Mohamed Khider University
of Biskra Faculty of Sciences
and Technology Department
of Electrical engineering

MASTER'S THESIS

Electrical Engineering
Telecommunications
Networks and Telecommunications

Submitted and Defended by:

Zidi Bouthaina

On: September 2020

Supervision and detection of faults in electrical machines using a network of XBee-based sensors

Board of Examiners:

Ms.	Tobbech Souad	**Pr**	Universityof Biskra	**President**
Mr.	Guesbaya Tahar	**MCA**	Universityof Biskra	**Examiner**
Mr.	Boumahrez Mouhamed	**Pr**	Universityof Biskra	**Supervisor**

Academic Year: 2019-2020

الجمهورية الجزائرية الديمقراطية الشعبية
People's Democratic Republic of Algeria
وزارة التعليم العالي و البحث العلمي
Ministry Of Higher Education and Scientific Research

Mohamed Khider Biskra
University Faculty of Sciences
and Technology Electrical
Engineering Department Field:
Telecommunication
Option: Networks and Telecommunication
A Dissertation for the Fulfillment of the Requirement of a

Master's Degree

Supervision and detection of faults in electrical machines using a network of XBee-based sensors

Presented by:

Zidi Bouthaina

Favorable opinion of the supervisor:

Pr. Boumahrez Mouhamed

Favorable opinion of the Jury President

Ms. Tobbech Souad

Stamp and signature

الجمهورية الجزائرية الديمقراطية الشعبيه
People's Democratic Republic of Algeria
وزارة التعليم العالي و البحث العلمي
Ministry Of Higher Education and Scientific Research

Mohamed Khider Biskra
University Faculty of
Sciences and Technology
Electrical Engineering
Department Field:
Telecommunication
Option: Networks and Telecommunication

Theme:

Supervision and detection of faults in electrical machines using a network of XBee-based sensors

Proposed by: Pr. Boumahrez Mouhamed
Directed by: Pr. Boumahrez Mouhamed

ABSTRACT

In this work, we will create a network of XBee-based sensors to send and receive data, whose goal is to supervise and detect faults in electrical machines using a network of XBee-based sensors.

The system is powered by Arduino, and wireless communication between each node is provided by the XBee module.

Keyword: Wireless sensor network, Supervision, Detection, Remote control, ZigBee, XBee, X-ctu.

Dedication

To my dear mother

Whatever I do or say, I cannot thank you properly, because your affection covers me, your kindness that keeps me, and being by my side has always been my strength to face various obstacles.

To my dear father

You have been by my side to support and encourage me; you have given me so much confidence and love. I hope this work reflects my gratitude, my passion and the fruit of your sacrifices that you have made for my education and training.

To my dear **grandfather** and **grandmother**

To my dear **brother Adam** and my dear **sisters Djalila**, **Nadjah**, **Israa** and my niece **Larine**.

To **Tata Malika** who did not stop encouraging me and praying for all my love and appreciation to you.

To all my dear family.

To all my friends who have always encouraged me and for whom I wish more success, especially: **Belguidoum Rym, Djenan Amel, Atamnia Tahani, and Atamnia Yousra**, **Kechem Nourelhouda, Ziadi Echikha.**

To all my teachers throughout my school life.

To everyone who tries to make an effort for the good of my country, Algeria, and the best of my religion: Islam.

For someone who believes in me.

I dedicate this humble work.

Thank's

"Success is based on the perseverance, the hard work and the Patience" **Thomas Edison**

First of all, I thank Almighty «**ALLAH**» God who granted us the will, the patience, and above all the health throughout our course to make this work come true.

Thank supervise Mr Boumahraz Mohamed, for providing us with his valuable advice during the preparation of this work.

I would also like to express my sincere thanks to the members of the Jury who were kind enough to judge this work:

Mrs.Tobbeche souad, President of the Jury.

Mr. Guesbaya Taher, Examiner of the Jury.

I thank my families who have always given me the opportunity to do what I want during my studies and who have always believed in me.

Finally, I thank everyone who has contributed to this work with their comments, suggestions, and support helped me directly or indirectly to the realization of this work.

TABLE OF CONTENTS

Chapter I: Wireless Sensor Networks

Chapter II: Study and development of technology ZigBee

Chapter III: Practical Realization

Figures List

CHAPTER I

CHAPTER II

CHAPTER III

Tables List

CHAPTER II

CHAPTER III

Acronyms List

AES: **A**dvanced **E**ncryption **S**tandard

ADC: **A**nalog to **D**igital **C**onverter

AF: **A**pplication **F**ramework

APL: **A**pplication **L**ayer

APO: **A**pplication **O**bjects

APS: **A**pplication **S**upport sub-layer

API: **A**pplication **P**rogramming

AT: **I**nterface **A**pplication **T**ransparent

BS: **B**ase **S**tations

CCM: **C**ompression **C**lient **M**anage

CH: **C**luster **H**ead

CSMA-CA: **C**arrier **S**ense **M**ultiple **A**ccess-**C**ollision **A**voidance

DIO: **D**igital **I**nput **O**utput

DSSS: **D**irect **S**equence **S**pread **S**pectrum

FFD: **F**ull **F**unction **D**evices

GPS: **G**lobal **P**ositioning **S**ystem

GTS: **G**uaranteed **T**ime **S**lot

ID: **ID**entificator

IDE: **I**ntegrated **D**evelopment **E**nvironment

IEEE: **I**nstitute of **E**lectrical and **E**lectronics **E**ngineers

ISM: **I**ndustrial, **S**cientific **M**edical

IP: **I**nternet **P**rotocol

LP-WPAN: **L**ow **P**ower-**W**ireless **P**ersonal **A**rea **N**etwork

MANETs: **M**obile ad hoc networks

MAC: **M**edia **A**ccess **C**ontrol

MIC: **M**essage **I**ntegrity **C**ode

NWK: **N**etwork **L**ayer

OSI: **O**pen **S**ystem **I**nterconnection

PAN: **P**ersonal **A**rea **N**etwork

PC: **P**ersonal **C**omputer

PER: **P**acket **E**rror **R**ate

PHY: **P**hysical **L**ayer

PWM: **P**ulse-**W**idth **M**odulation

QoS: **Q**uality **o**f **S**ervice

RF: **R**adio **F**requency

RFD: **R**educed **F**unction **D**evice

RX: **R**eception

SAR: Sequential **A**ssignment **R**outing

SN: **S**ensor **N**ode

SSID: Service Set **ID**entifier

TCP: **T**ransmission Control Protocol

TC: **T**rust **C**enter

TX: **T**ransmission

UDP: **U**ser **D**atagram **P**rotocol

USB: **U**niversal **S**erial **B**us

Wi-Fi: **W**ireless **F**idelity

WSN: **W**ireless **S**ensor **N**etwork

ZDO: **Z**igBee **D**evice **O**bject

Abstract

In this work, we will create a network of XBee-based sensors to send and receive data, whose goal is to supervise and detect faults in electrical machines using a network of XBee-based sensors.

The system is powered by Arduino, and wireless communication between each node is provided by the XBee module.

Keyword: Wireless sensor network, Supervision, Detection, Remote control, ZigBee, XBee, X-ctu.

ملخص

في هذا العمل، سنقوم بإنشاء شبكة من أجهزة الاستشعار المستندة إلى XBee لإرسال واستقبال البيانات، والغرض منها هو مراقبة واكتشاف الأعطال في الآلات الكهربائية باستخدام شبكة من أجهزة الاستشعار المستندة إلى XBee.

يتم تشغيل النظام بواسطة Arduino ويتم توفير الاتصال اللاسلكي بين كل عقدة بواسطة وحدة XBee.

الكلمات المفتاحية : شبكة الاستشعار اللاسلكية , الإشراف ,الكشف , التحكم عن بعد, X-ctu, XBee, ZigBee.

General Introduction

General Introduction

As wireless technology matured, Wireless Sensor Networks (WSN) began to emerge as an advantageous alternative to their wired counterparts due in part to easy deployment and scalability. The 802.15.4 IEEE communication standard was developed for use specifically with low-rate wireless personal area networks (LR-WPANs) with a focus on wireless sensor networks. In the early 2000s, the ZigBee alliance worked to construct the ZigBee protocols, communication protocols functioning on the 802.15.4 MAC and Physical layers. The main advantage of the ZigBee protocols over its competitor Bluetooth and Wi-Fi was ZigBee’s' highly efficient sleep mode; ZigBee devices use a basic master-slave configuration suited for low frequency data transmission star topologies, and can wake from sleep and transmit a packet in around 15 milliseconds. As a result, ZigBee devices can last for long periods on a single power supply.

In recent years, Digi incorporated the 802.15.4 standard and ZigBee protocols into a proprietary RF module known as the XBee. XBee devices have modular firmware capable of constructing various network topologies and have been utilized as end devices in wireless sensor network and monitoring applications.

However, XBee does not contain large processors for signal processing or local data analysis at the End Device. The limited processing capabilities of an XBee device can be addressed with the implementation of additional hardware for processing support. Current WSN designs utilize an Arduino, a low-cost, reliable microcontroller capable of functioning as a building block for data acquisition or control systems, to augment a sensor node processing capabilities.

The objective of this project is to create of an XBee-based sensor network to send and receive data and then supervision and detection of faults in electrical machines using a network of XBee-based sensors.

This project is made up of two parts; the first part uses the X-CTU application to configure the XBee RF modules for the ZigBee network using 802.15.4 TH from (Part Number: XB24-CZ7-WIT) from www.digi.com.The second part involves the use of Arduino development board to implement the connecting Network using the XBees RF modules.

This Dissertation is organized in three chapters:

Chapter I: In this chapter, we provide given a general view the wireless sensor networks by describing their architecture, types, characteristics, and then application act...

Chapiter II: In this chapter we will explain the ZigBee protocol with its various characteristics and topologies, and explain the XBee module that we will work with in our project.

Chapiter III: Finally, we conclude this work it with the practical realization it's to create of an XBee-based sensor network to send and receive data and then supervision and detection of faults in electrical machines using a network of XBee-based sensors.

This work ends with a general conclusion and a bibliography.

Chapter I

Wireless Sensor Networks

I.1 Introduction

Wireless sensor network (WSN) has emerged as one of the most promising technologies for the future. This has been enabled by advances in technology and availability of small, inexpensive, and smart sensors resulting in cost effective and easily deployable WSNs. With WSNs, we can more effectively analyze everything from rain forests and river deltas to the health and safety of buildings and bridges. A WSN consists of spatially distributed measurement devices that use sensors to monitor physical or environmental conditions. In addition to many wireless measurement nodes, a WSN system includes a gateway that collects data and provides connectivity back to a host application on a PC or embedded controller.

In this section we present WSNs: Their characteristics, their architectures and their applications, to understand them.

I.2 History of Wireless Sensor Network (WSN)

Though rapid interest and research in WSN fields have taken place only recently but, use of sensors for specialized services is not new. During the Cold War, quiet Soviet submarines were detected by deploying the Sound Surveillance System (SOSUS), which employed acoustic sensors. These systems are now adopted by National Oceanographic and Atmospheric Administration (NOAA) for sensing the events in the oceans. Simultaneously, Air defense radar networks were developed employing aerostats as sensors. The predecessor to the internet, Advanced Research Project Agency (ARPANET) formed by US DARPA in 1969, served as a test bed for new networking technologies connecting various universities and research centers. A sensor network can be assumed to have many spatially distributed autonomous sensing devices which route the information to a node which can make the best use of the acquired information. The actual WSN may be traced back to the Distributed Sensor Networks (DSN) program which started in 1980 at Defense Advanced Research Projects Agency (DARPA) **[1]**.

Recent advances in micro fabrication technologies have made it possible to produce tiny nodes which can house multiple sensors and have reasonable processing and communication capabilities. In addition to this, development of wireless networking standards having security, stability and minimum end to end delays have led to proliferation of WSN in to the field of control and monitoring the area which was unheard of earlier. The usage of WSN is increasing exponentially due to the features such as: Scalability, Adaptability, Convenience, Mobility, Accessibility, low cost etc **[1]**.

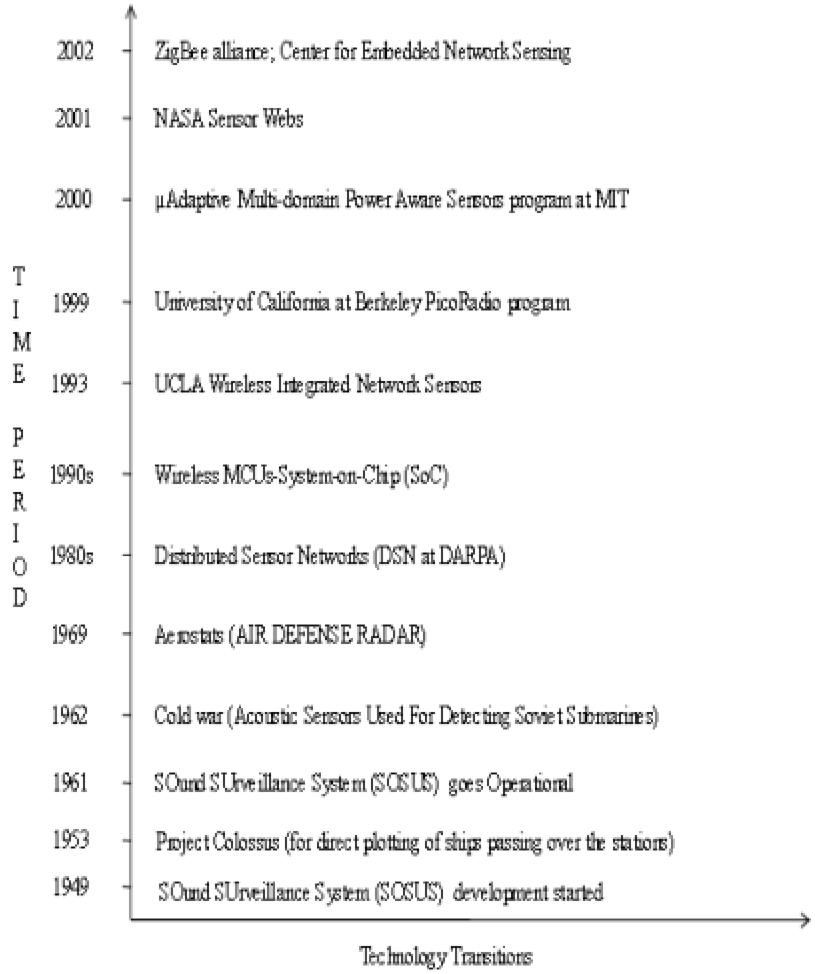

TIME PERIOD
2002
ZigBee alliance; Center for Embedded Network Sensing
2001
NASA Sensor Webs
2000
µAdaptive Multi-domain Power Aware Sensors program at MIT
1999
University of California at Berkeley PicoRadio program
1993
UCLA Wireless Integrated Network Sensors
1990s
Wireless MCUs-System-on-Chip (SoC)
1980s
Distributed Sensor Networks (DSN at DARPA)
1969
Aerostats (AIR DEFENSE RADAR)
1962
Cold war (Acoustic Sensors Used For Detecting Soviet Submarines)
1961
SOund SUrveillance System (SOSUS) goes Operational
1953
Project Colossus (for direct plotting of ships passing over the stations)
1949
SOund SUrveillance System (SOSUS) development started
Technology Transitions

Figure I.1 Technology transitions in WSN

I.3 presentation of sensor networks

I.3.1 definition Wireless Sensor Network

In a wireless sensor network, it is an important task to collect the data periodically from various sensor nodes for monitoring and recording the physical conditions of the environment. The sensed data must be transmitted and received between the nodes in the network **[2]**.

He Wireless sensor network (WSN) is a broadcast network **[3]** ; and is an ad hoc network that does not have a fixed infrastructure and consists of wirelessly connected sensor nodes, one or more base stations (BS), and a host computer that interfaces with the base station as seen in Figure I.2 **[4]**.

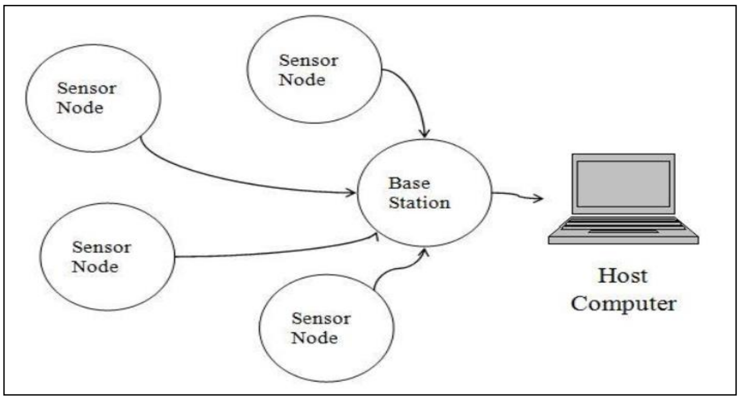


Figure I. 2 WSN Architecture

I.3.2 Definition of a wireless sensor and sensor node

Generally speaking, a sensor is a device which responds to physical stimulus (such as heat, light, sound, pressure, magnetism, etc.) and converts the quantity or parameter of a physical stimulus into recordable signals (such as electrical signals, mechanical signals, etc.)**[5]**.

The SNs are scattered to collect useful information from a specific geographical location and send it to a BS. The collected data can either be routed using single hop or multi-hop paradigms, depending on the specific routing protocol. SNs communicate information through wireless media such as radio signals and infra-red **[6] [7]**.

For the SNs to be able to sustain long term sensing capability over a large coverage set, they should conserve energy as much as possible since their battery life is limited. Replacement of a node battery is infeasible due to the harsh environment in which the SNs are usually deployed in. Hence, it becomes imperative to prioritise energy efficiency in order to maximise network lifetime and performance of WSN. In time-critical applications, routing delay also becomes a pressing factor **[8] [9]**.

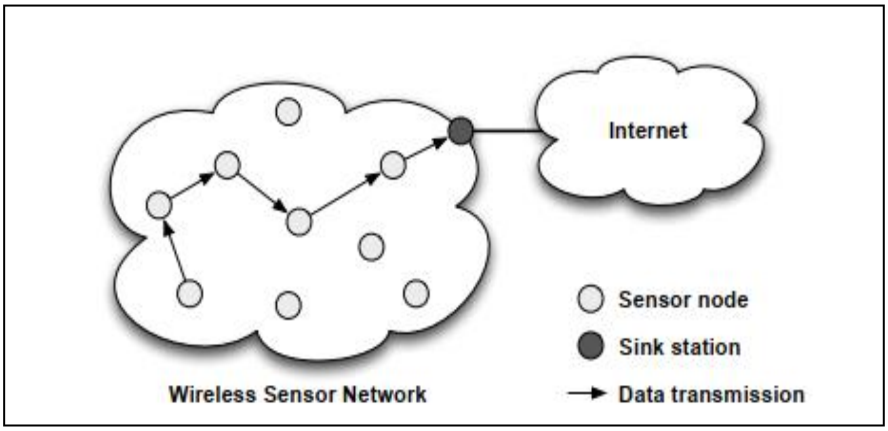


Figure I. 3 Sensor Node Architecture

I.3.3 Sensor Node Architecture

Wireless sensor nodes are equipped with **sensing unit**, **a processing unit**, **communication unit** and **power unit**. Each and every node is capable to perform data gathering, sensing, processing and communicating with other nodes **[10]**.

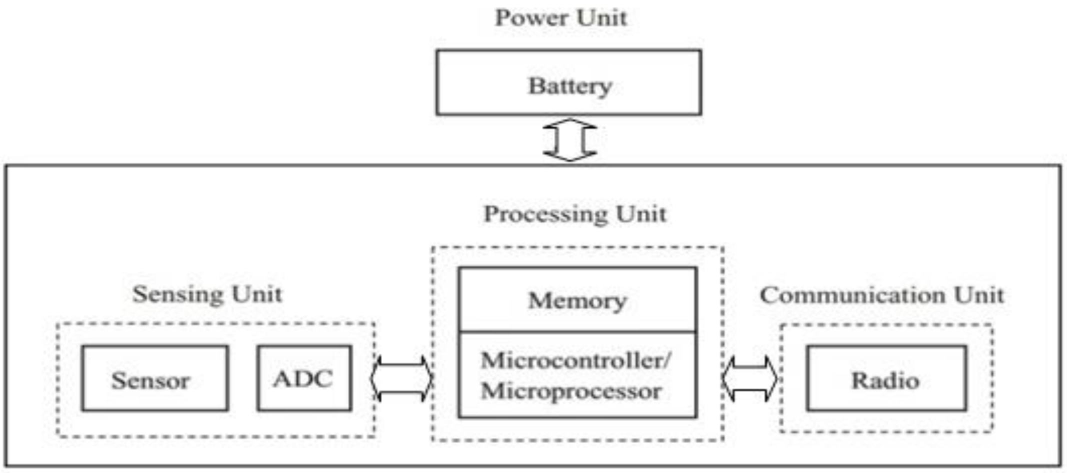


Figure I. 4 Sensor Node Architecture

I.3.3.1 The sensing unit

This component is composed of two subunits, a Sensor and an ADC (Analog to Digital Converter). The role of ADC is to convert the signal observed from the sensor to a digital signal which is delivered after the conversion operation to the Processing Unit for further analysis **[11]**.

I.3.3.2 Processing Unit

This unit is responsible for implementing the communication protocols that allow sensor nodes to collaborate with other nodes to accomplish the predefined tasks. It operates with an operating system designed specifically for micro-sensors **[12]**.

I.3.3.3 Communication Unit (Transceiver)

The role of the transceiver is to make all the transmission and reception operations. There are three deploying communication schemes in sensors including optical communication (laser), infrared, and Radio Frequency (RF) **[12]**.

I.3.3.4 Power Unit

This unit effectuates the operations of the remaining energy control, and measures the time to live of the sensor node **[13]**.

I.4 wireless sensor network architectures

According to the way that data are collected, WSNs can be classified into three types: homogenous sensor networks, heterogeneous sensor networks, and hybrid sensor networks **[14]**.

I.4.1 Homogenous Sensor Networks

A homogenous network consists of base stations and sensor nodes equipped with equal capabilities, for example, computational power and storage capacity. Data gathering in this type of networks is based on the structure of data dissemination. Flat and hierarchical topologies are two representative structures being widely studied for data dissemination and gathering in homogenous networks **[15]**.

- In **flat architecture**, except for sink node, the other nodes are identical, they have the same capacity in terms of energy and computing, also, they have the same role in sensing task. Node can directly communicate with a sink in single hop manner or communication with sink can be in multi hops manner. Simplicity presented by this architecture enables low communication latency. Furthermore, when the network becomes denser, the scaling problem arises mainly as regards routing. Figure I.5 presents WSNs flat architectures **[16]**.

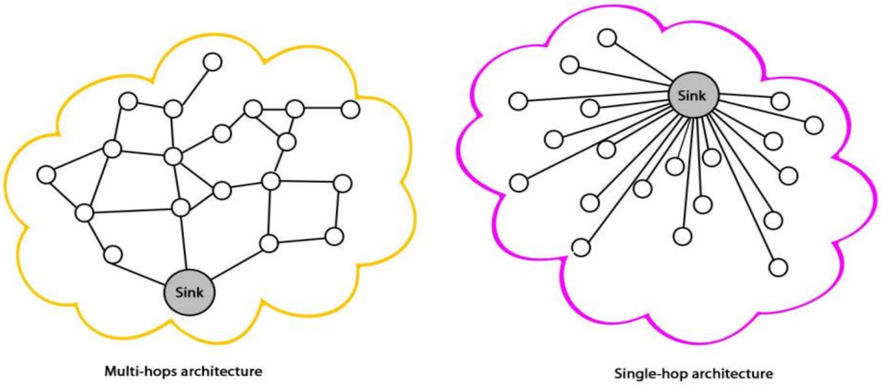


Figure I. 5 WSNs flat architectures

- A **hierarchical architecture** to deploy a large number of sensors. The network is divided into several groups or clusters which are the organizational unit of the network. Depending of cases, a more expensive cluster node type and more powerful than other nodes or a normal node in the cluster is designated as group leader called cluster head that is responsible of coordination of the sensors under its responsibility and act as a gateway to another cluster. The cluster head is responsible for the aggregation and/or compression of all the collected data in order to route it to the sink. This allows the reduction of the transmission data within the network. However, there may be more latency in communications due to the density of the network and higher energy consumption for the cluster heads. Figure I.6 highlights the clustered architectures for multi hops and single hop clusters **[16]**.

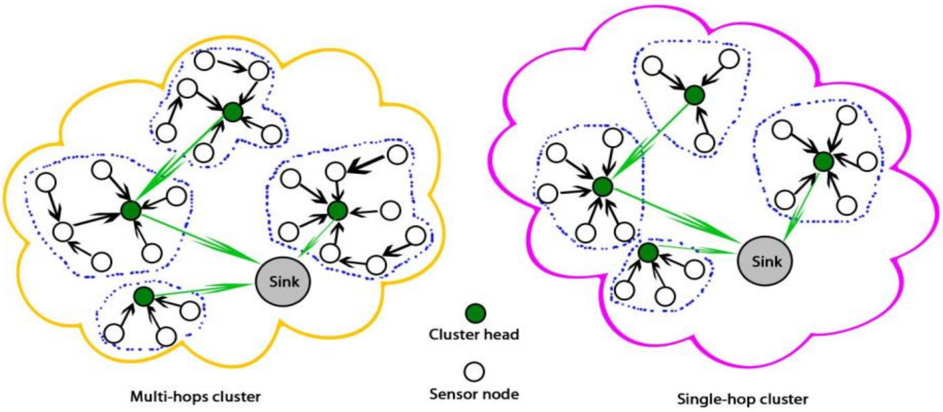


Figure I. 6 WSNs hierarchical architectures

I.4.2 Heterogeneous Sensor Networks

A heterogeneous sensor network consists of base stations (fixed and mobile), sensor nodes, and sophisticated sensor nodes with advanced embedded processing and communicating capabilities as compared to normal sensor nodes. Data gathering can be executed at the mobile base stations **[17]**. In such networks, mobile base stations move randomly in the area of the deployed network, collecting data directly from normal sensor nodes, or use some surrounding sensor nodes to relay the data (see Fig. I.7). Sometimes, sensor nodes may be distributed sparsely and the distance between any two sensor nodes can be far apart. The long distance among sensor nodes implies that more energy will be consumed for communication. Meanwhile, sensor nodes need to perform sensing and communication for as long as possible. As shown in many experimental results, data gathering with mobile sinks is able to prolong the lifetime of the system **[18]**.

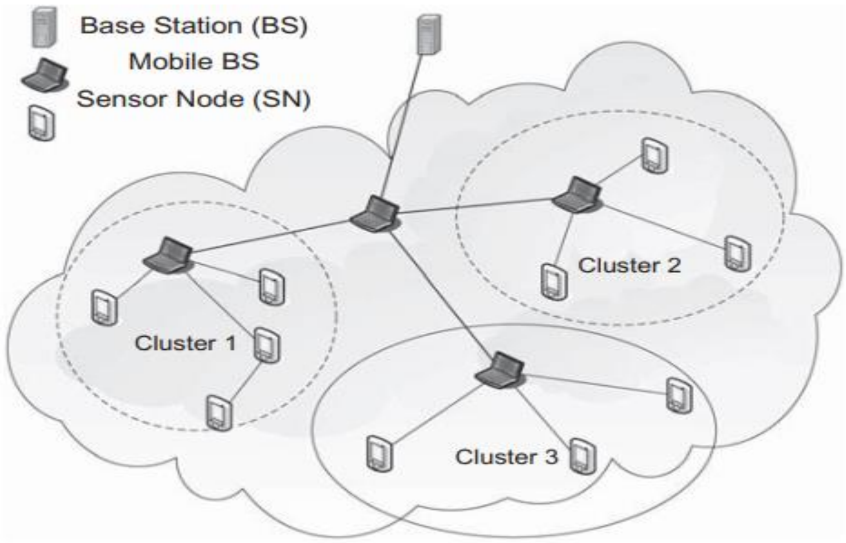


Figure I. 7 Heterogeneous sensor network topology

I.4.3 Hybrid Sensor Networks

In a hybrid sensor network, several mobile base stations work cooperatively to provide fast data gathering in a real time manner. In the scenario shown in Fig I.8, collected data will be relayed by several mobile base stations. The conventional and well - studied routing algorithms for ad hoc networks can be adopted as the routing protocols among these mobile base stations. Mobile ad hoc networks (MANETs) assume that every node is able to move at their own pace. Even though WSNs are more constrained than other wireless networks, for example, MANETs, in terms of energy, processing, transmission range, and bandwidth, routing from a source base station to a destination base station can be accomplished by using MANET protocols in hybrid sensor networks **[19]**. Accordingly, if the location of a base station is unpredictable or in case the base stations cannot communicate with each other on their own, it is reasonable to tailor techniques originally proposed for MANETs and apply them in WSNs. Hybrid sensor networks can achieve longer lifetime and can also improve the efficiency of data gathering **[20,21]** . As pointed out in Ref, a mobile base station prefers the hybrid architecture, by which a mobile base station can communicate with other sensor nodes by using a WSN protocol and with other base stations by using a MANET protocol.

While individual sensor nodes are not as powerful as normal computers, a large number of sensor nodes are required to provide high quality and reliable networking service, as well as easy deployment and fault tolerance in inaccessible environments where maintenance is inconvenient or impossible. Such unique operating environments and performance requirements of WSNs require fundamentally new approaches to networking design **[19]**.

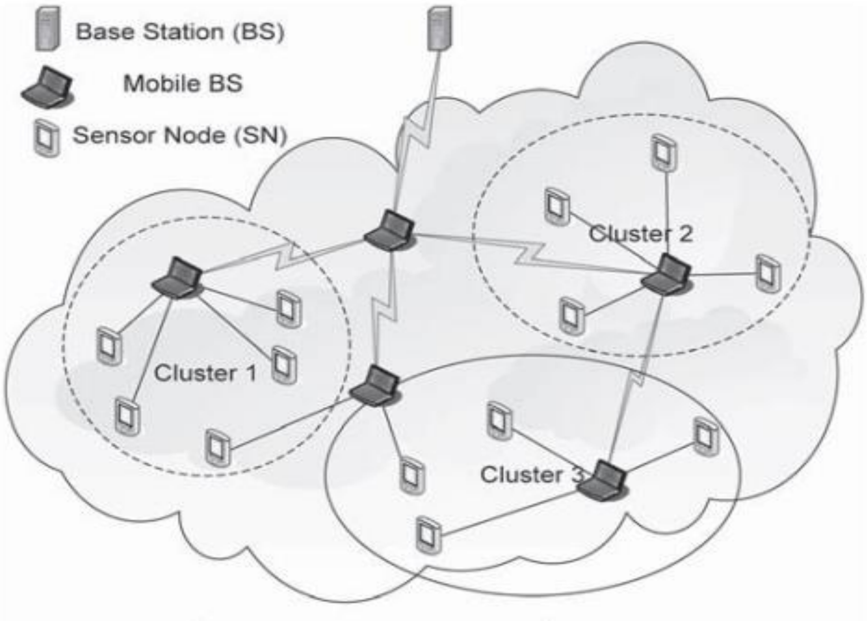


Figure I. 8 Hybrid sensor network topology

I.5 Protocol Stack of WSNs

Wireless sensor network shadows the OSI model. The architecture of WSN composed of five layers (physical, data link, transport, network and application) and three cross layers (power management, mobility management and task management). These layers collectively are used to complete the network and make the sensors work in an organized way, in order to raise the efficiency of the network **[22]**.

1. **Physical layer:** is accountable for frequency selection, modulation and data encryption and signal detection **[22]**.
2. **Data link layer:** function of Data link layer is to provide a pathway for multiplexing the data streams, data frame detection, error control, MAC **[22]**.
3. **Network layer:** for data supply network layer is used. It is accountable for routing the information received from transport layer i.e. finding the optimal path for data packet to travel from source to destination **[22]**.
4. **Transport layer:** transport layer helps to maintain the data flow. This layer is especially needed when the system is planned to be accessed through Internet or other external networks. Unlike protocols such as Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), the multi hop communication scheme of WSN is not based on global addressing **[23]**.
5. **Application layer:** is present at last and forms software and hardware transparent to the end user. It is responsible for presenting collected information and traffic management **[22]**.

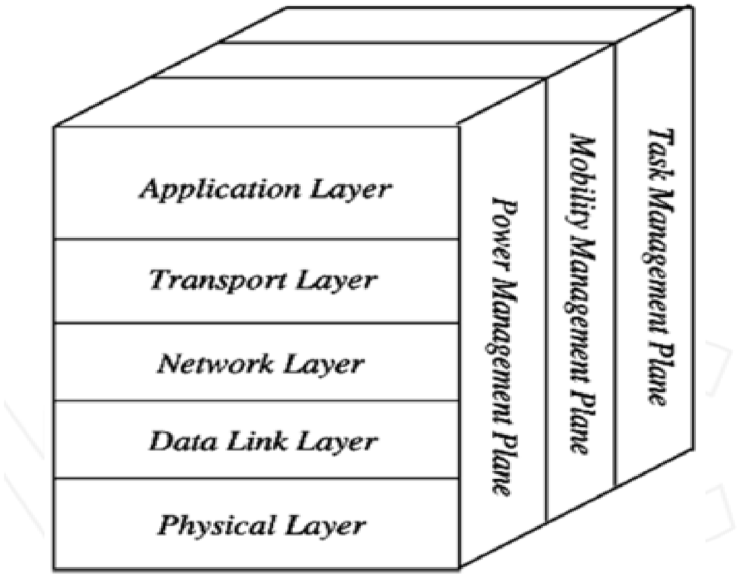


Figure I. 9 Wireless Sensor Network protocol stack.

I.6 Types of WSNs

Depending on the deployment environment on earth, underground or underwater, there are several types of wireless sensor network **[24]**.

- **Terrestrial**: In this type of sensor networks, hundreds to thousands of sensors deployed randomly or pre-deployed on a given area. This type of WSNs is mainly used in the field of environmental monitoring and presents a challenge to the sustainability of the network in terms of management of energy **[25]**.

- **Underground**: These very special sensor nodes are known for their high cost and the required logistics for maintenance and pre-planned deployment. Sensors are installed in the soil for agriculture or in the walls of a mine to monitor conditions in the soil. However, in this type of network, there is land node which has role of relaying sensed information by the underground nodes to the base station **[26]**.

- **Underwater**: This type of WSNs are still a great research challenge because of fact that environment in which the nodes are deployed is hostile and usually used for exploration. It's only possible to deploy a few nodes, these nodes are more expensive than terrestrial sensors, wireless communication is acoustic, the bandwidth is limited, the loss of signal is recurrent, propagation delays and synchronization problems are high **[27][28]**.

- **Multimedia**: This type of WSNs allows monitoring of a tracker in real-time events such as images, videos and sound. These sensors are equipped with cameras and microphones. Importance is given for: good bandwidth which implies a high energy consumption; processing and data compression; good quality of service (QoS). Advance planning is necessary for the deployment of these sensors **[28]**.

- **Mobile**: In this most recent type of WSNs, nodes are capable of repositioning and autonomously reorganize the network. After initial deployment, nodes disperse to collect information. There's also a hybrid network that consists of the combination of mobile sensors and fixed sensors **[29]**.

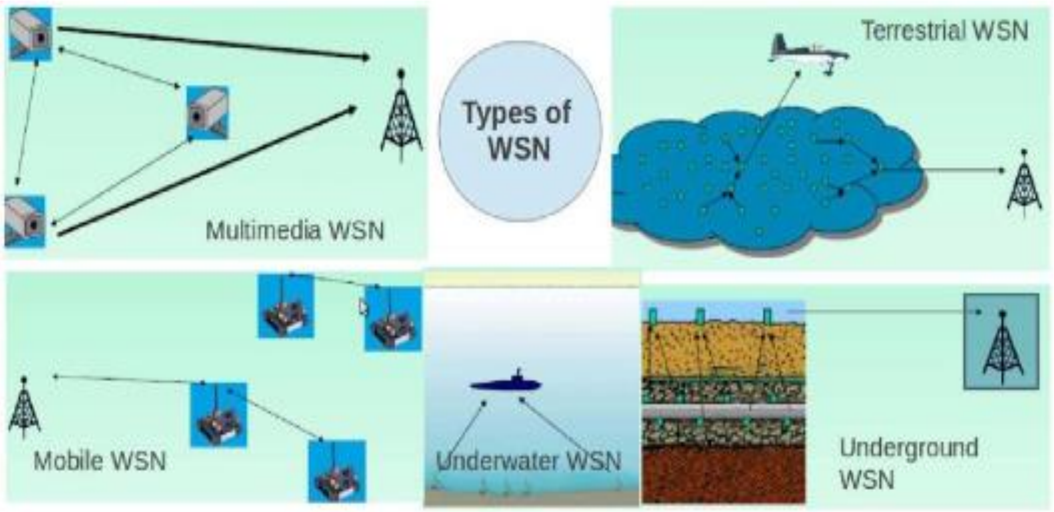


Figure I. 10 Types of WSNs.

I.7 Routing Protocols in wireless Sensor Networks

The surveys of different routing protocols that have been developed for secure sensor network and find their capabilities and deficiencies and suggest the most efficient among them. A wireless sensor network (WSN) consists of numerous tiny autonomous sensing nodes that are deployed across a wide geographical area. Routing in WSN differs according to the type of its network structure. The figure I.11 below shows the different network architectures and routing techniques used by the protocols to work **[2]**.

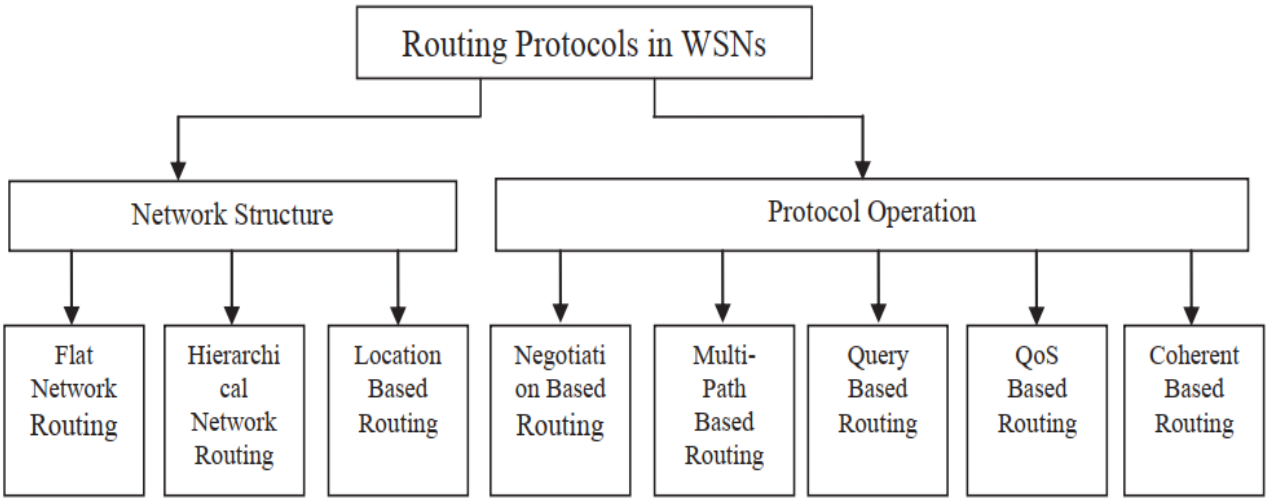


Figure I. 11 Routing protocols in WSN. [2]

I.7.1 Classification based on Network Structure

In a level topology, all nodes perform the same errands and have the same functionalities in the system. Information transmission is performed jump by bounce normally utilizing the type of flooding **[31]**.

I.7.1.1 Flat based Routing

Each sensor node behaves the same way and co-operates with other nodes to perform the sensing task. The network contains a large number of such nodes and a Base Station (BS) sends queries to certain regions and waits for data from the sensors located in the selected regions. Data-centric routing is used where there is no global identifier for nodes; instead data are identified using attribute based naming **[2]**.

- ✓ **Advantages**

The advantages of at routing protocols are scalability and simplicity. Flat network are scalable because each node participates equally in the routing task and the nodes only need information about their direct neighbours for routing. New nodes can easily be added to the network that use at routing protocols **[2]**.

- ✓ **Disadvantage**

The main disadvantage of at routing protocols is the creation of hotspots. The nodes around the base station will deplete their energy sources faster than the other nodes. This cannot be avoided because all the packets have to be routed to the base station eventually. This might not be a problem in network that has more than one base station. Network disconnectivity is another problem where certain sections of the network can become unreachable. If there is only node connecting a part of the network to the rest and if it fails, then that section would be cut off from rest of the network **[2]**.

I.7.1.2 Hierarchical Routing

Hierarchical routing protocols are based on cluster heads and the process by which the nodes decide which clusters to join. Routing path establishment is usually not considered because the nodes are one hop away from the cluster head and they always send data to the cluster head. Since most of the hierarchical routing protocols follow the same procedure, only the important hierarchical protocols are explained the following section. This routing method has special advantages related to scalability and efficient communication; they also provide energy-efficient routing in WSNs **[2]**.

- ✓ **Advantages**

The advantages of hierarchical routing are data aggregation and localized power consumption. Data from the entire cluster can be combined with the cluster head and then sent to the base station in a single packet. The amount of power consumed in a cluster is less than the network as a whole **[2]**.

- ✓ **Disadvantages**

The disadvantages of hierarchical routing are the creation of hotspots, special hardware requirements, complexity and non-scalable. Hotspots are created because the nodes elected as cluster heads consume more energy than other nodes in the network. If the cluster heads are not rotated regularly, the network becomes partitioned and this causes areas to become cut off from the network **[2]**.

Most of the hierarchical routing protocols require the cluster heads to have special hardware requirements like better radio equipments, higher processing power, more energy resources and etc. In network that rotate the cluster heads, the cluster head selection is very complex and requires high processing capability **[2]**.

Hierarchical protocols are not very scalable because the number of cluster heads increases as the network size increases. The message overhead also increases as the number of nodes in the network increase **[2]**.

I.7.1.3 Location-based Routing

Sensor nodes are addressed depending on their locations. Relative coordinates of neighboring nodes are obtained either by exchanging information between neighbor nodes or by directly communicating with a Global Positioning System (GPS) **[2]**.

The Hierarchal routing is the routine that operates in the hierarchal structures of WSNs. The main goal of the hierarchal routing protocols is to save the energy of sensor nodes as much as possible, and hence it prolongs the WSN lifetime. To design energy aware hierarchal routing protocol **[2]**.

I.7.2 Classification based on Protocol Operation

There are five different classifications as follows **[30]:**

I.7.2.1 Negotiation-Based

These protocols use high-level descriptions coded in high level so as to eliminate the redundant data transmissions. Flooding is used to disseminate data, due to the fact that flooding data are overlapped and collisions occur during transmissions. Nodes receive duplicate copies of data during transmission. The same data content is sent or exchanged again and again between the same set of nodes, and a lot of energy is utilized during this process. Negotiation protocols like SPIN are used to suppress duplicate information and prevent redundant data from being sent to the next neighbouring nodes or towards the base station by performing several negotiation messages on the real data that has to be transmitted **[30]**.

I.7.2.2 Multipath-Based

These protocols are efficient in handling multiple paths. Nodes send the collected data on multiple paths rather than using a single path. The reliability and fault tolerance of the network increases as there is, as long as it is possible, an alternative path when the primary path fails **[30]**.

I.7.2.3 Query-Based

Query-based routing propagates the use of queries issued by the base station. The base station sends queries requesting for certain information from the nodes in the network. A node, which is responsible for sensing and collecting data, reads these queries and if there is a match with the data requested in the query it starts sending the data to the requested node or the base station. This process is known as Directed Diffusion where the base station sends interest messages on to the network. These interest messages, which move in the network, create a path while passing through all the sensor nodes. Any sensor node, which has the data suitable to the interest message, sends collected data along with the interest message towards the base station. Thus, less energy is consumed and data aggregation is performed on a route **[30]**.

I.7.2.4 Quality of Service (QoS)-Based

In this type of routing protocol, both quality and energy have to be maintained within the network. Whenever a sink requests for data from the sensed nodes in the network, the transmission has to satisfy certain quality-of-service parameters, such as, for example, bounded latency (data has to be sent as soon as it is sensed without delaying any further) and bandwidth consumed. Sequential Assignment Routing (SAR) is one of the first routing protocols that use the notion of QoS in routing decisions. Routing decision in SAR depends on three factors: energy consumption within the network by the sink and the nodes, QoS of each path in the network, and priority level of each packet sent **[30]**.

I.7.2.5 Coherent-Based

In a WSN, the sensor nodes collect data and send it to the nearest neighbours or the sink within the network. In this process, the processing of the collected data is the most important event. There are two types of data-processing techniques followed within the network structure: coherent and non-coherent data processing based routing. All the nodes within the network collect the data and process it before sending to the next nearest node for further processing. This technique is called non-coherent data process routing and the nodes that perform further processing on the data are called aggregators. In coherent routing, after minimum processing, the data is forwarded to the aggregators. This minimum processing includes functions like time stamping or duplicate suppression. This technique is energy efficient as all the processing is done by the nodes, which reduces the total time and energy consumption **[30]**.

I.8 Routing Challenges in WSN

One of the fundamental configuration objectives of WSN is to do information correspondence while attempting to draw out the lifetime of the system and forestall integration debasement by utilizing forceful vitality administration methods. The configuration of steering conventions is affected by numerous testing elements as given underneath **[30]:**

I.8.1 Node Deployment

Node organization in WSN is application-particular and can be either manual (deterministic) or randomized. In manual sending, the sensors are physically put and information is directed through foreordained ways. On the other hand, in arbitrary arrangement, the nodes are scattered arbitrarily, making a specially appointed directing framework **[30]**.

I.8.2 Energy Consumption

Sensor node lifetime depends enormously on battery lifetime. In the multi-bounce WSN, every node assumes a double part as information sender and information switch. The breaking down of some sensor nodes emerging out of force disappointment can bring about critical topological changes and may oblige re-directing of parcels and revamping of the system **[30]**.

I.8.3 Fault Tolerance

Some sensor nodes may come up short or be obstructed because of absence of force, physical harm or natural impedance. The disappointment of sensor nodes ought not to influence the general functionalities of the sensor system. The steering convention needs to focus the other conceivable way to course the information to the sink node **[30]**.

I.8.4 Scalability

The quantity of sensor nodes conveyed in the detecting zone may be in the request of hundreds or thousands, or considerably more. Any directing plan must be scaled up to handle steering, among the immense number of sensor nodes. By and large nodes in the sensor system are in rest mode and at whatever point an occasion is detected, the nodes are changed over to dynamic state **[30]**.

I.8.5 Coverage

In WSN, every sensor nodes gets a certain perspective of the earth. A given sensor's perspective of the earth is constrained both in extent and exactness. It can just cover a constrained physical range of the earth. Subsequently, range scope is additionally a vital configuration parameter in WSN **[30]**.

I.9 Security in Wireless Sensor Networks

Wireless networks are vulnerable to security attacks due to the broadcast nature of the transmission medium. Furthermore, wireless sensor networks have an additional vulnerability because nodes are often placed in a hostile or dangerous environment where they are not physically protected. The main threads that address WSNs are as follows **[13]**.

1. **Passive Information Gathering**

An attacker can destroy the nodes by intercepting the messages and pick off the data stream. In addition to that, an adversary can detect the application specific content of messages including message IDs, timestamps and other fields. The application of encryption techniques can help to decrease the menace of this attack **[31]**.

2. **Subversion of a Node**

An adversary can capture a particular sensor and extract information stored on it. The information can be the key and in this case the whole sensor network is compromised **[32]**.

3. **False Node and malicious data**

A malicious node can be added to wireless sensors networks that send false data or prohibit a true data to be spread in the networks. This type of attack is one of the most dangerous attacks .In fact, the sent of false data can be received by all the nodes and as a consequence the whole network is destroyed or the network is taking over on behalf of an adversary **[31]**.

I.10 Characteristics of Wireless Sensor Networks (WSN)

The major characteristics of the sensor node used to evaluate the performance of WSN are:

1. **Fault tolerance**: Each node in the network is prone to unanticipated failure. Fault tolerance is the capability to maintain sensor network functionalities without any break due to sensor node failures **[33]**.
2. **Mobility of nodes**: In order to increase the communication efficiency, the nodes can move anywhere within the sensor field based on the type of applications **[33]**.
3. **Dynamic network topology**: Connection between sensor nodes follows some standard topology. The WSN should have the capability to work in the dynamic topology **[33]**.
4. **Communication failures**: If any node in the WSN fails to exchange data with other nodes, it should be informed without delay to the base station or gateway node.

5. **Heterogeneity of nodes:** The sensor nodes deployed in the WSN may be of various types and need to work in a cooperative fashion **[33]**.
6. **Scalability**: The number of sensor nodes in a sensor network can be in the order of hundreds or even thousands. Hence, WSN designed for sensor networks is supposed to be highly scalable **[33]**.
7. **Independency:** The WSN should have the capability to work without any central control point **[33]**.
8. **Programmability:** The option for reprogramming or reconfiguring should be available for the WSN to become adaptive for any dynamic changes in the network **[33]**.
9. **Utilization of sensors:** The sensors should be utilized in a way that produces the maximum performance with less energy **[33]**.
10. **Impracticality of public key cryptosystems:** The limited computation and power resources of sensor nodes often make it undesirable to use public key algorithms **[33]**.
11. **Lack of aprior knowledge of post-deployment configuration:** If a sensor network is deployed via random distribution, the protocols will not be aware of the communication status between each node after deployment **[33]**.

I.11 Applications of Wireless Sensor Networks

The rapid evolution of sensor technology has led to design very small and smart sensors, which are used for various applications. Selection of a given type of sensor depends of desired application. In fact, each application of WSN has a set of requested requirements as coverage, location, security, lifetime, etc. Classification of WSNs application can be mainly done into two categories: monitoring and tracking. In figure I.12 we summarize this classification **[24]**.

1.11.1 Monitoring

Monitoring is used to analyse, supervise and carefully control operations of a system or a process in real-time. Sensor network-based monitoring applications are various. Below some of them are briefly presented **[24]**:

- **Environment**: monitoring of water quality, weather, pressure, temperature, seismic phenomena, vibration, monitoring of forest fires.
- **Agriculture**: irrigation management, humidity monitoring.
- **Ecology**: monitoring of animals in their natural environments.
- **Industry**: supply chain, inventory monitoring, industrial processes, machinery, and productivity.

- **Smart house:** monitoring any addressable device in the house.
- **Urban**: transport and circulation systems, self-identification, parking management.
- **Health care**: organs monitoring, wellness, surgical operation.
- **Military**: intrusion detection.

I.11.2 Tracking

Tracking in WSN is generally used to follow an event, a person, animal or even an object. Existing applications in the tracking can be found in various fields **[24]**.

- **Industry:** traffic monitoring, fault detection.
- **Ecology:** tracking the migration of animals in various areas.
- **Public health:** monitoring of doctors and patients in a hospital.
- **Military:** a WSN can be deployed on a battlefield or enemy zone to track, monitor and locate enemy troop movements.

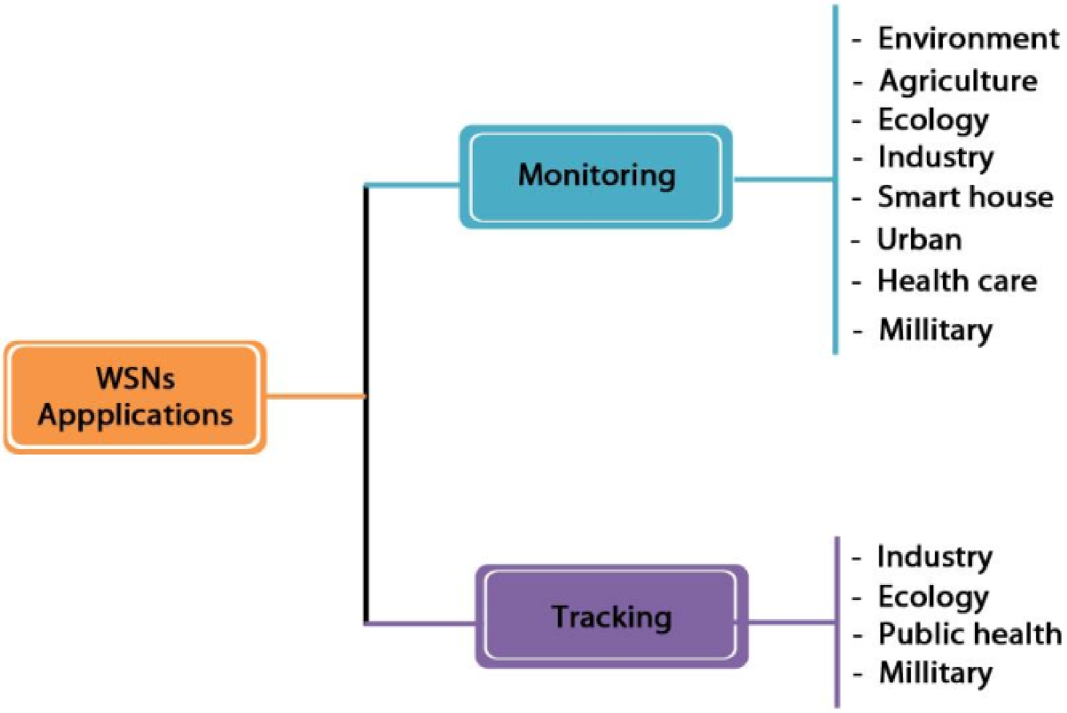


Figure I. 12 Classification of WSNs Applications

I.12 Conclusion

In this chapter we have gone through the wireless sensor networks, we have given a general view by describing their architecture, their applications, then we have described the characteristics of these networks, WSN is a field of research very answered, and becomes more and more vast and new research trends appeared especially concerning the optimization of energy and the application of these networks in specialized fields.

The next chapter we will explain the ZigBee protocol with its various characteristics and topologies, and explain the XBee module that we will work with in our project by mentioning the operating principle and mentioning types of XBee antennas and XBee adapter.

Chapter II

Study and development of technology ZigBee

II.1 Introduction

In this present communication world there are numerous high data rate communication standards that are available, but none of these meet the sensors' and control devices' communication standards. These high-data rate communication standards require low-latency and low-energy consumption even at lower bandwidths. The available proprietary wireless systems' ZigBee technology is low-cost and low-power consumption and its excellent and superb characteristics makes this communication best suited for several embedded applications, industrial control, and home automation, and so on.

This chapter will be entirely dedicated to explaining the ZigBee protocol with its various characteristics and topologies, explaining the XBee unit that we will work with in our project by mentioning the operating principle and mentioning types of XBee antennas and XBee adapter.

II.2 Presentation of IEEE802.15.4 / ZigBee

II.2.1 History [1]

- **1998:** From the arrival of wireless Wi-Fi and Bluetooth technologies, the first ZigBee-type network designs were introduced in applications where previous technologies were not usable. In particular, a lot of research has been carried out on networks that are automatically organized and composed of small radios. The ZigBee protocol was inspired by Bluetooth technology.
- **May 2003:** The IEEE 802.15.4 standard is announced (often wrongly associated with the ZigBee protocol).
- **Summer 2003:** Philips decides to abandon the grouping around ZigBee within the ZigBee Alliance. This is a blow to the project, which no longer enjoys the support of this large group.
- **October 2004:** The ZigBee Alliance announces that the number of registrations has doubled to more than 100 companies in 22 countries.
- **December 14, 2004:** Ratification of ZigBee's first specifications.
- **June 13, 2005:** The ZigBee Alliance releases the first official specifications of the ZigBee 1.0 version which are then available for free download.

II.2.2 General

Launched in the 2000s, ZigBee is a LP-WPAN (Low Power-Wireless Personal Area Network) **[1]** : ZigBee/IEEE 803.15.4 is a bidirectional wireless technology featured with short range, low cost, low power consumption, and low data rate, which makes it more suitable for applications associated with monitoring and remote control that are integrated with functional sensors and actuators **[2]**. ZigBee compliant wireless devices are expected to transmit 10-100 meters, supports many nodes (up to 65000) in a network, depending on the RF environment and the power output consumption required for a given application, and will operate in the unlicensed RF worldwide (2.4GHz global, 915MHz Americas or 868 MHz Europe). The data rate is 250kbps at 2.4GHz, 40kbps at 915MHz and 20kbps at 868MHz **[3]**.

Figure II. 1 ZigBee-logo

The ZigBee standard is developed by the ZigBee Alliance **[4]**, involving the collaboration of about 70 firms (over 200 firms are currently participating in the consortium). The aim was to develop a low-cost, low consumption wireless communication system with an average data transfer rate, for use in consumer electronics, computers and peripherals, control of residential lighting systems, industrial control, automatic construction systems and monitoring of medical variables**[5]-[6]**, etc. First release of the ZigBee specification was in the beginning of 2005 and the latest was released in the end of 2007 **[7]**.

ZigBee alliance offers several versions for the ZigBee protocol: ZigBee (2004/2006/2007), ZigBee PRO which defines a stack and additional characteristics (2007/2012), ZigBee 3.0 under development and specific protocols such as ZigBee IP, ZigBee RF4CE, and ZigBee Green Power **[8]**.

II.2.3 The architecture of ZigBee layers

Protocol architecture is based on Open system interconnection (OSI). ZigBee builds on IEEE standard 802.15.4 which defines the physical and media access control (MAC) layers. ZigBee alliance defines the network layer and application layer. Fig II.2 shows protocol stack of ZigBee system **[9]**.

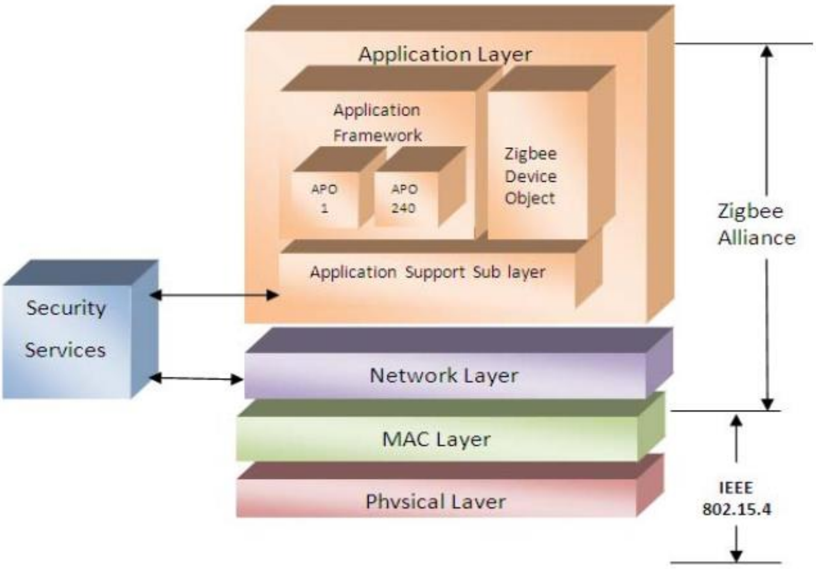
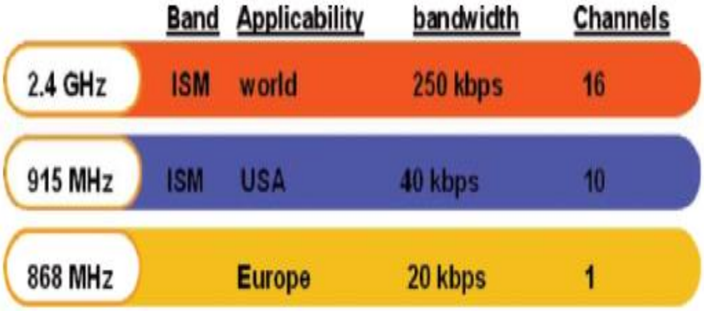


Figure II. 2 ZigBee Protocol Stack

- **Physical Layer**

Defines the physical operation of the ZigBee device including receives sensitivity, channel rejection, output power, number of channels, chip modulation, and transmission rate specifications. The physical layer of the IEEE802.15.4 standard is the closest layer to the hardware, supports three frequency bands, 2.45GHz band which using16 channels, 915MHz band which using 10 channels and 868MHz band using 1 channel **[10] [9]**.



	Band	Applicability	bandwidth	Channels
2.4 GHz	ISM	world	250 kbps	16
915 MHz	ISM	USA	40 kbps	10
868 MHz		Europe	20 kbps	1

Figure II. 3 Channel Numbering in Unlicensed Bands

The 868- and 915-MHz frequency bands offer certain advantages such as fewer users, less interference, and less absorption and reflection. However, the 2400-MHz band is a far more widely adopted for a number of reasons **[11]**:

- Worldwide availability for unlicensed use.
- Higher data rate (250 kbps) and more channels.
- Lower power (transmit/receive are on for shorter time due to higher data rate).
- RF band more commonly understood and accepted by the marketplace.

➢ The Medium Access Control (MAC) layer

Relies on the resources of the physical layer and Manages RF data transactions between neigh boring devices (point to point) **[9] [10]**. The MAC includes services such as: It generates the network beacons according to the device, if it is a coordinator also performs the function of synchronizing the beacons and uses the CSMA-CA channel access mechanism and It uses Guaranteed Time Slot (GTS) mechanism and It allows different mechanisms to conserve energy like collision avoidance using CSMACA and allowing the device to go into sleep mode **[12]**.

➢ Network Layer (NWK):

A feature of ZigBee such as the self-healing mechanism is acquired through this layer. As Figure 3 shows, this layer provides network management, routing management, network message broker, and network security management. The ZigBee Alliance defines this layer, which is an association of companies working together to enable reliable, cost-effective, and low-power wirelessly networked monitoring and control products based on an open global standard **[13]**.

➢ Application Layer

The application Layer is the highest protocol layer and it hosts the application objects. ZigBee specification separates the APL layer into three different sub-layers: the Application Support Sub layer (APS), the ZigBee Device Objects (APO), and Application Framework (AF) having manufacturer defined Application Objects **[9]**.

▪ The application objects (APO)

Control and manages the protocol layers in ZigBee device. It is a piece of software which controls the hardware. Each application objects assigned unique end point number that other APO’s can use an extension to the network device address to interact with it **[6]**. There can be up to 240 application objects in a single ZigBee device. A ZigBee application must conform to an existing application profile which is accepted ZigBee Alliance. An application profile defines message formats and protocols for interactions between application objects. The application profile framework allows different vendors to independently build and sell ZigBee devices that can interoperate with each other in a given application profile **[9]**.

- **ZigBee Device Object**

The key definition of ZigBee is the ZigBee device object, which addresses three main operations; service discovery, security and binding. The role of discovery is to find nodes and ask about MAC address of coordinator/router by using unicast messages. The discovery is also facilitating the procedure for locating some services through their profile identifiers. So profile plays an important role. The security services in this ZigBee device object have the role to authenticate and derive the necessary keys for data encryption. The network manager is implemented in the coordinator and its role is to select an existing PAN to interconnect. It also supports the creation of new PANs. The role of binding manager is to binding nodes to recourses and applications also binding devices to channels **[9]**.

- **Application support sub layer**

The Application Support (APS) sub layer provides an interface between the NWK and the APL layers through a general set of services provided by APS data and management entities. The APS sub layer processes outgoing/incoming frames in order to securely transmit/receive the frames and establish/manage the cryptographic keys. The upper layers issue primitives to APS sub layer to use its services. APS Layer Security includes the following services: Establish Key, Transport Key, Update Device, Remove Device, Request Key, Switch Key, Entity Authentication, and Permissions Configuration Table **[9]**.

- **Security service provider**

ZigBee provides security mechanism for network layer and application support layers, each of which is responsible for securing their frames. Security services include methods for key establishment, key transport, frame protection and device management **[9]**.

II.2.4 ZIGBEE STANDARD

ZigBee device are the combination of application (such as light sensor, lighting control etc), ZigBee physical device types (PAN Coordinator, Full Function Device and Reduced Function Device), and ZigBee logical (coordinator, router, end device) **[9] [12]**.

II.2.4.1 ZigBee physical device types

Based on data processing capabilities, two types of physical devices are provided in IEEE 802.15.4: PAN Coordinator (Personal Area Network) and FFD (Full Function Devices) and RFD (Reduced Function Devices) **[9].**

- **PAN (Personal Area Network) Coordinator**

PAN coordinator is the one which initializes the network, stores the information of the nodes in the network and also manages the network once it has been initiated. It handles the routing of data to different nodes and suggests what routing techniques to use to transfer the data to different nodes of the network. There can be only one PAN coordinator in a network; it has the ability to communicate with any device in the network. It can work in all star, mesh and cluster tree topologies. PAN coordinator is really the core component of the ZigBee network as proper working of this component is compulsory for the network to achieve the desired communication results **[14] [15]**.

PAN coordinator which manages one network can handle and manage all the nodes in that network by sending and receiving packets from them when needed. In case of communication between the two adjacent networks, the communication occurs from PAN to PAN, while in case, where one network wants to communicates with another, which is few networks away from the first one, then the PAN of first network gets to the destination network PAN through the series of PANs of different networks coming in the way for the required one **[12]**.

- **Full Function Device:**

Is a fully-functional device (FFD), which serves as coordinator, router or end device **[16]**. PAN coordinator can use it to fulfill the purpose of carrying out the multi-hop routing of messages across the network. It can communicate with the other FFDs and RFDs and work properly in any topology used to make the network. One other responsibility is that it searches the other FFDs and the RFDs to create the communication link so that the transfer of data can be made possible to reach the desire node **[14] [15].**

FFD can operate in three modes that are serving as a personal area network (PAN) coordinator, a coordinator, or a device. FFDs take charge of main data source transmission in a network, are able to talk to any other devices **[16]** .The current ZigBee standard requires FFDs to be always on, which in practice means that FFDs must be constantly powered. Battery-powered FFDs have a lifetime on the order of a few days [9]. IEEE 802.15.4 network need at least one FFD in the network to act as a coordinator **[17].**

- **Reduced Function Device**

Is a reduced-function (RFD) device, has the limited functionality as the name indicates which can act as end or the leaf node of the network and can only communicate with the FFD, but not with network coordinator. End devices can only communicate with routers or coordinator, but not to each other. Routers and coordinator can communicate with all network members **[16] [17]**.

RFDs act as end device that only can associate with one FFD at one time [15]. One of its other limitations is that it can only work in the star topology, as it only requires minimum RAM and ROM to be constructed. It is just for the purpose of sending or receiving. RFDs request the data from the network coordinator and can transfer it to some available network node, and then go to the sleep mode to conserve energy. It is generally battery powered **[14] [15]**.

II.2.4.2 ZigBee logical device types

There are three categories of nodes in a ZigBee system. They are Coordinator, Router and End devices.

- **Coordinator:** A coordinator is an FFD, forms the root of the network tree and might bridge to other networks **[16]**. Only one coordinator device is required in each network. It is responsible for creating the network, addresses handling **[16]** and selecting the network parameters such as radio frequency channel, unique network identifier and setting other operational parameters. It can also store the information about network, security keys **[9]**.

- **Router:** A router is an FFD. Router acts as intermediate nodes **[9]**, A router is used in tree and mesh topologies to expand network coverage. The function of a router is to find the best route to the destination over which to transfer a message **[19]**. Routing implies that it forwards a message to enable communications between other devices that are too far apart to receive information directly on their own **[18]**. A router performs all functions similar to a coordinator except the establishing of a network **[19]**.

- **End Devices:** An end device can be an RFD. An RFD operates within a limited set of the IEEE 802.15.4 MAC layer, enabling it to consume less power. The end device (child) can be connected to a router or coordinator (parent). It also operates at low duty cycle power, meaning it consumes power only while transmitting information. Therefore, ZigBee architecture is designed so that an end device transmission time is short **[19]**.

Their main function is to join networks and send and receive information. They cannot rout messages to any other devices; therefore, they can use less expensive hardware and can power themselves down intermittently, when it’s inactive thereby, saving energy by going temporarily into a nonresponsive sleep mode. However, end devices always require a router or the coordinator as a parent device which helps end devices join the network, and stores messages for them when they go to asleep. ZigBee networks can have any number of end devices. The network can comprise of one coordinator, multiple end devices, and no routers at all **[18]**. Table II.1 gives an overview of the 802.15.4/ZigBee device types.

Table II. 1 ZigBee device types

	Coordinator Network establishment and control	**Router** Supports routing functionality; can talk to other routers, coordinator, and end devices	**End Device** Can only talk to routers and the coordinator
Full Function Device (FFD): **Requires resources to handle all designated tasks.**	Yes	Yes	Yes
Reduced Function Device (RFD): Requires modest resources compared to FFD.	No	No	Yes

II.2.5 ZigBee topologies

Figure II.3 shows 3 types of topologies that ZigBee supports: star topology, peer-to-peer topology and cluster tree topology.

II.2.5.1 Star Topology

In the star topology, the communication is established between devices and a single central controller, called the PAN coordinator. The PAN coordinator may be mains powered while the devices will most likely be battery powered. Applications that benefit from this topology include home automation, personal computer (PC) peripherals, toys and games **[3]** .Also a master-slave network model is adopted where master is the ZigBee coordinator which is FFD and slave will be either FFD or RFD. ZigBee end devices are physically and electrically separated from each other end devices and pass information through coordinator. Devices can only communicate with the coordinator **[9].**

This type of topology is attractive because of its simplicity, but at the same time presents some key disadvantages. In the moment when the coordinator stops functioning, the entire network is functionless because all traffic must travel through the center of the star. For the same reason, the coordinator could easily be a bottleneck to traffic within the network, especially since a ZigBee network can have more than 6500 nodes [**20**].

II.2.5.2 Peer-to-peer/ Mesh Topology

The peer-to-peer topology is more complex, it allows end devices to communicate with each other within its radio sphere of influence. It takes use of a PAN coordinator with more network functions comparing with the coordinator in star topology. This topology makes it possible to achieve more complex mission and extendable **[11]**. And is the most flexible topology of the three. Flexibility is present because a message can take multiple paths from source to destination. If a particular router fails, then ZigBee’s self-healing mechanism will allow the network to search for an alternate path for the message to be passed **[16]**.

II.2.5.3 Cluster-tree Topology

In a Tree network, a coordinator initializes the network, and is the top (root) of the tree. The coordinator can now have either routers or end devices connected to it. For every router connected, there is a possibility for connection of more child nodes to each router. Child nodes cannot connect to end devices because it does not have the ability to relay messages. This topology allows different levels of nodes, with the coordinator being at the highest level. In order the messages to be passed to other nodes in the same network, the source node must pass the messages to its parent, which is the node higher up by one level of the source node, and the message is continually relayed higher up in the tree until it is passed back down to the destination node. Because the number of potential paths a message can take is only one, this type of topology is not the most reliable topology. If a router fails, then all of that router's children are cut off from communicating with the rest of the network **[20]**.

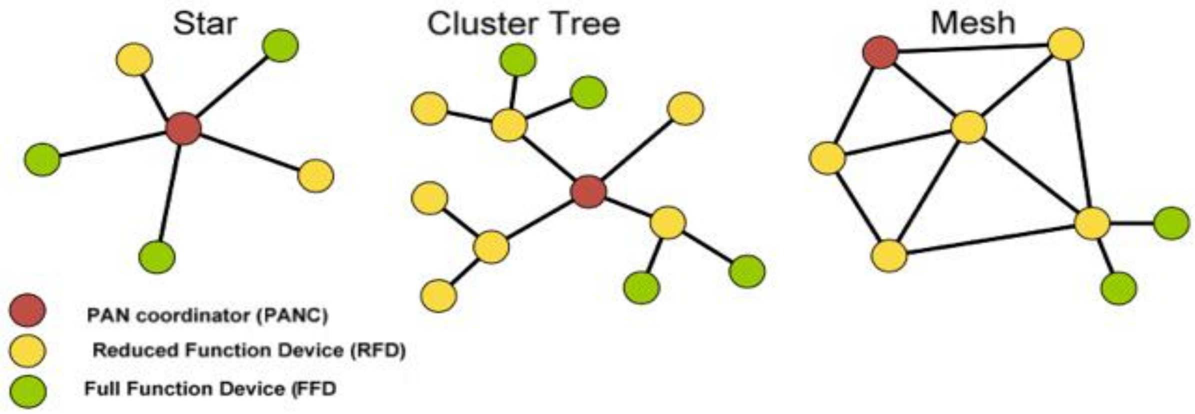


Figure II. 4 ZigBee Topology Models

II.2.6 ZigBee Security

The ZigBee standard includes complex security measures to ensure key establishment, secure networks, key transport and frame security **[21]**. Those services are implemented at the MAC, NWK and APS Layers of the protocol stack. Furthermore, the APS sub-layer provides services for the establishment, and maintenance of security relationships. The ZigBee Device Object (ZDO) manages the security policies and the security configuration of a device **[22]**.

The ZigBee protocol is based on an "open trust" model. This means all protocol stack layers trust each other. Therefore cryptographic protection only occurs between devices. Every layer is responsible for the security of their respective frames. The security of ZigBee networks is based on their encryption keys **[21]**. ZigBee uses certain security features of 802.15.4. It extends the functionality of this standard by using **[8]**:

- Uses 182-bit AES encryption.
- Definition of different keys to secure communications: master, link and network keys.
- Use of the CCM * algorithm
- Use of a Trust Center (TC)
- Security that can be customized by application.

1) MAC layer security

To provide security for the MAC Layer frames, ZigBee would use MAC Layer security specified in the 802.15.4 specifications .This will be used to secure the MAC Layer command, beacon, and acknowledgement frames. Securing MAC Layer data frames only provides security for messages transmitted over a single hop. But to provide security for multi-hop messages, ZigBee would rely on higher layer security, e.g. NWK Layer security. The MAC layer uses the Advanced Encryption Standard (AES) as its core cryptographic algorithm and describes a variety of security suites that use the AES algorithm. The MAC layer does the security processing, but the upper layers, which set up the keys determine the security levels to use **[22]**.

Fig II.4 shows ZigBee outgoing frame structure with the security fields used to provide MAC Layer security. As can be seen from the Figure, the MAC Layer adds an auxiliary header along with the MAC Layer header for carrying security information. The message integrity code (MIC) may take the values 0, 32, 64 or 128 and determines the level of data integrity **[22]**.

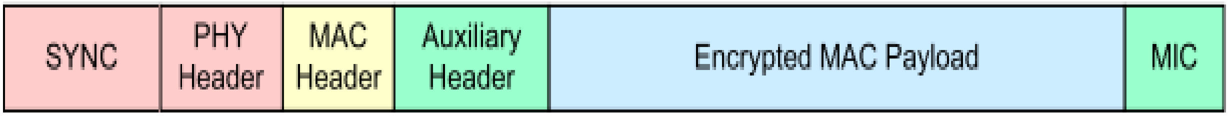


Figure II. 5 ZigBee frame with MAC layer security

2) NWK layer security

Like the MAC layer, the NWK layer’s frame protection mechanism shall make use of the Advanced Encryption Standard (AES). The NWK layer will broadcast route request messages and process received route reply messages to provide support for multi-hop routing of messages. Route request messages are simultaneously broadcast to nearby devices and route reply messages originate from nearby devices. If the appropriate link key is available, the NWK layer shall use the link key to secure outgoing NWK frames **[22]**.

Fig II.5 shows the security fields that are present when NWK Layer security is used to secure a NWK frame. As can be seen from the Figure, the NWK Layer adds an auxiliary header along with the NWK header for carrying security information. The MIC determines the level of data integrity provided **[22]**.

Another case may arise when the appropriate link key is not available. In this case the NWK layer shall use its active Network key to secure outgoing NWK frames in order to secure the messages while for the incoming NWK frames, either the active or the alternate Network key is used to secure incoming NWK frames. The security processing of the outgoing and incoming NWK frames with NWK Layer security is explained in **[22]**.

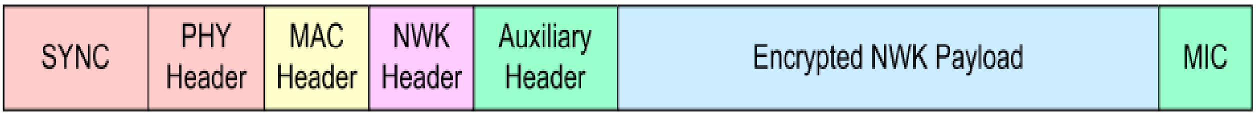


Figure II. 6 ZigBee frame with NWK layer security

3) APS layer security

The APS sublayer performs the security functions to provide security for the frames originating at the APL Layer. The APS layer frame security is based on link keys or the Network key. Fig II.6 shows the APL Layer frame with the security fields present when APL Layer security is applied. It can be seen in the Figure that the APS sublayer adds an auxiliary header along with the APS header for carrying security information. Here also the MIC is used which determines the level of data integrity provided **[22]**.

The APS layer has to also provide applications and the ZDO with key establishment, key transport, and device management services. The security processing of the outgoing and incoming APS frames with APS Layer security is explained in **[22]**.

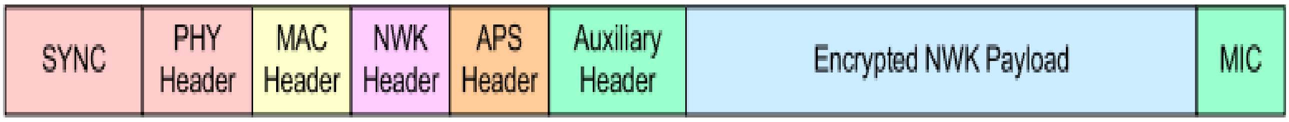


Figure II. 7 ZigBee frame with APS layer security

II.2.7 Comparison between ZigBee, Bluetooth and Wi-Fi

Due to the advantages of ZigBee technology like low cost and low power operating modes and its topologies, this short range communication technology is best suited for several applications compared to other proprietary communications, such as Bluetooth, Wi-Fi, etc. some of these comparisons such as range of ZigBee, standards, etc., are given below **[23]**.

Table II. 2 Comparison among Bluetooth, ZigBee & Wi-Fi

Standard	ZigBee 802.15.4	Bluetooth **802.15.1**	Wi Fi **802.11.a/b/g**
Frequency Band	868/915 MHz-2.4GHz	2.4 GHz	2.4 GHz-5 GHz
Max Data rate	250 kbps	1 Mbps	54 Mbps
Topology	Star, Mech, cluster tree	Star , point-to-point	Star , point-to-point
Network size	65536	7	32
Range (m)	10-100	1-10	100
Number of RF channels	1,10,16	79	14,23
Battery life	1+ year	Regular charging	Hourly charging
Security	128-AES	128-AES	SSID

II.2.8 Applications of ZigBee Technology [23]

- **Industrial Automation:** In manufacturing and production industries, a communication link continually monitors various parameters and critical equipments. Hence ZigBee considerably reduce this communication cost as well as optimizes the control process for greater reliability.
- **Home Automation:** ZigBee is perfectly suited for controlling home appliances remotely as a lighting system control, appliance control, heating and cooling system control, safety equipment operations and control, surveillance, and so on.
- **Smart Metering:** ZigBee remote operations in smart metering include energy consumption response, pricing support, security over power theft, etc.
- **Smart Grid monitoring:** ZigBee operations in this smart grid involve remote temperature monitoring, fault locating, reactive power management, and so on.

II.3 Presentation of the XBee module

II.3.1 General

XBee is a module produced by Digi International mainly use as a radio communication transceiver and receiver **[24]**. It is created for control and sensor networks based on the IEEE 802.15.4 standard. It is created by the ZigBee Alliance which runs in low data rate with low power consumption and small packet devices. XBee is intended to be simpler and low cost because low cost allows the technology to be widely used in wireless control and sensor networks **[25].**

The XBee comes in two flavours, XBee-STD and XBee-PRO. Table II.3 shows the major differences between them [**26].**

Table II. 3 XBee comparison

Specification	**XBee-STD**	**XBee-PRO**
Transmit Power Output	1 mW (0 dBm)	60 mW (18 dBm)
Receiver Sensitivity	-92 dBm (1% PER)*	-100 dBm (1% PER)*
Transmit Current (typical)	45 mA (@ 3.3V)	215 mA (@ 3.3V, 18 dBm)
Idle / Receive Current (typical)	50 mA (@ 3.3V)	55 mA (@ 3.3V)
Number of Channels (DSSS)	16	12

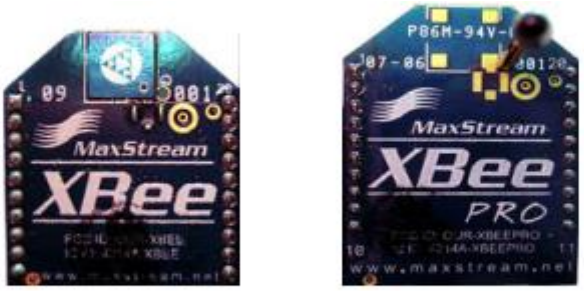


Figure II. 8 XBee-STD with chip antenna and XBee-PRO with whip

II.3.2 the main characteristics of the XBee: [27]

- Carrier frequency: 2.4 GHz
- Varied ranges:
  - quite low for the XBee 1 and 2 (10 - 100m)
  - large for the XBee Pro (1000m)
- Low speed: 250kbps
- Low consumption: 3.3V @ 50Ma
- Inputs / outputs: 6 Input pins, 8 digital I / O pins
- Security: reliable communication with a 128-bit encryption key
- Low cost: 25 €
- Ease of use: communication via the serial port
- set of AT and API commands
- Network flexibility: its ability to cope with an out-of-service node or to integrate new nodes quickly
- Large number of nodes in the network: 65000
- Various network topologies: mesh, point to point, point to multipoint

II.3.3 XBee Categories

There are two basic varieties of XBee radio physical hardware: Series 1 and Series 2.

1. **Series 1**

The S1is based on the IEEE 802.15.4 standard. This standard defines the physical (PHY) and medium access control (MAC) layers for low-rate wireless personal area network (PAN) **[28]**. These modules are simple to use, as they do not require any configuration to operate in peer to peer communication. That means you can directly replace a wired serial connection with these devices **[29]**.

2. **Series 2**

The S2 is based on the ZigBee technology. ZigBee is a network layer protocol on top of the 802.15.4 standard. This means that besides the PHY and MAC layers, the ZigBee also has Network and Application Support Sublayer (APS) layers **[28]**. These layers made mesh networking and multi-hop intrinsically available to the XBee module. These modules are slightly complex to use as they require some configuration even to work in peer to peer communication. If you are just starting with XBee and your requirement is just replace a serial wired connection it is recommended you work with Series 1, though Series 2 is more power efficient than Series 1**[29]**.

Series 2 modules are available in several versions: XBee ZNet 2.5 (obsolete), ZB (current) and 2B (most recent). You also have some XBee Pro, which do the same thing, but with greater capabilities, notably the range which seems to be able to go up to 1000 meters **[27]**.

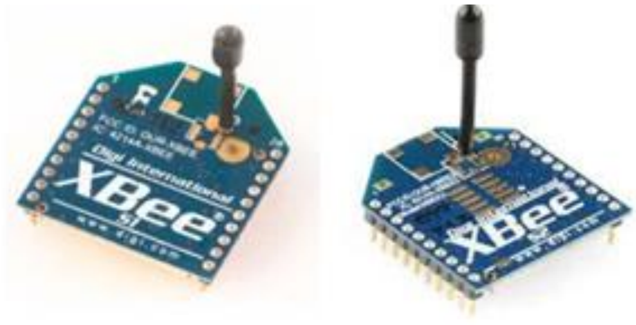
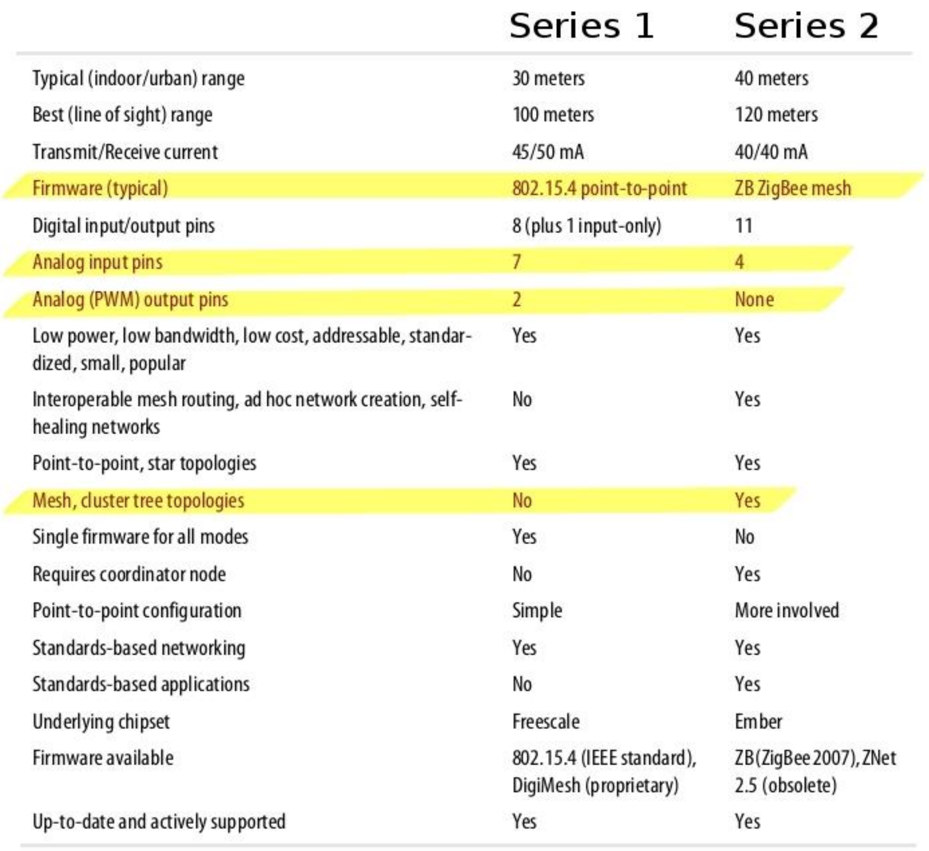


Figure II. 9 XBee Series 1and XBee Series 2

Table II.4 illustrates the modules of series 1 and 2 are not compatible with each other ; range and consumption are much the same; the number of inputs and outputs is different and especially the series 2 does not have PWM analog outputs; the possible network topologies are not the same. With the series 1, the architecture is simple: point to point (peer) or multipoint (star). Series 2 also allows you to create more complex networks: mesh (mesh) or "tree" (cluster tree) **[27]**.

Table II. 4 comparison of XBee modules S1 and S2



	Series 1	Series 2
Typical (indoor/urban) range	30 meters	40 meters
Best (line of sight) range	100 meters	120 meters
Transmit/Receive current	45/50 mA	40/40 mA
Firmware (typical)	802.15.4 point-to-point	ZB ZigBee mesh
Digital input/output pins	8 (plus 1 input-only)	11
Analog input pins	7	4
Analog (PWM) output pins	2	None
Low power, low bandwidth, low cost, addressable, standardized, small, popular	Yes	Yes
Interoperable mesh routing, ad hoc network creation, self-healing networks	No	Yes
Point-to-point, star topologies	Yes	Yes
Mesh, cluster tree topologies	No	Yes
Single firmware for all modes	Yes	No
Requires coordinator node	No	Yes
Point-to-point configuration	Simple	More involved
Standards-based networking	Yes	Yes
Standards-based applications	No	Yes
Underlying chipset	Freescale	Ember
Firmware available	802.15.4 (IEEE standard), DigiMesh (proprietary)	ZB (ZigBee 2007), ZNet 2.5 (obsolete)
Up-to-date and actively supported	Yes	Yes

II.3.4 XBee antennas

Radios need antennas to transmit and receive signals. There's more than one way to build an antenna, each with advantages and disadvantages. Here are the kinds of antenna options currently available **[30]:**

1. **Whip or wire antenna**

This is just what it sounds like a single piece of wire sticking up from the body of the radio. In most cases, the wire antenna is just what you need. It's simple and offers Omni-directional radiation, meaning the maximum transmission distance is pretty much the same in all directions when its wire is straight and perpendicular to the module **[30]**.

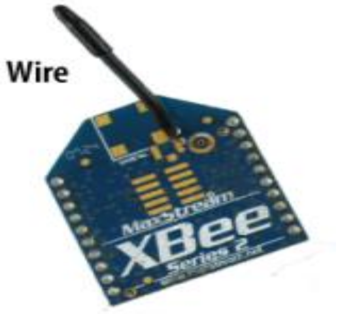


Figure II. 10 Whip or wire antenna **[30]**

2. **Chip antenna**

Again, this is pretty much what it sounds like. The chip a te0000nna i00s a flat ceramic chip that's flush with the body of the XBee. That makes it smaller and sturdier, but those advantages come at a price. Chip antennas have a cardioids (heart-shaped) radiation pattern, meaning that the signal is attenuated in many directions. However, if you're making a device where mechanical stress to the wire antenna might break it, or you need to put the radio in a very small space, then the chip antenna may be your best bet. Chip antennas are often the right choice for anything wearable **[30].**

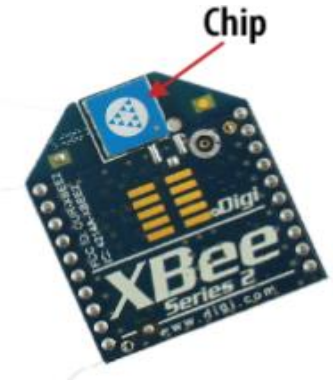


Figure II. 11 Chip antenna [30]

3. PCB antenna

Introduced with the XBee-PRO S2B, the PCB antenna is printed directly on the circuit board of the XBee. It is composed of a series of conducting traces laid out in a fractal pattern. The PCB antenna offers many of the same advantages (and disadvantages) as the chip antenna with a much lower cost to manufacture **[30].**

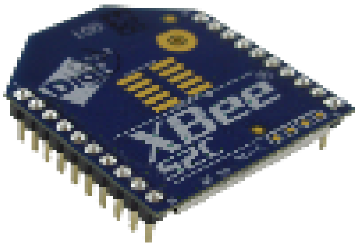


Figure II. 12 PCB antenna

4. U.FL connector

This is the smaller of the two types of external antenna connectors. More often than not, an external antenna is not needed: it is an additional expense if a simple wire antenna will do. However, when your radio is going to live on the inside of a metal box then the antenna will need to live on the outside. That way the signal is not attenuated by the enclosure. Also, it is sometimes advantageous to orient an external antenna differently than the XBee itself to or use a special-purpose antenna with a specific radiation pattern, such as a high-gain antenna that passes signals in a single direction over a broader distance. The U.FL connector is small, somewhat fragile, and almost always used with a short connecting cable that carries the signal from a remotely mounted antenna **[30].**

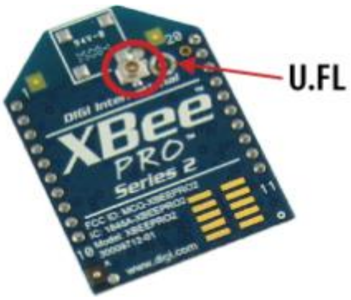


Figure II. 13 U.FL connector [30]

5. RPSMA connector

The RPSMA connector is just a different type of socket from the U.FL connector. It’s bigger and bulkier, but you can use it with an external antenna, mounted directly to the XBee without a connecting cable. For most introductory projects, you’re still best off with the simple wire antenna that is smaller, cheaper, and usually just as good **[30]**.

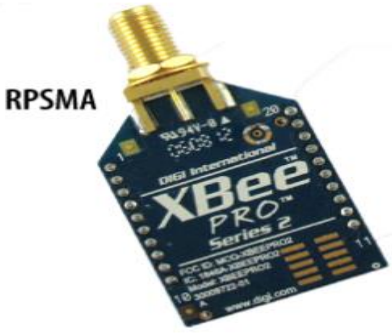


Figure II. 14 RPSMA connector [30]

II.3.5 XBee Adapter

The XBee RF module needs adapters to connect with other devices. There are four different kinds of adapters:

A. XBee USB Explorer Board:

All XBee radios have 20 connection pins each spaced 2 mm apart. As the pins on the XBee are separated differently than the holes in the breadboard. This tight spacing of the pins helps to keep the radios very small, but doesn't allow them to fit into a breadboard .Simple XBee breakout boards that adapt to 0.1″ breadboard spacing **[29]**.

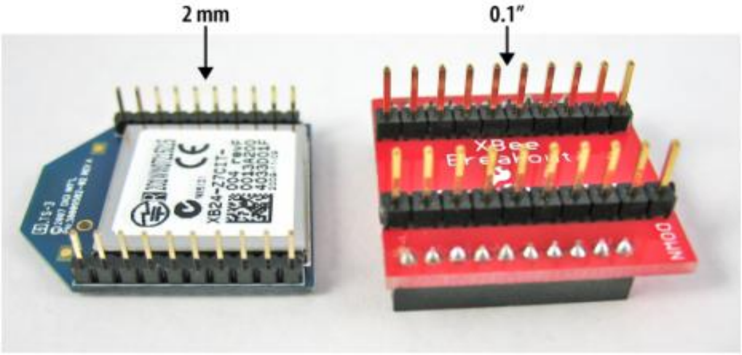


Figure II. 15 Breakout board showing pin spacing [30]

An Explorer Board shown in Fig. 5 is used connect XBee to breadboard and then to computer's USB port. The XBee connected to Explorer board is shown in Fig II. 6, when XBee radio connected to Explorer board the pin functions changes **[29]**.

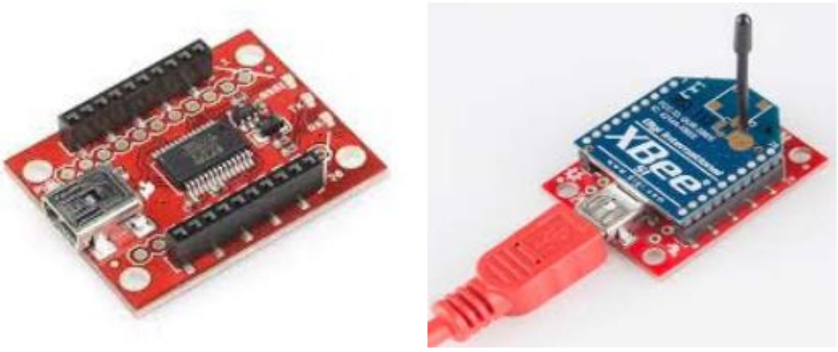


Figure II. 16 Explorer Board with XBee Radio

B. XBee to FTDI cable adapter

This is an inexpensive board that you'll need to solder together yourself. It also must be used with a special USB cable called the FTDI USB TTL-232, which can attach to its pin headers. The cable can be used with certain Arduino-type boards as well. Male headers can be added so that this adapter can be used in a breadboard **[30]**.

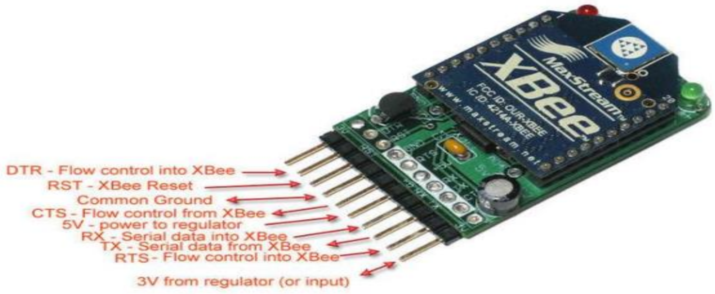

DTR - Flow control into XBee
RST - XBee Reset
Common Ground
CTS - Flow control from XBee
5V - power to regulator
RX - Serial data into XBee
TX - Serial data from XBee
RTS - Flow control into XBee
3V from regulator (or input)

Figure II. 17 XBee to FTDI cable adapter

C. XBee Explorer Dongle

One of the smallest adapters, it needs no external cable. The Dongle does not provide any access to the radio beyond USB. Also, because it has no cable, its shape sometimes interferes with other cables or the computer casing. On the other hand, it's a very small all-in-one device that's easy to carry in a pocket. It's terrific for use on the go **[30]**.

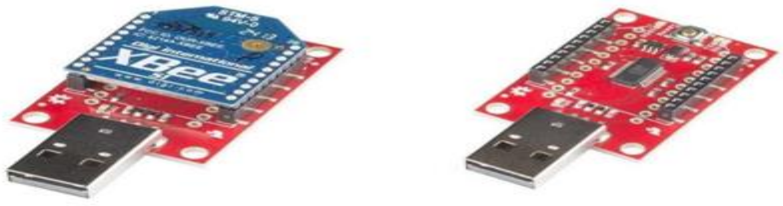


Figure II. 18 XBee Explorer Dongle

D. XBee to Arduino shield

These attach an XBee directly to an Arduino microcontroller. Shields are printed circuit boards engineered to seat directly on top of an Arduino board. When you are not including other components, the shield eliminates the need for breadboards and wiring **[30]**.

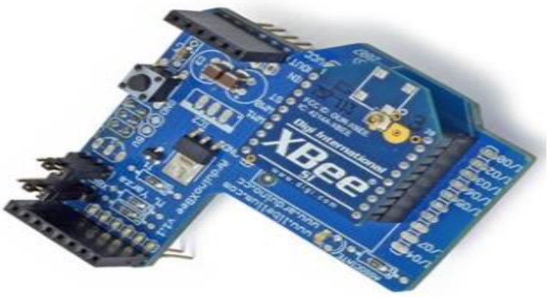


Figure II. 19 XBee to Arduino shield

II.3.6 XBee Module Pin Description

XBee modules use DSSS (Direct Sequence Spread Spectrum) modulation technique for communication. XBee has on board features like Digital I/O pins, analog ADC (10-bit) input pins, PWM output etc. It has serial (TX, RX) pins for communication with PC and Microcontrollers serially **[31]**.

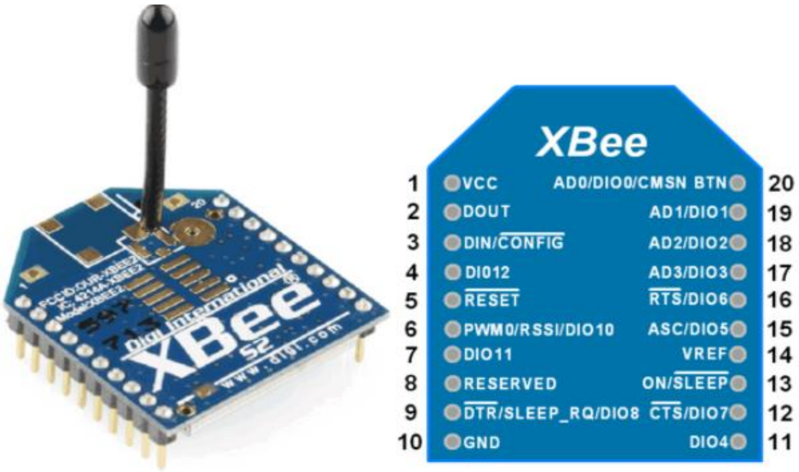


Figure II. 20 XBee Series 2 RF Module Pin Number

As shown in above figure II.20 of XBee module, it has 20 pins. Each pin function is described in table given below **[31]**.

Table II. 5 Pin Assignments for the XBee Series 2 Modules [30]

Pin No.	Name	Description
1	VCC	3.3 V Power Supply
2	DOUT	Data out (TX)
3	DIN/CONFIG	Data in (RX)
4	DIO12	Digital I/O 12
5	RESET	Module Reset (asserted low by bringing pin to ground)
6	PWM0/RSSI/DIO10	Pulse-width modulation analog Output 0 ,Received Signal Strength Indicator
7	DIO11	Digital I/O 11
8	Reserved	Do not connect
9	DTR/SLEEP_RQ/DIO8	Sleep Control or Digital Input 8
10	GND	Ground
11	DIO4	Digital I/O 4
12	CTS/DIO7	Clear-To-Send Flow Control or Digital I/O 7
13	ON/SLEEP	Module Status Indicator, High = ON, Low = SLEEP
14	VREF	No used in Series 2
15	ASSOCIATE /DIO5	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS/ DIO6	Request-To-Send, Digital I/O 6
17	AD3/DIO3	Analog Input 3 or Digital I/O 3
18	AD2/DIO2	Analog Input 2 or Digital I/O 2
19	AD1/DIO1	Analog Input 1 or Digital I/O 1
20	AD0/DIO0/COMMIS	Analog Input 0 or Digital I/O 0, Commissioning Button

II.3.7 XBee addresses and channels

Each RF data packet sent over-the-air contains a source address and destination address field in its header. XBee modules have a unique and permanent address on earth, this address is a 64-bit serial number assigned by the manufacturer. XBee modules also have a 16-bit short address assigned within the network. Finally, Node Identifier can be assigned to each module as a string of text. Table II.5 shows the type of addresses **[32]**:

Table II. 6 Type of address in ZigBee Network

Type	Example	Unique
64-bit	0013A20043E0750	Yes, on the earth
16-bit	23F7	Yes, only within a network
Node identifier	Node1	Uniqueness not guaranteed

- **XBee ZigBee**

The 16-bit short address of a node is assigned by a coordinator or router at the time of joining the network. For this reason, it is also called "network address" the 16-bit address of 0x0000 is therefore reserved for coordinator. This way of resolving the identity of a node in a network can be unreliable since the 16-bit address is not static. To solve this problem, the 64-bit destination address is often included in data transmissions to guarantee data is delivered to the correct destination **[32]**. Although both modules support the same type of addresses, the configuration of addresses of nodes within the network is different. XBee ZigBee supports mesh topology; in this case the addresses of nodes which wish to join or leave the network are assigned by the coordinator, thus they are dynamically assigned. Meanwhile XBee IEEE 802.15.4 basically supports peer-to-peer network in which the address is configured manually **[32]**.

- **PAN Address**

This is another 16-bit address relevant to the Personal Area Network. The 16-bit PAN address gives the possibility of 65,536 different PAN addresses along with 65,536 16-bit module address for each PAN. This addressing scheme gives the capability for more than 4 billion total radios **[32]**.

- **Channels**

Assuming that all the addressing is correct in the network, the communication still may not be established unless the modules work on the same frequency. Thus, all modules in the network must use the same channel. Generally 12 channels, or more based on the type of radio, are available, but the programmer doesn't have to worry about selecting a specific channel since the XBee modules select it automatically **[32]**.

Figure II.10 shows a demonstration example of how a message is being delivered within a network or between two XBee modules that belong to different networks. In both scenarios, the XBee modules have to communicate on the same channel **[32]**.

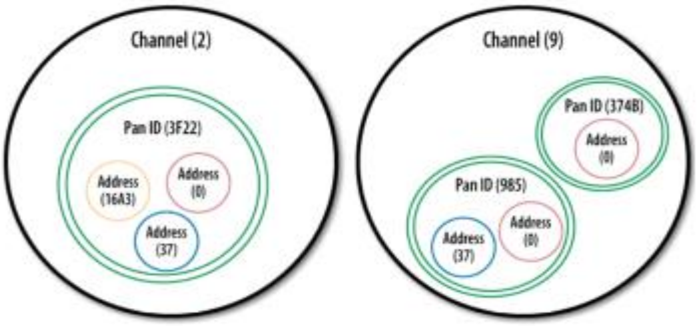


Figure II. 21 Venn diagram showing channel, PAN and addressing

In peer-to-peer network, if any pair of XBee modules wishes to communicate in the same area where a network already exists, they have to choose a different address, PAN ID or frequency channels **[32]**.

For troubleshooting, it is important to know that all modules within a PAN should operate using the same firmware version **[32]**.

II.3.8 XBee firmware

The XBee modules can be used with two main types of firmware, 802.15.4 firmware or ZigBee firmware. The next two subsections describe the two firmwares in more detail [26].

- **802.15.4**

The 802.15.4 firmware makes the module act as a standard 802.15.4 device. This makes the module capable of peer-to-peer communication and star networking, as specified in the IEEE 802.15.4 standard [**26**].

- **ZigBee**

The ZigBee firmware makes the module act as a ZigBee device. With this firmware the module has support for both mesh networking and the standard 802.15.4 network techniques [26].

II.3.8 Configuration Modes for XBee Modules

XBee devices use either AT (transparent) or API operating mode to send and receive data over the serial interface. The network could also have a mixture of devices using either AT or API mode **[18]**.

- **AT Mode**

AT mode is synonymous with transparent mode, each module has a single network destination and personal address. Configuration of network parameters must be done through command mode, either by the user or a micro controller. AT mode is useful for simple point-to-point communication or non-modular network topology. All the data is received through the serial input and queued up for radio transmission. The data received wirelessly is sent to the serial output exactly as it is received, without additional information. When a device operates in transparent mode, it cannot identify the source of a wireless message it receives. In order to identify the sources of all the data it receives from different devices, the devices sending the message must include extra identifier information known by all the devices which can be extracted later. This can be achieved by defining a robust protocol that comprises of all the information needed for the transmissions **[33] [18].**

- **API Mode:**

API mode is recommended for larger networks with more complicated overall topologies. Messages contain a personal and destination address, which can be changed during run time depending on message type, making mesh topology simple to implement. Additionally, API mode contains packet delivery confirmation messages and has the option of escape parameters. Also API mode provides a structured interface where data is communicated through the serial interface in organized packets and in a determined order. This enables you to establish complex communication between devices without having to define your own protocol. Since the data destination is included as part of the API frame structure, you can use API mode to transmit messages to multiple devices. The API frame includes the source of the message, so it is easy to identify where data is coming from **[33] [18].**

II.4 Conclusion

In this chapter, we briefly presented the ZigBee protocol with its various characteristics and topologies, and then touched upon the XBee unit that we will work with in our project by mentioning the operating principle and mentioning types of XBee antennas and XBee adapter.

Chapter III

Pratical Realization

III.1 Introduction

This last chapter is divided into two parts: a software part in which we will define the software used as well as the configuration steps, and a second part which describes the practical process of our system.

III.2 THE SOFTWARE PART

In this part we used the X-CTU software for the configuration of the XBee modules before moving on to programming with Arduino.

III.2.1 Arduino IDE

Arduino IDE is an open-source microcontroller system which has been designed for easy learning and fast development. It is easy to use and flexible. Arduino can interact with the environment through data received from a variety of sensors and can be used to control its surroundings by controlling lights, motors, and other actuators **[1]**. Application Arduino written in Java, and derives from the IDE for the Processing programming language and the Wiring projects. The Arduino development environment contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them. It includes a code editor with features such as syntax highlighting, brace matching, and automatic indentation, and is also capable of compiling and uploading programs to the board with a single click **[2]**.

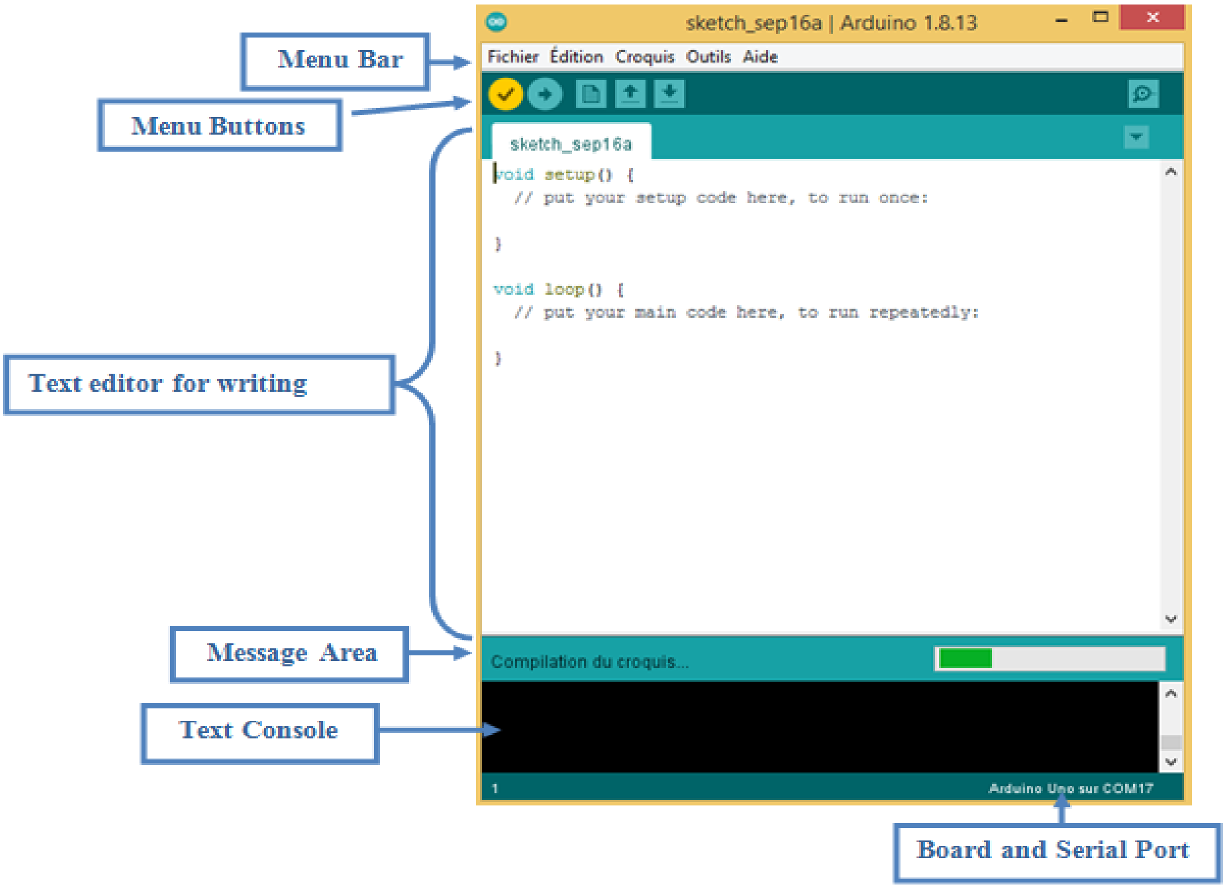
Arduino programs are written in C or C++. The Arduino IDE comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier **[2]**.

The Arduino is found in several types of card:

- Arduino UNO.
- Arduino LEONARDO.
- Arduino 101.
- Arduino Micro.
- Arduino NANO.
- Arduino MEGA… etc.

III.2.1.1 Arduino development environment:

The Arduino development environment contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them **[3]**.



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

}

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

}
```

Compilation du croquis...

1 Arduino Uno sur COM17

Figure III. 1 Description of the working window of the IDE environment.

III.2.1.3 Structure of a program

Arduino code (also called Arduino sketch) includes two main parts: setup code and loop code **[5]**.

➢ **Setup Code**

- Is code in setup () function.
- Executed right after power-up or reset.
- Executed only one time.
- Used to initialize variables, pin modes, and start using libraries.

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

- **Loop Code**

- Is code in loop () function.
- Executed right after setup code.
- Executed repeatedly (infinitely).
- Used to do the main task of application

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

III.2.1.4 Avantages of the Arduino

The five major benefits of using Arduino **[6]**:

1. **Inexpensive:** Arduino boards are relatively inexpensive compared to other microcontroller platforms. The least expensive version of the Arduino module can be assembled by hand, and even the pre-assembled Arduino modules cost less than $50.

2. **Cross-platform:** The Arduino software runs on Windows, Macintosh OSX, and Linux operating systems. Most microcontroller systems are limited to Windows.

3. **Simple, clear programming environment:** The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino.

4. **Open source and extensible software:** The Arduino software is published as open source tools, available for extension by experienced programmers. The language can be expanded through C++ libraries, and people wanting to understand the technical details can make the leap from Arduino to the AVR C programming language on which it's based. Similarly, you can add AVR-C code directly into your Arduino programs if you want to.

5. **Open source and extensible hardware:** The Arduino is based on Atmel's ATMEGA8 and ATMEGA168 microcontrollers. The plans for the modules are published under a Creative Commons license, so experienced circuit designers can make their own version of the module, extending it and improving it. Even relatively inexperienced users can

build the breadboard version of the module in order to understand how it works and save money.

III.2.2 X-ctu software

The X-CTU software is the official configuration program for XBee modules provided by digiinc. X-CTU is available only for Windows operating system; if the user has Linux then X-CTU should be running under WINE windows emulator. X-CTU is used mainly for setting the firmware to XBee module. And allows MaxStream customers to test the radio modems in the actual environment with just a computer and the items include with the radio modems. This application can configure XBee RF module as the requirement of users directly, such as destination address, coordinator/ end devices, AT/ API Mode, etc **[7]**.

Figure III. 2 X-ctu software.

III.2.2.1 Configuration steps for X-ctu

To make the XBee S2C modules communicate, it is necessary to follow the following configuration steps:

1) By default, the serial communication parameters are set:

   Baud Rate = 9600 (Equivalent to 9600 bits/second)

   Data Bits = 8

   Parity = NONE

   Stop Bits = 1

   Flow Control = NONE

2) Define the type of module (COORDINATOR, ROUTER or END DVICE) and the operating mode (AT or API).
3) Choose always update firmware

There are two modes of operation of the XBee modules first, the AT mode which uses AT commands to configure the module whose sending as well as receiving is done in a simple way. The second mode is called API mode, requires designing the frames to be sent by the user himself. And for reasons of simplicity (configuration time and size of the program) we have adopted the AT mode for the rest of our configuration.

4) Configure the necessary parameters:

- **PAN ID:** Represents an identifier that will allow the module to communicate only with modules which are on the same channel and which carry the same PAN ID, thus preventing interference with other modules not concerned by the communication.
- **DH and DL:** Contains the upper and lower parts of the destination addresses (for broadcast communication, DH = 0x0000 and DL = 0xFFFF must be set).

There are also two other important parameters on the list, which are:

- **CH:** Contains the number of the channel used by the modules to communicate, it is the coordinator who scans the communication channels until it finds its correspondent (ROUTER / END-DEVICE).
- **SH and SL:** Contains the factory serial numbers of the modules. And which also represents their own addresses (source address).

5) Finally, Click on **write**.

III.3 communication system of point-to-multipoint for project

This is the project's communication system for point-to-multipoint communication and then controlling the electrical machines by the XBee module, and the unit can connect to one or multiple devices on the network. This type of communication generally involves a central coordinator with multiple remote nodes connected back to the central host. The network topology is called star, Mech, or cluster tree.

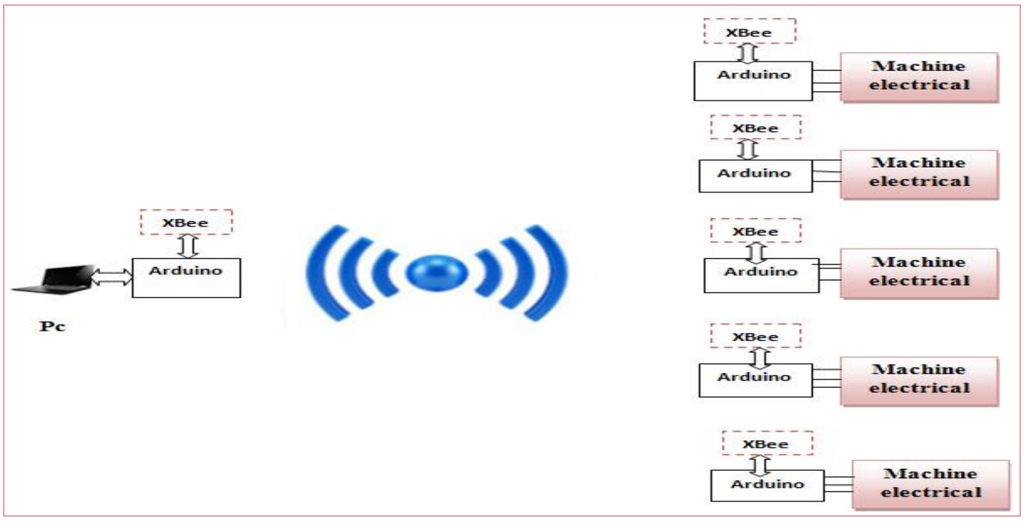
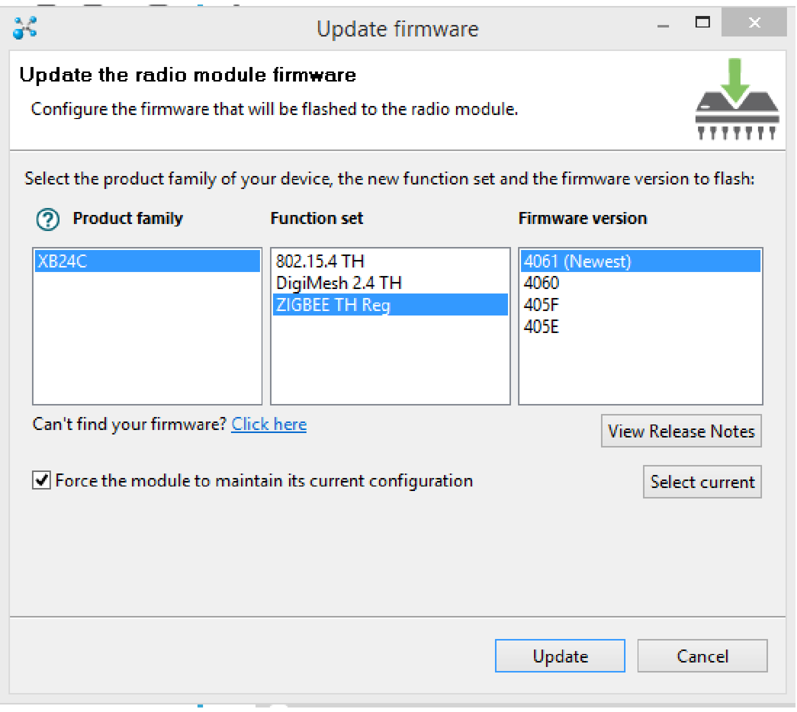


Figure III. 3 communication system of point-to- multipoint

III.3.1 Configuration of point-to-multipoint

First, update each unit with the latest version of its intended firmware (ZC, ZR, or ZED).



Update firmware

Update the radio module firmware

Configure the firmware that will be flashed to the radio module.

Select the product family of your device, the new function set and the firmware version to flash:

Product family	Function set	Firmware version
XB24C	802.15.4 TH	4061 (Newest)
	DigiMesh 2.4 TH	4060
	ZIGBEE TH Reg	405F
		405E

Can't find your firmware? Click here

View Release Notes

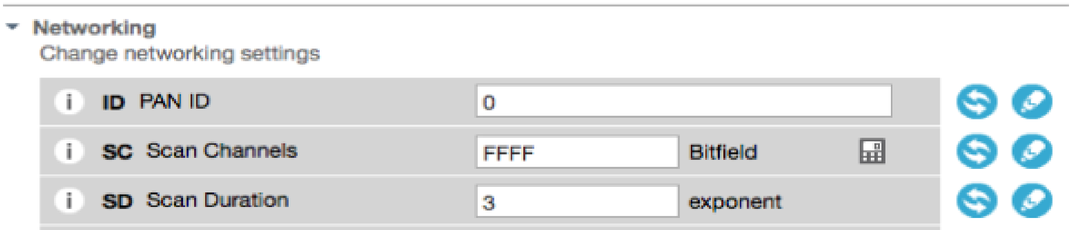
☑ Force the module to maintain its current configuration

Select current

Update Cancel

Figure III. 4 XCTU interface for Firmware Download

Each device must specify its Personal Area Network ID as a hexadecimal string on the range x0000 to xFFFF; all devices must operate on the same channel to communicate.

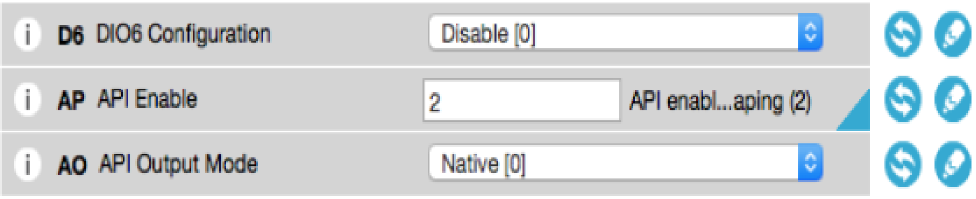


Networking
Change networking settings

ID PAN ID	0	
SC Scan Channels	FFFF	Bitfield
SD Scan Duration	3	exponent

Figure III. 5 XCTU Interface for Setting PAN ID

Finally, API mode is used with escape (API = 2) that can be modified depending on the number of final devices.



D6 DIO6 Configuration	Disable [0]	
AP API Enable	2	API enabl...aping (2)
AO API Output Mode	Native [0]	

Figure III. 6 XCTU Interface for Setting API mode 2

III.4 Equipment used in the project

III.4.1 Arduino UNO R3

Arduino Uno R3 is one kind of ATmega328P based microcontroller board. It includes the whole thing required to hold up the microcontroller; just attach it to a PC with the help of a USB cable, and give the supply using AC-DC adapter or a battery to get started. The term Uno means “one” in the language of “Italian” and was selected for marking the release of Arduino’s IDE 1.0 software. The R3 Arduino Uno is the 3rd as well as most recent modification of the Arduino Uno. Arduino board and IDE software are the reference versions of Arduino and currently progressed to new releases. The Uno-board is the primary in a sequence of USB-Arduino boards, the reference model designed for the Arduino platform **[7]**.

- **Arduino UNO R3 Pin Diagram :**

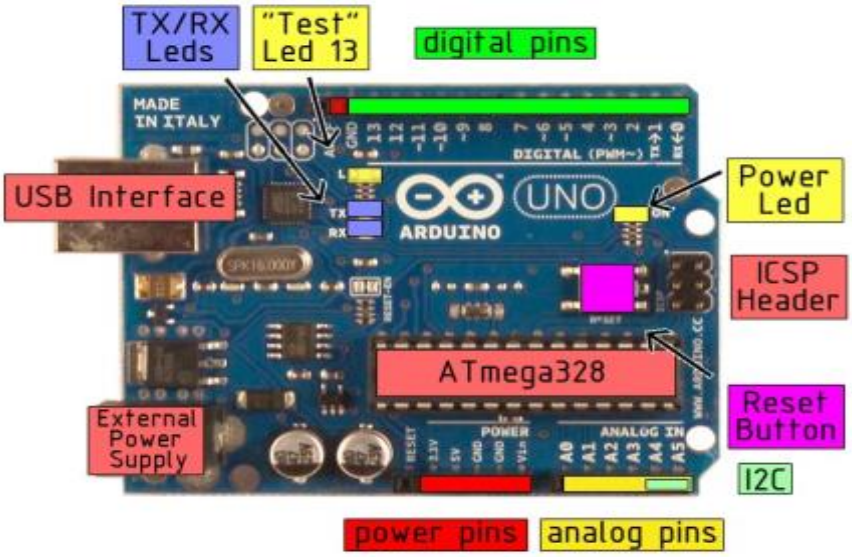


Figure III. 7 Arduino UNO Pin Diagram.

➢ **Power Supply**

The Arduino Uno can be powered via the USB connection or with an external power supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts **[8]**.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and VIN pin headers of the POWER connector **[8]**.

The power pins are as follows **[8]**:

- **VIN:** The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V:** The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.
- **3V3:** A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND:** Ground pins.

➢ Memory

The Atmega328 has 32 KB of flash memory for storing code (of which 0, 5 KB is used for the bootloader); it has also 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library) **[8]**.

➢ Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode (), digitalWrite (), and digitalRead () functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 KOhms. In addition, some pins have specialized functions **[8]**:

- **Serial: 0 (RX) and 1 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- **External Interrupts: 2 and 3.** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt () function for details.
- **PWM: 3, 5, 6, 9, 10, and 11.** Provide 8-bit PWM output with the analogWrite () function.

- **SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK).** These pins support SPI communication, which, although provided by the underlying hardware, is not currently included in the Arduino language.
- **LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, each of which provides 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference () function. Additionally, some pins have specialized functionality **[8]**:

- **I2C:** 4 (SDA) and 5 (SCL). Support I2C (TWI) communication using the Wire library.

There are a couple of other pins on the board:

- **AREF:** Reference voltage for the analog inputs. Used with analogReference ().
- **Reset:** Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

➢ Communication

The communication protocols of an Arduino Uno include SPI, I2C, and UART serial communication **[7].**

- **UART:** An Arduino Uno uses the two functions like the transmitter digital pin1 and the receiver digital pin0. These pins are mainly used in UART TTL serial communication.
- **I2C**: An Arduino UNO board employs SDA pin otherwise A4 pin & A5 pin otherwise SCL pin is used for I2C communication with wire library. In this, both the SCL and SDA are CLK signal and data signal.
- **SPI Pins:** The SPI communication includes MOSI, MISO, and SCK.

Table III. 1 Technical specification for Arduino

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

III.4.3 XBee S2C

Figure III. 8 XBee S2C.

XBee S2C is a RF module designed for wireless communication or data exchange and it works on ZigBee mesh communication protocols that sit on top of IEEE 802.15.4 PHY. The module provides wireless connectivity to end-point devices in any ZigBee mesh networks including devices from other vendors **[9]**.

➢ **Features and Electrical Characteristics [9]:**

- Transmission Frequency: 2.4GHz to 2.5GHz
- Number of Channels: 16 Direct Sequence Channels
- Featured with UART (250 Kb/s maximum) and SPI (5 Mb/s maximum) interface
- Featured with software adjustable transmitting power
- Indoor/Urban Range: 200ft
- Outdoor RF line-of-sight Range: up to 4000ft

- Transmit Power Output: 6.3mW (8dBm) in Boost mode,2mW (3dBm) in Normal mode
- RF Data Rate: 250,000 bps
- Receiver Sensitivity: -102dBm in Boost mode, -100dBm in Normal mode
- Supply Voltage Range: +2.1V to +3.6V
- Operating Current: 33mA (at3.3V, for Normal mode) , 45mA (at 3.3V,for Boost mode)
- Idle Current: 9mA
- Maximum output current on all pins together: 40mA
- Power-down Current: <1uA @25C
- ESD protection: 3000V
- Operating Temperature: -40ºC to 85° C

III.4.4 Bees Shield

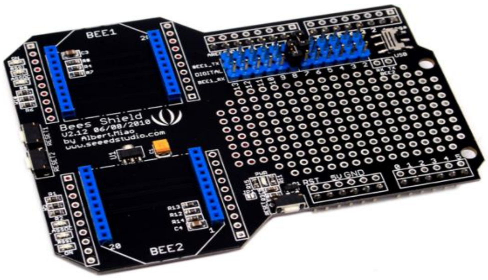


Figure III. 9 Bees Shield.

Bees Shield will make interfacing multiple Bee-styles (XBee, GPRS Bee, Bluetooth Bee and etc) easier than ever before. Aside from two Bee-style 20p 2.0 pitch sockets, it also has a large prototyping area, and a customizable software serial port for easier prototyping [10].

➢ **Features [10]:**

- Dual Bee type socket.
- 3 indicators LED (ON/Sleep, RSSI, ASSOC) for each XBee.
- Full size with free drills.
- Reset button for each XBee.
- Reset button for base board.
- Provide maximal 500mA under 3.3V
- Full break out for each Bee.
- Switchable of communication with FTDI-USB /Base board.

III.4.5 Adapter Board “UartSBee V4"

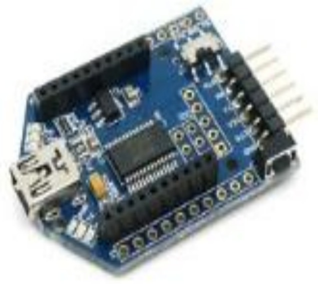


Figure III. 10 Adapter board.

The adapter board is designed to make adding wireless point-to-point or mesh networking easy. The latter has an integrated 3.3 V regulator, 6 display LEDs and 2 female connectors with a pitch of 2 mm (on which the XBee ™ modules can be inserted) **[11]**.

- I / O 3.3V or 5V compatible (selection by mini-switch)
- Dual 3.3V and 5V power output
- RESET button

III.5 Communication system point-to-point

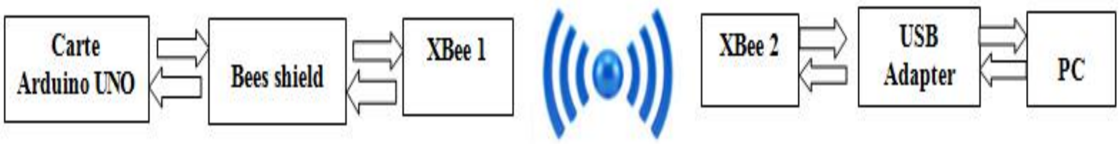


Figure III. 11 Communication diagram.

The communication process for implementing the coordinator and End-device apparatus is fully explained in the figure above. .Using Arduino XBee shield that has the role of sender and XBee adapter which is a receiver , We did a project looking at configuring a chat environnement to send and receive data and then to control Arduino's LED through XBee communication will be implemented by a wireless communication system via serial link .

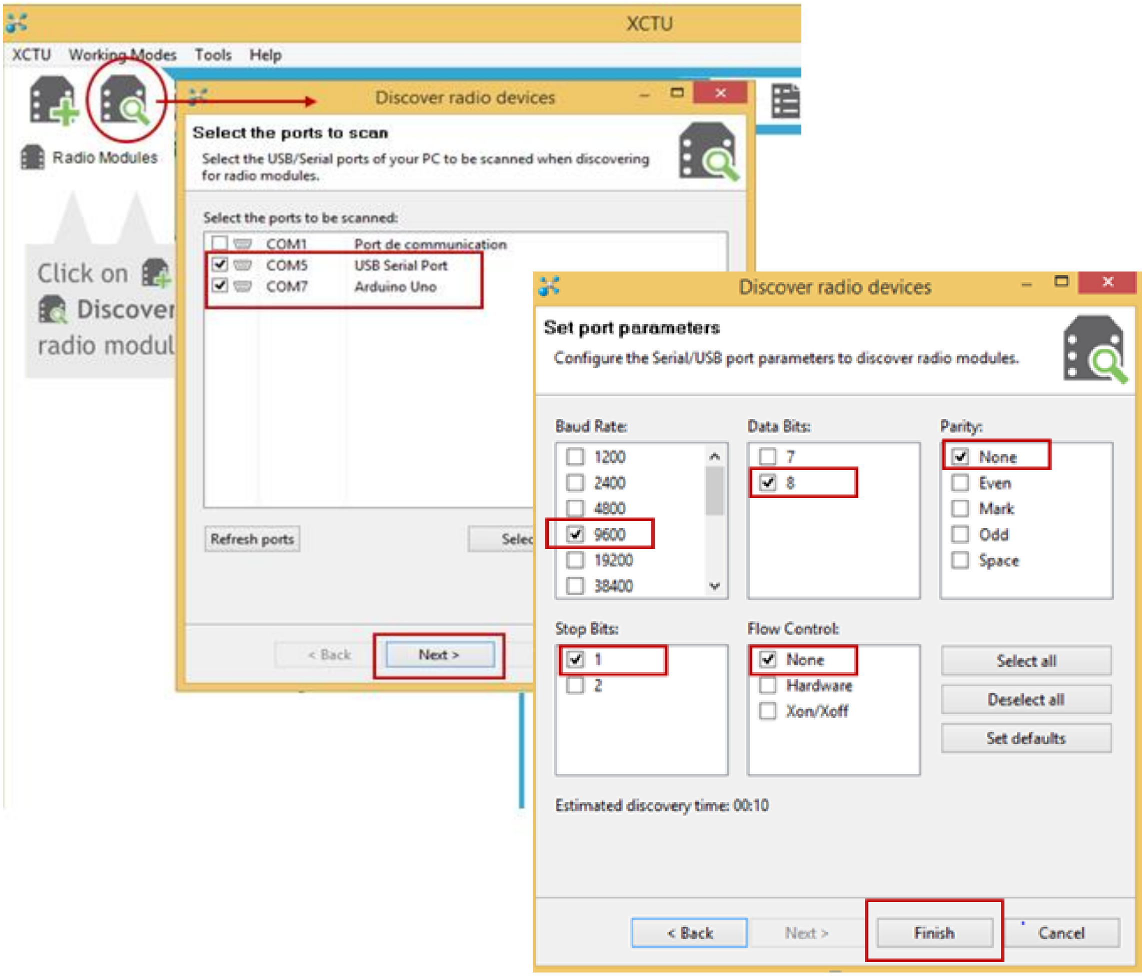
II.6 XBee configuration for Coordinator and End-device

In order to get the two XBee S2C units to talk to each other, we need to set up one as moderator and one as end device, we will need a serial connection from PC to the two XBee modules.

When the X-CTU is installed correctly, XBee can be connected to the Arduino shield and USB adapter of the computer. We must know the COM number granted to these devices in order to specify it in the X-CTU.

Therefore, the following steps must be followed for each XBee unit:

- ✓ Connect devices to a computer
- ✓ We choose the serial ports then set port parameter.



Discover radio devices

Select the ports to scan

Select the USB/Serial ports of your PC to be scanned when discovering for radio modules.

Select the ports to be scanned:

□	COM1	Port de communication
☑	COM5	USB Serial Port
☑	COM7	Arduino Uno

Discover radio devices

Set port parameters

Configure the Serial/USB port parameters to discover radio modules.

Baud Rate: □ 1200 □ 2400 □ 4800 ☑ 9600 □ 19200 □ 38400

Data Bits: □ 7 ☑ 8

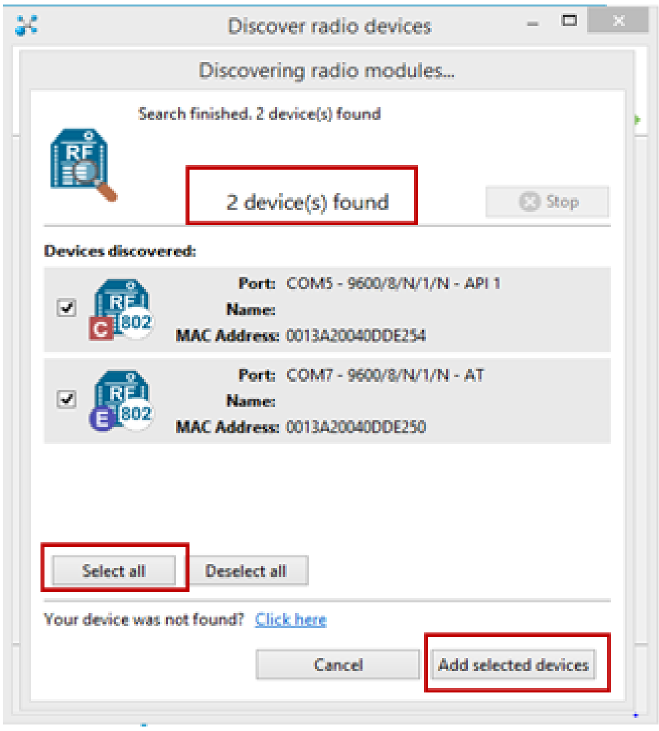
Parity: ☑ None □ Even □ Mark □ Odd □ Space

Stop Bits: ☑ 1 □ 2

Flow Control: ☑ None □ Hardware □ Xon/Xoff

Estimated discovery time: 00:10

-a-



Discover radio devices

Discovering radio modules...

Search finished. 2 device(s) found

2 device(s) found

Devices discovered:

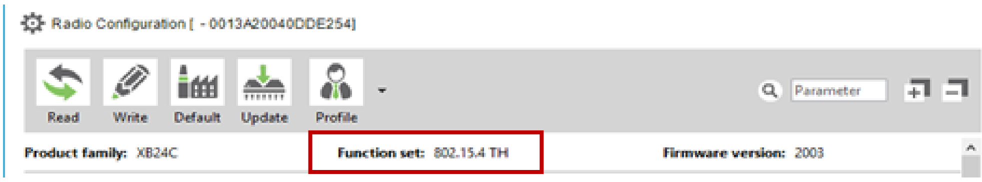
☑ Port: COM5 - 9600/8/N/1/N - API 1
Name:
MAC Address: 0013A20040DDE254

☑ Port: COM7 - 9600/8/N/1/N - AT
Name:
MAC Address: 0013A20040DDE250

Your device was not found? Click here

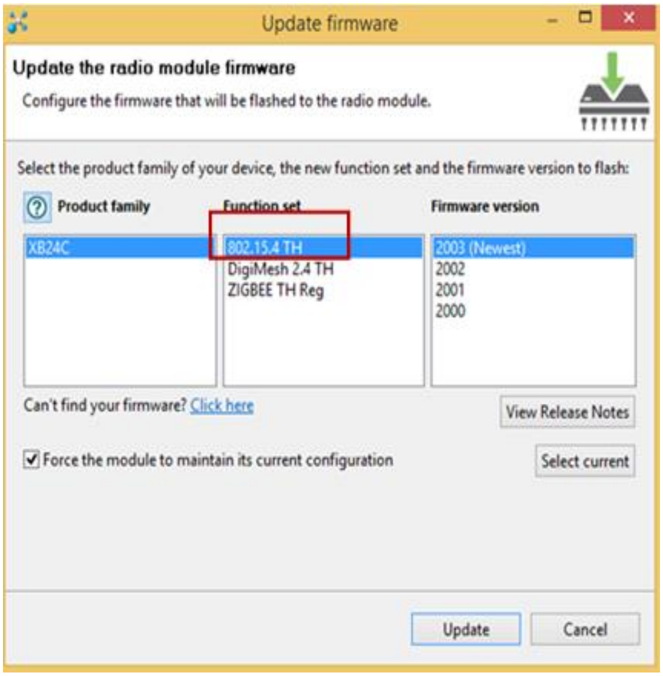
-b-

- We setup using firmware 802.15.4 TH



Radio Configuration [- 0013A20040DDE254]

Product family: XB24C Function set: 802.15.4 TH Firmware version: 2003



Update firmware

Update the radio module firmware

Configure the firmware that will be flashed to the radio module.

Select the product family of your device, the new function set and the firmware version to flash:

Product family	Function set	Firmware version
XB24C	802.15.4 TH DigiMesh 2.4 TH ZIGBEE TH Reg	2003 (Newest) 2002 2001 2000

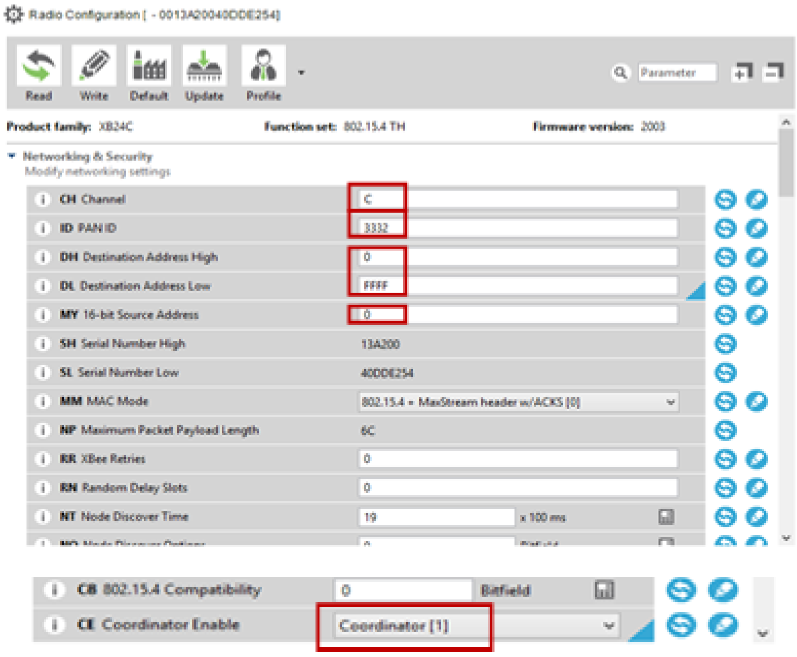
Can't find your firmware? Click here

☑ Force the module to maintain its current configuration

Figure III. 12 configuration step XBee.

III.5.1 XBee configuration for Coordinator:

The XBee S2C module implemented in Coordinator mode according to the following parameters:



Radio Configuration [- 0013A20040DDE254]

Product family: XB24C Function set: 802.15.4 TH Firmware version: 2003

Networking & Security
Modify networking settings

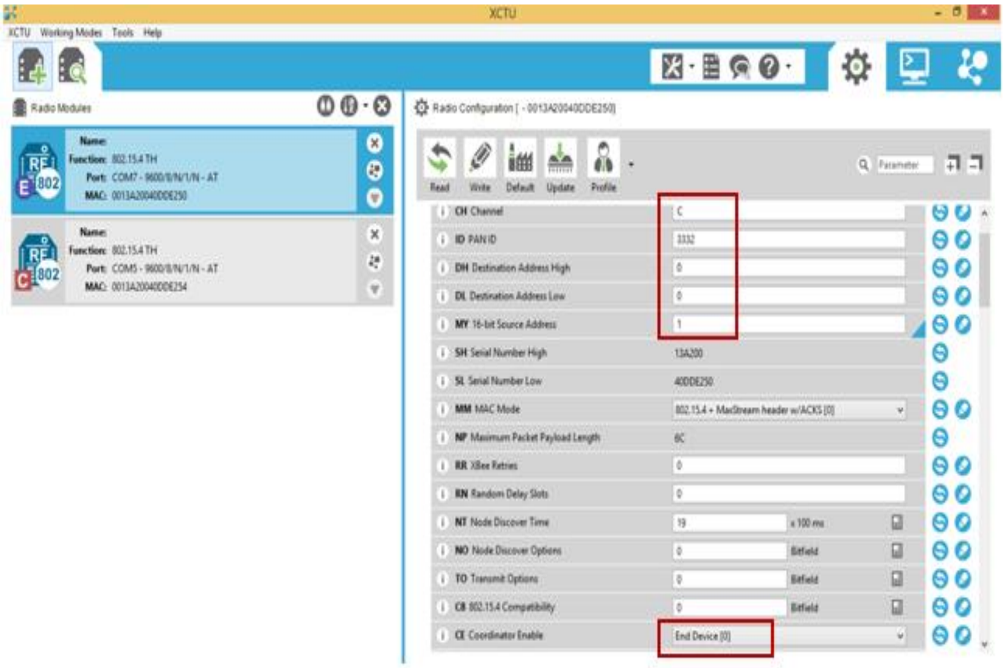
CH Channel C
ID PAN ID 3332
DH Destination Address High 0
DL Destination Address Low FFFF
MY 16-bit Source Address 0
SH Serial Number High 13A200
SL Serial Number Low 40DDE254
MM MAC Mode 802.15.4 + MaxStream header w/ACKS [0]
NP Maximum Packet Payload Length 6C
RR XBee Retries 0
RN Random Delay Slots 0
NT Node Discover Time 19 x 100 ms

C8 802.15.4 Compatibility 0 Bitfield
CE Coordinator Enable Coordinator [1]

Figure III. 13 XBee configuration step for Coordinator.

- **CH Channel**: This defaults to C and can be left alone.
- **ID PAN ID**: This is a number that is unique to your network setup and needs to be the same on all the XBee modules. To avoid potential conflict with other XBee radios, change this to any hexadecimal number between 0 and FFFF. Here i have selected **3332**.
- **DH Destination Address High** : Set this to 0
- **DL Destination Address Low**: Because the XBee device we are setting up is a Coordinator, we need to allow it to communicate with all other XBee radios on the same network. To do this we set this value to FFFF which is the Broadcast Address to all other modules.
- **MY 16-bit source address**: Leave this at 0
- **CE Coordinator Enable:** We put it on the coordinator [1] so that this unit becomes coordinated.

III.5.2 XBee configuration for End-device:



Name:
Function: 802.15.4 TH
Port: COM7 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE250

Name:
Function: 802.15.4 TH
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE254

Radio Configuration [- 0013A20040DDE250]

CH Channel: C
ID PAN ID: 3332
DH Destination Address High: 0
DL Destination Address Low: 0
MY 16-bit Source Address: 1
SH Serial Number High: 13A200
SL Serial Number Low: 40DDE250
MM MAC Mode: 802.15.4 + MaxStream header w/ACKS [0]
NP Maximum Packet Payload Length: 6C
RR XBee Retries: 0
RN Random Delay Slots: 0
NT Node Discover Time: 19 x 100 ms
NO Node Discover Options: 0 Bitfield
TO Transmit Options: 0 Bitfield
C8 802.15.4 Compatibility: 0 Bitfield
CE Coordinator Enable: End Device [0]

Figure III. 14 XBee configuration step End-device.

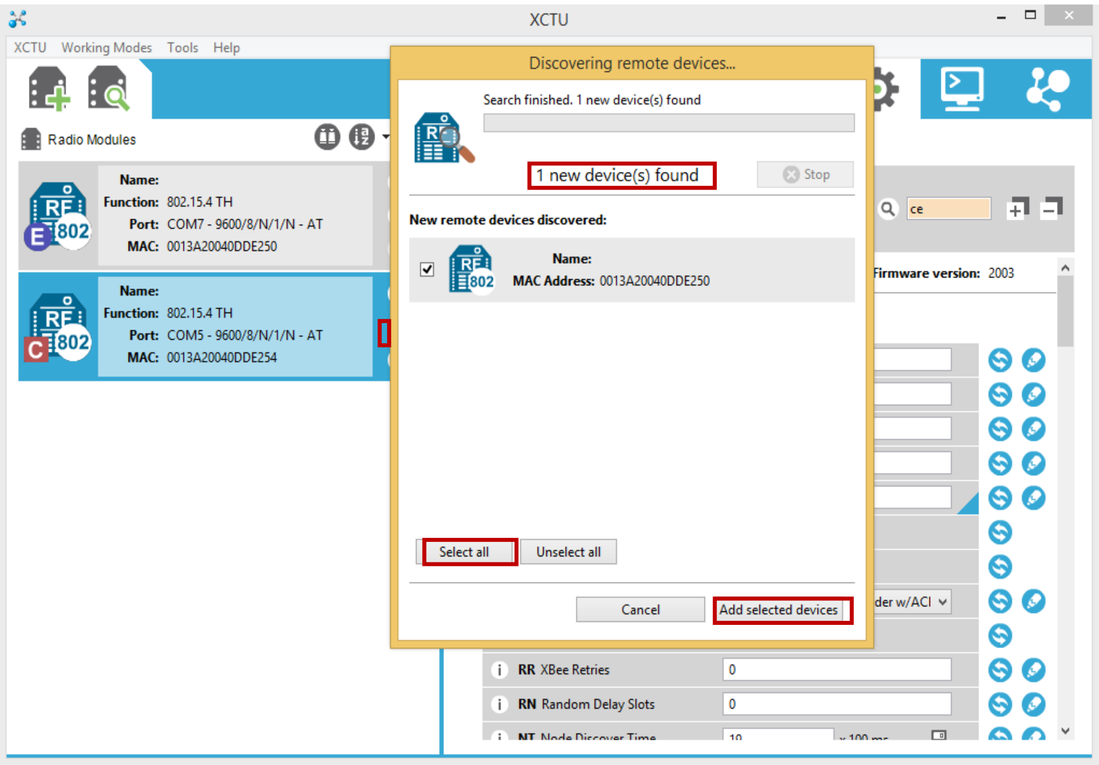
- **CH Channel**: This defaults to **C** and can be left alone, but should match the Coordinator
- **ID PAN:** I have selected **3332-** the same as the Coordinator.
- **DH Destination Address High**: Set this to **0.**
- **DL Destination Address Low**: Set this to 0.

  Along with the DH high address, this is the address of the coordinator module.

  (**MY** setting)
- **MY 16-bit source address**: Set this to **1**
- **CE Coordinator Enable:** Is set to **End Device [0]**

III.7 Test result

1. **1.** Testing communication between XBee modules using XBee Shield with Arduino Uno:



Name:
Function: 802.15.4 TH
Port: COM7 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE250

Name:
Function: 802.15.4 TH
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE254

Search finished. 1 new device(s) found

1 new device(s) found

New remote devices discovered:

☑ Name:
MAC Address: 0013A20040DDE250

Firmware version: 2003

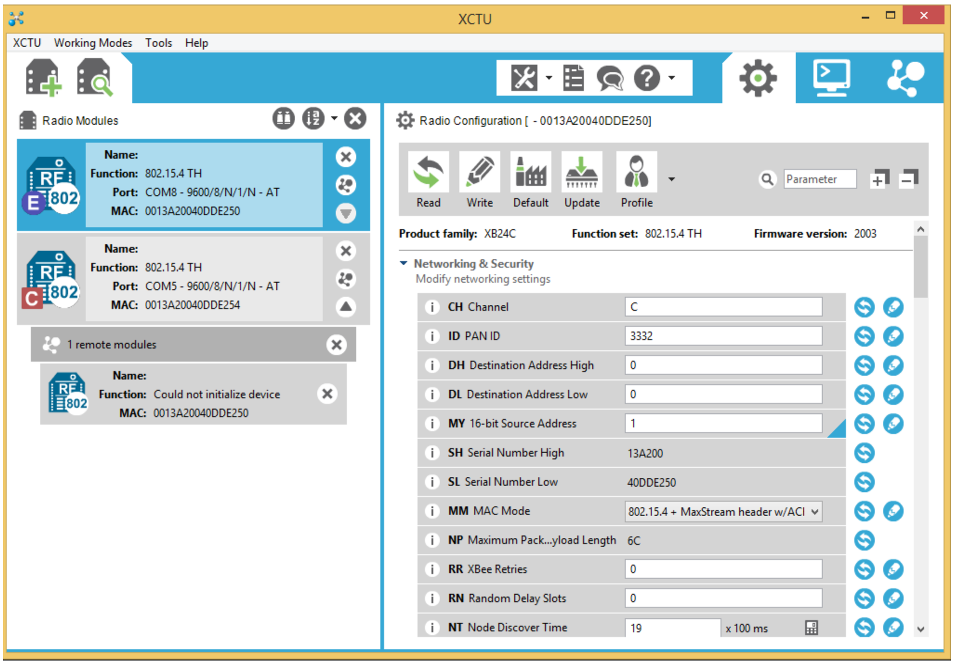
RR XBee Retries 0

RN Random Delay Slots 0

NT Node Discover Time 10 x 100 ms

Figure III. 15 Testing communication between XBee.

- ✓ As in the image above, there is a remote controller underneath the second XBee unit. If you look at the MAC address, you can see that it is the first XBee module for the End-device.



Name:
Function: 802.15.4 TH
Port: COM8 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE250

Name:
Function: 802.15.4 TH
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE254

1 remote modules

Name:
Function: Could not initialize device
MAC: 0013A20040DDE250

Radio Configuration [- 0013A20040DDE250]

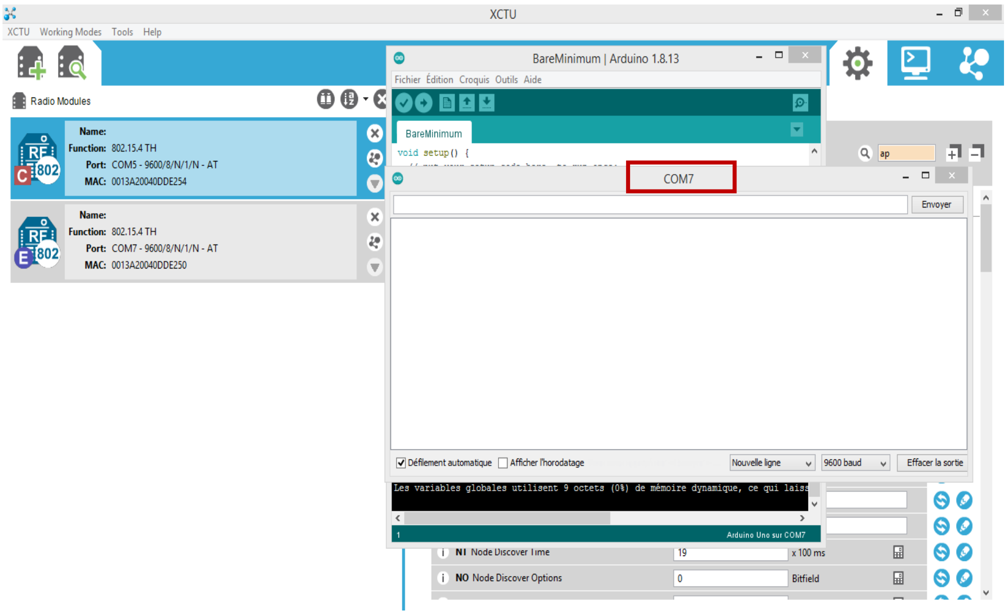
Product family: XB24C Function set: 802.15.4 TH Firmware version: 2003

Networking & Security
Modify networking settings

CH Channel C
ID PAN ID 3332
DH Destination Address High 0
DL Destination Address Low 0
MY 16-bit Source Address 1
SH Serial Number High 13A200
SL Serial Number Low 40DDE250
MM MAC Mode 802.15.4 + MaxStream header w/ACI
NP Maximum Pack...yload Length 6C
RR XBee Retries 0
RN Random Delay Slots 0
NT Node Discover Time 19 x 100 ms

Figure III. 16 Testing communication between XBee.

- ✓ Next, turn on the serial monitor in the Arduino IDE you can see that it is COM7 connected the Arduino Uno, and the baund rate is set to 9600.


XCTU
XCTU Working Modes Tools Help
Radio Modules
BareMinimum | Arduino 1.8.13
Fichier Édition Croquis Outils Aide
BareMinimum
void setup() {
Envoyer

Name:
Function: 802.15.4 TH
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE254

Name:
Function: 802.15.4 TH
Port: COM7 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE250

COM7

☑ Défilement automatique □ Afficher l'horodatage   Nouvelle ligne   9600 baud   Effacer la sortie

```
Les variables globales utilisent 9 octets (0%) de mémoire dynamique, ce qui laiss
```

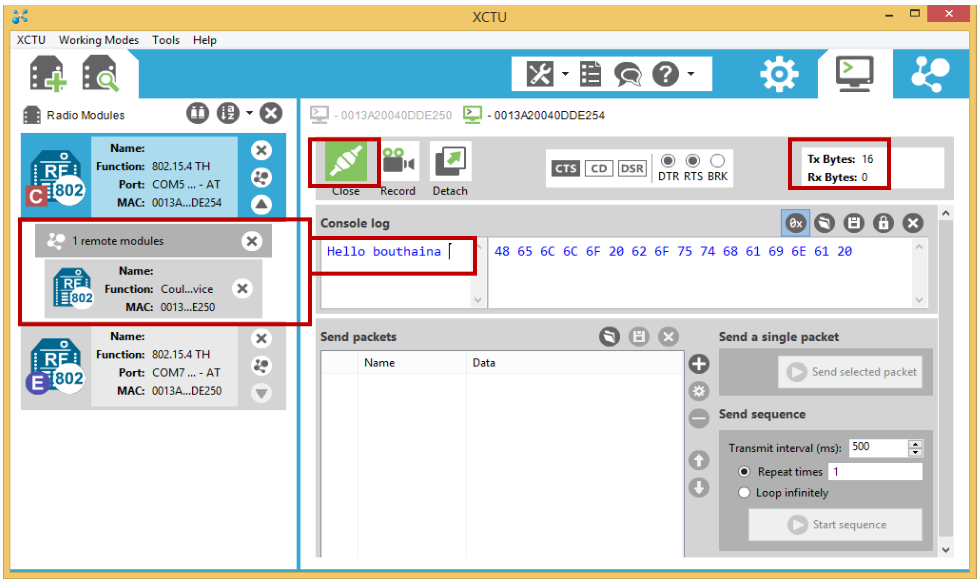
Arduino Uno sur COM7

NT Node Discover Time 19 x 100 ms
NO Node Discover Options 0 Bitfield

Figure III. 17 Testing communication between XBee.

- ✓ Go back to X-ctu and select the menu in the shape of the monitor on the top right to switch the window. Then, we select the second XBee module and click the right connect button.
- ✓ If the connection is successful, it turns green as shown in the picture above and the Close text appears. If the connection is successful, try entering text in the console log (you can distinguish between sending and receiving with text color).

Enter “Hello bouthaina” => Blue text is transmission, red text is reception.



Name:
Function: 802.15.4 TH
Port: COM5 ... - AT
MAC: 0013A...DE254

1 remote modules
Name:
Function: Coul...vice
MAC: 0013...E250

Name:
Function: 802.15.4 TH
Port: COM7 ... - AT
MAC: 0013A...DE250

- 0013A20040DDE250   - 0013A20040DDE254

Tx Bytes: 16
Rx Bytes: 0

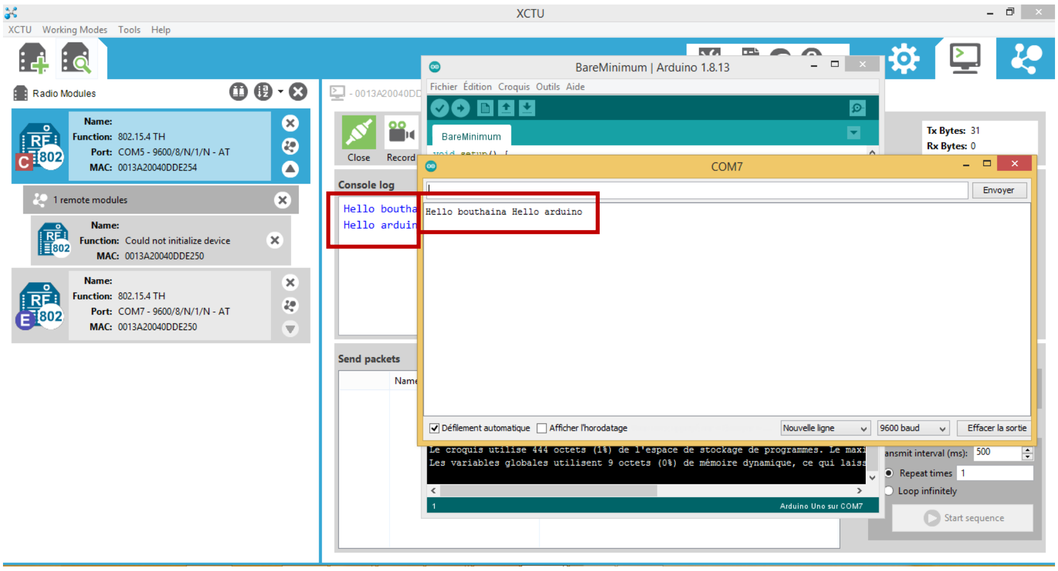
Console log

```
Hello bouthaina     48 65 6C 6C 6F 20 62 6F 75 74 68 61 69 6E 61 20
```

Transmit interval (ms): 500
● Repeat times 1
○ Loop infinitely

Figure III. 18 Testing communication between XBee.

- ✓ Then the Arduino IDE data received from the serial monitor lets you putput a "hello bouthaina and hello Arduino" transmit to XBee connected to XBee adapter (transmit to X-ctu) and recieve to XBee connected to XBee shield (receive to IDE).



BareMinimum | Arduino 1.8.13

COM7

```
Hello bouthaina Hello arduino
```

Console log

```
Hello boutha
Hello arduin
```

```
Le croquis utilise 444 octets (1%) de l'espace de stockage de programmes. Le maxi
Les variables globales utilisent 9 octets (0%) de mémoire dynamique, ce qui laiss
```

Tx Bytes: 31
Rx Bytes: 0

Name:
Function: 802.15.4 TH
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE254

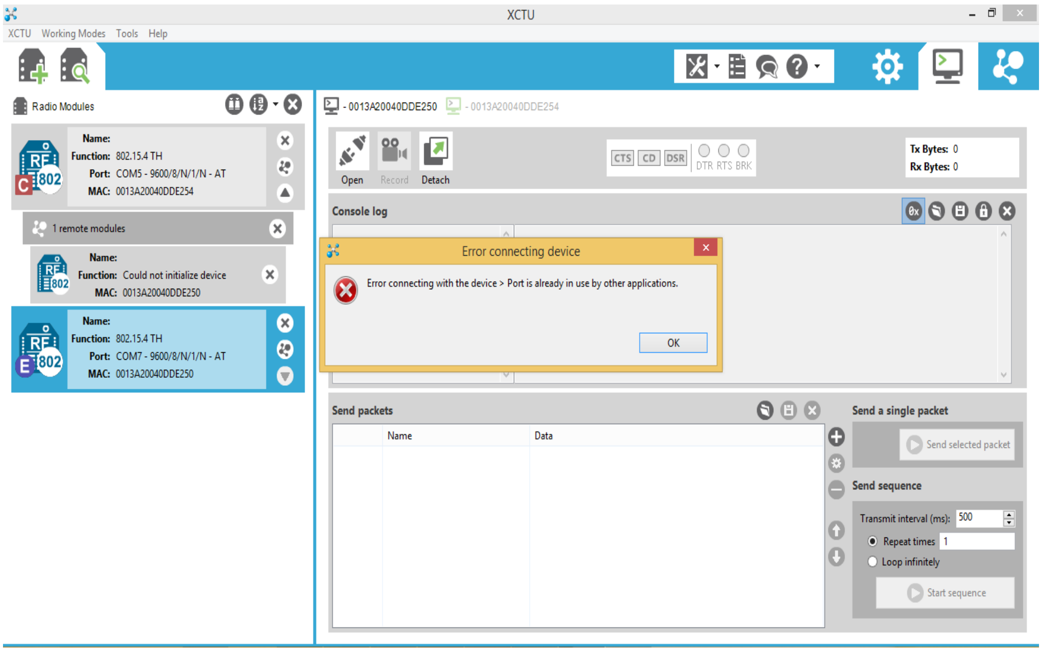
Name:
Function: Could not initialize device
MAC: 0013A20040DDE250

Name:
Function: 802.15.4 TH
Port: COM7 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE250

☑ Défilement automatique □ Afficher l'horodatage Nouvelle ligne 9600 baud

Figure III. 19 Testing communication between XBee.

- ✓ In this state, if you press the connect button on the right menu of end device module with the mind that you need to transmit and receive via X-ctu. The connection fails and error appears.
- ✓ In this case, can send/receive via X-CTU by terminating the serial monitor of Arduino IDE.



Error connecting device

Error connecting with the device > Port is already in use by other applications.

- 0013A20040DDE250 - 0013A20040DDE254

Tx Bytes: 0
Rx Bytes: 0

Name:
Function: 802.15.4 TH
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE254

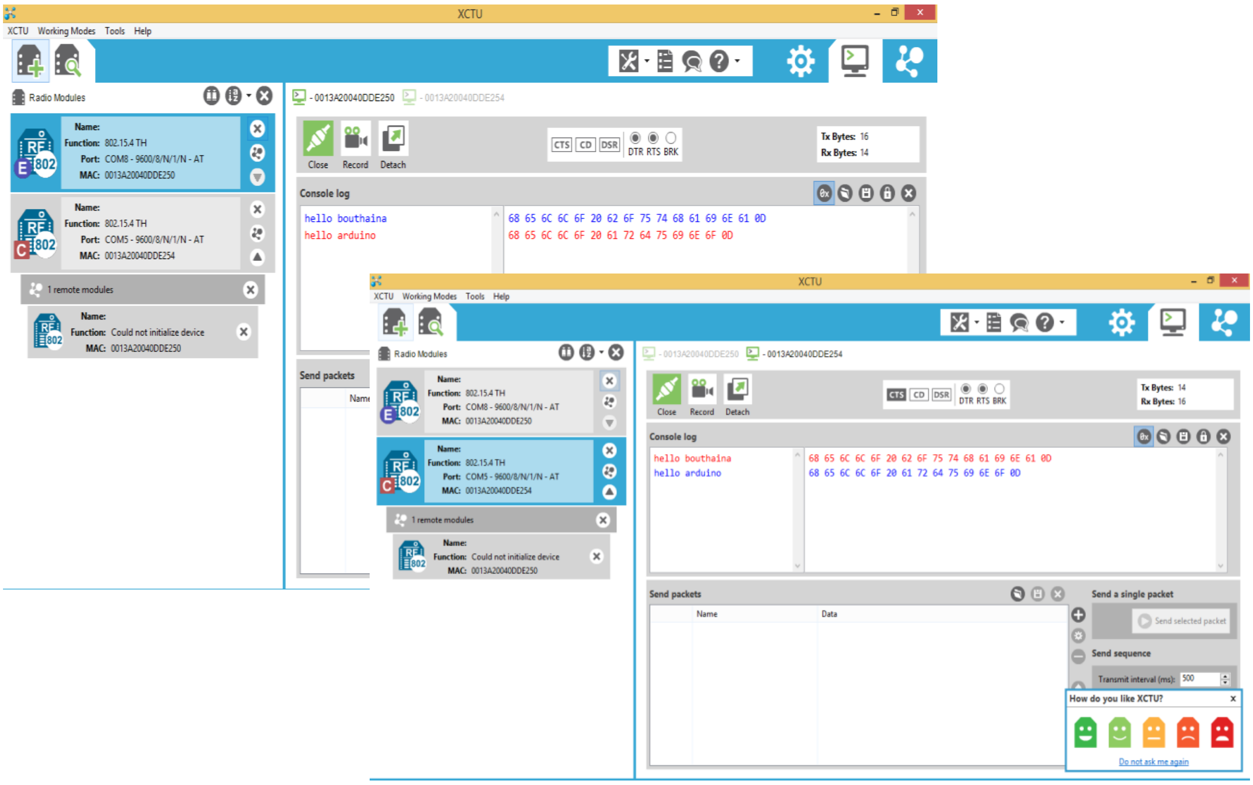
Name:
Function: Could not initialize device
MAC: 0013A20040DDE250

Name:
Function: 802.15.4 TH
Port: COM7 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE250

Transmit interval (ms): 500
● Repeat times 1
○ Loop infinitely

Figure III. 20 Testing communication between XBee.

- ✓ Every time we locate the XBee module, connect it in, and enter "hello bouthaina and hello Arduino" into the console log.



Name:
Function: 802.15.4 TH
Port: COM8 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE250

Name:
Function: 802.15.4 TH
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE254

Name:
Function: Could not initialize device
MAC: 0013A20040DDE250

Transmit interval (ms): 500

- 0013A20040DDE250 — Tx Bytes: 16, Rx Bytes: 14

Console log

```
hello bouthaina     68 65 6C 6C 6F 20 62 6F 75 74 68 61 69 6E 61 0D
hello arduino       68 65 6C 6C 6F 20 61 72 64 75 69 6E 6F 0D
```

- 0013A20040DDE254 — Tx Bytes: 14, Rx Bytes: 16

Console log

```
hello bouthaina     68 65 6C 6C 6F 20 62 6F 75 74 68 61 69 6E 61 0D
hello arduino       68 65 6C 6C 6F 20 61 72 64 75 69 6E 6F 0D
```

Figure III. 21 Testing communication between XBee.

2. Control the LED wired to Arduino

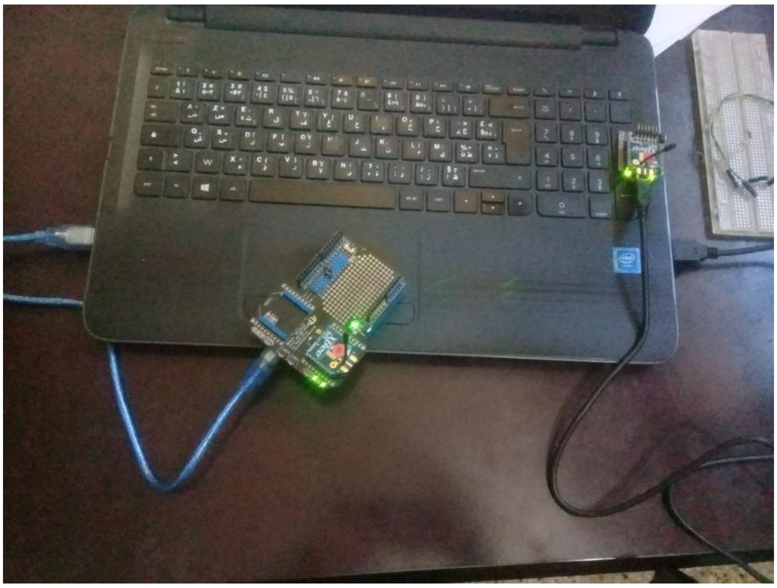


Figure III. 22 Arduino XBee shield connecting with XBee adapter.

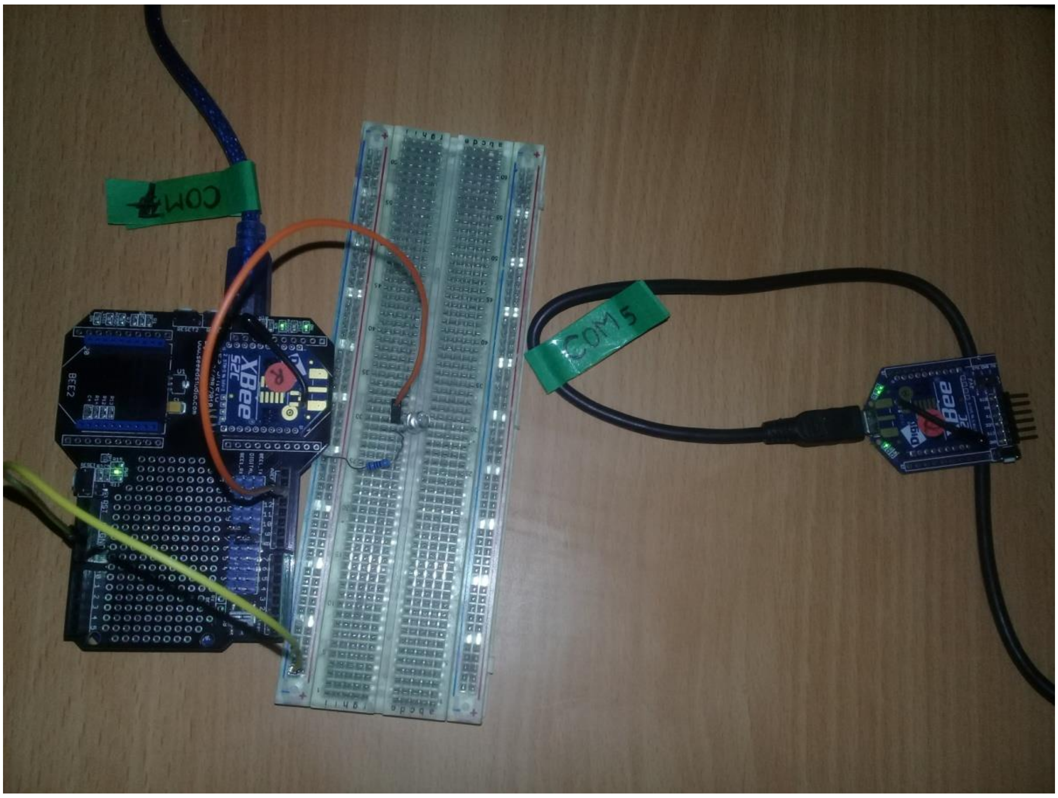
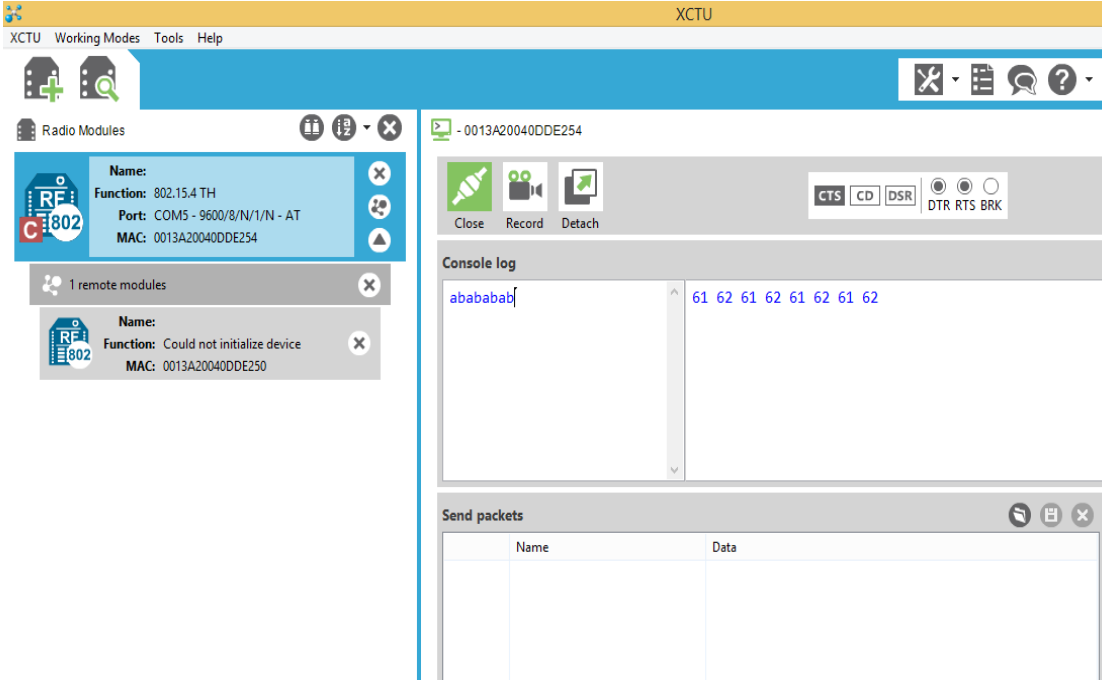


Figure III. 23 control the LED wired to Arduino using Arduino XBee Shield and USB adapter.

Here to control the LED wired to Arduino through XBee connections using Arduino XBee Shield and USB adapter we worked as follows:



- 0013A20040DDE254

Name:
Function: 802.15.4 TH
Port: COM5 - 9600/8/N/1/N - AT
MAC: 0013A20040DDE254

Name:
Function: Could not initialize device
MAC: 0013A20040DDE250

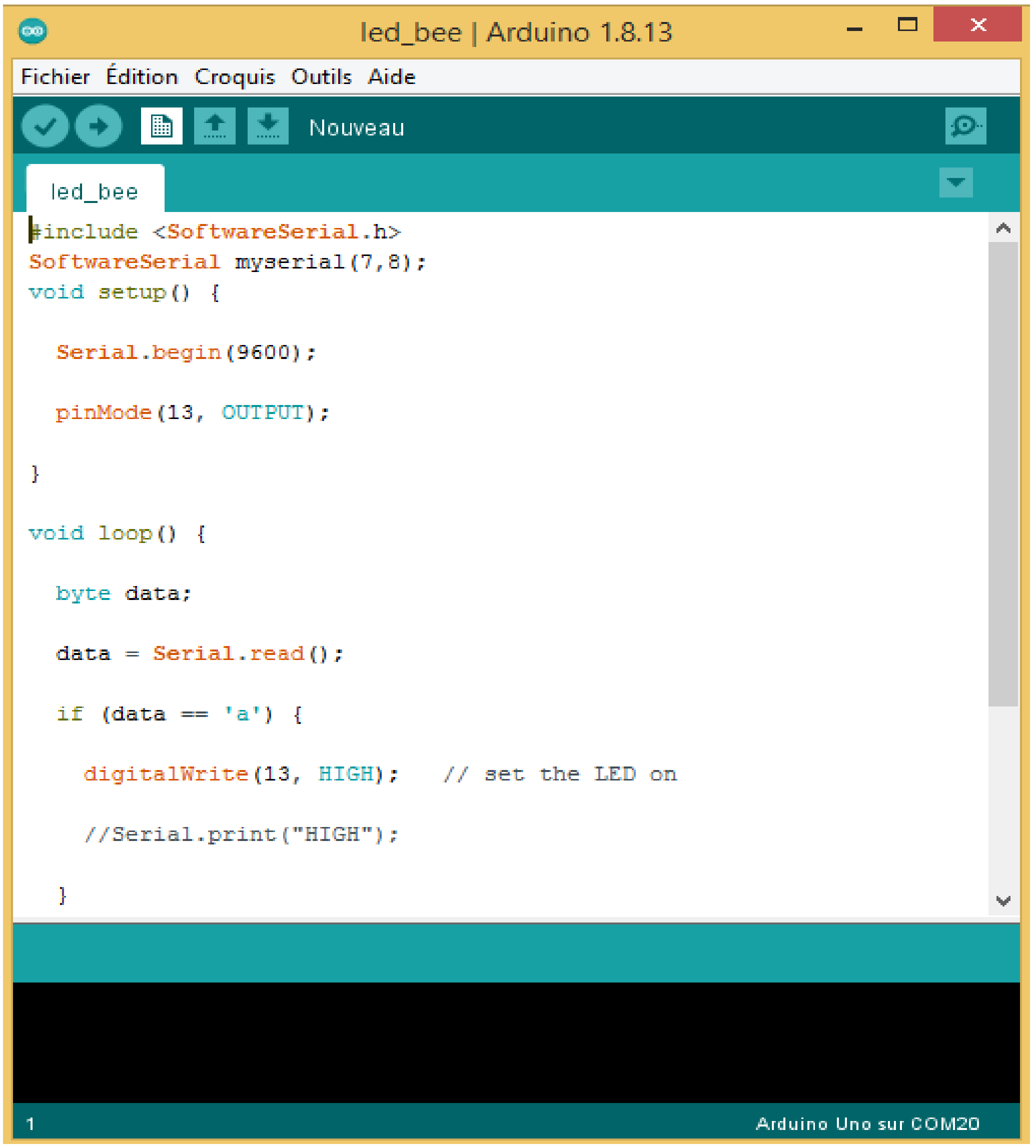
```
abababab
```

```
61 62 61 62 61 62 61 62
```

Figure III. 24 control the LED wired.

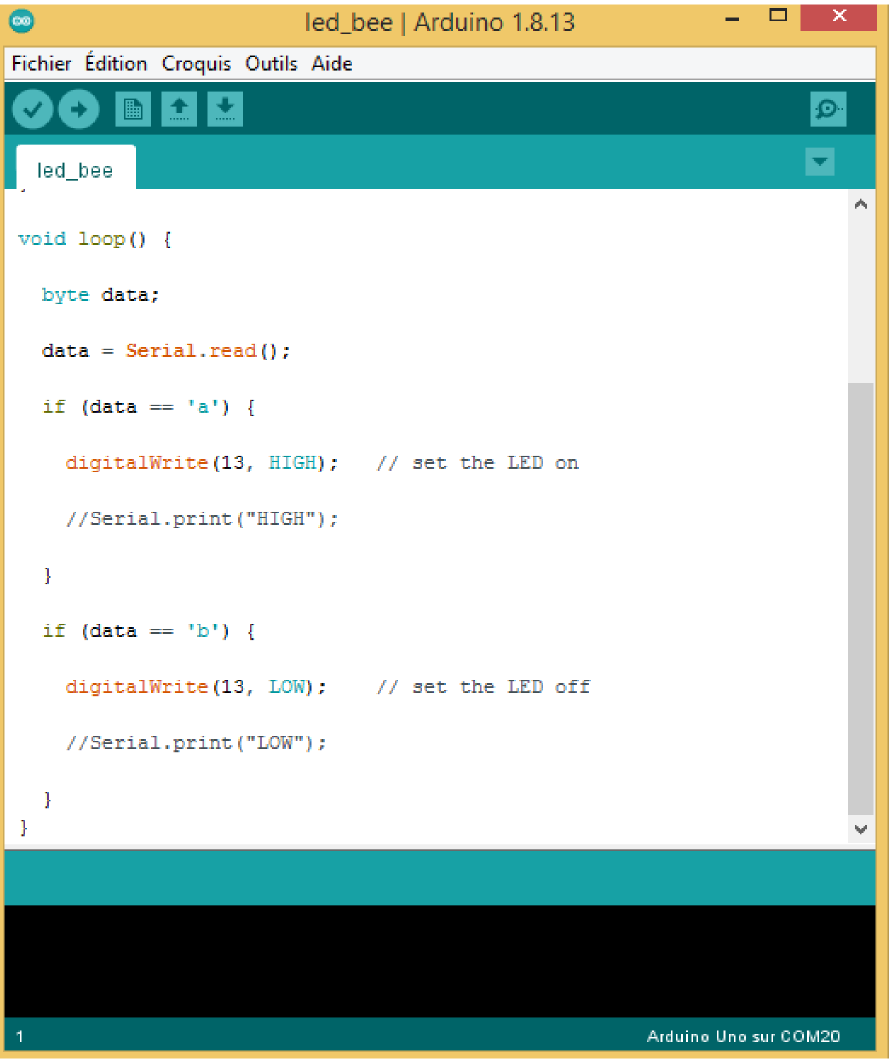
As shown in the figure III.18, we connect two XBees to Arduino Uno, XBee shield and XBee adapter and connect them to computer. Then in figure III.19 connecting pinMode 13 and the LED is turned on If you enter alphabet a in the console window, the LED connected to the Arduino will turn on and if you enter alphabet b in the console window the LED connected to the Arduino turns off.

Code Arduino:



```
#include <SoftwareSerial.h>
SoftwareSerial myserial(7,8);
void setup() {

  Serial.begin(9600);

  pinMode(13, OUTPUT);

}

void loop() {

  byte data;

  data = Serial.read();

  if (data == 'a') {

    digitalWrite(13, HIGH);   // set the LED on

    //Serial.print("HIGH");

  }
```

Figure III. 25 code Arduino control the LED wired



```
void loop() {

  byte data;

  data = Serial.read();

  if (data == 'a') {

    digitalWrite(13, HIGH);   // set the LED on

    //Serial.print("HIGH");

  }

  if (data == 'b') {

    digitalWrite(13, LOW);    // set the LED off

    //Serial.print("LOW");

  }
}
```

Figure III. 26 code Arduino control the LED wired

III.8 Practical Realization

Here we applied the practical side using the Arduino Uno and two XBee usb adapter project looking to control the brightness of the LED wireless XBee communication in the following diagram:

1. **Wiring diagram-transmitter:**

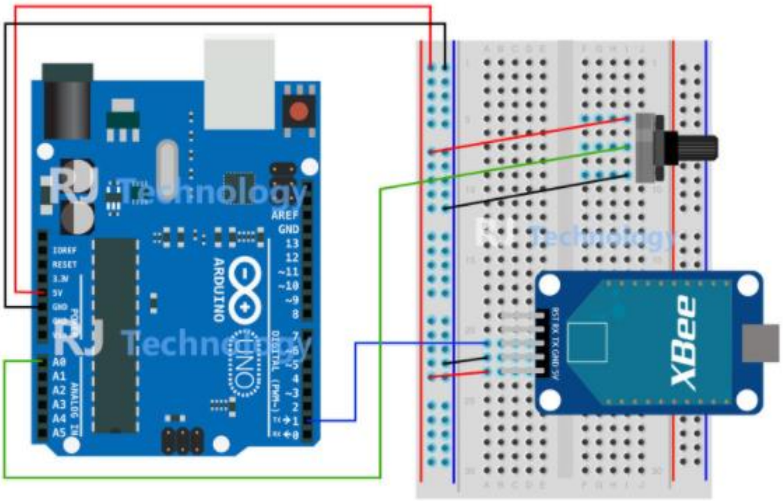


Figure III. 27 wiring diagram-transmitter

2. **wiring diagram-receiver:**

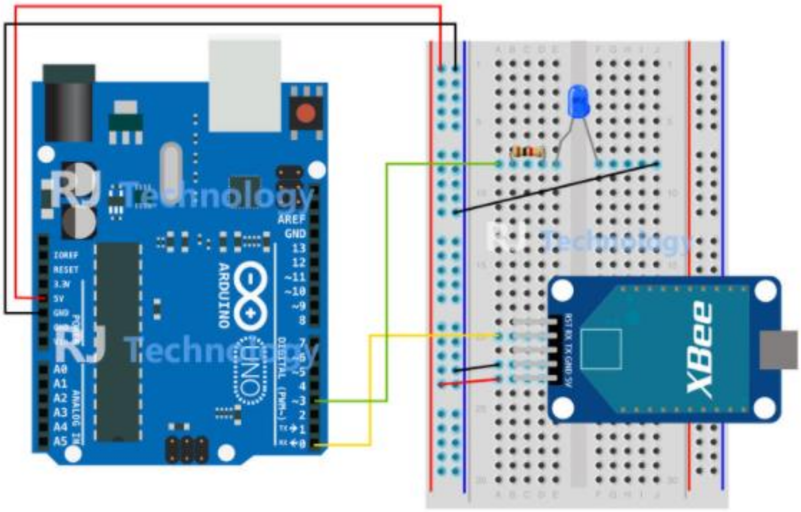


Figure III. 28 wiring diagram-receiver

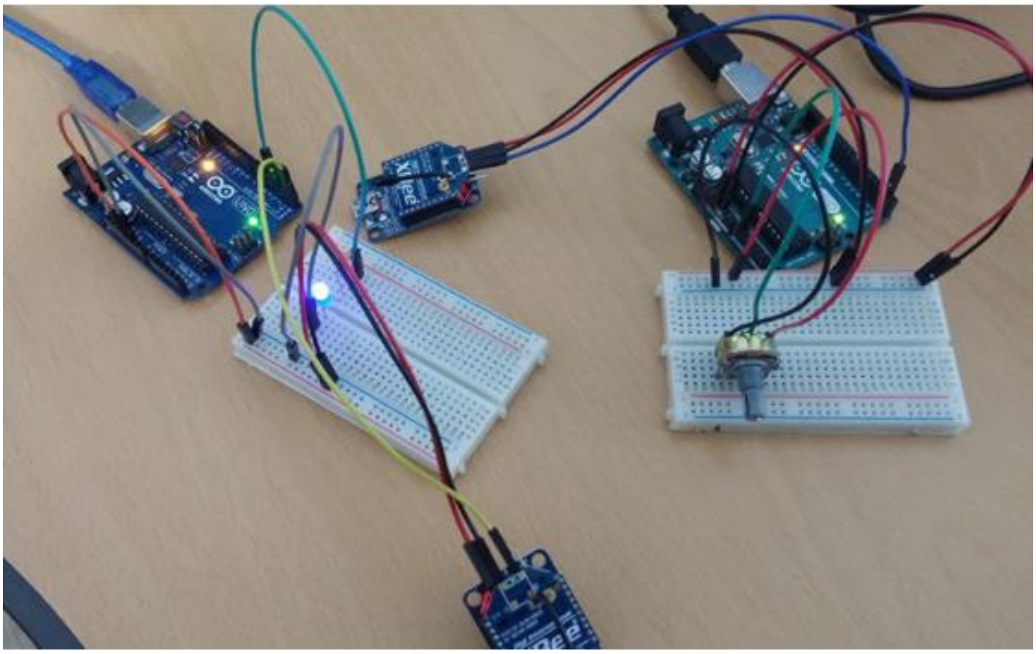
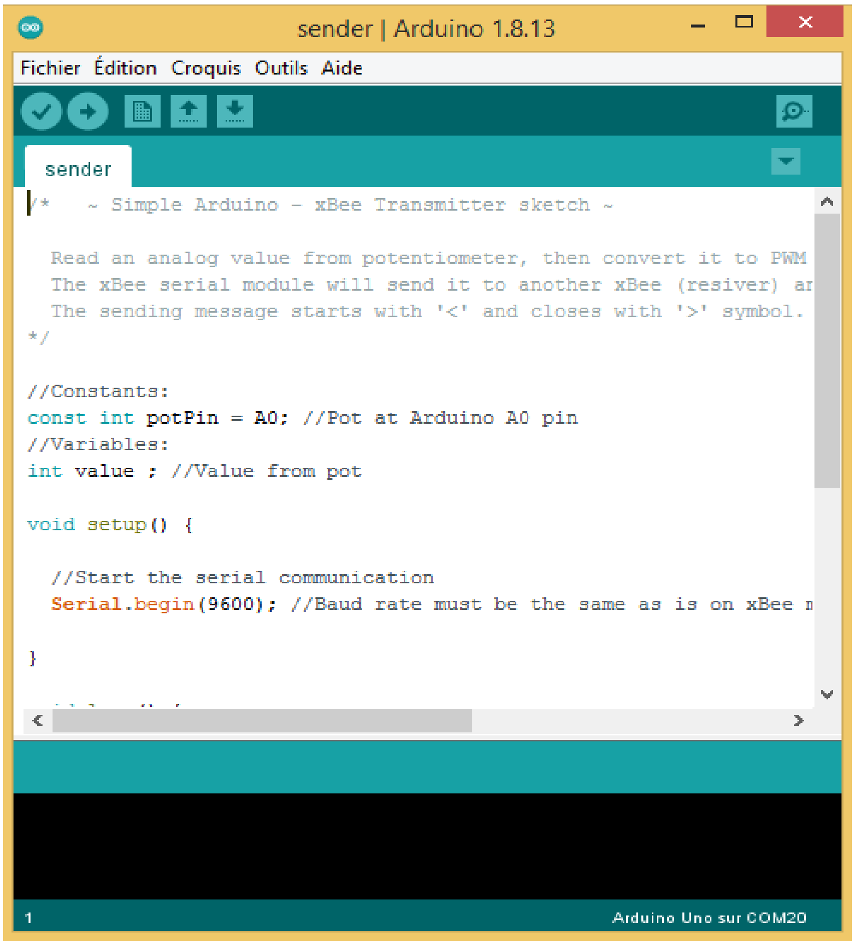


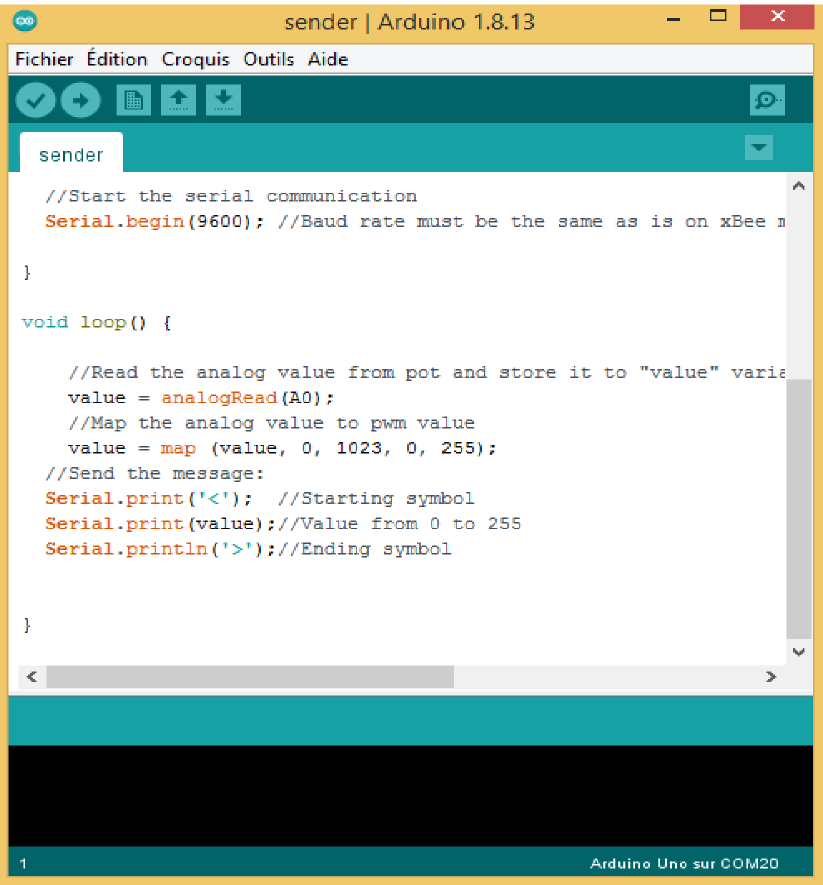
Figure III. 29 Actual assembly

- **Code Arduino sender :**



```
/*   ~ Simple Arduino - xBee Transmitter sketch ~

  Read an analog value from potentiometer, then convert it to PWM
  The xBee serial module will send it to another xBee (resiver) ar
  The sending message starts with '<' and closes with '>' symbol.
*/

//Constants:
const int potPin = A0; //Pot at Arduino A0 pin
//Variables:
int value ; //Value from pot

void setup() {

  //Start the serial communication
  Serial.begin(9600); //Baud rate must be the same as is on xBee m

}
```

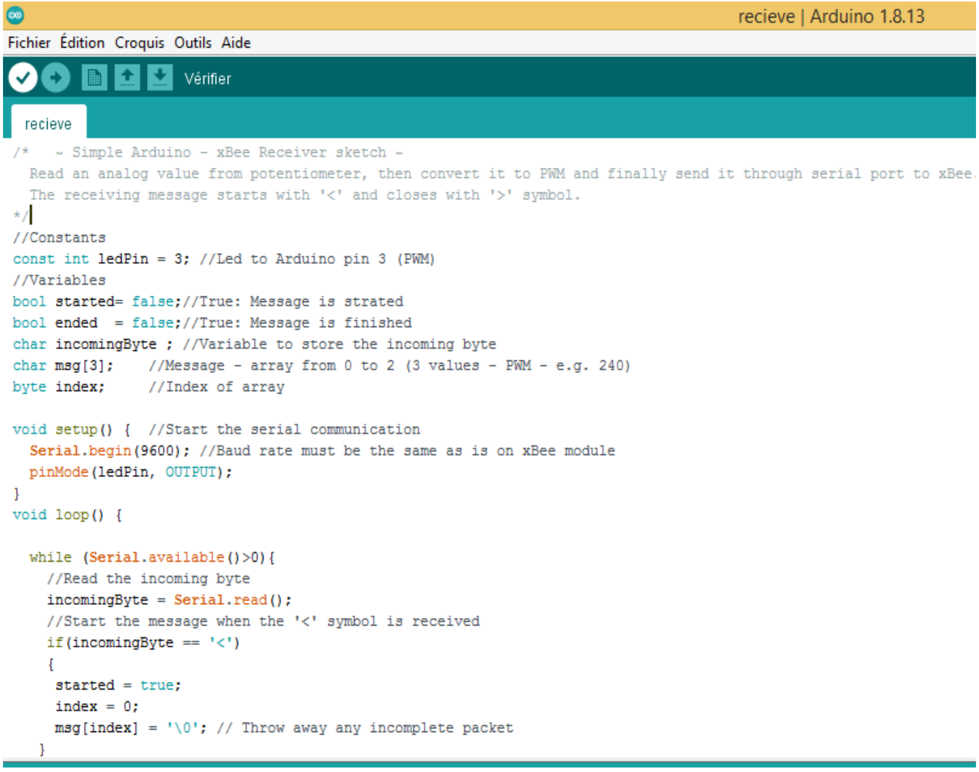
Figure III. 30 Code Arduino sender



```
  //Start the serial communication
  Serial.begin(9600); //Baud rate must be the same as is on xBee m

}

void loop() {

    //Read the analog value from pot and store it to "value" varia
    value = analogRead(A0);
    //Map the analog value to pwm value
    value = map (value, 0, 1023, 0, 255);
  //Send the message:
  Serial.print('<');  //Starting symbol
  Serial.print(value);//Value from 0 to 255
  Serial.println('>');//Ending symbol


}
```

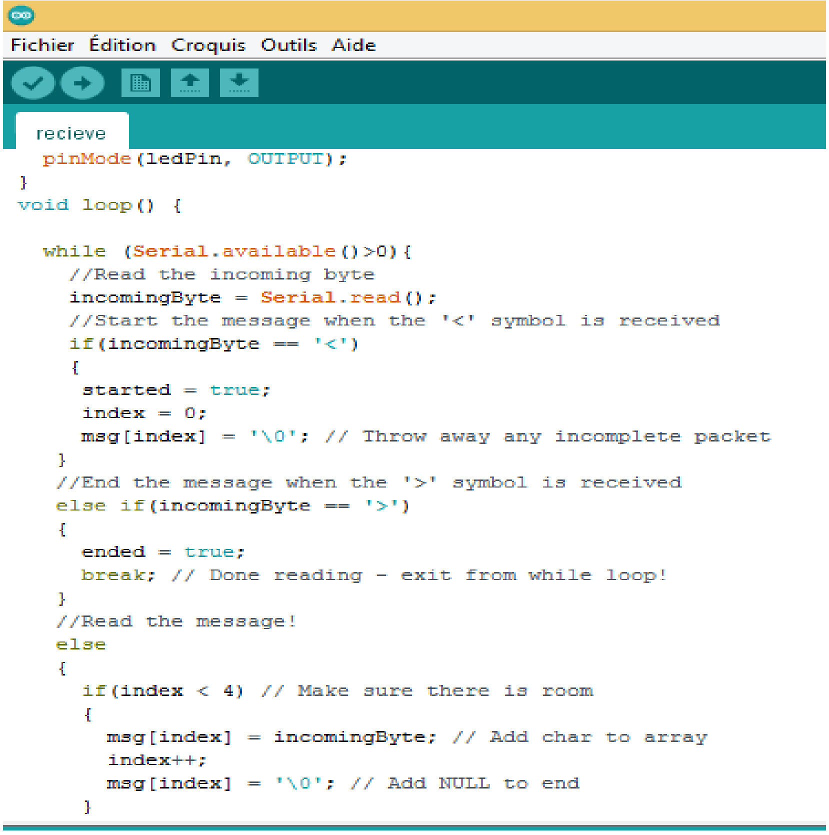
Figure III. 31 Code Arduino sender

- **Code Arduino recieve**



```
/*   ~ Simple Arduino - xBee Receiver sketch ~
  Read an analog value from potentiometer, then convert it to PWM and finally send it through serial port to xBee.
  The receiving message starts with '<' and closes with '>' symbol.
*/
//Constants
const int ledPin = 3; //Led to Arduino pin 3 (PWM)
//Variables
bool started= false;//True: Message is strated
bool ended  = false;//True: Message is finished
char incomingByte ; //Variable to store the incoming byte
char msg[3];    //Message - array from 0 to 2 (3 values - PWM - e.g. 240)
byte index;     //Index of array

void setup() {  //Start the serial communication
  Serial.begin(9600); //Baud rate must be the same as is on xBee module
  pinMode(ledPin, OUTPUT);
}
void loop() {

  while (Serial.available()>0){
    //Read the incoming byte
    incomingByte = Serial.read();
    //Start the message when the '<' symbol is received
    if(incomingByte == '<')
    {
     started = true;
     index = 0;
     msg[index] = '\0'; // Throw away any incomplete packet
   }
```

Figure III. 32 Code Arduino recieve



```
   pinMode(ledPin, OUTPUT);
 }
 void loop() {

   while (Serial.available()>0){
     //Read the incoming byte
     incomingByte = Serial.read();
     //Start the message when the '<' symbol is received
     if(incomingByte == '<')
     {
      started = true;
      index = 0;
      msg[index] = '\0'; // Throw away any incomplete packet
    }
    //End the message when the '>' symbol is received
    else if(incomingByte == '>')
    {
      ended = true;
      break; // Done reading - exit from while loop!
    }
    //Read the message!
    else
    {
      if(index < 4) // Make sure there is room
      {
        msg[index] = incomingByte; // Add char to array
        index++;
        msg[index] = '\0'; // Add NULL to end
      }
```

Figure III. 33 Code Arduino recieve

This Arduino code will read the analog value from the potentiometer, then convert it to PWM and finally send it through the serial port to XBee. The XBee serial module will send to another XBee (receiver) and the Arduino will turn on (fade) the LED.

III.7 Conclusion

In this chapter we explained the project's connection scheme to point-to-point connectivity and then control the electrical bales that were supposed to be in place but the lack of power in our conditions we changed the system where we relied on a point-to-point system first we created an environment Chat to send and receive data and then control the Arduino LED through the XBee connection as its first line to test whether the connection from sender to receiver is correct. This will be implemented by a wireless connection system via a point-to-point serial link and the sinteration of weaning between Arduino XBee Shield and XBee switch via XBees2C modules.

Also on the application side we used side using the Arduino Uno and two XBee USB adapter project looking to control the brightness of the LED wireless XBee communication.

Programming XBee units requires a lot of effort because of the lack of references to programming that we have been able to collect from both sides through search engines and the DIGI platform, the XBee manufacturer.

General Conclusion

General conclusion

This project allowed us to discover and strengthen our knowledge of telecommunications and microcontroller. To do this, we designed and built a network of XBee-based sensors for sending and receiving data, designed to supervision and detection of faults in electrical machines using a network of XBee-based sensors.

It was supposed to design and create a multi-point network based on XBee which was either a star or mech or cluster tree, but due to the lack of capabilities in the current situation of the country and the world, we were unable to provide all the equipment needed for the project and tried to change from multi-network disruption to one of its nodes to one node to complete my project.

This project is made up of two parts; the first part uses the X-CTU application to configure the XBee RF modules for the ZigBee network using 802.15.4 TH from (Part Number: XB24-CZ7-WIT) from www.digi.com.The second part involves the use of Arduino development board to implement the connecting Network using the XBees RF modules. Also we used the module XBee S2C with ZigBee protocols, which are based on the IEEE 802.15.4 standard, as the connection bridge for the Wireless Personal Area Network (WPAN). XBee S2Cs are programmed to communicate in AT mode with point-to-point (coordinator and End device).

In this project, we have achieved many of our goals, such as:

- Study and understand the system.
- Configuring XBee modules.
- Arduino programming.
- Create a network of XBee-based point-to-point sensor devices.
- Successful communication and exchange of information between the sender and receiver or between the facilitator and the final device.
- Remote control of the Arduino corded valve.

Also among the difficulties we encountered during this project are:

- The problem is the lack of equipment that was planned for the project to form a multi-node network.

- a problem with the configuration of XBee modules where programming of XBee modules requires a lot of effort because there are no references to the programming we have been able to collect from both sides through search engines and the DIGI platform, XBee manufacturer.

❖ **Perspectives:**

- However, it is likely to be better developed in the future to be certified to control several different modules such as LED lights, sensors, drives, etc. connected to Arduino through better XBee connectivity and better equipment.
- I also propose the adoption of a multi-contract communication system, which is more two nodes provide people with more than one way to communicate with each other. If one contract in this network is down, it is still possible to establish communication with the other nodes. It reduces the possibility of wireless communication outages and improves the reliability of this network.
- Also, AT mode can be changed to API mode, and the latter is more convenient for an interlaced network. In a single-transmit connection, if at mode is used, more power is consumed. In addition.

Bibliography

Bibliography

CHAPTER I

[1] Jindal, V. (2018). **History and architecture of wireless sensor networks for ubiquitous computing**. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 7(2), 214-217.

[2] https://shodhganga.inflibnet.ac.in/bitstream/10603/38341/9/11_chapter%202.pdf

[3] Gaura, E., Girod, L., Brusey, J., Allen, M., & Challen, G. (Eds.). (2010). **Wireless sensor networks: Deployments and design frameworks**. Springer Science & Business Media

[4] Andal, I., Shirley, J., & Vierra, H. (2013). Poly_Sense: Modular Wireless Sensor Network.

[5] Wang, B. (2010). **Coverage control in sensor networks**. Springer Science & Business Media.

[6] Charan, P., Paulus, R., Kumar, M., & Jaiswal, A. K. (2012). **A survey on the Performance Optimization in Wireless Sensor Networks using Cross Layer Approach**. Journal of Scientific and Research Publications, 2(5), 1-6

[7] Jaladi, A. R., Khithani, K., Pawar, P., Malvi, K., & Sahoo, G. (2017). **Environmental monitoring using wireless sensor networks (WSN) based on IOT**. Int. Res. J. Eng. Technol, 4(1), 1371-1378.

[8] Khan, O., Khan, F. G., Nazir, B., & Wazir, U. (2016). **Energy efficient routing protocols in wireless sensor networks: a survey**. International Journal of Computer Science and Information Security, 14(6), 398.

[9] Ramluckun, N., & Bassoo, V. (2020). **Energy-efficient chain-cluster based intelligent routing technique for Wireless Sensor Networks**. Applied Computing and Informatics.

[10] Aiswariya, S., Rani, V. J., & Suseela, S. **Challenges, Technologies and Components of Wireless Sensor Networks**.

[11] Ng, H. S., Sim, M. L., & Tan, C. M. (2006). **Security issues of wireless sensor networks in healthcare applications**. BT Technology Journal, 24(2), 138-144

[12] http://www.site-naheulbeuk.com/utbm/sr04/SR04_dossier_WSN.pdf

[13] Manel ELLEUCHI. (2012). **Improvements of the RiSeG Secure Group Communication Scheme for WSNs**. University in Sfax-Tunis

[14] Nakayama, H., Ansari, N., Jamalipour, A., & Kato, N. (2007). **Fault-resilient sensing in wireless sensor networks**. Computer Communications, 30(11-12), 2375-2384.

[15] Rajagopalan, R., & Varshney, P. K. (2006). **Data aggregation techniques in sensor networks: A survey.**

[16] Ari, A. A. A., Gueroui, A., Labraoui, N., & Yenke, B. O. (2015). **Concepts and evolution of research in the field of wireless sensor networks**. arXiv preprint arXiv:1502.03561.

[17] Shah, R. C., Roy, S., Jain, S., & Brunette, W. (2003). **Data mules: Modeling and analysis of three-tier architecture for sparse sensor networks**. Ad Hoc Networks, 1(2-3), 215-233.

[18] Chatzigiannakis, I., Kinalis, A., & Nikoletseas, S. (2006, October). **Sink mobility protocols for data collection in wireless sensor networks**. In Proceedings of the 4th ACM international workshop on Mobility management and wireless access (pp. 52-59).

[19] Zheng, J., & Jamalipour, A. (2009). **Wireless sensor networks: a networking perspective**. John Wiley & Sons.

[20] Al-Karaki, J. N., & Kamal, A. E. (2004). **Routing techniques in wireless sensor networks: a survey**. IEEE wireless communications, 11(6), 6-28.

[22] Royer, E. M., & Toh, C. K. (1999**). A review of current routing protocols for ad hoc mobile wireless networks**. IEEE personal communications, 6(2), 46-55.

[23] I. Stojmenovic ,(2005). **Handbook of Sensor Networks Algorithms and Architectures**. John Wiley & Sons, Inc., Hoboken, NJ,

[24] Ari, A. A. A., Gueroui, A., Labraoui, N., & Yenke, B. O. (2015). **Concepts and evolution of research in the field of wireless sensor networks**. arXiv preprint arXiv:1502.03561.

[25] Jiang, F., Frater, M., and Ling, S. S. (2011, June**). A distributed smart routing scheme for terrestrial sensor networks with hybrid Neural Rough Sets**. In Fuzzy Systems (FUZZ), IEEE International Conference on (pp. 2238-2244).

[26] Yu, X., Wu, P., Han, W., and Zhang, Z. (2014). **Overview of wireless underground sensor networks for agriculture**. African Journal of Biotechnology, 11(17), 3942-3948.

[27] Potdar, V., Sharif, A., and Chang, E. (2009, May**). Wireless sensor networks: A survey. In Advanced Information Networking and Applications Workshops**. International Conference on (pp. 636-641). IEEE.

[28] Misra, S., Reisslein, M., and Xue, G. (2008). **A survey of multimedia streaming in wireless sensor networks**. Communications Surveys & Tutorials, IEEE, 10(4), 18-39.

[29] Tagne-Fute, E. (2013). **Une approche de patrouille multi-agents pour la détection d'évènements** (Doctoral dissertation, Université de Technologie de Belfort-Montbeliard, France).

[30] Ganesh, S. (2017). **Efficient and Secure Routing Protocol for WSN-A Thesis**. arXiv preprint arXiv:1708.04500.

[31] info2myfriends.blog.com/.../ Security protocol in wireless sensor

[32] G. Padmavathi and Shanmugapriya : A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks :https://arxiv.org/ftp/arxiv/papers/0909/0909.0576.pdf

[33] https://shodhganga.inflibnet.ac.in/bitstream/10603/22912/7/07_chapter_01.pdf

CHAPTER II

[1] https://blog.domadoo.fr/guides/generalites-sur-le-zigbee/

[2] El-Bendary, M. A. (2015). Developing security tools of WSN and WBAN networks applications. Springer Japan.

[3] Ergen, S. C. (2004). ZigBee/IEEE 802.15. 4 Summary. UC Berkeley, September, 10, 17.

[4] Farahani, S. (2011). ZigBee wireless networks and transceivers. Newnes.

[5] Jen-Wei, C. (2006). Deve-lopment of a Novel Power Monitoring System based on DSP and ZigBee Technologies (Doctoral dissertation, Master Thesis, Chung Cheng Institute ofTechnology, National Defense University, Tao-Yuan, Taiwan, ROC).

[6] Lee, J. S., Chuang, C. C., & Shen, C. C. (2009, May). Applications of short-range wireless technologies to industrial automation: A ZigBee approach. In 2009 Fifth Advanced International Conference on Telecommunications (pp. 15-20). IEEE.

[7] ZigBee Specification 2007 by ZigBee Alliance
http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf

[8]https://connect.ed-diamond.com/MISC/MISC-086/Tout-tout-tout-vous-saurez-tout-sur-le-ZigBee

[9] Ramya, C. M., Shanmugaraj, M., & Prabakaran, R. (2011, April). Study on ZigBee technology. In 2011 3rd International Conference on Electronics Computer Technology (Vol. 6, pp. 297-301). IEEE

[10]https://www.digi.com/resources/documentation/Digidocs/90002002/Content/Reference/r_zb_stack.htm?TocPath=zigbee%20networks%7C_____3

[11] Abdelmalek, O. (2009). DESIGN AND IMPLEMENTATION OF SMART WIRELESS SENSOR NETWORK (Doctoral dissertation). **Université de setif. fftel-00487384ff**

[12] Javed, K. (2006). ZigBee suitability for wireless sensor networks in logistic telemetry applications. Halmstad University.

[13] Leung, S., Gomez, W., & Kim, J. J. (2009). ZigBee mesh network simulation using OPNET and study of routing selection. SFU ENSC, 427.

[14] ZigBee Alliance website. (2005). “Network Layer Technical Overview” Retrieval Date September, (http://www.zigbee.org/en/documents/zigbee-network-layer-technical-overview.pdf)

[15] William C. Craig. (2005). Zigbee: "Wireless Control That Simply Works», White paper, ZigBee Alliance < www.zigbee.org >.

[16] Wang, C., Jiang, T., & Zhang, Q. (Eds.). (2014). ZigBee® network protocols and applications. CRC Press.

[17] Meyer, R. (2012). Security issues and vulnerability assessment of Zigbee enabled home area network implementations. California State University, Sacramento

[18] Siemuri,A.(2019). Wireless Mesh Network Implementation (A Proof-of-Concept using ZigBee Wireless Mesh Network). University of vaasa

[19] Elahi, A, & Gschwender, A. (2009). ZigBee wireless sensor and control network. Pearson Education.

[20] Mihajlov, B, & Bogdanoski, M. (2011). Overview and analysis of the performances of ZigBee-based wireless sensor networks. International Journal of Computer Applications, 29(12), 28-35.

[21] Tobias Z. "ZigBee exploited: The good, the bad and the ugly", version 1.0, August 6th 2015

[22] Baronti, P., Pillai, P., Chook, V. W., Chessa, S., Gotta, A., & Hu, Y. F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards. Computer communications, 30(7), 1655-1695.

[23] https://www.elprocus.com/what-is-zigbee-technology-architecture-and-its-applications/

[24] https://www.arduino-tutorials.com/what-is-xbee/

[25] Ling, T. H. Y., Wong, L. J., Tan, J. E. H., & Lee, C. K. (2015, February). XBee Wireless Blood Pressure Monitoring System with Microsoft Visual Studio Computer Interfacing. In 2015 6th International Conference on Intelligent Systems, Modelling and Simulation (pp. 5-9). IEEE.

[26] Wettergren, A. (2007). ZigBee in Industry.

[27] Jérôme Abel. Jan 23, 2013: http://jeromeabel.net/files/ressources/xbee-arduino/xbee-arduino.xhtml

[28] bin Fadhlullah, S. Y. (2016). 2.4 GHz Xbee Pocket Manual.

[29] Kumbhar, H. (2016, August). Wireless sensor network using Xbee on Arduino Platform: An experimental study. In 2016 International Conference on Computing Communication Control and automation (ICCUBEA) (pp. 1-5). IEEE.

[30] Faludi, R. (2010). Building wireless sensor networks: with ZigBee, XBee, arduino, and processing. " O'Reilly Media, Inc.".

[31] https://www.electronicwings.com/sensors-modules/xbee-module

[32] HAJI MAHMOUD, K. (2013). Data collection and processing from distributed system of wireless sensors (Doctoral dissertation, Masarykova univerzita, Fakulta informatiky).

[33] Cassero, S. (2016). Design and Analysis of Arduino, Raspberry Pi, and Xbee Based Wireless Sensor Networks (Doctoral dissertation, UC Santa Barbara).

[34] http://www.libelium.com/development/waspmote/documentation/x-ctu-tutorial/

CHAPTER III

[1] Siemuri, A. (2019). Wireless Mesh Network Implementation (A Proof-of-Concept using ZigBee Wireless Mesh Network). University of Vaasa

[2] Rodríguez Pallares, J. E. (2015). Wireless sensor network implementation with Arduino and Xbee (Master's thesis, Universitat Politècnica de Catalunya).

[3] https://www.arduino.cc/en/guide/environment

[4] https://www.theengineeringprojects.com/2018/10/introduction-to-arduino-ide.html

[5] https://arduinogetstarted.com/tutorials/arduino-code-structure

[6] https://www.arduino.cc/en/guide/introduction

[7] http://www.science.smith.edu/~jcardell/Courses/EGR328/Readings/XCTU%20Guide.pdf

Annex

Code Arduino of control the LED wired

```
#include <SoftwareSerial.h>
SoftwareSerial myserial(7,8);
void setup() {

  Serial.begin(9600);

  pinMode(13, OUTPUT);

}

void loop() {

  byte data;

  data = Serial.read();

  if (data == 'a') {

    digitalWrite(13, HIGH);   // set the LED on

    //Serial.print("HIGH");

  }

  if (data == 'b') {

    digitalWrite(13, LOW);    // set the LED off

    //Serial.print("LOW");

  }
}
```

Code Arduino sender

```
/*   ~ Simple Arduino - xBee Transmitter sketch ~

  Read an analog value from potentiometer, then convert it to PWM
and finally send it through serial port to xBee.
  The xBee serial module will send it to another xBee (resiver) and
an Arduino will turn on (fade) an LED.
  The sending message starts with '<' and closes with '>' symbol.
*/

//Constants:
const int potPin = A0; //Pot at Arduino A0 pin
//Variables:
int value ; //Value from pot

void setup() {

  //Start the serial communication
  Serial.begin(9600); //Baud rate must be the same as is on xBee
module

}

void loop() {

    //Read the analog value from pot and store it to "value" variable
    value = analogRead(A0);
    //Map the analog value to pwm value
    value = map (value, 0, 1023, 0, 255);
  //Send the message:
  Serial.print('<');  //Starting symbol
  Serial.print(value);//Value from 0 to 255
  Serial.println('>');//Ending symbol



}
```

Code Arduino recieve

```
/*   ~ Simple Arduino - xBee Receiver sketch ~
  Read an analog value from potentiometer, then convert it to PWM and finally send it through serial port to xBee.
  The receiving message starts with '<' and closes with '>' symbol.
*/
//Constants
const int ledPin = 3; //Led to Arduino pin 3 (PWM)
//Variables
bool started= false;//True: Message is strated
bool ended  = false;//True: Message is finished
char incomingByte ; //Variable to store the incoming byte
char msg[3];    //Message - array from 0 to 2 (3 values - PWM - e.g. 240)
byte index;     //Index of array

void setup() {  //Start the serial communication
  Serial.begin(9600); //Baud rate must be the same as is on xBee module
  pinMode(ledPin, OUTPUT);
}
void loop() {

  while (Serial.available()>0){
    //Read the incoming byte
    incomingByte = Serial.read();
    //Start the message when the '<' symbol is received
    if(incomingByte == '<')
    {
     started = true;
     index = 0;
     msg[index] = '\0'; // Throw away any incomplete packet
   }
   //End the message when the '>' symbol is received
   else if(incomingByte == '>')
   {
     ended = true;
     break; // Done reading - exit from while loop!
   }
   //Read the message!
```

```
else
{
  if(index < 4) // Make sure there is room
  {
    msg[index] = incomingByte; // Add char to array
    index++;
    msg[index] = '\0'; // Add NULL to end
  }
```