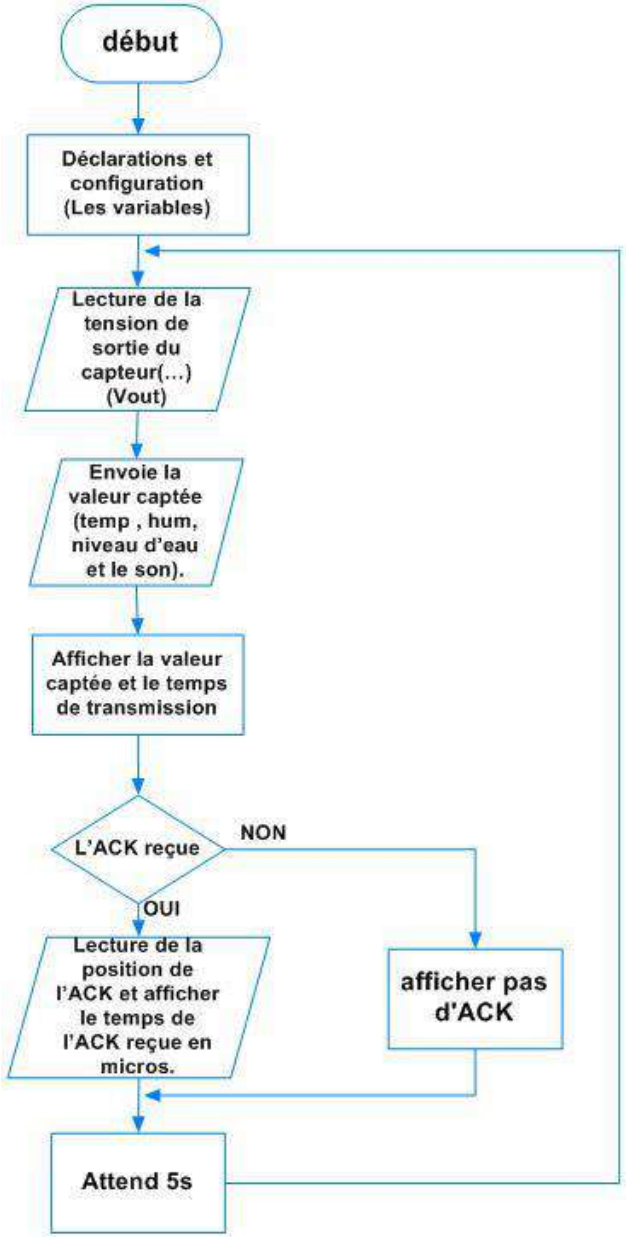


Figure III.2. *L'organigramme de l'émission.*

III.3.1.2 Description de mode émission

Dans cette partie de programme, chaque Arduino reçoit les valeurs de (température, l'humidité, son et le niveau d'eau), ces valeurs sont envoyées vers le Nœud centrale par une liaison RF. Le temps de transmission ainsi les données de chaque capteur sont affichées sur l'écran de logiciel (moniteur série (COM)). Pour la vérification de la bonne réception de donnée chaque Arduino attend les retours de l'ACK envoyé par le nœud central.

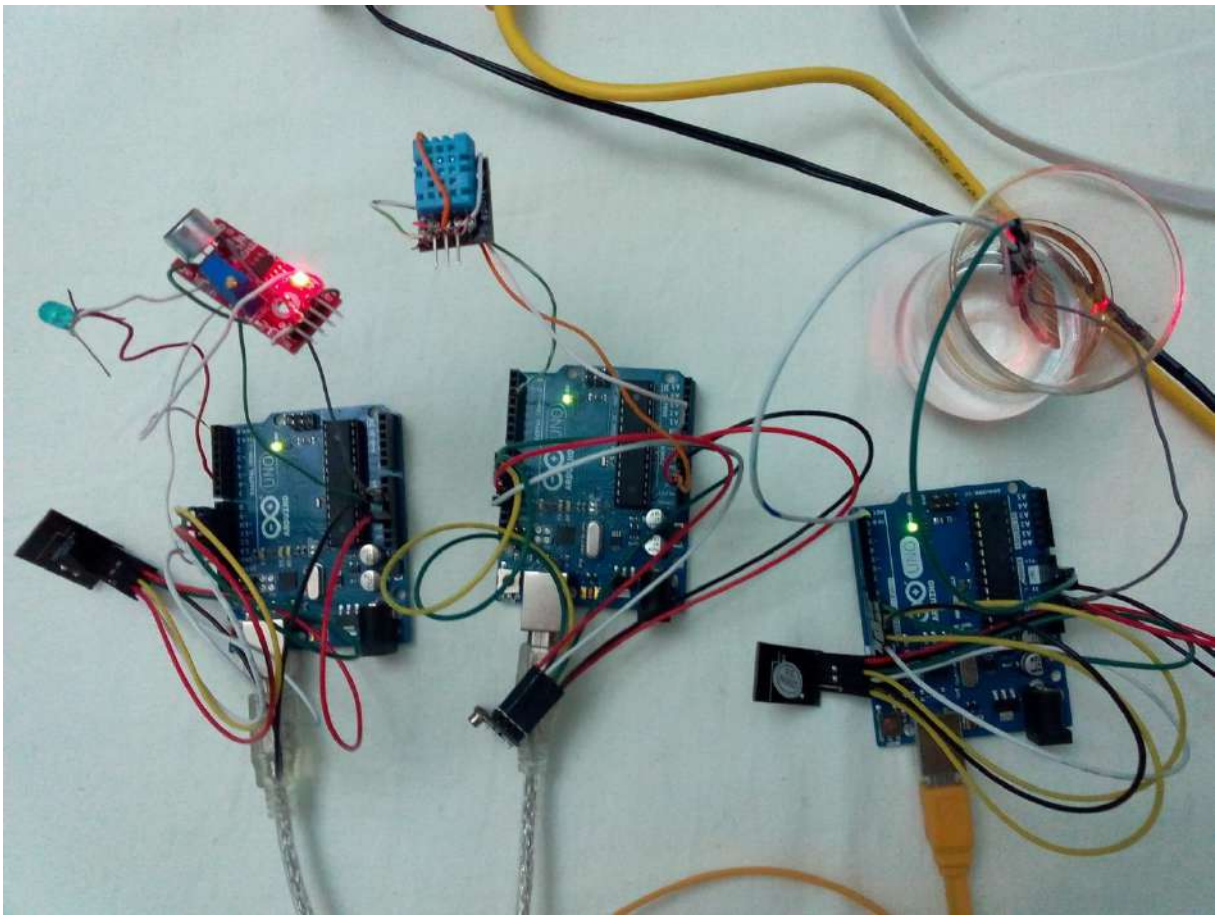


Figure III.3. *Partie émission.*

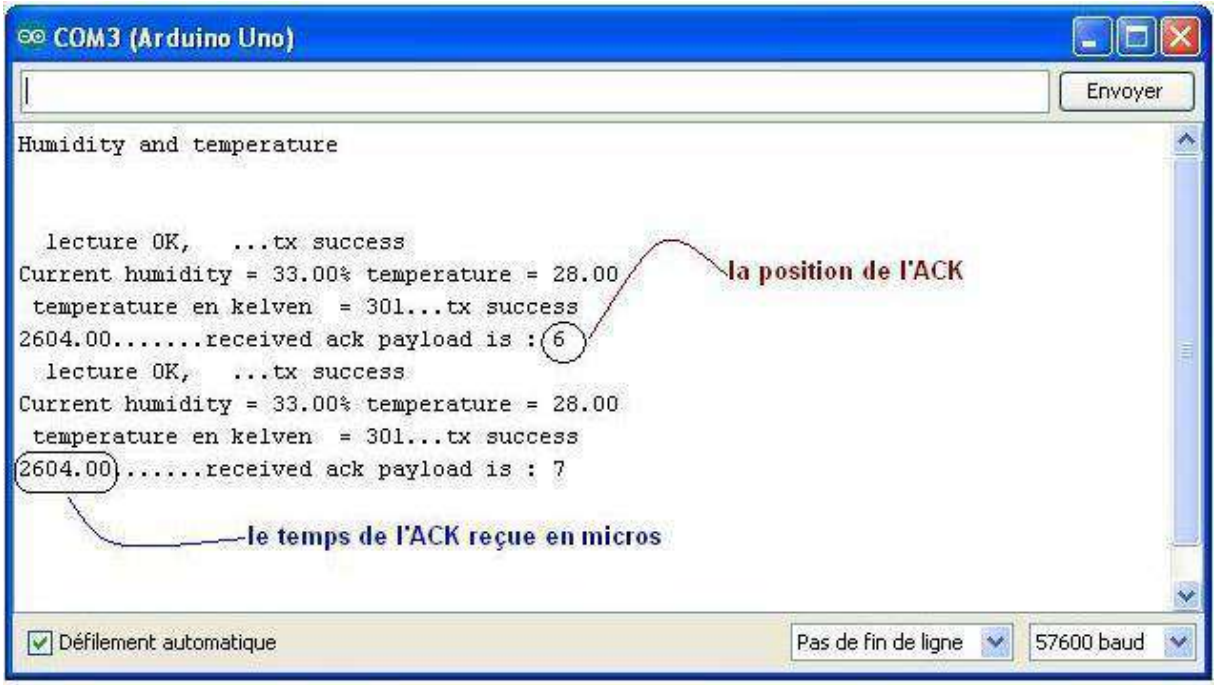


Figure III.4.A *Le résultat de DHT11.*

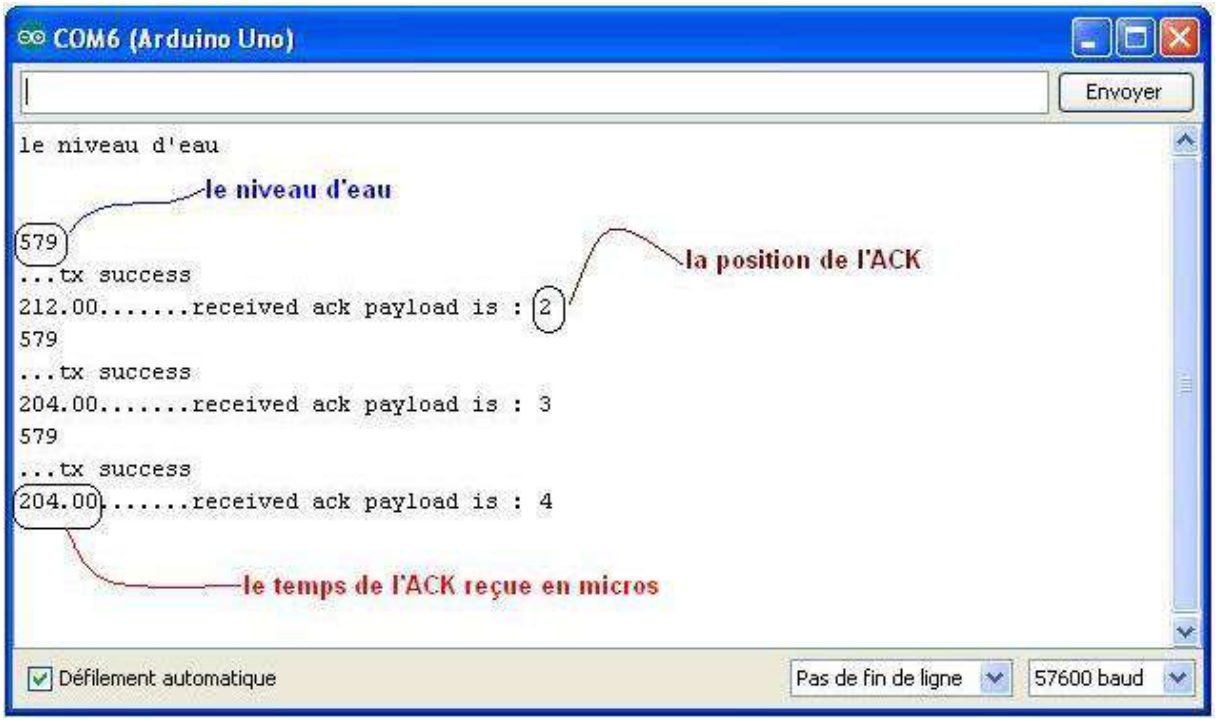


Figure III.4.B *Le résultat du capteur de niveau d'eau.*

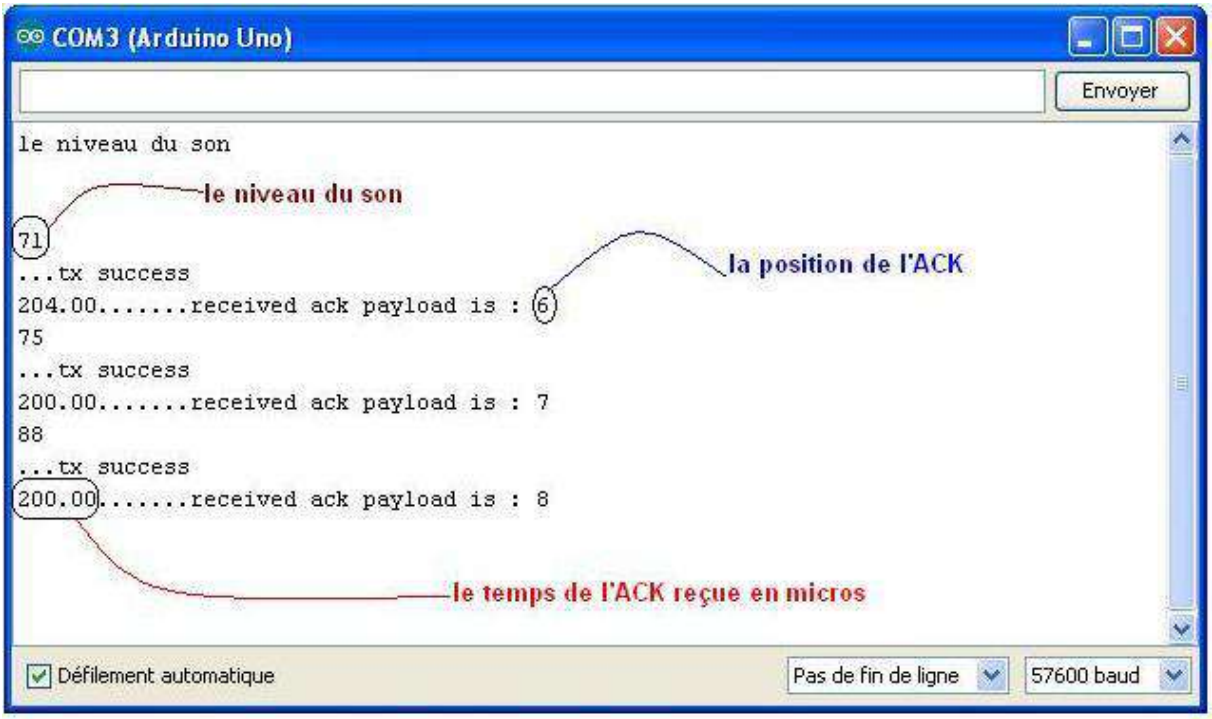


Figure III.4.C *Le résultat du capteur de détection du son.*

III.3.2 Le mode réception

Pour cette partie les constituants utilisés sont :

- Un module Arduino Uno.
- Shield Ethernet.
- Le module RF : nRF24L01+.

III.3.2.1 L'organigramme de la réception

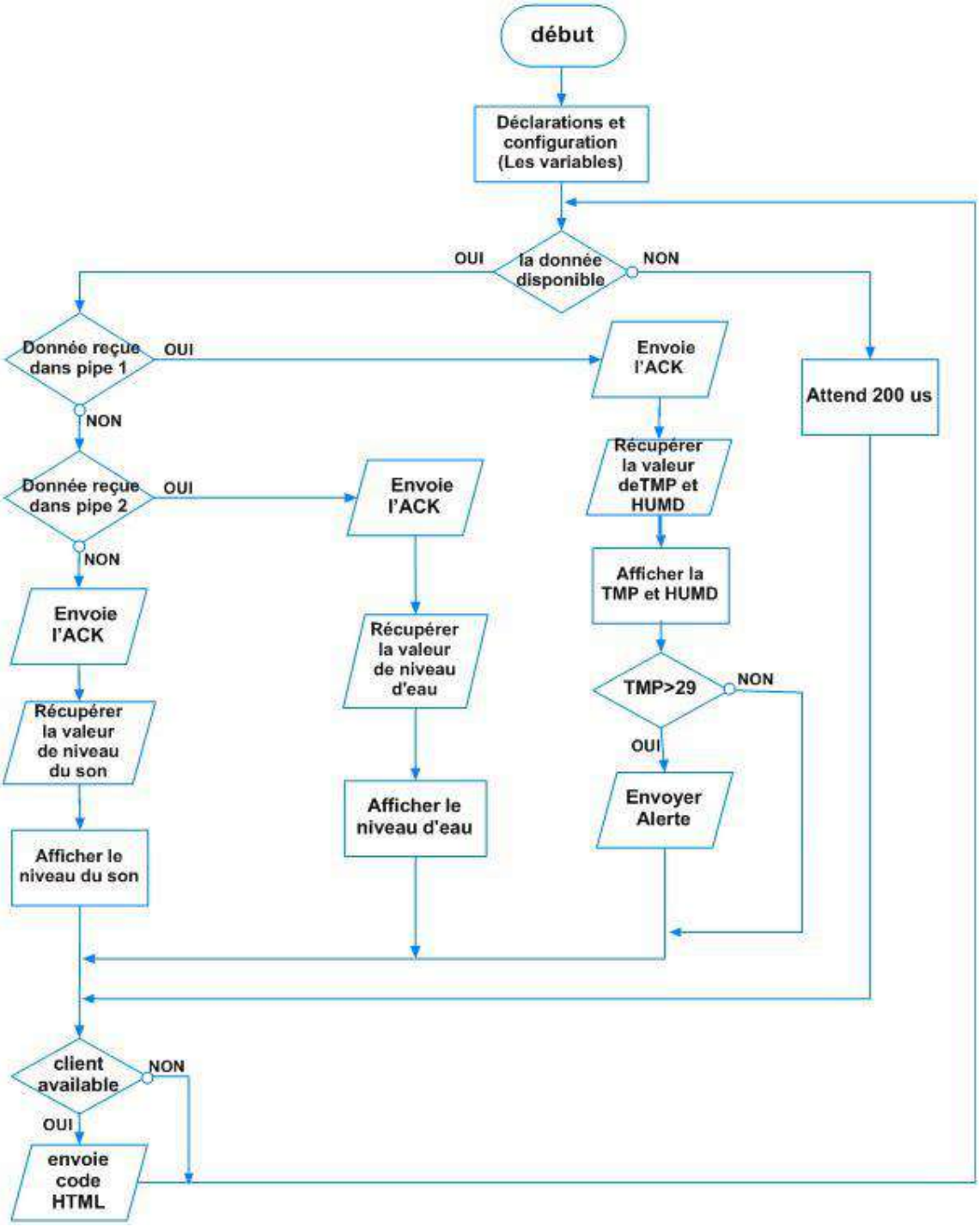


Figure III.5. *L'organigramme de la réception.*

III.3.2.2 Description de mode réception

Dans cette partie l'objective principale est de réaliser un nœud central qui permet la réception multi canal pour cela, l'Arduino vérifier l'état des pipes de son nRF24L01 ou chaque pipe est dédié à un émetteur (on peut communiquer avec 5 nœuds en même temps), si une donnée est présente, la nRF vérifier la pipe qui corresponde à l'émetteur.

Si la pipe est vide donc le nœud centrale reçoit la donnée et répond par un message d'ACK pour indique la bonne réception de donnée. Sinon attend 200 us. Les données collectées sont envoyé par la suite sur Intranet pour l'affichage.

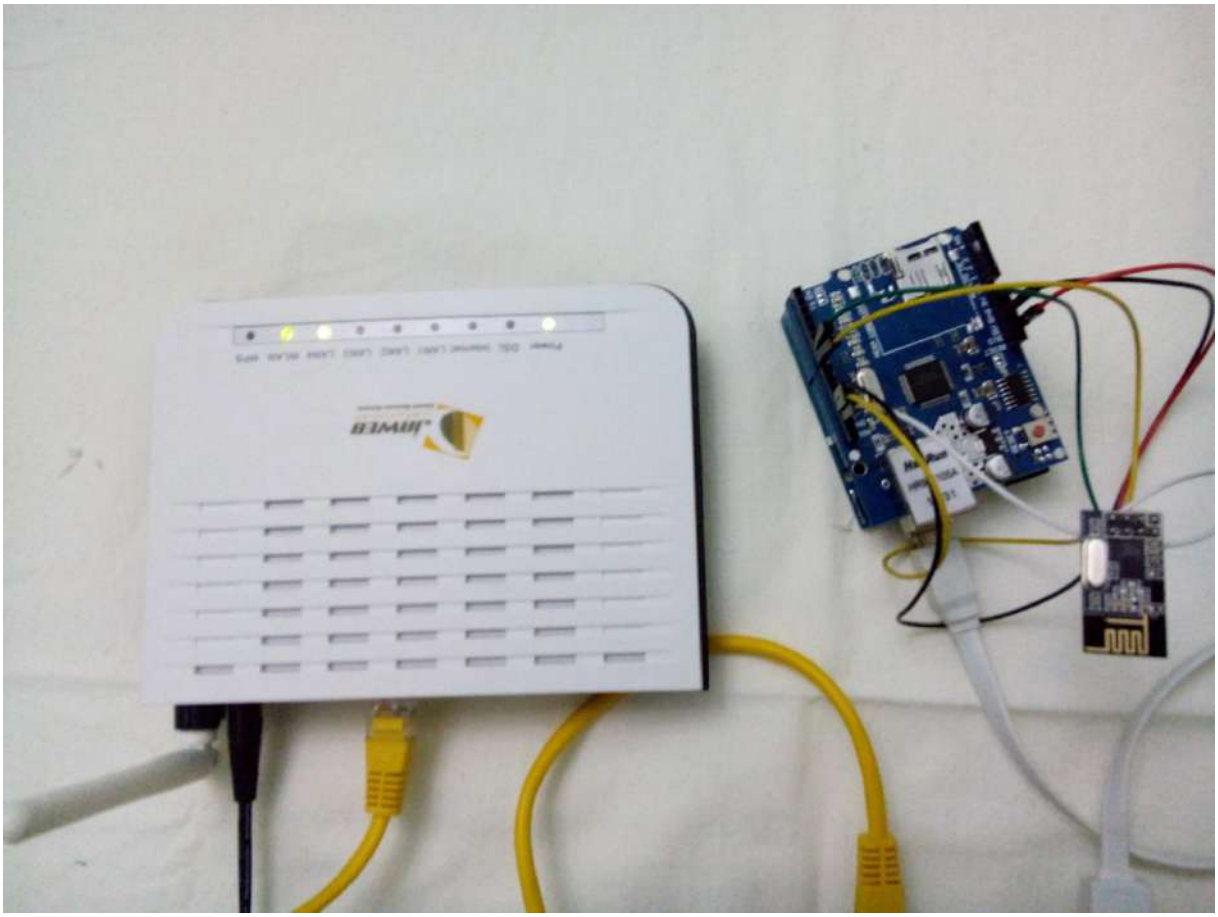


Figure III.6. *Partie réception.*

III.3.3 Le mode actionneur

Pour cette partie les constituants utilisés sont :

- Un module Arduino Uno.
- Le module RF : nRF24L01.
- Une LED.

III.3.3.1 L'organigramme d'actionneur

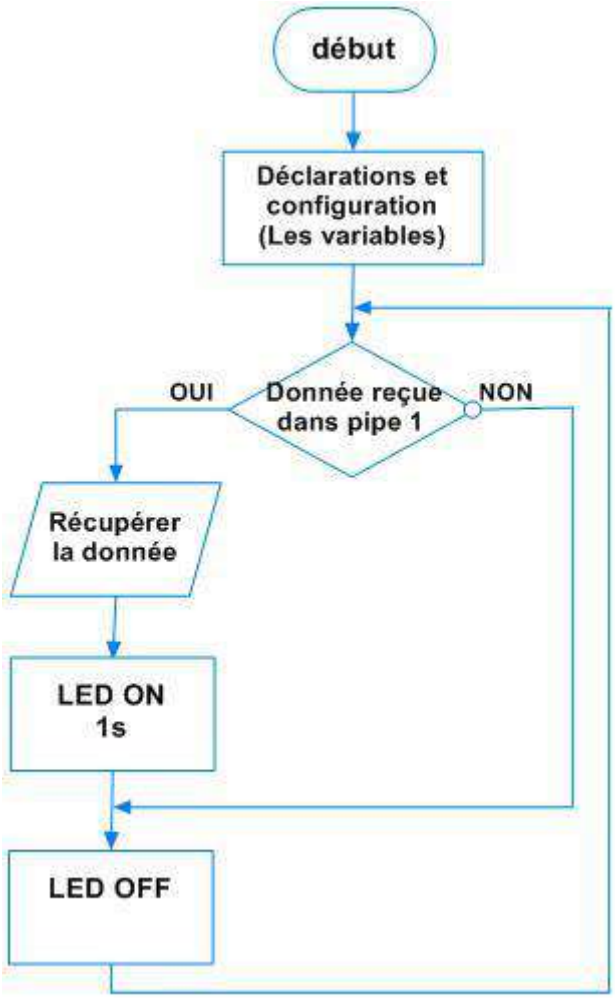


Figure III.7. *L'organigramme de l'actionneur.*

III.3.3.2 Description de mode actionneur

Concernant la partie d'actionneur, l'Arduino vérifier l'état de la pipe de la nRF, si la pipe est pleine donc l'Arduino va récupérer la donnée c'est-à-dire récupérer une valeur (1), donc la LED allumée pendant une seconde, sinon la LED (diode électroluminescente) resteront éteinte, l'objectif de cette partie est commandé un nœud prédéfinie dans se cas par exemple (commander une LED), parce que nous considérons le nœud central est aussi comme un émetteur.

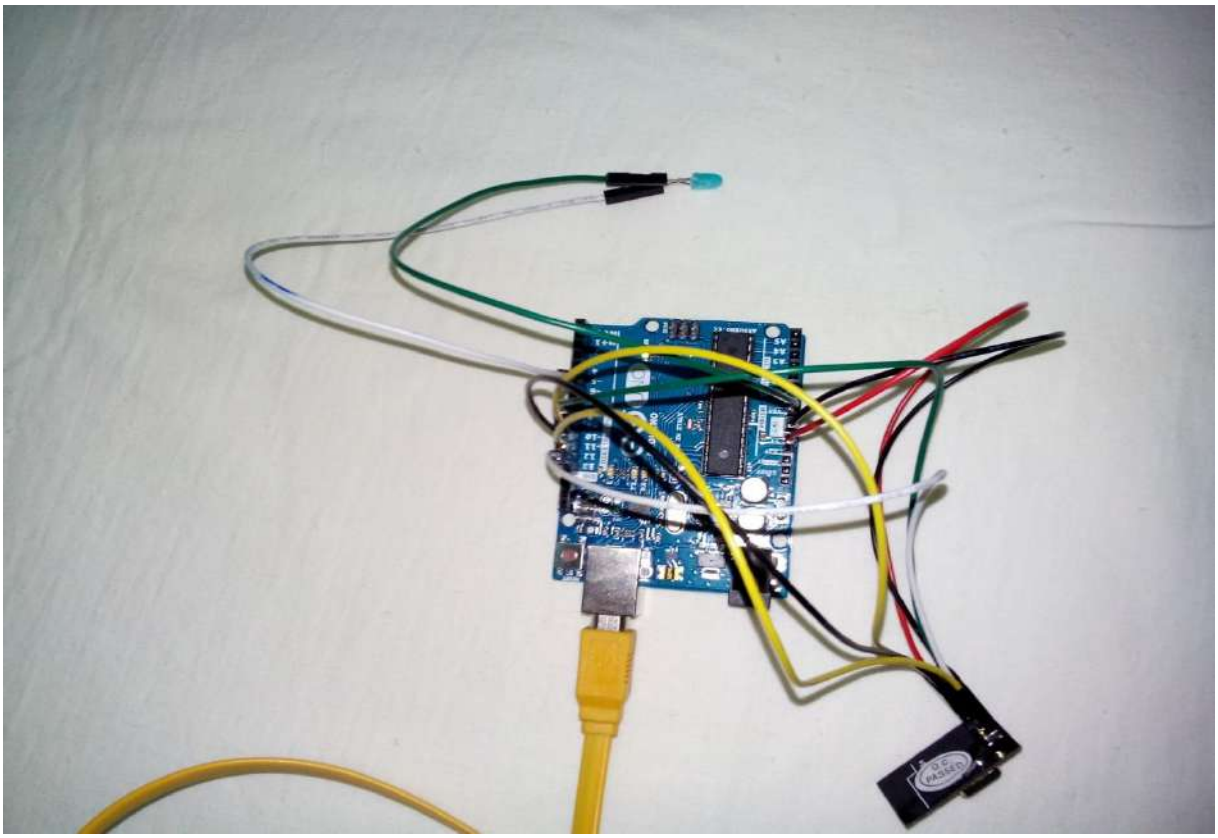


Figure III.8. *Partie d'actionneur.*

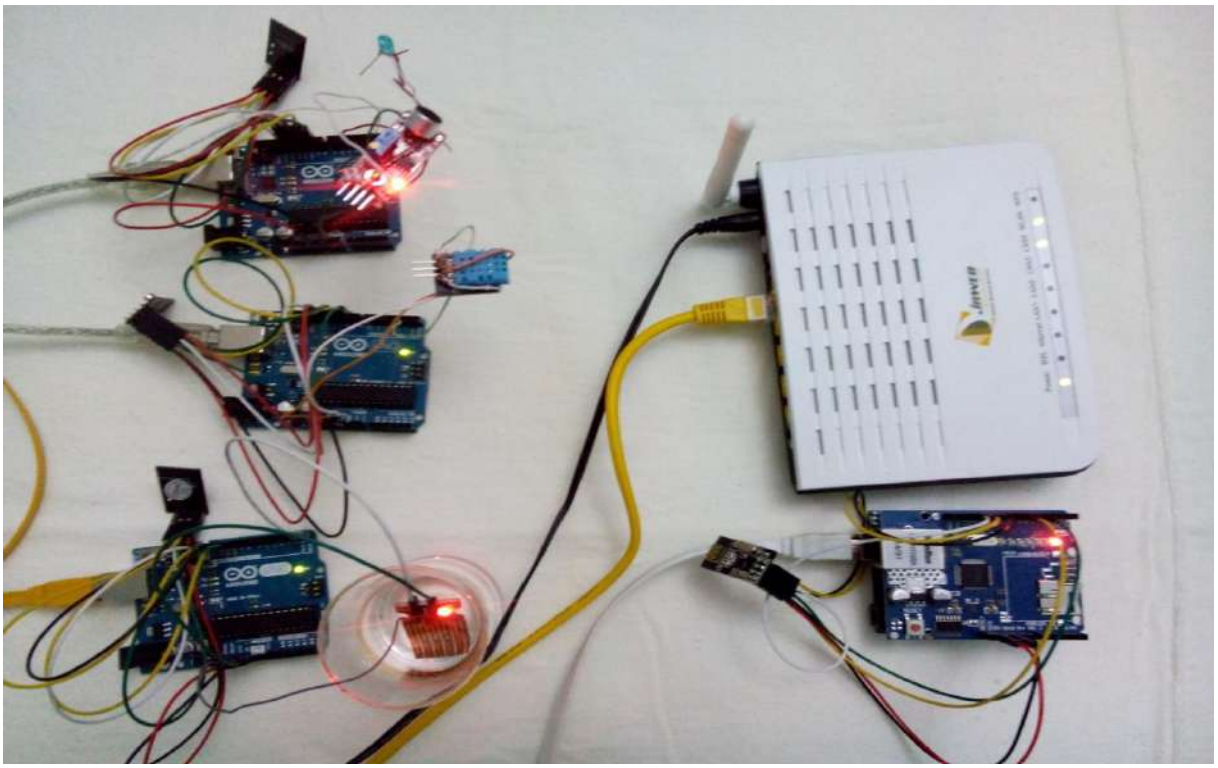


Figure III.9. *Partie émission réception.*

III.4 Résultats

Pour voir les résultats de cette expérience nous avons utilisé le réseau internet pour l'affichage des données à distance comme il est présenté sur la figure III.10.

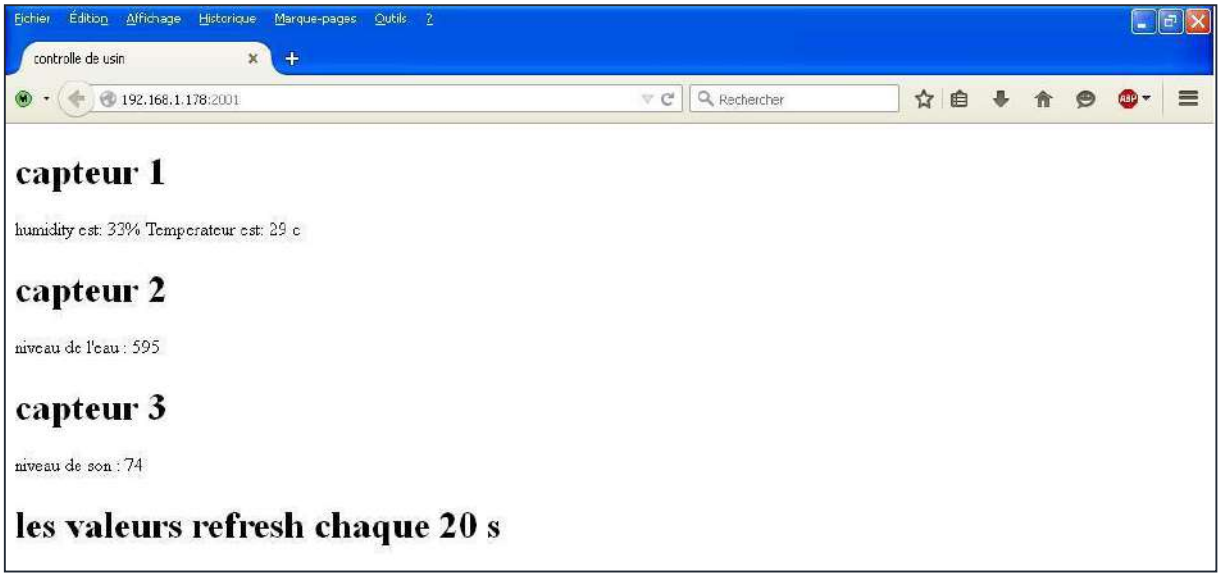


Figure III.10. *Résultats.*

III.5 Conclusion

Nous avons décrit dans ce chapitre tous les étapes de notre réalisation telle que les modules utilisés ainsi que le schéma bloc de réseau a réalisé et les différents diagrammes de chaque partie. Ce travail permet la réalisation d'un nœud de capteur sans fil à faible coût basé sur des modules Arduino qui peut communiquer avec une station de réception via un module de communication RF de type nRF24L01+.

Les résultats obtenus par les différents capteurs sont comparés par des instruments de mesure comme dans le cas de la température et l'humidité ou nous avons obtenu des valeurs avec une incertitude faible comparé avec les valeurs réelles du milieu de mesure.

Conclusion générale

Dans ce projet nous avons réalisés un nœud de capteur sans fil basé sur des composants simples comme l'Arduino, le circuit de transmission RF nRF24L01+ et le Shield Ethernet. La présence des bibliothèques des différents composants utilisés dans ce travail 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. On a choisi ce sujet avec des composants étudiés auparavant. Cela nous a permis de se familiariser avec ce type de composants. Ce fût une expérience enrichissante, car nous avons pu réaliser un projet personnel de manière autonome, Ce projet aussi, nous a permis de rassembler des domaines différents.

En fait, ce projet a été une occasion pour découvrir plusieurs domaines d'études comme la programmation (Arduino, language C). Nous espérons par la suite que le système développé sera l'objet d'une amélioration par d'autres projets.

[1] RAMLI.M et *AGHA.B* (étude et réalisation d'un nœud de capteur sans fil à base d'Arduino Uno) Université Dr. YAHIA Farés de Médéa 2014.

[2] Jean-François PILLOU « Transmission de données - Les modes de transmission.pdf »

[3] « Transmission de données, Interface sérielle et émetteur-récepteur asynchrone universel (UART) », [en ligne]. Disponible : http://www.technologuepro.com/transmission/chapitre2.htm.

[4] Lycées E. Pérochon et J. Desfontaines, section Sciences de l'Ingénieur Cours origine Camille Diou, université de Metz.pdf

[5] Xavier HINAULT (2010) « Le module Ethernet » Arduino, [en ligne]. Disponible :

Http //www.arduino.cc/en/Main/ArduinoBoardEthernet.

Et [en ligne]. Disponible : http://www.mon-club-elec.fr/pmwiki_reference_arduino/

Pm wiki.php? n=Main.Materiel Ethernet Shield.

[6] Product specification (Single chip 2.4 GHz TransceivernRF24L01.pdf)

[En ligne]. Disponible: http://www.nordicsemi.no.

[7] «Single Chip 2.4GHz Transceiver with USB Microcontroller and Flash Memory.pdf**»**

[En ligne]. Disponible: http://www.nordicsemi.no.

[8] « Module Capteur Humidité et Température DHT11 », [en ligne]. Disponible : http://www.zartronic.fr/module-capteur-humidit%C3%A9-et-temp%C3%A9rature-dht11-p-98.html.

Et D- Robotics UK.pdf et [en ligne]. Disponible : http://www.droboticsonline.com.

[9] « Capteur de niveau d'eau », [en ligne]. Disponible : http://Capteur de niveau d'eau - Letmeknow.fr La boutique des objets connectés.htm.

[10] « Capteur de détecteur de son **»,** [en ligne]. Disponible :

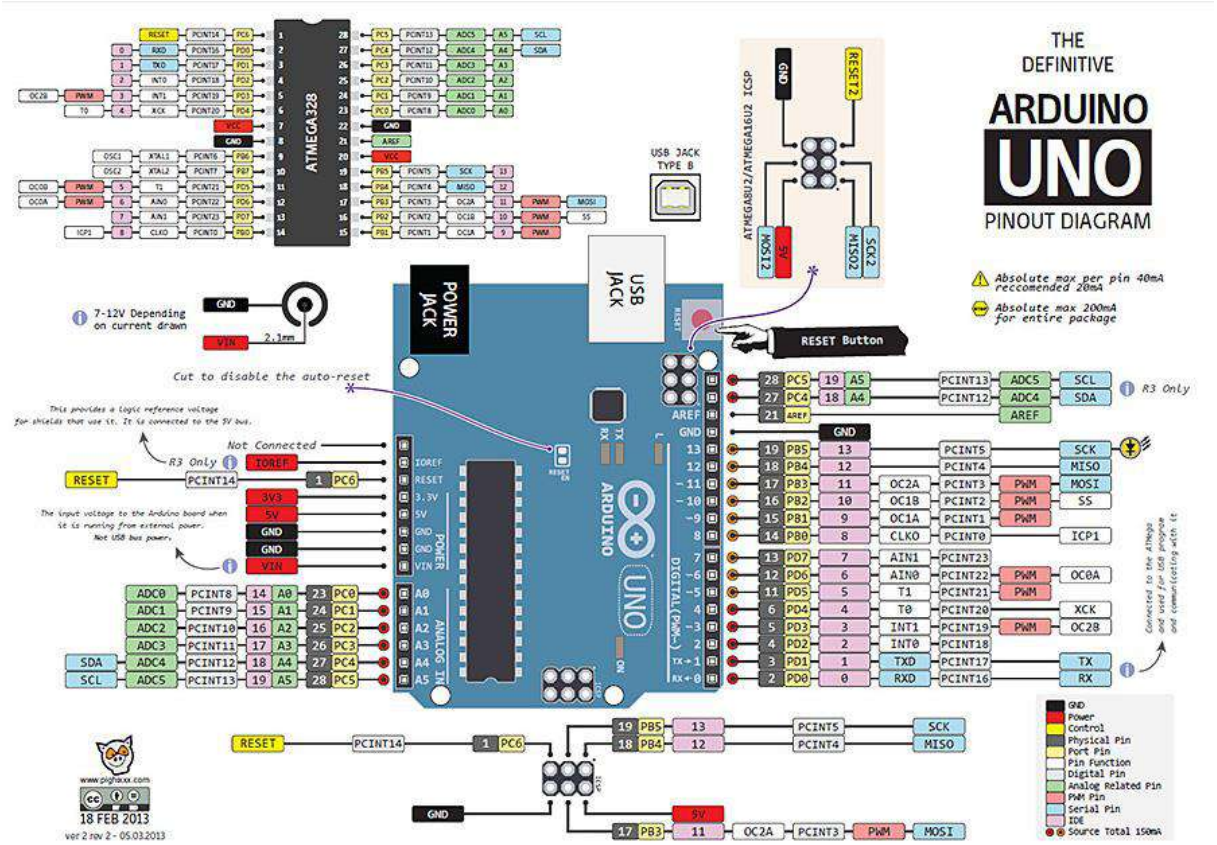
http://fr.aliexpress.com/item/lightbid-Sound-Detection-Sensor-Module-LM393-Chip-Electret-Microphone-For Arduino/32315469066.html.



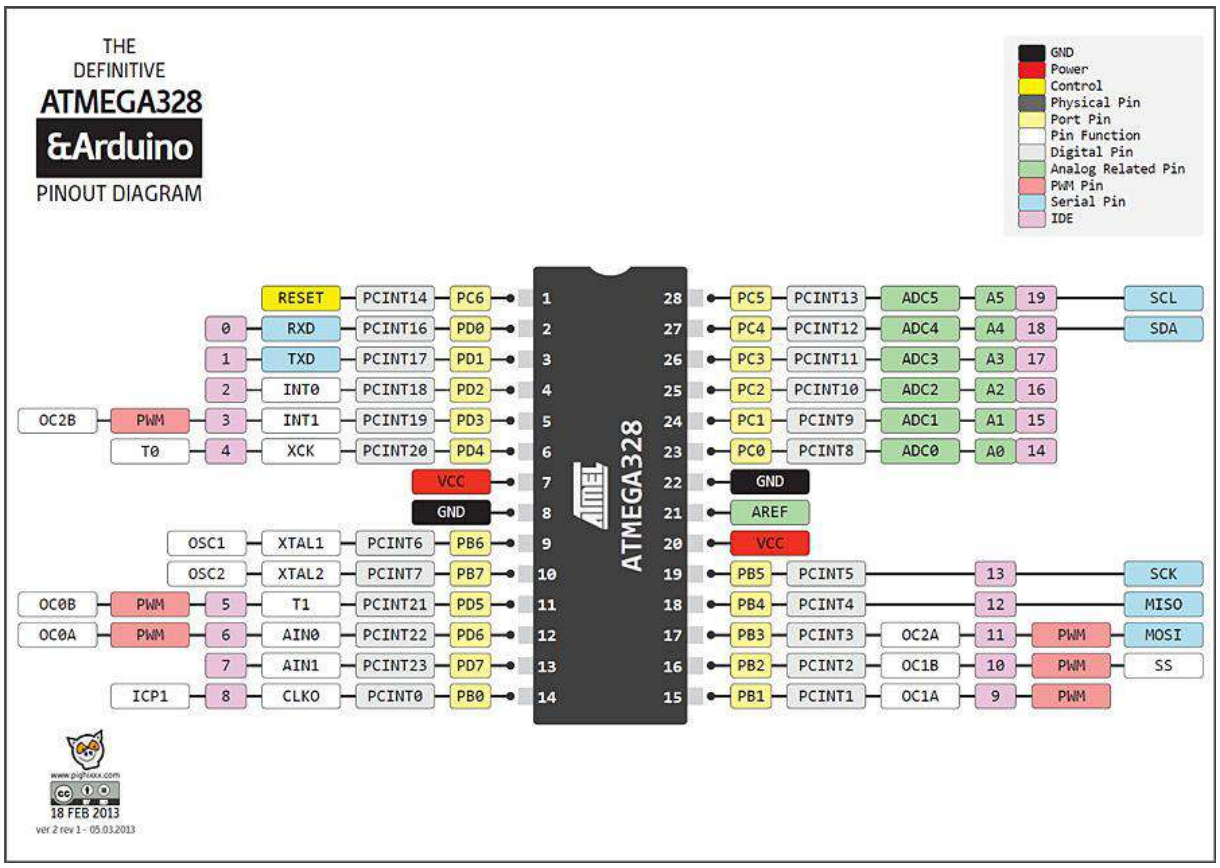
ANNEXE



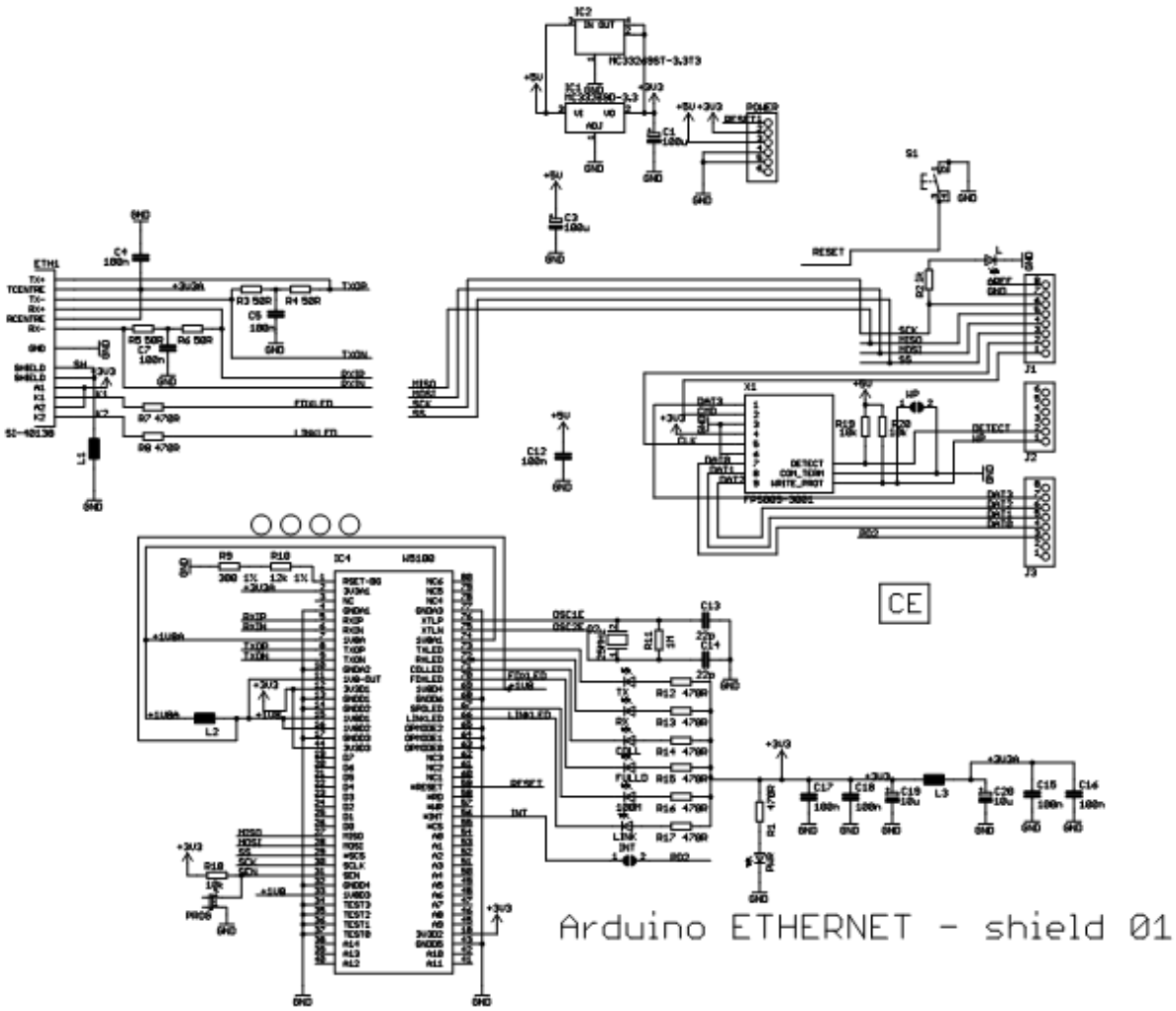
ANNEXE A : Le schéma des pins de l'Arduino UNO



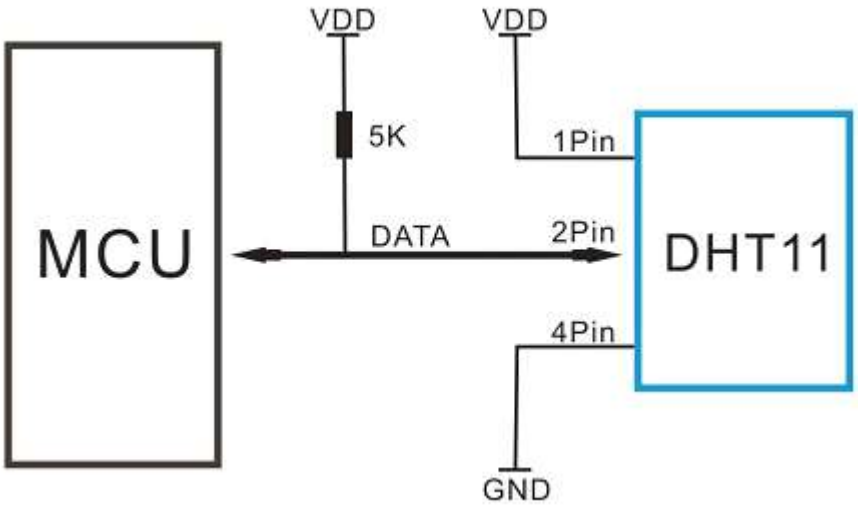
<u>ANNEXE B :</u> Le schéma des pins de l'atmega 328

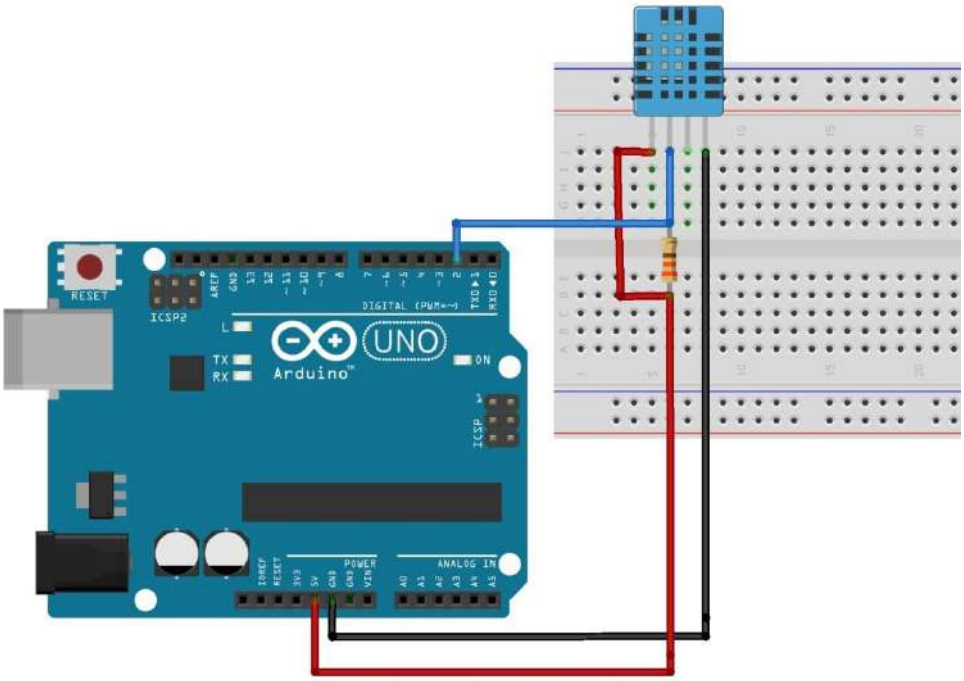


<u>ANNEXE C :</u> Le schéma de l'Arduino Ethernet



ANNEXE D : Le capteur DHT 11





ANNEXE E : Programme de l'émission

1- le capteur DHT11

```
#include<SPI.h>
#include<nRF24L01.h>        } On utilise 3 bibio (pour la gestion de spi ….)
#include<RF24.h>
#include<dht.h>          // la bibio de DHT
#definedht_dpin 2   //communication avec le capteur DHT11 Port Analogique 2
dht DHT; intrec[1] = {5}; // vecteur de type int pour numéroter les ACK reçues
float time; // pour défini le temps de transmission
float time1;  // pour défini le temps de réception de l'ACK
byte input[32]="";  //vecteur des caractère .
const uint64_t pipe[1] = {0xE8E8F0F0E2LL};  // l'adresse du capteur
RF24 radio(9, 10);     //  CE sur broche 9 et CSN sur broche 10
voidsetup(){      // l'initialisation de l'Arduino au démarrage
Serial.begin(57600);  // Initialisation du port série
radio.begin();
delay(100);  // l'attente entre chaque tentatives
radio.setAutoAck(true);    //activer l'auto ACK
radio.enableAckPayload();    // permettre l'ACK de la charge
radio.enableDynamicPayloads();    //la charge dynamique
radio.setDataRate(RF24_250KBPS);   //définie la vitesse de transmissions
radio.setCRCLength(RF24_CRC_16);     // pour calculer le CRC (8 ou 16).
radio.setPALevel(RF24_PA_HIGH);    //puissance maximum
radio.stopListening();     //désactiver la réception
radio.openWritingPipe(pipe[0]);   //activer l'émission dans la pipe 0.(ouvrir l'écriture)
radio.setRetries(15,15);      // définie le nbr et le retard de retransmission.
Serial.println("Humidity and temperature\n\n");
delay(2000);}
voidloop(){      // fct en permanence
intchk=DHT.read11(dht_dpin);    // la fonction de lecture d'un octet retourne l'octet lu
```

```
switch (chk){
case DHTLIB_OK:
Serial.print("  lecture OK,\t");
input[0]=DHT.humidity;
input[1]=DHT.temperature;  // input [] -> buffer contenant les données du capteur.
// *Affichage de la température et du taux d'humidité*//
radio.write(input, sizeof(input));   // émis le caract qui place dans l'input
Serial.println("...tx success");
Serial.print("Current humidity = ");
Serial.print(DHT.humidity);
Serial.print("% ");
Serial.print("temperature = ");
Serial.print(DHT.temperature);
time = micros();     //le temps en micro sec.
Serial.println("...tx success");    // dit le prog TX est transmis
if (radio.isAckPayloadAvailable()) {    // vérifié que l'ACK est reçue.
radio.read(rec, sizeof(int));    //lire le num de l'ACK dans le vecteur rec.
Serial.print(micros() - time);
Serial.print(".....received ack payload is : ");      // pour dire l'ACK est recue.
Serial.println(rec[0]); }
else  {
time1 = micros();
Serial.print(micros() - time1);
Serial.println(" pas de ACK ...."); }    // l'ACK n'est pas reçue.
break;
case DHTLIB_ERROR_CHECKSUM:
Serial.print("Checksum error,\t");
break;
case DHTLIB_ERROR_TIMEOUT:
Serial.print("Time out error,\t");
```

```
break;
default:
Serial.print("Unknown error,\t");
break; }       // la fin de Switch
delay(5000);}        //*Pas besoin de rafraichir l'affichage très souvent
```

2- capteur de niveau d’eau

```
#include<SPI.h>
#include<nRF24L01.h>
#include<RF24.h>
int capteur =0;      // Le capteur est sur la pin A0
intval[32]={0};int rec[1] = {5};
float time;float time1;
const uint64_t pipe[1] = {0xE8E8F0F0E3LL};
RF24 radio(9, 10);
void setup(){
Serial.begin(57600);
radio.begin();
delay(100);
radio.setAutoAck(true);
radio.enableAckPayload();
radio.enableDynamicPayloads();

radio.setDataRate(RF24_250KBPS);
radio.setCRCLength(RF24_CRC_16);
radio.setPALevel(RF24_PA_HIGH);
radio.stopListening();
radio.openWritingPipe(pipe[0]);
radio.setRetries(15,15);
delay(2000);}
```

```
void loop(){
val[0] = analogRead (capteur);
radio.write(val, sizeof(val));
Serial.println (val[0]);
time = micros();
Serial.println("...tx success");
if (radio.isAckPayloadAvailable()){
radio.read(rec, sizeof(int));
Serial.print(micros() - time);
Serial.print(".....received ack payload is : ");
Serial.println(rec[0]);}
else{
time1 = micros();
Serial.print(micros() - time1);
Serial.println(" pas de ACK ....");}
delay(5000);
}
```

3- capteur de détection de son

```
#include<SPI.h>
#include<nRF24L01.h>#include<RF24.h>
int volume[32]={0};int rec[1] = {5};
float time;float time1;
int led = 8;int threshold = 69; // le seil
const uint64_t pipe[1] = {0xE8E8F0F0E4LL};
RF24 radio(9, 10);
void setup(){
Serial.begin(57600);
radio.begin();delay(100);
radio.setAutoAck(true);
```

```
radio.enableAckPayload();
radio.enableDynamicPayloads();
radio.setDataRate(RF24_250KBPS);
radio.setCRCLength(RF24_CRC_16);
radio.setPALevel(RF24_PA_HIGH);
radio.stopListening();
radio.openWritingPipe(pipe[0]);
radio.setRetries(15,15);
pinMode(led, OUTPUT);
delay(2000);}
void loop(){
volume[0] = analogRead (A0);
if(volume[0]>=threshold){    //la condition pour allumer la LED
digitalWrite(led, HIGH);
radio.write( volume, sizeof(volume));
Serial.println (volume[0]);
time = micros();
Serial.println("...tx success");
if (radio.isAckPayloadAvailable()){
radio.read(rec, sizeof(int));
Serial.print(micros() - time);
Serial.print(".....received ack payload is : ");
Serial.println(rec[0]);}
else{
time1 = micros();
Serial.print(micros() - time1);
Serial.println(" pas de ACK ....");}}
else{
digitalWrite(led, LOW);    // Eteignez la Led.
}}
```

ANNEXE F : Programme de réception

```
#include<SPI.h>// bibliothèque de communication SPI avec le Shield Ethernet
#include<nRF24L01.h>  //bibliothèque de communication de nRF.
#include<RF24.h>
#include<Ethernet.h>      // bibliothèque de communication Ethernet sur un réseau local
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED }; //Adresse MAC du shield Ethernet
byte ip[] = { 192,168,1,178};    // Adresse IP donné au Shield Ethernet
byte gateway[] = {192,168,1,1 };        // accès à internet via un routeur passerelle par défaut
byte subnet[] = { 255, 255, 255, 0 };              //masque de sous réseau
EthernetServer server(2001);          //Le Shield devient un serveur sur le port 80 (port HTTP)
String readString;
const uint64_t pipe[6]
={0xF0F0F0F0E1LL,0xE8E8F0F0E2LL,0xE8E8F0F0E3LL,0xE8E8F0F0E4LL,0xE8E8F0F
0E5LL,0xE8E8F0F0E6LL};     // les adresses des pipes
RF24 radio(6, 7);      // les broches de CE et CSN pour le sheild.
int rec1[1]={0};int rec2[1]={0};int rec3[1]={0};int rec4[1]={0};int rec5[1]={0};
int rec6[1]={0};
byte red1[32];int red2[32];int red3[32];int red4[32];byte red5[32];int red6[32];
int input[32]={1};
inti;
void setup(){
Serial.begin(57600);  // initialisation du port série à 57600 bauds

radio.begin();delay(100);
radio.enableAckPayload();
radio.setAutoAck(true);
radio.enableDynamicPayloads();
radio.setDataRate(RF24_250KBPS);
radio.setCRCLength(RF24_CRC_16);
radio.setPALevel(RF24_PA_HIGH);
radio.openReadingPipe(0,pipe[0]);    // ouvrir la réception dans la pipe n=.
radio.openReadingPipe(1,pipe[1]);
```

```
radio.openReadingPipe(2,pipe[2]);
radio.openReadingPipe(3,pipe[3]);
radio.openReadingPipe(4,pipe[4]);
radio.openReadingPipe(5,pipe[5]);
radio.startListening();    //commence à écouter(la réception)
radio.setRetries(15, 15);
while (!Serial) {
; // attendre pour le port série pour vous connecter.
}
/*Lancement connexion Ethernet avec l'adresse MAC et l'adresse IP */
Ethernet.begin(mac, ip, gateway, subnet);
server.begin();
Serial.print("server is at ");
Serial.println(Ethernet.localIP());}
void loop() {
uint8_t pipe_num;
if(radio.available(&pipe_num) ) {   // vérifie l’état de la pipe.
i= pipe_num ;
switch (i) {case 0:break;
case 1:
radio.writeAckPayload( i,rec1,sizeof(int));    //envoi  ACK
radio.read(&red1,sizeof(red1) );      //lire le message
Serial.print("le message recue dans le Capteur--> ");
Serial.print(i);
Serial.print(" humidity = ");
Serial.print(red1[0]);
Serial.print("% ");
Serial.print("    temperature = ");
Serial.println(red1[1]);
Serial.print("  c");
```

```
rec1[0] += 1;
if(red1[1]>21){    //la condition pour arrêter  la réception.
radio.stopListening();
delay(1000);
radio.openWritingPipe(pipe[0]);
radio.write(input, sizeof(input));
radio.startListening();}
if(rec1[0]==100){rec1[0]=0;}
break;
case 2:
radio.writeAckPayload( i,rec2,sizeof(int));
radio.read(&red2,sizeof(red2) );
Serial.print("le niveau d'eau  dans le Capteur--> ");
Serial.print(i);
Serial.print(" est: ");
Serial.print(red2[0]);
Serial.println("");
rec2[0] += 1;
if(rec2[0]==100){rec2[0]=0;}
break;
default:
radio.writeAckPayload( i,rec3,sizeof(int));
radio.read(&red3,sizeof(red3) );
Serial.print("niveau son  dans le Capteur--> ");
Serial.print(i);
Serial.print(" est: ");
Serial.print(red3[0]);
Serial.println("");
rec3[0] += 1;
if(rec3[0]==100){rec3[0]=0;}break;}}
```

```
else{
delay(0.2);}
//Pour récupérer les nouveaux clients qui viennent se connecter à votre Arduino
EthernetClient client = server.available();
if (client) {// s'éxectute s’il y a un client

Serial.println("new client");   // une requête http se termine par une ligne vide
booleancurrentLineIsBlank = true;
while (client.connected()) {     //le client est disponible.
if (client.available()) {
char c = client.read();
Serial.write(c);
// si vous avez obtenu à la fin de la ligne (reçu une nouvelle ligne de caractère) et  si la ligne est vide, la requête http est terminée,
// vous  pouvez donc envoyer une réponse
if (c == '\n' &&currentLineIsBlank){
// envoyer un en-tête de réponse http standard.
client.println("HTTP/1.1 200 OK");
client.println("Content-Type: text/html");
client.println("Connection: close");  // la connexion sera fermée après l'achèvement delaréaction
client.println("Refresh: 20");  // actualiser la page automatiquement toutes les 20 secondes
//* l’affichage*//
client.println();
client.println("<!DOCTYPE HTML>");
client.println("<html>");
client.print("<head><title>controlle de usin</title></head><body><h1>capteur 1 </h1><p> humidity est:  ");
client.print(red1[0]);
client.print("%  ");
client.println("     Temperateur est:");
```

```
client.print(red1[1]);
client.print(" c");
client.print("</head><body><h1>capteur 2 </h1><p>niveau de l'eau  :  ");
client.print(red2[0]);
client.print("</head><body><h1>capteur 3 </h1><p>niveau de son :  ");
client.print(red3[0]);
client.print("</head><body><h1> les valeurs refresh claques  20 s ");break;}
if (c == '\n') {
// Vous commencez une nouvelle ligne.
currentLineIsBlank = true;}
else if (c != '\r') {
// *Vous avez obtenu un caractère sur la ligne actuelle*//
currentLineIsBlank = false;
}}}
// Laisser le temps au navigateur Web pour recevoir les données.
delay(1);
// * fermer la connexion *//
client.stop();
Serial.println("client disonnected");
}}
```

ANNEXE G : Programme de la partie d'actionneur

```
#include<SPI.h>#include<nRF24L01.h>#include<RF24.h>
const uint64_t pipe[1] ={0xF0F0F0F0E1LL};
RF24 radio(9, 10);int rec[1]={0};int red[32];
inti;int led = 6;
void setup(){
Serial.begin(9600);
radio.begin();
delay(100);
radio.setAutoAck(false);    //désactiver l'auto ACk
radio.enableDynamicPayloads();
radio.setDataRate(RF24_250KBPS);
radio.setCRCLength(RF24_CRC_16);
radio.setPALevel(RF24_PA_HIGH);
radio.openReadingPipe(0,pipe[0]);
radio.startListening();
radio.setRetries(15, 15);
digitalWrite(led,LOW);}   //la LED est déjà éteinte.
voidloop() {
uint8_t pipe_num;
while (radio.available(&pipe_num) ) {     //vérifier l'état de la pipe.
i= pipe_num +1;
radio.read(&red,sizeof(red) );
digitalWrite(led, HIGH);
delay(1000);
digitalWrite(led,LOW);}}
```

ANNEXE H : Le Bronchement de nRF24L01 avec l'Arduino Uno

Broche nRF24L01	Broche Arduino
GND	GND
V_{CC}/3.3v	3.3v
CE	9 (configurable)
CSN	10 (configurable)
SCK	13
MOSI	11
MISO	12
IRQ	non connectée

ANNEXE I : Le Bronchement de nRF24L01 avec l'Arduino Ethernet

Broche nRF24L01	Broche Arduino
GND	GND
V_{CC}/3.3v	3.3v
CE	6 (configurable)
CSN	7 (configurable)
SCK	13
MOSI	11
MISO	12
IRQ	non connectée

<u>ANNEXE J:</u> Une figure qui présente le module avec les broches

