الجمهـوريــة الجزائـــريــة الديـمقراطيـــة الشعبيــة

PEOPLAR DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION	وزارة التعليم العـالي
AND SCIENTIFIC RESEARCH	و البحـــث العلمـــــي
UNIVERSITY FERHAT ABBAS – SETIF	جامعـــة فرحـــات عبـــاس سطيـــف

Submitted to the department of electronic
In partial fulfilltiment of requirement for the degree of
“INGENIEUR D’ETAT EN ELECTRONIQUE”
OPTION: COMMUNICATION

DESIGN AND IMPLEMENTATION OF SMART WIRELESS SENSOR NETWORK

PRESENTED BY:	**PROPOSED AND SUPERVISED BY:**
Mr.: ABDELMALEK Omar	Dr.: H. Chemali

Promotion June 2009

الجمهـوريــة الجزائــريــة الديـمقراطيــة الشعبيــة

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L’ENSEIGNEMENT SUPERIEUR	وزارة التعليم العـالي
ET DE LA RECHERCHE SCIENTIFIQUE	و البحــث العلمـــي
UNIVERSITE FERHAT ABBAS – SETIF	جامعــة فرحــات عبــاس سطيــف
FACULTE DES SCIENCES DE L’INGENIEUR	كــــلـيـــــــة: علوم المهـنــــد س
DEPARTEMENT D’ELECTRONIQUE	قـــســـــم: الإلـكــتـــــرونــيــــك

MEMOIRE DE FIN D’ETUDES

EN VUE DE L’OBTENTION

DU DIPLOME D’INGENIEUR D’ETAT EN ELECTRONIQUE

OPTION : **COMMUNICATION**

Thème

CONCEPTION ET IMPLANTATION D’UN RESEAU DE CAPTEURS SANS-FIL « SMART WIRELESS SENSOR NETWORK »

Présenté par :	**Proposé et encadré par :**
Mr.: ABDELMALEK Omar	Dr.: H. Chemali

Promotion juin 2009

Dedication

"Success is based on the perseverance, the hard work and the patience."

Thomas Edison

To my mother, the most wonderful mother in the whole world.

To my brother.

.To all of my teachers in my entire career

To anyone who tries to make an effort for the best of my country: ALGERIA, and for the best of my religion: the Islam.

To anyone who believed in me.

I dedicate this humbled work

Thank’s

First of all, I am grateful to ALLAH, the almighty god, that he gave me the strength and the wisdom to make this project come true.

Then, I thank my supervisor, Dr. Chemali HAMIMI, for his assistance and the serious help that he gave me, and for the precious time that he dedicated with me along the duration of my project, and in addition, for his kindness.

*My thanks are addressed to (***Croissance et caractérisation de nouveaux semi-conductrices***) CCNSC laboratory which provided financial support and equipments and my best greetings are directed to Professor A. ZEGADI, head of this (CCNSC) laboratory for accepting to evaluate this work and mainly for coordinating the complex purchase operation of Zigbee Kits with Prof N. ABOUCHI (France).*

My greetings are also addressed to N.KERKAR for here interest to this work and accepting to evaluate it.

I express my best thanks to Prof N.ABOUCHI (CPE LYON France) for his help and encouragements.

Special thanks are directed to my friend M.Felouah who assisted me during this work

Great thanks and gratefulness are due to my family for its complete support.

Finally, I thank all my friends and any person who has helped me to achieve this humble work.

Abstract

The past several years have witnessed a rapid development in the wireless network area. So far wireless networking has been focused on high-speed and long range application. However, there are many wireless monitoring and control application for industrial and home environments which require longer battery life, lower data rates and less complexity than those from existing standard. What the market need is globally defined standard that meet reliability, security, low power and low cost.

For such wireless application a new standard called ZigBee has been developed by the ZigBee Alliance based upon The IEEE 802.15.4 standard.

The aim of this thesis diploma work is to design full functional nodes and to evaluate an application in wireless sensor network. The resulting designed product could form a reliable support for transferring data between nodes and a computer. MaxStream XBee ZNet 2.5 modules are chosen as the radio platform and LabVIEW as the virtual instrument for user interface.

The designed system is constituted of five battery powered functional Zigbee modules (temperature, pressure sensor modules…etc) structured to be wirelessly networked.

A system prototype is implemented using MaxStream XBee ZNet 2.5 modules, microcontroller Microchip PIC18LF452, LM35DZ temperature sensor and MPX2200A pressure sensor to build a network and test its main parameters.

KEYWORDS: WIRELESS SENSOR NETWORK, ZIGBEE, IEEE 802.15.4, Xbee, LM35, MX220A, LABVIEW.

ملخص

شهدت السنوات القليلة الماضية تطورا هائلا في مجال الشبكات اللاسلكية حيث ركزت على السرعة الفائقة و التطبيقات البعيدة المدى من ناحية أخرى،بينما هناك العديد من أجهزة الرصد و المراقبة اللاسلكية الموجهة للمجالات الصناعية و المنزلية ،التي تحتاج إلى بطاريات ذات عمر أطول ومعدا تبادل البيانات منخفض،وأقل تعقيدا ، شيئا فشيئا. إن حاجة السوق اليوم هي معيار عام يجمع بين الثقة و الأمان و انخفاض لتكلفة واستهلاك الطاقة.

كمثال على التطبيقات اللاسلكية و المعيار الجديد المسمى ZigBee الذي من طرفة تحالف ZigBee مرتكز على المعيارIEEE802.15.4

إن الهدف من هذا العمل هو تصميم عقد كاملة الوظائف وتقييم تطبيقاتها في شبكة الملتقطات اللاسلكية.هذا التصميم لهذا المنتج الجدي باستطاعته تكوين دعامة متباينة لنقل لبيانات والحاسوب.

وحدات MaxStream XBee ZNet 2.5 كدعامة للاسلكي بينما اخترنا البرنامج LABVIEW كوسيلة افتراضية لإنشاء واجهات للمستخدم.

إن النظام المصمم يحتوي 5 وحدات ZigBee مغذاة بالبطارية ، أنشئت بها شبكة لا سلكية لقياس الحرارة الضغط الجوي الرطوبة.........الخ.

كما استعملنا الحاسبات الدقيقة من عائلة MICROCHIP نوع PIC18452, مستشعر الحرارة LM35DZ، مستشعرات الضغط الجوي MPX2200A.

مفاتيح: MaxStream XBee ZNet 2.5، IEEE802.15.4، LM35DZ. MPX2200A, MICROCHIP ZigBee

Résumé

Au cours de ces dernières années, un développement rapide est signalé dans le domaine des réseaux sans fil. Jusqu'à lors, il y eu principalement focus sur des débits élevés et une grande portée des applications. Cependant, il y a beaucoup d’applications sans-fil dans le domaine du contrôle industriel et de la gestion environnementale qui exigent une longue vie des batteries, des débits réduits et beaucoup moins de complexité que ceux des normes et standards. La fiabilité,

la sécurité, la faible consommation et le faible coût sont les nouveaux paramètres globalement recherchés par le marché moderne.
Pour de telles applications sans fil, un nouveau standard ZigBee a été développé par l'Alliance ZigBee basé sur le standard IEEE 802.15.4.
L'objectif de cette thèse est de concevoir les fonctionnalités de nœuds de réseaux sans fil, et d'en évaluer une application dans le domaine réseaux de capteurs sans fil. Le produit résultant formera un support de transfert de données entre ces nœuds et un ordinateur.

Les modules MaxStream XBee ZNet 2.5 constituent les éléments de la plate-forme de la radio alors que LabVIEW est exploité comme instrument virtuel pour l'interface utilisateur.

Le système conçu est constitué de cinq modules ZigBee (module de température, de pression,…etc.) structurés en réseau sans fil.
Un prototype est finalement construit autour de modules MaxStream XBee ZNet 2.5, un microcontrôleur Microchip PIC18LF452, des capteurs de température LM35DZ, et des capteurs de pression MPX200A pour former un véritable réseau. La gestion et la configuration du réseau obtenu sont étudiées et ses principales caractéristiques sont testées.

Mots clés : Réseaux de capteur sans-fil, Zigbee, IEEE 802.15.4, Xbee, LM35, MPX220a, LabVIEW.

Table of contents

List of Table

Table of figures

Abreviation

AES Advanced Encryption standard

AODV Ad-hoc On-demand Distance Vector (routing protocol)

ASK - Amplitude Shift Key

BER - Bit Error Rate/Ratio

BPSK - Binary Phase Shift Keying

CCA -Clear Channel Assessment

CBC-M- Cipher Block Chaining - Message Authentication code

CSMA- Carrier Sense Multiple Access and Collision avoidance

DSSS - Direct Sequence Spread Spectrum

EDR - Enhanced Data Rate (Bluetooth technology)

EEP-Extensible Authentication Protocol

EVM- Error Vector Magnitude

FFD –Full function device

FH Frequency Hopping

FHSS - Frequency Hopping Spread spectrum

GPIO - General Purpose Input/output

HID - Human Interface Device

HDR-HEADER

HR/DS- High Rate Direct Sequence Spread Spectrum

IEEE - Institute of Electrical and Electronics Engineers, Inc

GTS- Guaranteed Time Slot

ISM - Industrial, scientific medical (usually refers to RF band)

IT- Information technology

LFBG - Low Profile Fine Pitch Ball Grid Array (packaging type)

LLC - Logical Link Control

LQI Link Quality Indicator

LR-WPAN Low-Rate Wireless Personal Area Network

MAC - Medium Access Control

MB-OFDM- Multi-band Orthogonal Frequency-Division Multiplexing

MCPS - Medium access control Common Part Sub-layer

MIMO - Multiple In Multiple Out

MLME - Medium access control sub- layer Management Entity

MSDU - Medium Access Control sub-layer Service Data Unit

NLDE - Network Layer Data Entity

NLME - Network Layer Management Entity

OEM original Equipment Manufacturer

OFDM - Orthogonal Frequency frequency-Division Multiplexing

OSI Open Systems Interconnection

PER Packet Error Rate

PA –Power amplifier

PD - Physical layer Data

PER - Packet Error Rate/Ratio

PHY - Physical Layer

PLME - Physical Layer Management Entity

PSSS - Parallel Sequence Spread Spectrum

QAM - Quadrature Amplitude modulation

QPSK - Quadrature Phase Shift keying

QOS Quality of Service

RoC - Radio-on-Chip

RX Receiver

RFD- reduced Function Device

SAP - Service Access Point

SFD Start-of-Frame Delimiter

SHR Synchronization Header

SiP - System-in-Package

SMA-sub-miniature version A connector(coaxial RF connector)

SoC - System-on-Chip

SIG - Special Interest Group

SSCS - Service-Specific Convergence Sub-layer

TDD - Time Division Duplex

TX Transmitter

TFBGA - Thin Fine Pitch Ball Grid Array (packaging type)

USB - Universal Serial Bus connector which is flat

USB-A - A version of USB

USB-B - A version of USB connector which is square

UWB - Ultra Wide Band

WLCSP - Wafer Level Chip Scale Package (packaging type)

WEP-Wireless Encryption Protocol

WLAN Wireless Local Area Network

ZDO ZigBee Device Object

Introduction and background

A sensor network is a network of many smart devices, called nodes, which are spatially distributed in order to perform an application-oriented global task. The primary component of the *network* is the sensor, essential for monitoring real world physical conditions or variables such as temperature, humidity, presence (absence), sound, intensity, vibration, pressure, motion, and pollutants, among others, at different locations.

Each smart device within the network is small and inexpensive, so that it can be manufactured and deployed in large quantities. The important design and implementation requirements of a typical sensor network are energy efficiency, memory capacity, computational speed and bandwidth.

The smart device has a microcontroller, a radio transmitter, and an energy source. Sometimes a central computer is integrated onto the network in order to manage the entire networked system.

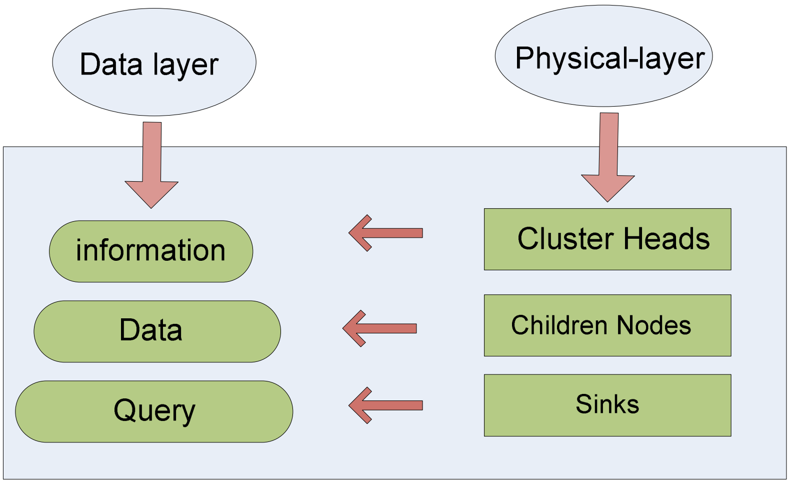
Regardless of achieving its global task, a sensor network essentially performs three basic functions: sensing, communicating and computation by using the three fundamental components: hardware, software and algorithms, respectively.

Conventionally, a sensor network is considered a wireless network, however, some sensor networks are wired or hybrid types.

A wireless sensor network (WSN), as its name implies, consists of a number of microcontroller-integrated smart devices. Each node is equipped with a variety of sensors, such as acoustic, seismic, infrared, still/motion video-camera, and so on. The node has some degree of intelligence for signal processing and management of network data. The basic goals of a WSN are to:

- determine the value of physical variables at a given location,
- detect the occurrence of events of interest,
- estimate parameters of the detected event or events,
- classify a detected object, and
- track an object

Thus the important requirements of a WSN are: use of a large number of sensors, attachment of stationary sensors, low energy consumption self-organization capability, collaborative signal processing, and querying ability.



Wireless sensor architecture

Figure 1.1 shows the general architecture of a sensor network. As can be seen in the figure, the three important layers are the services-layer, data-layer, and physical- layer. The layers provide routing protocol, data dissemination, and aggregation.

The physical-layer containing the node defines itself as either a sink node, children node, cluster head, or parent node. Parent nodes are integrated to more than two cluster heads. Messages are modeled in the data-link layer. Broadcasting of a query is carried out by the use of sink nodes. The broadcasting can be either to the sensor network or to a designated region depending on the way the query is being used. In response to a change in the physical parameter the sensor nodes, which are close to the sensed object, broadcast this information to their neighboring sensor nodes. In effect, a cluster head will receive this transmission. The role of a cluster head is to process and aggregate this data and broadcast it to the sink node(s) through the neighboring nodes. This is due to the fact that cluster head receives many data packets from its children.

The aim of this work is to design full functional nodes and to evaluate an application in wireless sensor network. The resulting designed product could form a reliable support for transferring data between nodes and a computer. MaxStream XBee ZNet 2.5 modules are chosen as the radio platform and LabVIEW as the virtual instrument for user interface.

The designed system is constituted of five battery powered functional Zigbee modules (temperature, pressure sensor modules…etc) structured to be wirelessly networked.

A system prototype is implemented using MaxStream XBee ZNet 2.5 modules, microcontroller Microchip PIC18LF452, LM35DZ temperature sensor and MPX2200A pressure sensor to build a network and test its main parameters.

The thesis is organized into five chapters. After an introduction to wireless sensor networks technology and application in chapter 2, a brief comparison of wireless technologies is provided the third chapter. The fourth chapter is devoted to Zigbee specification. A brief comparison of ZigBee hardware is provided the fifth chapter. The different steps of our designed system are explained in the sixth chapter. A conclusion terminates the thesis, presents the obtained results and proposes future improvements and perspectives.

Chapter 1

Introduction to wireless sensor network and application

Contents:

This chapter introduces the topic of wireless sensor networks from the applications perspective. A wireless sensor network consists of a possibly large number of wireless devices able to take environmental measurements such as temperature, light, sound, and humidity. These sensor readings are transmitted over a wireless channel to a running application that makes decisions based on these sensor readings.

1.1 Introduction

A wireless sensor network (WSN) is an infrastructure consisting of several sensing, computing, and communication elements wirelessly connected, that provides the ability to observe and react to events and phenomena in a specific environment, such as environmental monitoring, health applications, home automation, inventory control, vehicle tracking and detection.

Sensor nodes or wireless nodes which are also called motes are connected via series of multi-hop or single-hop short distance and low-power wireless link.

Implementations of WSNs have to address a set of technical challenges. One of the current research challenges is to develop low-power and low-cost wireless nodes. Low power consumption is a key factor in ensuring long operation time for battery powered nodes. Today’s commercially available radio transceivers consume typically several tens of milliwatts. To maintain the required power consumption, the nodes must sleep most of the time.

This can be realized on using low duty cycle operation such as a 1% or 0.1% duty cycle. In addition to low duty cycle operations, utilizing a power efficient transceiver can also increase the power efficiency.

An example of a wireless sensor network using RF links is shown in Figure 1-2, where the sensor nodes gather data autonomously.

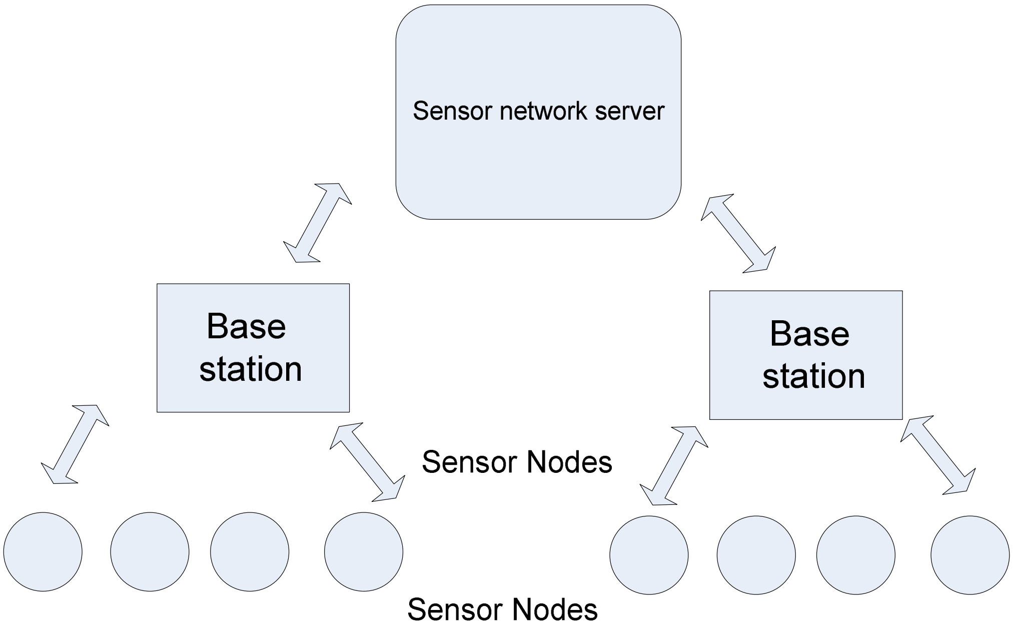


Figure 1.1: Sensor network architecture

The basic goals of a WSN are to:

- Determine the value of physical variables at a given location.
- Detect the occurrence of events of interest, and estimate parameters of the detected event or events.
- Classify a detected object and Track an object.

And the important requirements of a WSN are:

- Use of a large number of sensors.
- Attachment of stationary sensors.
- Low energy consumption.
- Self-organization capability.
- Collaborative signal processing.
- Querying ability

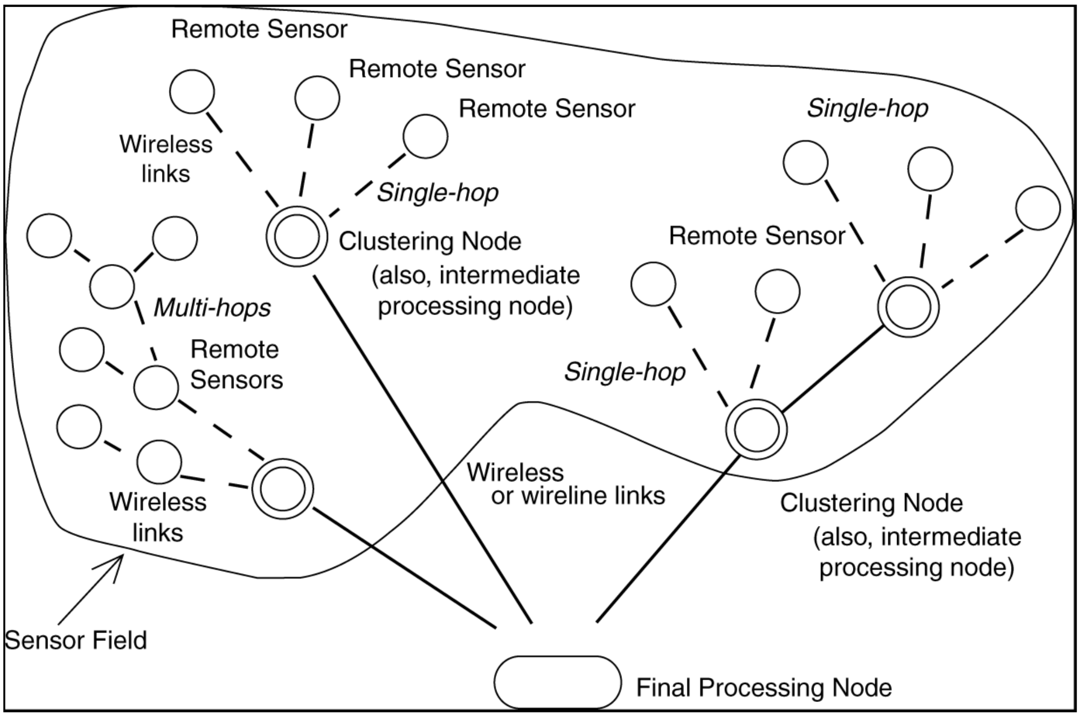


Figure 1.2: Typical sensor network arrangements

1. 2 Applications

There can be thousands of applications for WSNs. This chapter deals with the fundamental aspects of sensor networks. To justify the applicability of sensor network technology, we have included some real-field case studies. They are listed below:

1.2.1 General Engineering

a) Automotive telematics

Cars, which comprise a network of dozens of sensors and actuators, are networked into a system of systems to improve the safety and efficiency of traffic.

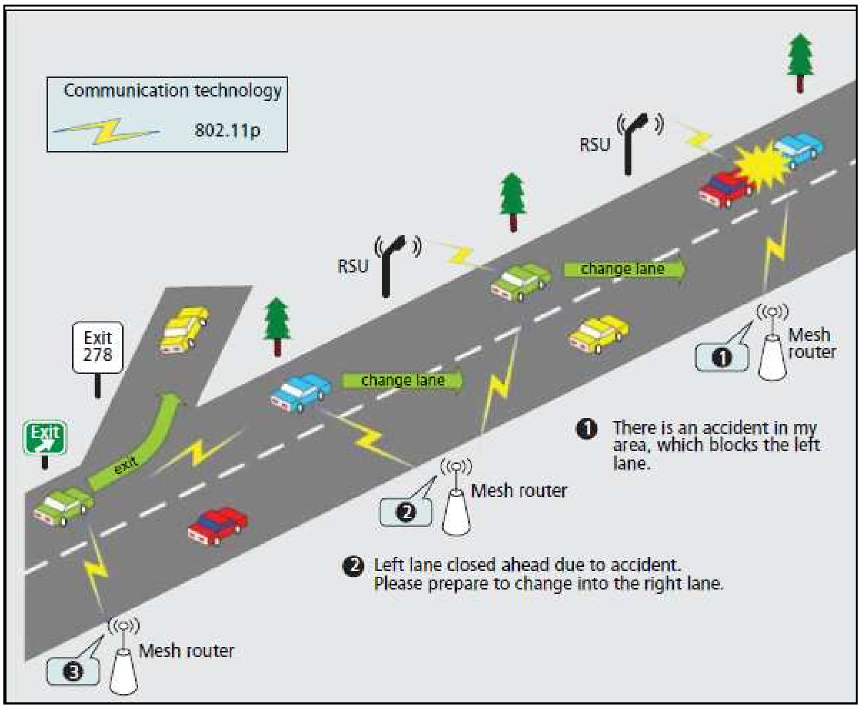

Communication technology
802.11p
RSU
RSU
change lane
change lane
Exit 278
Exit
exit
Mesh router
Mesh router
Mesh router
1
2
3
1 There is an accident in my area, which blocks the left lane.
2 Left lane closed ahead due to accident. Please prepare to change into the right lane.

Figure1.3: Wsn for traffic monitoring

b) Sensing and maintenance in industrial plants

Complex industrial robots are equipped with up to 200 sensors that are usually connected by cables to a main computer. Because cables are expensive and subject to wear and tear caused by the robot’s movement, companies are replacing them by wireless connections.

By mounting small coils on the sensor nodes, the principle of induction is exploited to solve the power supply problem.

c) Aircraft drag reduction

Engineers can achieve this by combining flow sensors and blowing/sucking actuators mounted on the wings of an airplane.

d) Smart office spaces

Areas are equipped with light, temperature, and movement sensors, microphones for voice activation, and pressure sensors in chairs. Air flow and temperature can be regulated locally for one room rather than centrally.

e) Social studies

Equipping human beings with sensor nodes permits interesting studies of human interaction and social behavior.

1.2.2 Agriculture and Environmental Monitoring

a) Precision agriculture

Crop and livestock management and precise control of fertilizer concentrations are possible.

b) Planetary exploration

Exploration and surveillance in inhospitable environments such as remote geographic regions or toxic locations can take place.

c) Geophysical monitoring

Seismic activity can be detected at a much finer scale using a network of sensors equipped with accelerometers.

d) Monitoring of freshwater quality

The field of hydrochemistry has a compelling need for sensor networks because of the complex spatiotemporal variability in hydrologic, chemical, and ecological parameters and the difficulty of labor-intensive sampling, particularly in remote locations or under adverse conditions. In addition, buoys along the coast could alert surfers, swimmers, and fishermen to dangerous levels of bacteria.

e) Disaster detection

Forest fire and floods can be detected early and causes can be localized precisely by densely deployed sensor networks.

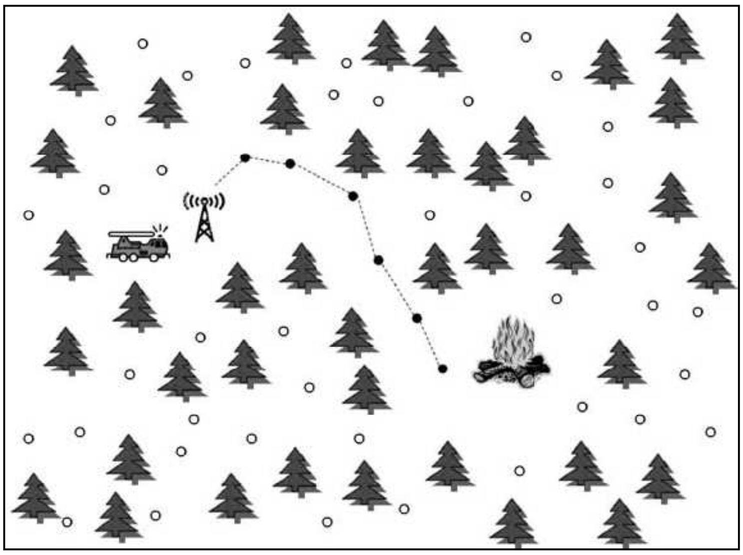


Figure 1.4: Forest-fire monitoring application

1.2.3 Civil Engineering

a) Monitoring of structures

Sensors will be placed in bridges to detect and warn of structural weakness and in water reservoirs to spot hazardous materials. The reaction of tall buildings to wind and earthquakes can be studied and material fatigue can be monitored closely.

b) Urban planning

Urban planners will track groundwater patterns and how much carbon dioxide cities are expelling, enabling them to make better land-use decisions.

c) Disaster recovery

Buildings razed by an earthquake may be infiltrated with sensor robots to locate signs of life.

1.2.4 Military Applications

a) Asset monitoring and management

Commanders can monitor the status and locations of troops, weapons, and supplies to improve military command, control, communications, and computing.

b) Surveillance and battle-space monitoring

Vibration and magnetic sensors can report vehicle and personnel movement, permitting close surveillance of opposing forces.

c) Urban warfare

Sensors are deployed in buildings that have been cleared to prevent reoccupation; movements of friend and foe are displayed in PDA-like devices carried by soldiers. Snipers can be localized by the collaborative effort of multiple acoustic sensors.

d) Protection

Sensitive objects such as atomic plants, bridges, retaining walls, oil and gas pipelines, communication towers, ammunition depots, and military headquarters can be protected by intelligent sensor fields able to discriminate between different classes of intruders. Biological and chemical attacks can be detected early or even prevented by a sensor network acting as a warning system.

e) Self-healing minefields

The self-healing minefield system is designed to achieve an increased resistance to dismount and mounted breaching by adding a novel dimension to the minefield. Instead of a static complex obstacle, the self-healing minefield is an intelligent, dynamic obstacle that senses relative positions and responds to an enemy’s breaching attempt by physical reorganization.

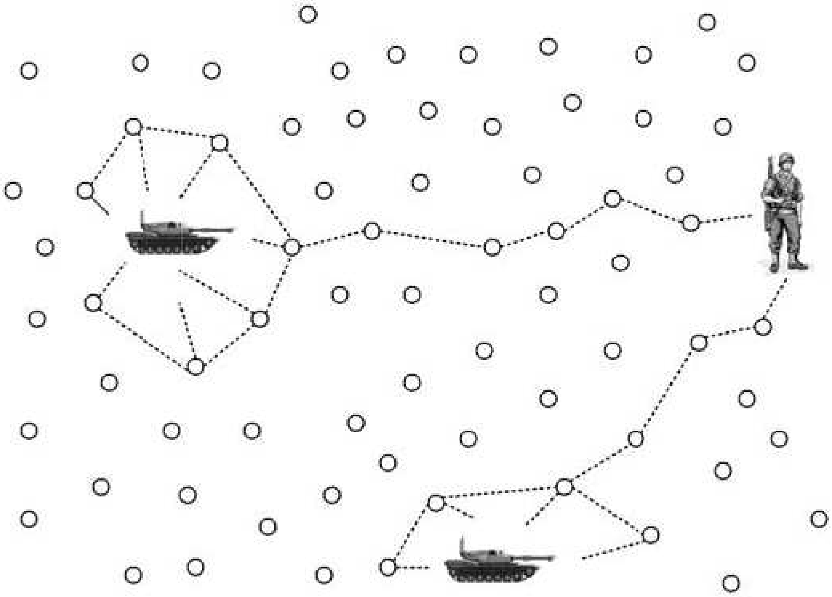


Figure 1.5: Enemy target localization and monitoring

1.2.5 Health Monitoring and Surgery

a) Medical sensing

Physiological data such as body temperature, blood pressure, and pulse are sensed and automatically transmitted to a computer or physician, where it can be used for health status monitoring and medical exploration. Wireless sensing bandages may warn of infection. Tiny sensors in the blood stream, possibly powered by a weak external electromagnetic field, can continuously analyze the blood and prevent coagulation and thrombosis.

b) Microsurgery

A swarm of MEMS-based robots may collaborate to perform microscopic and minimally invasive surgery.

The opportunities for wireless sensor networks are ubiquitous. However, a number of formidable challenges must be solved before these exciting applications may become reality

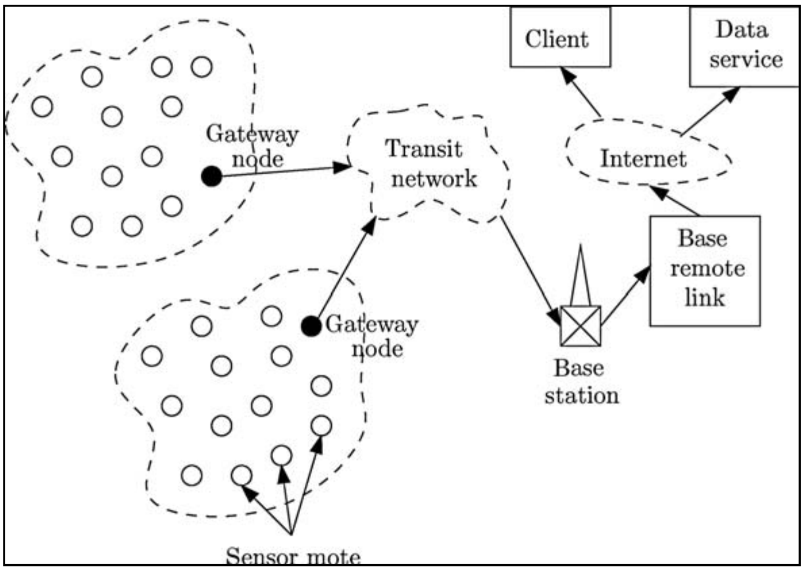


Figure 1.6 System architecture for habitat monitoring

1.3 Conclusion

In this chapter we provided a sample of possible WSN applications, and we should be able to envision dozens of other potential applications, as they appear to be almost unlimited. Basically, wherever one wants to instrument, observe, and react to events and phenomena in a specified environment, one can use WSNs; the environment can be the physical world, a biological system, or an IT framework.

Chapter 2

Wireless Technologies Comparaison

Contents:

My main focus in this chapter is to decide which of nowadays offered standards technologies is the most suitable for a large application using measuring devices. In brief, the concerns points are:

- Hardware dimensions
  - Software / hardware demands for embedded usage (Memory requirements)
  - Data rate and other advantageous in network features (‘Bandwidth and range’)
- Power consumption
- Reliability and security
- Costs

To be precise, for measuring in small devices the data rate demand is an insignificant parameter compared to e.g. the reliability or power consumption, (which, in most cases, are in contradiction with high data rates). This is mainly because nowadays, commonly considered data rates are usually in the scope of megabytes or more. For devices much lower demands are implied. In other words, higher data rate would be beneficial for the final product, but it should enhance other parameters bandwidth it can be to some extent sacrificed.

Three of the remaining technologies were chosen to be further discussed. They are all well-know widely commercially exploited LAN/PAN standards with industry-driven development and support, manufactured in large-scale and equipped in wide range of applications. They are:

- **Bluetooth**
- **Wi-Fi**
- **ZigBee**

2.1 Bluetooth

Bluetooth is a wireless standard that belongs to the PAN (respectively WPAN) protocol family. It operates in the 2.4GHz band divided into 79 sub-channels with 1MHz spacing, employing FHSS. GFSK and/or PSK modulations are used, depending on the Bluetooth version used. Full duplex transfers are realized via TDD. It is defined by the IEEE 802.15.1 [10] standard and extended by the Bluetooth Special Interest Group.

Figure 2.1: Official Bluetooth logo

First IEEE standard of Task Group 802.15.1 was released in 2002, based on Bluetooth 1.1. Last version of Bluetooth (as of June 2008) is 2.1, announced in mid-2007 by Bluetooth SIG, but has not yet been approved by IEEE standard. Nowadays, most devices use Bluetooth 2.0/ 2.1 with EDR, which offers higher transfer rates (described in chapter 2.1.3).

When compared to Wi-Fi, a single Bluetooth device occupies the entire 83.5MHz-wide band, but uses only a single channel at any instant moment (a result of FHSS). A Wi-Fi device, on the other hand, takes up only a single 22MHz-wide channel at any instant moment (a result of DSSS).

The well-known feature of Bluetooth is the services. Services are the result of Bluetooth profiles that define an interface for transferring specific type of user data via Bluetooth. The profiles are e.g. the “serial port profile”, “LAN access profile”, “Dial-up networking profile”, hands-free profile’ and tens of others. Standard profiles are defined by the Bluetooth SIG and available at [12].

The primary purpose of Bluetooth is a low cost, low power and robust replacement of short range cable connections in both portable and fixed devices. This means that it is primarily not intended to create large networks but only for temporary peer to peer links and spontaneous data transfers.

2.1.1 Hardware dimensions

The smallest Bluetooth devices currently available on the market are utilizing the v2.1 EDR chips manufactured by CSR in the Blue Core series [13]. The dimensions of these chips in the WLLCSP are no larger than 4x4mm, thus they can be easily integrated e.g. into SB-A type connectors.

2.1.2 Memory requirements

A disadvantage of Bluetooth usage in small devices is the fact that a full Bluetooth stack is too large for embedded applications. Because of that, most simple applications only include a fraction of the stack in their firmware. Full stack is then implemented in a device with more computational power (and usually acts as a host), e.g. PC or PDA.

2.1.3 Bandwidth and range

Both bandwidth and ranges have been extended along with the popularization of Bluetooth. The latest IEEE 802.15.1-2005 specification defines Bluetooth 1.2. Bluetooth 2.0/2.1 is developed under Bluetooth SIG only.

The bandwidth limits and maximum distances are described in the following table.

Bluetooth version	Data rate	Bluetooth class	Range
1.0	500 Kbit/s	Class 3 (1 mW)	1m
1.2	720 Kbit/s	Class 2 (2.5 mW)	10m
2	1Mbit/s	Class 1 (100 mW)	100m
2.0+EDR	3Mbit/s		

List of **Table2.1: Bluetooth versions and classes**

The EDR technology allows higher speeds by using PSK modulation for parts of the transmission instead of GPSK and also uses different packet structure.

2.1.4 Power consumption

Power consumption has become an important concern of end device manufacturers recently. Bluetooth SIG responded by enhancing the feature set of new Bluetooth versions.

The power consumption can be primarily lowered by adjusting network scanning and transmission parameters, which, however, sacrifices bandwidth and prolongs the link initialization.

The EDs (except from higher data rate) three new measures to enhance power consumption

- **Sniff mode:**

In this mode a slave in a piconet is allowed to listen only in certain timeslots, lowering the Radio power when not listening.

- **Hold mode**:

Salve device can use hold mode to tell the master it is not listening for asynchronous data. Device may enter low-power mode during the hold mode.

- **Parked state:**

Devices that wish to remain synchronized with the piconet but do not want to transfer any data for a certain time period can enter park state. In this mode, they receive a special pair of addresses and only wake up at defined intervals.

An average Bluetooth chip usually drains about 50mW for both receive and transmit modes. In sleep mode (or equivalent mode, e.g. parked state), some of the chips (e.g. CRS's BlueCore) can lower the consumption below 1mW.

2.1.5 Network features

In basic mode, Bluetooth creates a temporary device link in a process called "pairing" a process of establishing point-to-point ad-hoc connection between two devices.

So called "piconet" is a network created on a temporary basis (although usually the intention is to create a permanent network) and is an extension to the point-to-point topology. It forms a 'star' master-slave topology.

To create a piconet, in the process of device pairing one device has to be elected as the network master, while other devices join the network as slaves. Master defines the physical layer parameters of the network.

The maximum number of active devices in piconet is limited by the structure of Bluetooth's MAC layer to 7 slaves and one master. "Scatternet" is a tree topology network super-structure to piconet. A scatter net can be formed when the master device of a piconet joins another piconet as a slave. This way the overage of the final network can be extended beyond the standard 10m/100m Bluetooth range. However, not all of the chips on market support this feature.

For an illustration of how Scatternet/piconet can be formed, see figure 2.2.

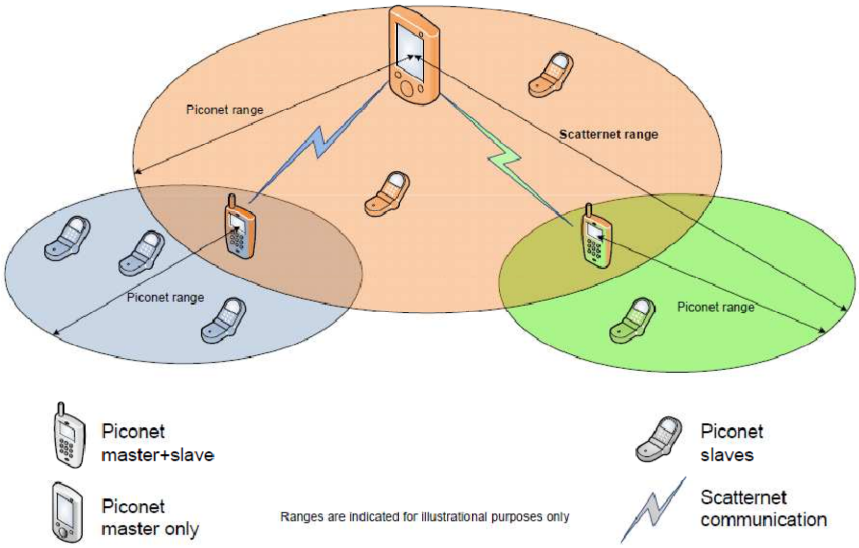


Figure 2.2: Bluetooth's piconet and Scatternet

2.1.6 Security and reliability

Bluetooth offers an authentication method using 128 bit key.

Bluetooth is more resistant to eavesdropping than other technologies mentioned in this project, because in difference to e.g. Wi-Fi or ZigBee, the modulation it uses on the physical layer is FHSS and not DSSS, which means you need to follow the frequency hopping in order to continue receiving data.

Reliability is a rarely discussed attribute of Bluetooth. The technology used on physical layer makes Bluetooth slightly more likely to suffer from interferences and intentional jamming. The Scatternet is also not a topology that can guarantee reliable multi-hop data transfers. Moreover, the spontaneous nature of all connections is not an attribute that benefits the reliability.

2.1.7 Costs

There are several manufacturers of Bluetooth chips available on the market. Broadcom offers the BCM2046 SoC, however, at unknown price and probably only for wholesale.[14]

Texas-instrument's portfolio contains a number of Bluetooth chips (e.g. the BlueLink series)

All of them are available to 'high-volume wireless OEMs.

The only manufacturer of which I have found retail prices is CSR. They also offer more information about their solutions than other manufacturers. Their BlueCore4 SoC and BlueCore5 SoC (Bluetooth transceivers without ROM/Flash).

2.1.8 Future development

The next generation of Bluetooth is developed under the supervision of Bluetooth Special Interest Group and WiMedia Alliance. The aim of devices in version 3.0 is enhancing the peed by widening the frequency spectrum used. The speeds mentioned are 480 Mbit/s for short distances (up to one meter) and 100mbit/s at the range of 10m. The disadvantage of this approach is that the frequencies used to gain speed are not in the European Public Spectrum Range, thus can’t be use in Europe.

2.2 Wi-Fi

Wi-Fi is probably the most exploited wireless technology nowadays. It belongs to the family of (W) LAN networks, but with latest amendments it could also be belonging to the (W) MAN family. It is built on the IEEE 802.11 standard, which first version was announced in 1997 and first successful commercial standard (the 802.11b and 802.11a) was adopted in 1999.

Figure 2.3: Official Wi-Fi logo

The physical layer defines the operation frequency to 2.4GHz (802.11b/g/n) and 5GHz (802.11a/n) employing DSSS (802.11b) or OFDM (for higher speeds in 802.11a/g). In difference to Bluetooth, the Wi-Fi spectrum is divided into only 13 partly overlaying sub channels (14th available in Japan only), each occupying the band of 22MHz. At an instant moment, the 802.11a/b/g versions are always occupying only a single channel. (See appendices for detailed demonstration of Wi-Fi channels spectrum).

Channel modulations used in Wi-Fi are CCK for 802.11b and several variants of QAM for 802.11a/g.

802.11n (not yet released as standard, referred as ‘draft’) is using the MIMO technology (multi-way signal propagation phenomenon) for enhanced data rate, passing 100 Mbit/ transfer rate.

Despite of the majority in DSSS usage in contemporary Wi-Fi applications, 802.11 defines the physical layer as any of

- FHSS
- DSSS
- INFRARED
- OFDM
- HR/DSSS

2.2.1 Hardware dimensions

The average size of a Wi-Fi SoC (usually means Wi-Fi + Radio-MCU + Caches) is about 8x8mm. Most of the modern chips that are being introduced now are even smaller, some of them, mostly coming in the TFBGA packaging, have the size of 5x3 mm.

The disadvantage of Wi-Fi is its requirement of external memories and processors, which in the end means that general Wi-Fi device is much larger (scope of centimeters) than all of the other devices mentioned in this thesis.

2.2.2 Memory requirements

Wi-Fi is, compared to Bluetooth, and ZigBee, the most memory and computing-power demanding. Modern chips contain 400MHz RISC processors with 64KB-128KB caches, and are using external RAM and ROM in the sizes of MB.

Common Wi-Fi USB dongles do not contain the Wi-Fi stack in their firmware.

2.2.3 Bandwidth and range

The theoretical maximum for Wi-Fi range is in the units of kilometers. There are working installations on the distance of above 2 km in the 5GHz band and above 1 km using a standard hardware.

Standard ranges achieved by stock antennas and standard output power are 100 m, for outdoor (line-of-sight) range, and about 10 m for indoor use.

The maximum bandwidths are shown in the table below:

Standard	**Band**	**Data rate**
802.11a	5 GHz	54 Mbit/s
802.11b	2,4 GHz	11 Mbit/s
802.11g	2,4 GHz	54 Mbit/s
802.11n (Draft 4)	2,4 / 5 GHz	Up to 540 Mbit/s

Table 2.2: Wi-Fi standards, bands and data rates

2.2.4 Power consumption

The main purpose of Wi-Fi is to deliver enhanced data rate. The power consumption issue is in this technology not significant, and there is no main intention in the development to try to lower it.

There were some attempts to create low power Wi-Fi devices, but the results are not as persuasive as e.g. Bluetooth or ZigBee can be. Broadcom has developed a chip that drains 270 mW in full speed, and calls it low power. Gainspan has developed another chip, GS1010,

This is truly SoC, with radio, MCU, RAM and Flash integrated in single chip, thus lowering it.

The final power needed. They claim that single AA battery will last years in their product. [16]

In numbers, an average full speed power consumption of an 802.11g device ranges from 400mW up to 1W.

2.2.5 Network features

Wi-Fi defines two types of networks – ad-hoc and infrastructure:

In **ad-hoc network**, there is no master device, all devices have equal roles and all of the connections made are peer-to-peer. These are mostly used for temporary purposes.

Infrastructure is a network with one (or more) master devices. These devices define the parameters of the network. Devices that join the masters and are not providing connection to the network to other devices are called slaves.

The topologies possible to form using Wi-Fi are star or tree (using multiple AP). There are also proprietary mesh network applications being developed, with the aim on metropolitan networks.

2.2.6 Security and reliability

Wi-Fi is well known for its former weak WEP encryption. In last few years, as its popularity grew up, new security algorithms were applied in Wi-Fi. Nowadays, the security standard is defined by 802.11i. WPA2 supplies most features defined by that standard (using AES for enciphering) and together with authentication protocols (various forms of Extensible Authentication Protocol (EAP)) it is at least for now considered secure.

2.2.7 Costs

It is difficult to obtain prices for single chips for Wi-Fi, as most of the SoCs / RoCs are currently under development, and the others are not for sale separately and wholesale prices are kept as confidential.

2.2.8 Future development

The future potential of Wi-Fi lies in the enhancement of speed beyond the speed of Ethernet and in improving the network topology (in order to create and maintain easily accessible metropolitan networks) and reliability.

The power consumption is in concern of only few companies nowadays, and it does not seem likely that it will become a main trend.

2.3 ZigBee

ZigBee is an extension to the IEEE 802.15.4 (low-rate (W) PAN) standard. It is focused on embedded platforms-low power consumption and very low complexity is the main concerns, as well as security and jamming resistivity

The ZigBee Alliance is an association of companies working on a standard, which would enable low-power, cost-effective, reliable, wireless communication. Its members are for example Freescale, Motorola, Texas Instruments, Honeywell, Samsung, Philips, Siemens and over hundred others. First release of the ZigBee specification was in the beginning of 2005 and the latest was released in the end of 2007[17].

Detailed features of all ZigBee layers features are described in chapter 4.

Figure 2.4: Official ZigBee logo

Based on firmware and capabilities, there are two types of devices that can participate in an 802.15.4 network. First one is a fully-functional device (FFD), which serves as coordinator, router or end device. Second one is a reduced-function (RFD) device, which can act as an end device only. End devices can only communicate with routers or coordinator, but not to each other. Routers and coordinator can communicate with all network members.

There are two types of addresses in ZigBee – the 64 bit (long) 802.15.4 address and the 16 bit (short) network layer address. The 64 bit address is similar to MAC address in e.g. Ethernet or Wi-Fi networks – it is the address that is defined by manufacturer and is unique for each device. The 16 bit address is comparable to an IP address in IP networks. It represents a temporary address of a device in the current network. It is assigned by the coordinator and is unique for the current network. If a device does not have its 16 bit address, it can be still addressed by the 64 bit address if it is associated to the network.

The relation between IEEE 802.15.4 and the ZigBee is analogical to the relation of IEEE 802.11 and the Wi-Fi, but for example the standard created by ZigBee Alliance adds more layers to 802.15.4 than Wi-Fi to 802.11.

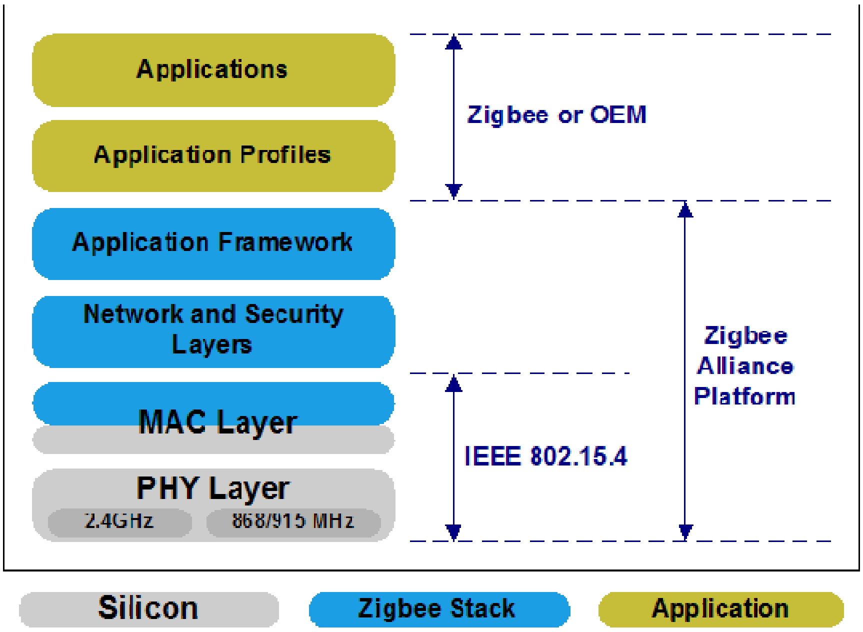


Figure 2.5: ZigBee Networking Protocol Layers

2.3.1 Hardware dimension

The dimensions of both ZigBee SoCs and ZigBee transceivers are approximately the same, ranging from 4x4 mm to about 7x7 mm mostly in QFN packages.

(Detailed description of ZigBee hardware is given in chapter 4)

Additionally, because of a low number of external components, the final product can reach similar sizes to Bluetooth, as the only larger external part required is an antenna.

2.3.2 Memory Requirements

When compared to other technologies, memory requirements of ZigBee are the least demanding. The total required memory is based on the particular ZigBee stack you want to use. But, in global terms, 100kB of ROM and units of kilobytes of RAM should be sufficient for most of the stacks. Also the computational power is reasonable – usually RISC MCUs at frequency below 30 MHz are used.

2.3.3 Bandwidth and Range

ZigBee belongs to the low-speed WPAN network family, thus its bandwidth is limited. The maximum data rate is 250kb/s at every physical layers defined in the latest 802.15.4 standard.

The standard range for indoor application is about 30 m for indoor applications and about 100 m for outdoor use. The range can be also extended by using higher power, whip / external antennas, and clearing the Fresnel zone [18].

2.3.4 Power consumption

The most valuable feature of ZigBee is its power consumption. It was designed to be the least power consuming wireless standard. The networks are expected to be on low-duty. The topology should be consisting of mains powered (means always up) backbone and battery powered end devices. The main purpose of the backbone should be only to provide reliable and always available connectivity to end devices. End devices, on the other hand, should be snoozing for most of their time, waking up only for data transfer, thus saving their battery power. By this approach, end devices can live for many years on a single pair of AA batteries.

Most of the ZigBee modules/chips mentioned in this thesis offer two modes of sleeping – pin sleep and periodical sleep. Pin sleep is the mode in which the wake up / snooze cycle of the radio is controlled by another device, usually external microcontroller. In periodical sleep, the device uses an internal timer to control its duty cycle. A common maximum for a single sleep period in periodical sleep mode is beyond an hour.

Power drainage can be also controlled by lowering the radio’s transmit power. This can be most significant for the end devices, which usually require less transmit power than routers.

Some devices also offer various levels of sleep mode, distinctively light sleep, in which the radio is active for receiving only and thus requires less power, and deep sleep, from which the device can only be waken by external event or by internal clock.

In numbers, typical ZigBee radio uses from 30 to 50 mW for receive and about one quarter more for transmit. The sleep power is about (but mostly lower than) 0,001 mW.

2.3.4.1 Power down and wake up cycles

As most of the devices in the network are sleeping endpoints, the duration of a power down and wake up cycle is important factor that influences the length of battery life.

Speaking about the SoCs or modules, the time required for them to wake up is mostly dependent on the software (firmware) used. E.g. a radio running only the 802.15.4-compliant software does not need to undergo the initialization process of a ZigBee stack, which can in case of very low duty cycle save valuable milliseconds. This fact should be considered when deciding between ZigBee and proprietary 802.15.4 application.

The power down is not a computationally difficult task. Generally, the end devices can be turned down instantly at any time. Nevertheless, a timeout is usually implemented for applications which use duplex transfers.

There were also some rumors about the wake up / sleep performance of various stacks, but as there is no serious research on this topic, no conclusions can be made for now.

2.3.5 Network features

ZigBee networks can be established by a coordinator only. Upon correct PAN parameters settings, other devices may join the network, forming one of the following topologies.

2.3.5.1 Star topology

It is a basic topology, with one coordinator and a number of end devices that are all connected directly to the coordinator. It offers some advantages concerning for example latency, but on the other hand all devices have to be in the range of coordinator.

It is suitable for a small network – for example a scenario of a single data collector and several sensor endpoints within close range (figure2.6).

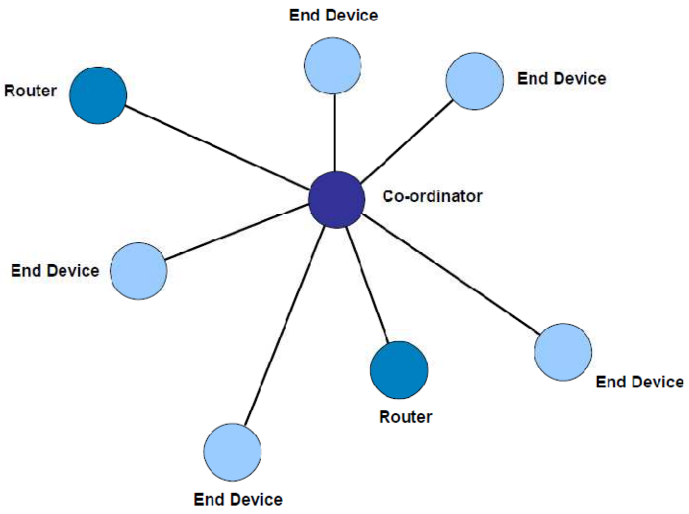


Figure 2.6: Star topology

2.3.5.2 Peer to peer topology

It is the type of topology in which every device has the ability to communicate directly to any other device within its range. For this network, the beacon mode is not the most efficient, thus it is recommended to use non-beacon mode.

In some cases it has even lower latency than star topology network, but requires most of the devices to be routers, thus it is not Due to the range requirements this approach usually leads to a small size network (figure2.7).

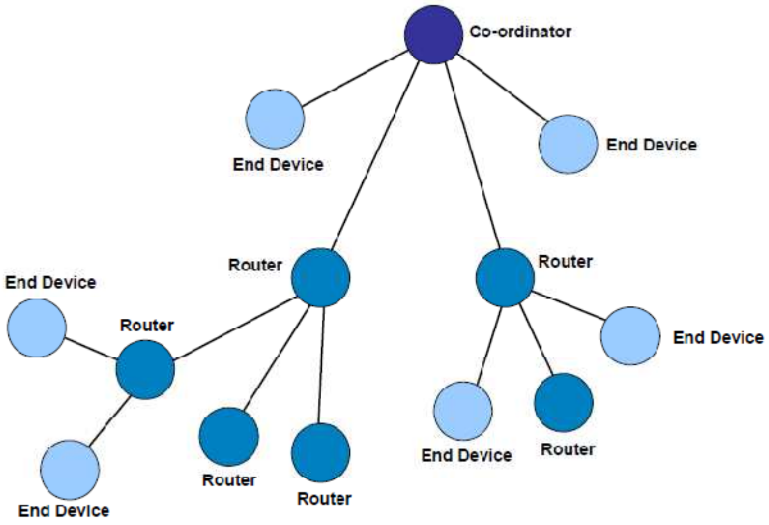


Figure 2.7: Peer topology

2.3.5.3 Mesh Topology

The extraordinary feature of ZigBee is its ability to create a large, self healing network – a mesh network. The foundation of this type of network is the fact that each router in the network is able to communicate directly with any other router in its proximity. Also, any end device in the network is able to choose any of its nearby routers as its parent (figure2.8).

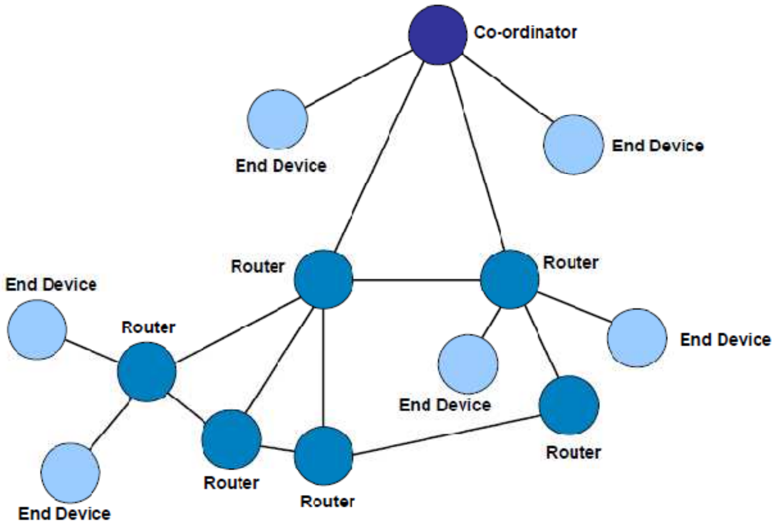


Figure 2.8: Mesh topology

This allows designing an advanced, highly reliable network where there exist several routes to every device of the network. The disadvantage of this topology is the propagation of broadcast packets.

2.3.5.4 Cluster tree topology

It is a downgrade of a mesh network. In this network, no router can have multiple parents. The structure of the network is more straight-forward but offers (compared to mesh network) only limited self-healing capabilities. It might also cause bottlenecks in backbone links of the network.

2.3.6 Security and Reliability

Both security and reliability are a significant concern of ZigBee.

- Reliability is guaranteed by:
- Self healing network topology (chapter2.3.5.3).
- Cautious access to physical medium (chapter 3.1.1 and 3.1.2).
- QoS measures and real time transfer provided by GTS (chapter 3.1.4.2).

The above mentioned features together with the physical layer structure also ensure better resistance to intentional jamming.

A sufficient level of security is guaranteed by the possibility of 128bit symmetric block AES ciphering and by measures to deny access to the network.

2.3.7 Costs

The costs of ZigBee chips are thoroughly discussed in chapter 4.1

2.3.8 Future development

Promising future lies in front of ZigBee as a wireless technology for its unique specialization. While others are more and more focused on achieving higher data rate, ZigBee is still most likely to keep its own priorities – low power, cheap hardware, small size and high reliability.

ZigBee is the youngest technology discussed in this project, it is already supported by companies such as Freescale, Texas Instruments, Panasonic or Siemens, ZigBee may be able to fill the free space in the market of wireless home automation, wireless control and low-rate wireless data collection.

Furthermore, ZigBee chips are nowadays easier to acquire than any chips for Wi-Fi, or Bluetooth, and also there is a variety of these chips. These facts might make it possible not only for commercial sphere but also for radio-amateurs to experiment with ZigBee, which might be a solid foundation for future open-source software for ZigBee. (There are already source codes available e.g. for the TI's Z-Stacks, but the license does not allow to use or modify the stack without TI's permission)

2.4 Combined technologies network

There is a great potential in using a combination of all the technologies mentioned in one network, however that type of network is out of the scope of this project.

2.5 Comparison conclusion

We would like to start by the elimination of the obviously inadequate candidate – Wi-Fi. Although Wi-Fi offers a high bandwidth, above-average range, enhanced security and decent network topology options, the fact that chips supporting it are not yet available on market makes it impossible to create a simple custom device.

Also the need of extensive external computation power and other components means a significant disadvantage when compared to ZigBee, or Bluetooth.

According to these facts Wi-Fi cannot be considered as a possible foundation for hardware in this project.

Bluetooth se resemblance to each other, especially in the means of

- Low power
- Small hardware sizes
- Limited external components required

- Modest computational power demands
- Low cost
- Prospective potential

But still there are some differences. Theoretically, for small battery-powered devices both technologies could be used. The selection in this project was made upon a more precise definition of requirements, where transmission speed was sacrificed for enhancing battery life and other benefits, as explained in following paragraph

The maximum data rate is the main advantage of Bluetooth over ZigBee – 3 Mbit/s of Bluetooth 2.0 vs. 250 Kbit/s of IEEE 802.15.4. Bluetooth also has some minor advantages, such as smaller size of some Bluetooth SoCs, which, however, cannot outperform the advantages of ZigBee.

ZigBee offers (in addition to the similarities mentioned) superb network topologies potential, ranging from point-to-point to mesh self-healing networks, while networks formed by Bluetooth have temporary character and also in Bluetooth there is no support for reliable network topologies.

Moreover, ZigBee allows vast options in power saving and has already shown it’s promising outlook by releasing several new generations of ZigBee SoCs.

The benefits ZigBee can bring in the final network led to the final decision to use ZigBee as the base platform in this project.

The results of this comparison are depicted in graphs and tables in appendices.

Chapter 3

ZigBee Specifications

Contents:

The ZigBee stack is wider than stacks of most other technologies mentioned in this thesis (only Bluetooth covers similar layer range). It is based on the Open Systems Interconnection (OSI) model and reaches up to its Application Layer (but the last 4 layers of SI model transport, session, presentation and application are grouped into a single application layer with ZigBee specific sub-layers). In following chapters, the most important features of each layer will be explained. There will be parts where the standard offers implementation specific features.

The foundation layers (physical and network) are defined by the IEEE 802.15.4 standard [17]. Each layer, as in other stacks, provides a specific set of services for the layer above. Each service entity exposes an interface to the upper layer through a service access endpoint SAP). There are two types of entities in every layer: a data entity, which provides a data transmission service, and a management entity, which provides all other services. The foundation layers (physical layer and MAC sub-layer of the Data Link layer) are defined y the IEEE 802.15.4 standard. ZigBee defines the network, transport and part of the application layer.

3.1 ZigBee and IEEE 802.15.4 Networking layers

3.1.1 PHY Features

Except from receiving and sending data via air, the physical layer is also responsible for the following tasks.

3.1.1.1 Energy detection

This feature of the physical layer enables the radio to detect power level on the current channel. During energy detection data aren’t collected nor processed, only a passive energy level measurement is done. The precision of the measurement should be 6 dB according to IEEE 802.15.4. The results of this process are essential for the Clear Channel Assessment algorithm.

3.1.1.2 Link Quality Indication (LQI)

The LQI measurement is a characterization of the strength and/or quality of a received packet. It can be based on either signal to noise ration or on the results of energy detection.

The data collected in this measurement are used only in the MAC sub-layer.

3.1.1.3 Clear Channel Assessment (CCA)

This process gets information about the availability of the current channel and reports the state (busy or idle) to the MAC layer, which uses it for the CSMA/CA process.

IEEE 802.15.4 defines three methods of performing CCA:

Energy above threshold: busy medium is reported upon detecting any energy above the ED threshold.

Carrier sense only: busy medium is only reported if a signal compliant with 802.15.4 is found with the same modulation and spreading characteristics. The energy might be above or below ED threshold.

Carrier sense with energy above threshold: a combination of above mentioned techniques.

3.1.2 MAC features

The MAC layer uses a CSMA/CA to access the radio channel. It is also responsible for transmitting network beacon frames, synchronization and transfer reliability.

3.1.2.1 CSMA/CA (Listen before send)

Carrier Sense Multiple Access and Collision Avoidance is a common mechanism used for accessing a wireless physical medium. Prior to any transmission, the radio has to listen for a defined period for a transmission on the selected medium. If there is no transmission in progress, it starts its own transmission. Otherwise it listens until the existing transfer is over.

The purpose of this method is to provide faster and reliable transfers, even if the medium is densely occupied.

In IEEE 802.15.4, two types of CSMA/CA exist:

- Slotted CSMA/CA for beacon networks
- Unslotted CSMA/CA for non-beacon networks

In beacon mode, every router transmits a network beacon in a set period. Numerous events bounded to the beacon event are described below.

In Non-Beacon-enabled mode, the devices use CSMA/CA for all transfers. Neither time intervals nor time slots are defined.

3.1.2.2 Beacon frame

Each beacon contains basic information about the network from which it originates mainly PAN identifier and source coordinator / router address.

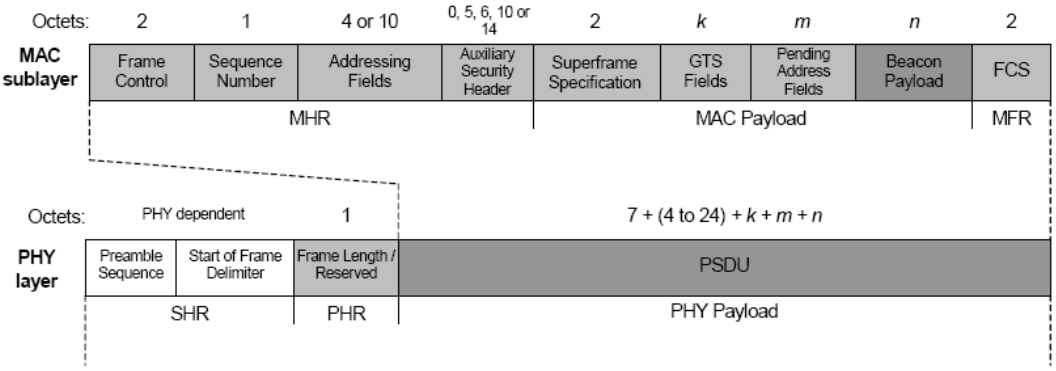


Figure 3.1: Beacon frame MAC structure and PHY encapsulation

For detailed description see [17]

3.1.2.3 Superframe structure

The coordinator of each PAN has the option of bounding its channel time using a superframe structure.

The superframe is divided into 16 equally sized slots. The first slot is always occupied by a beacon frame that is used for synchronization, PAN identification and to describe the superframe structure. *(See the appendices for an illustration o superframe structure).*

Superframes act as containers for data transfers and specific purposes, such as the Guaranteed Time Slots 3.1.2.4.

The time between two beacons is divided into *active* and *inactive* period. An active period is represented by the superframe; an inactive period is free for use by any other device in the network, e.g. for routers in the network to transmit their superframe. The active period is further divided into **Contention Access Period** and **Contention Free Period.**
Any transfers required can be done during the Contention Access Period. They should always use CSMA/CA and should finish before the Contention Free Period starts.

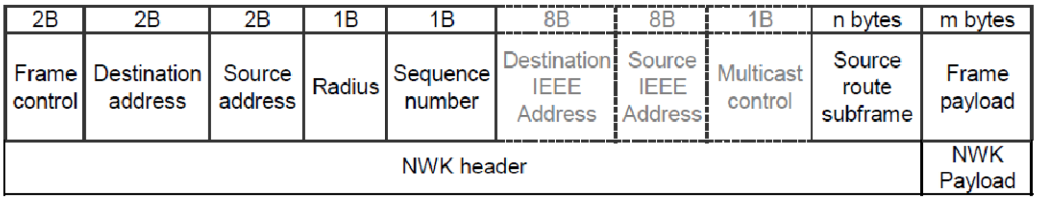
Devices transferring during the Contention Free Period do not need to use the CSMA/CA mechanism as the channel has already been pre-assigned to them (i.e. to those who have the right to transfer during this period). The pre-assigned intervals within the Contention Free Period are called Guaranteed Time Slots.

3.1.2.4 Guaranteed Time Slots

Guaranteed Time Slots are a special feature granting high reliability, low latency transfers and QoS in IEEE 802.15.4/ZigBee networks. They represent a portion of a superframe structure that is dedicated exclusively to a certain device. By standard, there can be maximally 7 GTSs in a single superframe and they should be created by the coordinator only and the transfers done in these periods should use only short (16 bit) addressing.

3.1.3 Network layer

ZigBee network layer defines several mechanisms similar to those in IP networks. Devices in ZigBee networks are using different addressing system. The basic address for all devices is the 64-bit address as defined by the MAC layer. The address used in network layer is 16-bit long, which means the maximum number devices in single network, is 65536.



2B	2B	2B	1B	1B	8B	8B	1B	n bytes	m bytes
Frame control	Destination address	Source address	Radius	Sequence number	Destination IEEE Address	Source IEEE Address	Multicast control	Source route subframe	Frame payload
NWK header									NWK Payload

Figure 3.2: Network layer frame structure

Similar to TCP/UDP transfers in ZigBee network can be either acknowledged or not-acknowledged. The acknowledge request is implemented same way as in TCP by setting an acknowledge sign.

3.1.3.1 Broadcasts

ZigBee networks, due to the network topology used, have to provide a special broadcast handling to prevent broadcast storms.

Each node (router) has to maintain a broadcast table. When a node receives a broadcast frame, it adds a new entry to the table that identifies the packet and broadcasts the frame. If there already is an entry for the current frame, it won’t be broadcasted any further. Usually, the node also keeps the whole packet buffered for a certain time, in case a retransmission is needed.

Both the size of the table and the time an entry is kept are limited, which might cause unexpected behavior when numerous broadcasts are sent/received at once.

3.1.3.2 Routing and route discovery

In ZigBee networks, all routing is done dynamically and on-demand - no routes are created by default and no manual route configuration is required.

Route discovery is based on the Ad-hoc On-demand Distance Vector (AODV) routing protocol / algorithm. The main objectives of this algorithm are to exchange / request routing information only when it is necessary. This means that when a network node needs a connection to another node, it has to broadcast a connection request, containing its address (16-bit), destination node address (16-bit) and path cost field (used as metric for AODV). This broadcast is propagated through the network; each node retransmitting the packet updates the path cost field by one and creates an entry in its route discovery table. When the destination node receives the route request, it responds with reply packet using the route with lowest path cost. It can send multiple replies if a better route has been discovered since its last response.

The ZigBee standard defines that neighbor tables should be maintained by all devices and shall contain information on every device within transmission range (the number of the devices can be limited depending on implementation). During a network discovery or during rejoining the table contains a list of candidate parents of the current device. After the device has joined a network, the table is used to store relationship and link-state information about its neighbors within the joined network as collected by the AODV protocol.

3.1.3.3 Association

When a device wants to join a new network, it has to undergo several steps. First one is the active scan, which will retrieve the list of available ZigBee networks in its proximity.

These ZigBee networks can be open or closed. Open networks allow new devices to join, closed do not. Coordinator should prevent device joining if its network is marked as closed and end devices should not attempt to join a network that is closed.

Upon discovery of a suitable PAN to associate with, joining device should set the following attributes according to the information obtained by scanning.

- Chanel
- PAN ID
- PAN Coordinator’s 16-bit (always 0000) and 64-bit address

By this point, the joining device may already start tracking the PAN beacons for optimization. When parameters have been set, joining device should send the association request and receive an acknowledged by the coordinator. However, an acknowledgment to an association request command does not mean that the device has successfully associated. Only after the coordinator decides that there are enough resources for a new device to join the network, it sends an association response command that has to be acknowledged by the joining device. According to the response, the joining device will set its 16-bit network address. This will complete the association procedure.

Depending on implementation, a network key may be passed to the end device during association. Other implementations recommend not sending plaintext keys over-the-air, but rather direct programming.

3.1.3.4 Disassociation

Coordinator may order a device to leave the network. For this it uses the disassociation command frame. The receiving device shall confirm its receipt by sending an acknowledgment frame, but even when it is not received by the coordinator the device is considered disassociated.

When an associated device willingly wants to leave the current PAN, it should send a disassociation notification to its PAN coordinator. This should be acknowledged by the coordinator, but even if it is not, the device should consider itself disassociated.

Leaving network implies the leaving device to set its PAN ID, 16-bit address, 64-bit address and coordinators addresses to default values. Upon disassociation, network coordinator should remove all references to that device.

3.1.3.5 Orphaned device

A device might find itself orphaned when it looses communication with the device through which it has its PAN membership. When such an event is detected, the device should perform (as defined by IEEE 802.15.4) the “orphaned device realignment” procedure or reset the MAC layer and then perform association procedure.

Orphaned device realignment is based on channel scanning (orphan scan). If the desired PAN has been found during the scan, the device should update its PAN information with the data received from the coordinator.

3.1.4 Application layer

The application layer consists of three sub-layers / object containers – the Application support sub-layer (APS), the ZigBee Device Objects (ZDO) and the manufacturer-defined application objects.

The Application Support sub-layer supervises an interface between application layer objects and the network layer.

Following points outline the application layer objects as well as some of the resulting features of extended ZigBee application layer addressing.

3.1.4.1 ZigBee Device Objects

ZDO are applications which implement ZigBee End Devices, ZigBee Routers, and ZigBee coordinators. ZDO represent an interface between the application objects, the device profile, and the APS.

They are responsible for initializing the APS (and consequently the network layer) and the Security Service Provider. Its interface provides address management of the device, discovery, binding, and security functions (figure 3.3).

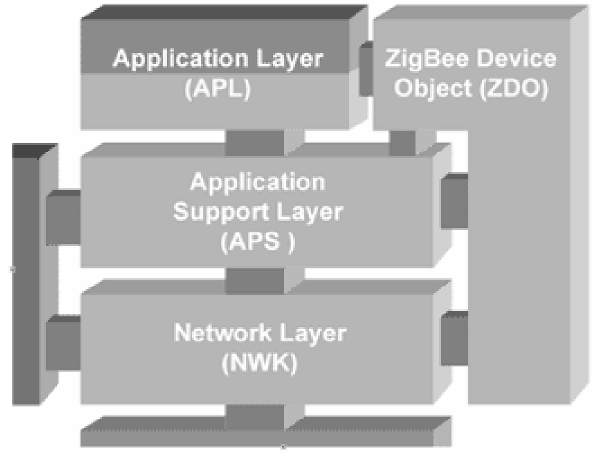


Figure 3.3: Position of ZigBee Device Object inside the ZigBee stack

3.1.4.2 Service Discovery

Services discovery is the process whereby the capabilities of a given device are discovered by other devices. It bears resemblance to profiles announcing in Bluetooth.

Service discovery can be accomplished by querying each endpoint on a given device or by using a match service feature (either broadcast or unicast). Various service descriptors are returned by this discovery. This is generally implementation dependent feature.

3.1.4.3 Device Discovery

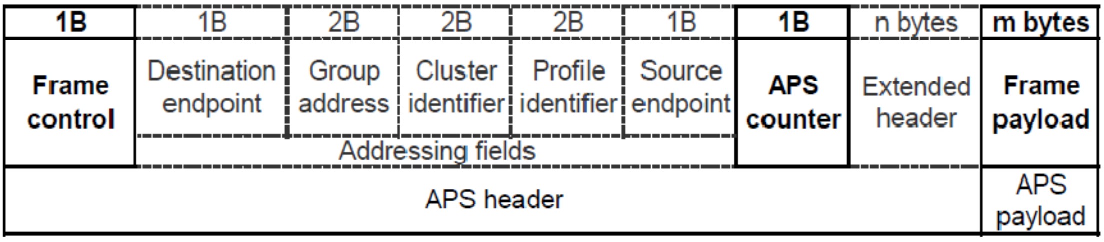
Address of devices in ZigBee networks can be discovered by two means:

- Network layer (16-bit) address discovery, in case the 64-bit address is know
- IEEE (MAC) (64-bit) address discovery, in case the 16-bit address is known

Over more, all devices in network can be discovered by sending broadcast IEEE address discovery. In general, the way all the devices are discovered is subject to implementation.

3.1.4.4 Endpoints

Endpoints are single byte values that provide application multiplexing in ZigBee. They have basically the same role as do ports in TCP / UDP protocols. Both source and destination endpoints are optional items incorporated into the APDU frame.



1B	1B	2B	2B	2B	1B	1B	n bytes	m bytes
Frame control	Destination endpoint	Group address	Cluster identifier	Profile identifier	Source endpoint	APS counter	Extended header	Frame payload
	Addressing fields							
APS header								APS payload

Figure 3.4: ZigBee application layer frame structure

a) Destination endpoints

For destination endpoints, the ZigBee standard defines that endpoint 0x00 is the address of ZigBee Device Object, 0xFF is endpoint broadcast (is delivered to all endpoints). Endpoints from 0xF1 to 0xFE are reserved. All other endpoints (0x01-0xF0) are available for custom implementations (table 3.1).

Destination Endpoint Address	**Destination Interpretation**
0x00	ZigBee Device Object
0x01-0xF0	Not defined (application specific)
0xF1-0xFE	Reserved
0xFF	Broadcast (all endpoints)

Table 3.1: ZigBee destination endpoints

b) Source endpoints

For source endpoints, the ZigBee standard defines endpoint 0x00 as ZigBee Device Object, endpoints from 0xF1 to 0xFF reserved and all other endpoints (0x01-0xF0) as free for use.

Source Endpoint Address	Source Interpretation
0x00	ZigBee Device Object
0x01-0xF0	Not defined (application specific)
0xF1-0xFF	Reserved

Table 3.2: ZigBee source endpoints

3.1.4.5 Clusters

The cluster identifier field is 16 bits in length and specifies the identifier of the cluster to which a frame relates. It can be used e.g. for enhancing routing effectiveness or frame filtering, but its general purpose is to distinguish between groups of different device types in a network.

For example, the cluster ID might identify a group of light switches, another one the group of lighting in single room. Or perhaps one cluster might be a group of temperature sensors and another group of conditioning control devices.

3.1.4.6 Profiles

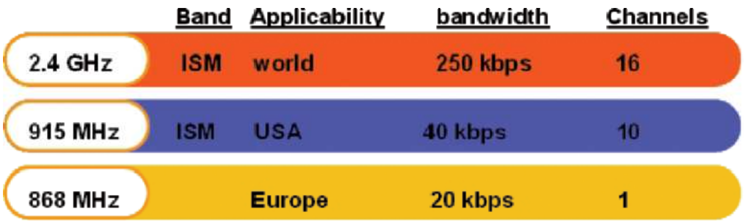
The profile identifier is represented by two bytes and specifies the ZigBee profile identifier for which the frame is intended.

ZigBee profiles are analogical to Bluetooth profiles. They are maintained by the ZigBee alliance and should guarantee interoperability between various manufacturers' devices in single network by defining (in general) offered services and data presentation within the profile.

3.2 RF Propagation and radio theory of IEEE 802.15.4

After description of Zigbee protocol functional radio comportment is then analyzing, in order to get inside view of spread spectrum and type of modulation used in IEEE 802.15.4.

There are several radio bands defined in the physical layer of 802.15.4. The narrowest is the band at 868 MHz, offering 1 channel. The second is the 915 MHz band, with 10 channels. The widest band is at 2,4GHz, offering 16 channels.



	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 3.5: 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:

- 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.

3.2.1 Receiver Sensitivity

In the IEEE 802.15.4 standard, receiver sensitivity is defined as the lowest received signal power that yields a packet error rate (PER) of less than 1%, IEEE 802.15.4 requires only -85 dBm of sensitivity for operations in the 2.4 GHz ISM band. In the 868/915 MHz band, if the BPSK modulation is used, the required sensitivity is -92 dBm. The optional modes of operation in the 868/915 MHz band (using ASK and OQPSK modulation) must meet - 85 dBm of sensitivity.

If a 50 Ohm single-ended antenna is used, the sensitivity level of -85 dBm translates to a signal with effective (or rms) voltage of 12.6 μV:

$$\text{Signal power} = \text{-85dBm}=10^{(\frac{-85-30}{10})}=3.16\text{X}10\text{-}12=3.16\text{pW}$$

This means that any received signal with root-mean-square (rms) voltage of 12.6 μV or higher can be detected and the data can be extracted with a PER of less than 1%.

3.2.2 The Modulation and Spreading Methods for 2.4 GHz Operation

Any time domain signal has an equivalent representation in the frequency domain. The graph of the signal power versus frequency is referred to as the *signal power spectral density* (PSD). Figure 3.6 represents a typical IEEE 802.15.4 signal PSD centered at 2450 MHz this center frequency is known as the *carrier frequency*. The signal bandwidth is normally considered the frequency band that contains the majority of the Signal power.

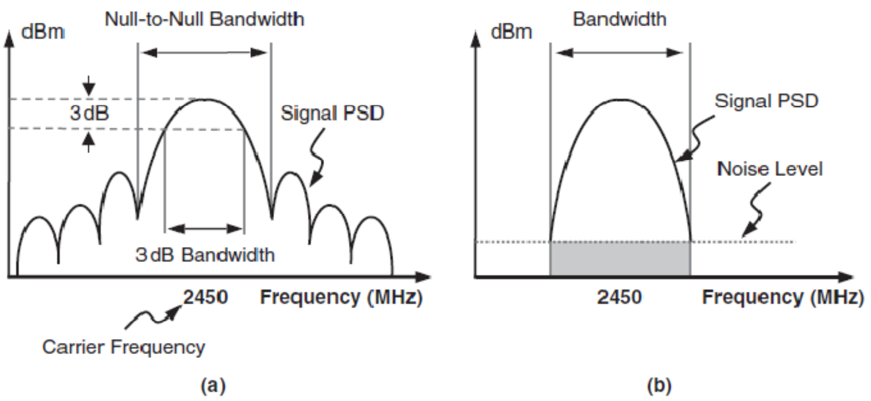


Figure 3.6: (a) Signal Bandwidth Definitions and (b) Signal-to-Noise Ratio (SNR) Definition

The spreading method required by IEEE 802.15.4 for the 2.4 GHz frequency band is the Direct Sequence Spread Spectrum (DSSS). In an IEEE802.15.4-specific implementation of DSSS, every 4 bits of each octet of a PHY Protocol Data Unit (PPDU) are grouped together and referred to as a symbol (figure 3.7).

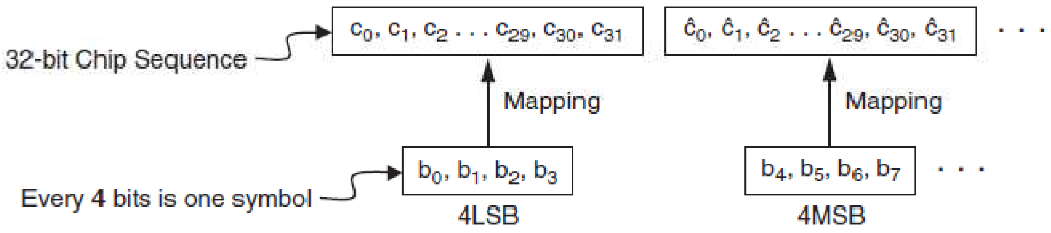

32-bit Chip Sequence
c0, c1, c2 . . . c29, c30, c31
ĉ0, ĉ1, ĉ2 . . . ĉ29, ĉ30, ĉ31
. . .
Mapping
Mapping
Every 4 bits is one symbol
b0, b1, b2, b3
b4, b5, b6, b7
. . .
4LSB
4MSB

Figure 3.7: The Signal Spreading in DSSS

Then a lookup table is used to map each symbol to a unique 32-bit sequence. This 32-bit sequence is also known as the *chip sequence* or the *pseudorandom noise (PN) sequence.* The table of these symbol-to-chip mappings is provided in Appendix.

With calculating the *cross-correlation functions of two sequences minimize* its similarity to the other 15 sequences.

Since each 4 bits of the actual data are mapped to a 32-bit chip sequence, the effective over-the-air bit rate is increased by a factor of eight. The bit rate is proportional to the signal bandwidth, we know that the signal bandwidth will also increase by a factor of eight, If the original signal before spreading has a bandwidth of 250 KHz, after the spreading the bandwidth will be increased to 2 MHz, the processing gain for the 2.4 GHz mode of operation in IEEE 802.15.4 is equal to 9 dB:

$$\text{Processing Gain} = 10\text{X}\log 10 \quad \left(\frac{2Mbps}{250\text{Kbps}}\right) \cong \; 9\text{dB}$$

IEEE 802.15.4 requires the use of offset-quadrature phase shift keying (OQPSK) modulation for the 2.4 GHz mode of operation. In a phase shift keying (PSK) modulation, the signal phase is used as a way to transport binary information between the transmitter and the receiver. Each signal phase corresponds to a binary number. The transmitter sets the signal phase before transmission and receiver can recover the binary information by detecting the signal phase.

The simplest PSK modulation is BPSK (binary PSK). In BPSK, the phase θ (t*)* can only take two discrete values, zero and 180 degrees. These two phases represent binary levels of zero and one. The quadrature PSK (QPSK) allows for four phase options: 45 °, 135 °, –45 °, and –135 °. Therefore, each phase can represent 2 bits. The four phases in QPSK correspond to binary numbers 00, 01, 10, and 11 (figure 3.8).

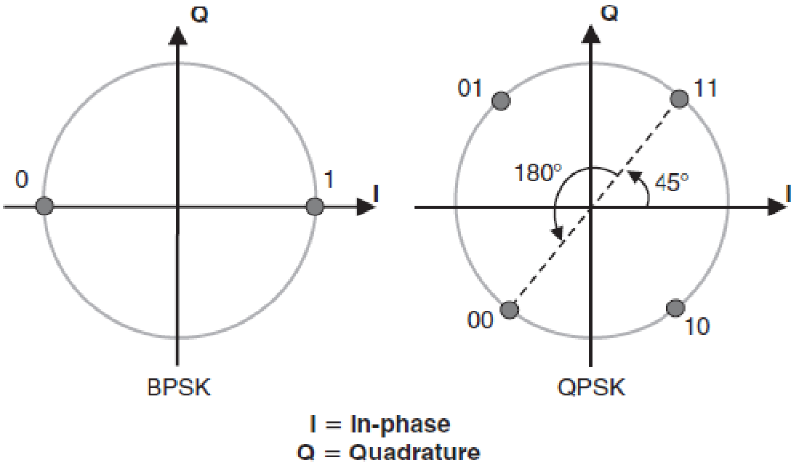


Figure 3.8: The BPSK and QPSK Constellations

One observation in QPSK modulation is the following:

$$\theta = \frac{\pi}{4} \rightarrow s(t) = A \times \sin\left(2\pi fct + \frac{\pi}{4}\right) \; = \; \frac{A}{\sqrt{2}} \times (\sin(2\pi fct) + \cos(2\pi fct))$$

$$\theta = \frac{3\pi}{4} \rightarrow s(t) = A \times \sin\left(2\pi fct + \frac{3\pi}{4}\right) \; = \; \frac{A}{\sqrt{2}} \times (\sin(2\pi fct) + \cos(2\pi fct))$$

$$\theta = \frac{-3\pi}{4} \rightarrow s(t) = A \times \sin\left(2\pi fct + \frac{-3\pi}{4}\right) \; = \; \frac{A}{\sqrt{2}} \times (\sin(2\pi fct) - \cos(2\pi fct))$$

$$\theta = \frac{-\pi}{4} \rightarrow s(t) = A \times \sin\left(2\pi fct + \frac{-\pi}{4}\right) \; = \; \frac{A}{\sqrt{2}} \times (\sin(2\pi fct) - \cos(2\pi fct))$$

This means that all four possible phase options in QPSK can be built by summation or subtraction of sin ($2\pi fC\ t$) and cos ($2\pi f\ C\ t$) functions.

Figure 3.9 shows the basic design of a QPSK modulator in a transmitter. The train of pulses with -1 and -1 values are multiplied by sin (2πfct) and cos (2πfct) functions, and the results are added together to form the modulated signal s(t). The signal s(t) will be amplified by a power amplifier (PA) before transmission. The modulated signal s(t) consists of two components:

An **in-phase component** (- cos (2πfct)) and a **quadrature component** (-sin (2πfct)).

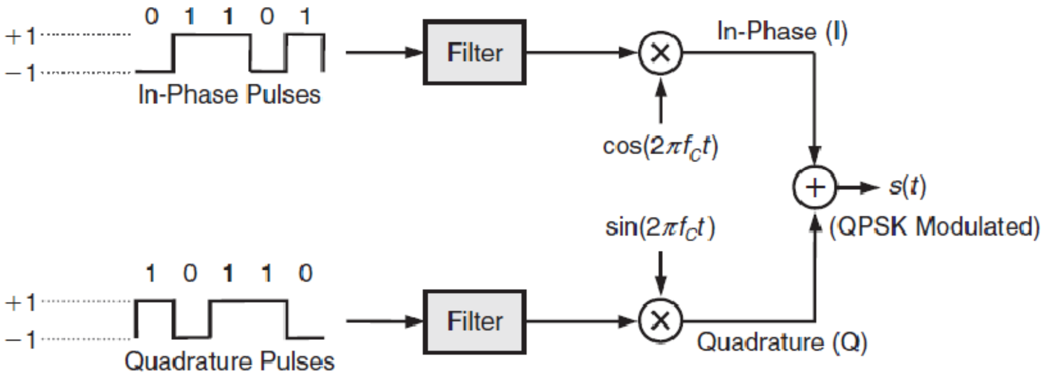


Figure 3.9: The Simplified Diagram of the QPSK Modulator

To minimize the signal bandwidth and suppress the signal power outside the frequency band of interest. The QPSK modulator in Figure 3.9 use of specific filters (half-sin filter, raised cosine filter, and square-root raised cosine filter). To shape the signal PSD. Figure 3.10 shows the effect of the half-sin pulse-shaping filter.

The half-sin filter replaces each sharp pulse with one half of a sinusoid signal:

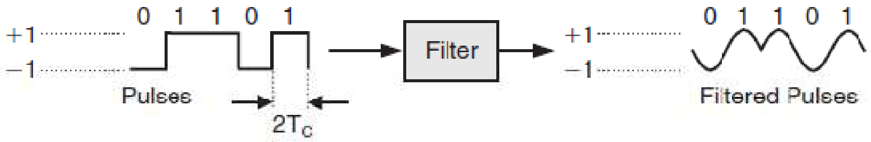


Figure 3.10: The Effect of the Half-Sin Pulse-Shaping Filter

The second problem with the QPSK, is the maximum phase shift, the abrupt phase change causes large amplitude variation in s(t). The smaller the amplitude variations, the easier the implementation of the PA. Therefore, the QPSK modulation itself is slightly modified to an Offset-QPSK (OQPSK) to limit the maximum abrupt phase shift in s(t).

In QPSK there is no time delay or offset between the in-phase and quadrature pulses that enter the modulator. However in an OQPSK there is a time offset equal to one-half of the pulse period between the in-phase and quadrature pulses, with this limit the maximum instantaneous phase shift of a signal modulated using OQPSK to 90 degrees

Figure 3.11 represents the operations performed in a transmitter for the 2.4 GHz mode of operation in IEEE 802.15.4. The individual bits are grouped together to form a symbol. Then the symbols are modulated, amplified, and transmitted.

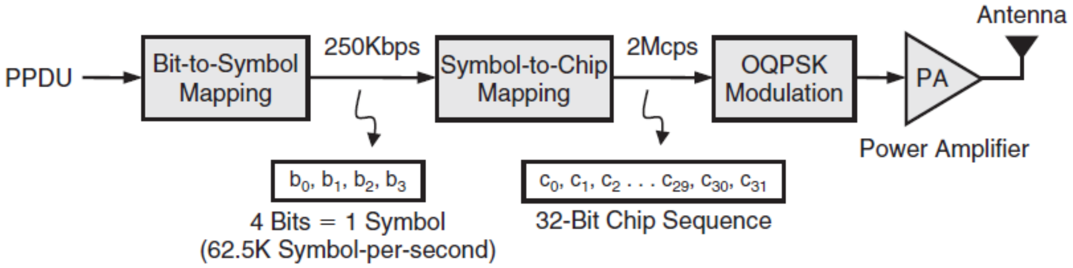


Figure 3.11: The Spreading and Modulation for the 2.4 GHz Operation

3.2.3 Error Vector Magnitude

The *error vector magnitude* (EVM) is an indication of the modulation accuracy.

In an ideal transmitter, the output signal has all the constellation points at the ideal locations.

But in reality, the location of the actual signal on the I-Q diagram may be drifted from the ideal position. We always expect a receiver to tolerate some level of modulation error. To quantify this modulation error, let's start by calculating the amplitude of the error:

$$\text{Error Amplitude} = \sqrt{(\mathrm{Ii} - \mathrm{I_A})^2 + (\mathrm{Qi} - \mathrm{Q_A})^2} = \sqrt{\Delta I^2 + \Delta QI^2}$$

I_i and Q_i are the In-phase and Quadrature values of an ideal signal. The actual location of the signal is shown by I A and Q_A . The root mean square (rms) value of the error amplitude for *N* symbols is:

$$\text{Error amplitude (rms)} = \sqrt{\frac{1}{N}\{\sum_{K=1}^{N} \Delta I^2 + \Delta QI^2}$$

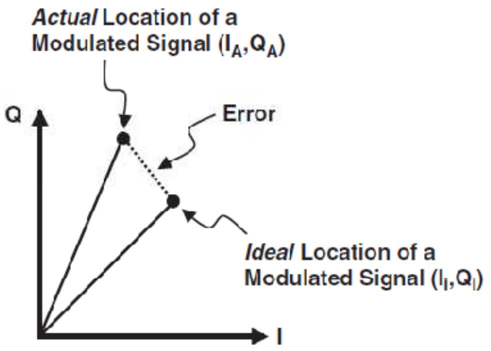

Actual Location of a
Modulated Signal (I_A,Q_A)
Q
Error
Ideal Location of a
Modulated Signal (I_I,Q_I)
I

Figure 3.12: Error Vector Magnitude

If the ideal signal amplitude is S, the EVM is defined as:

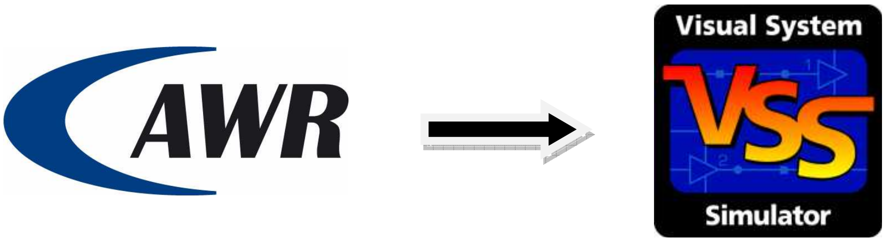
$$EVM(\%) = \frac{Error\ Amplitude}{Ideal\ Signal\ Amplitude} \times 100 = \frac{\sqrt{\frac{1}{N}(\sum_{K=1}^{N} \Delta I_K^2 + \Delta Q_K^2}}{S} \times 100$$

IEEE 802.15.4 requires measuring the error amplitude for **1000 chips**. The result must show an EVM of less than **35%** to pass the EVM requirement of the IEEE 802.15.4 standard.

3.3 VSS Visual System Simulator

VSS is a sampled time-domain simulator that uses a fixed time step which is set either by the default system settings for every system diagram, or by individual blocks inside a system diagram (usually sources) and inherited by subsequent blocks. (www.awrcorp.com).

VSS software enables you to design and analyze end-to-end communication systems. You can design systems composed of modulated signals, encoding schemes, channel blocks and system level performance measurements. You can perform simulations using VSS’s predefined transmitters and receivers, or you can build customized transmitters and receivers from basic blocks. Based on your analysis needs, you can display BER curves, ACPR measurements, constellations, and power spectrums, to name a few. VSS provides a real-time tuner that allows you to tune the designs and then see your changes immediately in the data display.



3.3.1 IEEE 802.15.4 ZigBee 2.4 GHz Test Bench

This simulation illustrates the use of basic VSS blocks to construct a 2.45 GHz ZigBee transmitter that has the same functionality as the system shown in the figure directly below. It also shows how to introduce amplitude imbalance, phase imbalance, and DC-offset to the TX signal through use of the "Input Imbalance" subcircuit. When the simulation is running, an EVM measurement is performed as you tune on DC-offset, Amplitude Imbalance, and Phase Imbalance.

The 2Mchip/s signal is created in VSS by multiplying a 250 kHz data signal (RND_D) with a PRBS that is operating at 2MHz. The DAC blocks are used to ensure that the sampling frequency registered at both inputs of the multiplier is the same as well as change the incoming signal's polarity from {0,1} to {1,-1} . (Sampling Frequency = Data Rate * Samples per Symbol). The real to digital block takes the {1,-1} real data and converts it to digital data prior to going into the OQPSK_TX block.

The OQPSK_TX block has its pulse shape parameter (PLSTYP) set to Half Sine and its center frequency set to 0GHz. The output of the transmitter is passed through an "Input Imbalance" block that allows you to simulate imbalance on the in-phase (I) and quadrature (Q) rails.

The output of the imbalance block is connected to an RF attenuator of 15dB. A VSS behavioral mixer (MIXER_B) and a TONE source are used to create the complex envelope of the 2.45GHz modulated signal. MIXER_B internally splits the complex baseband waveform coming into it into individual I and Q rails

The output of the mixer is then fed into a linear gain block as well as a non-linear amplifier (AMP_B). The signal align block (ALIGN) is used to scale the measured signal relative to the reference signal prior to making an EVM measurement. The EVM calculation is performed via use of the VSA and the EVM measurement window.

As you tune on the imbalance parameters and DC-offset you can improve or degrade EVM results. As the simulation is running you can monitor the data, the average RMS EVM in percent, the pulse shaped signal, the reference IQ plot versus the measured IQ plot, and the output spectrum.

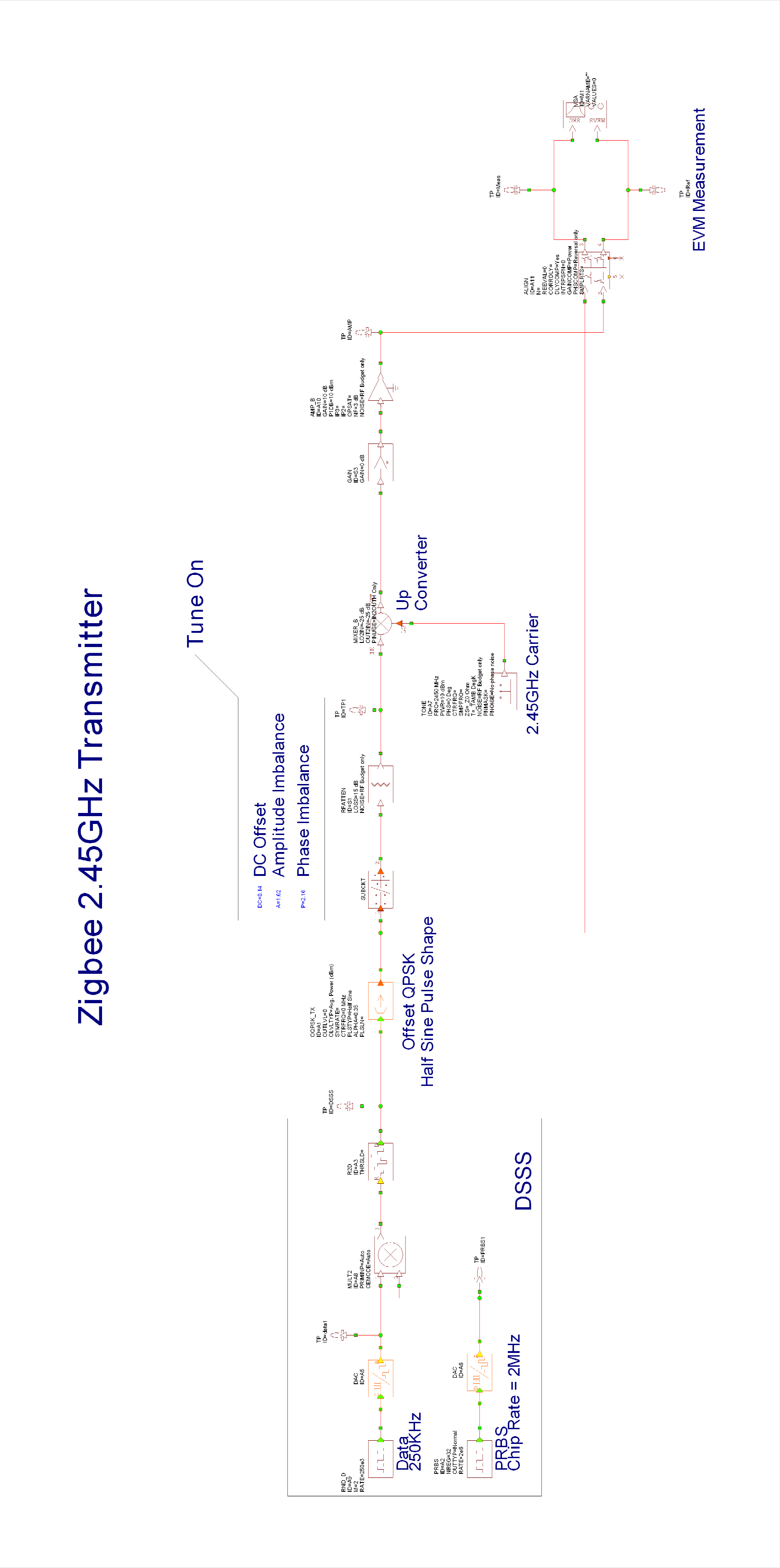


Figure 3.13: bloc diagram of Zigbee transmitter

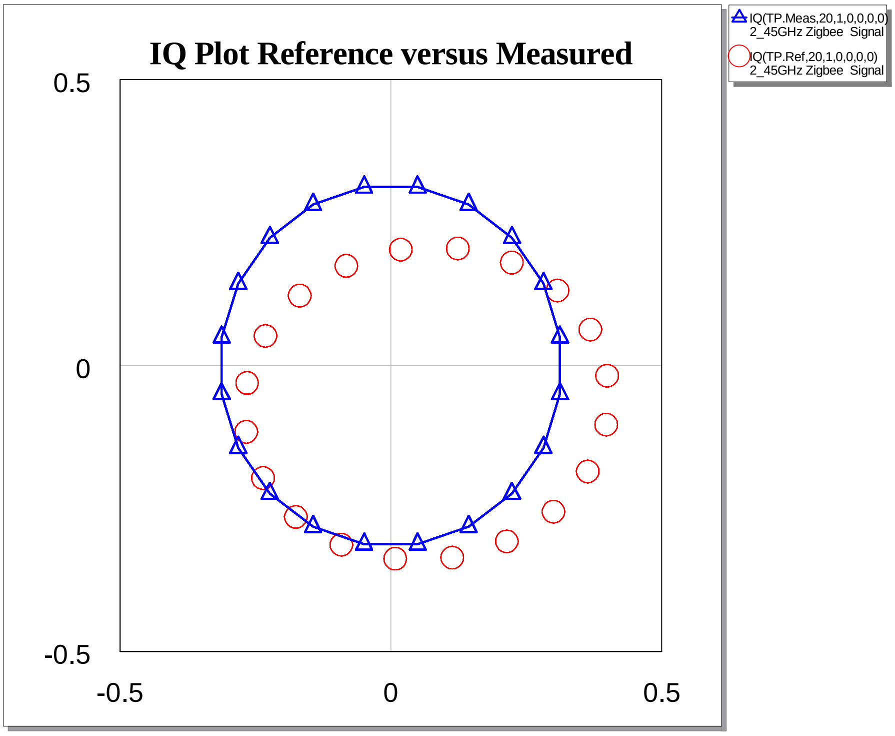


Figure 3.14: IQ plot reference versus measured

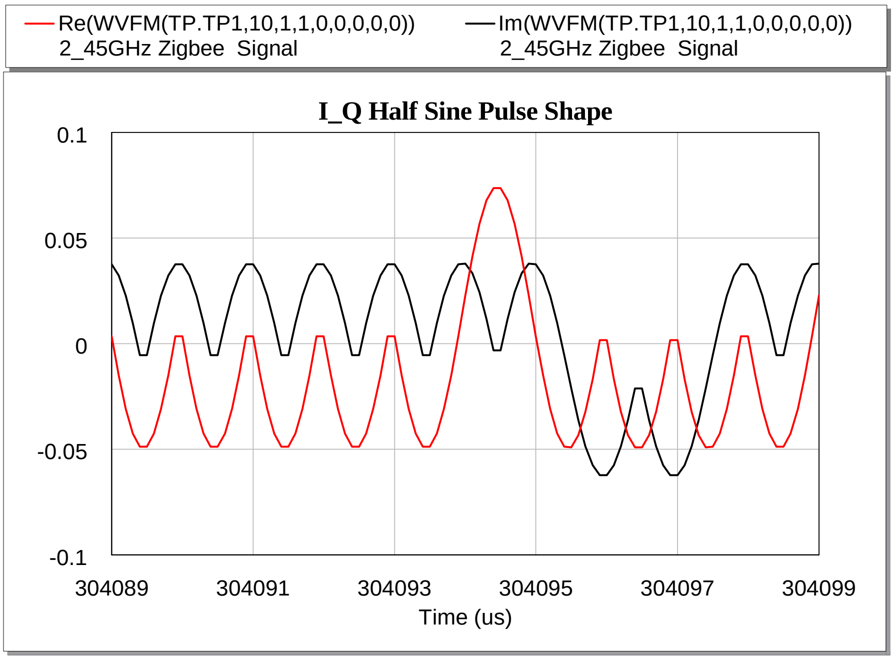


Figure 3.15: IQ half sine pulse Shape

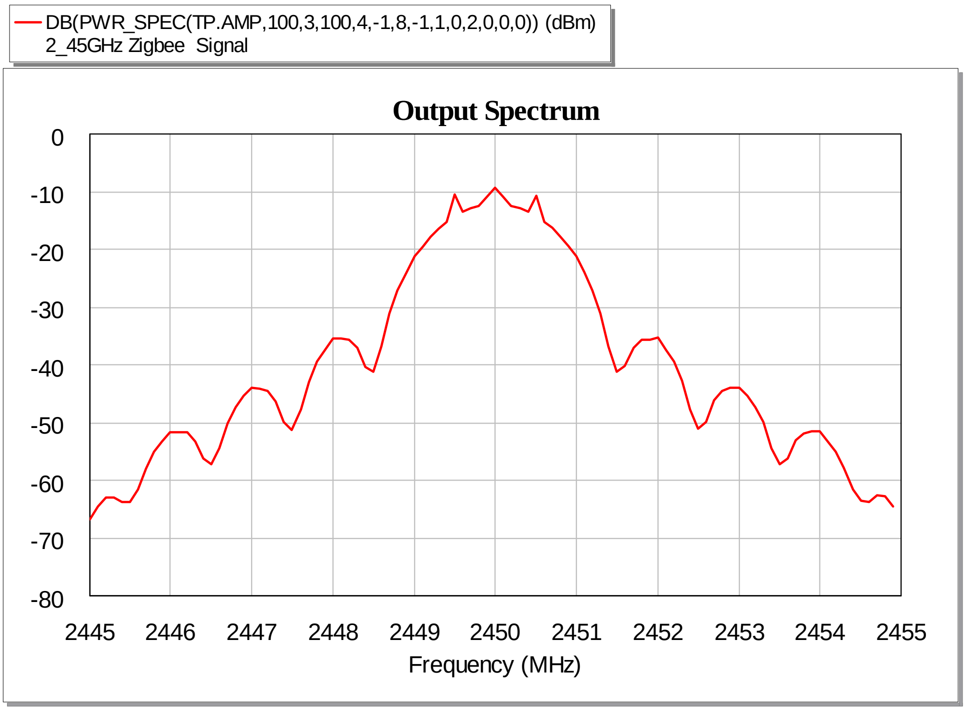


Figure 3.16: Output spectrum of IEEE 802.15.4

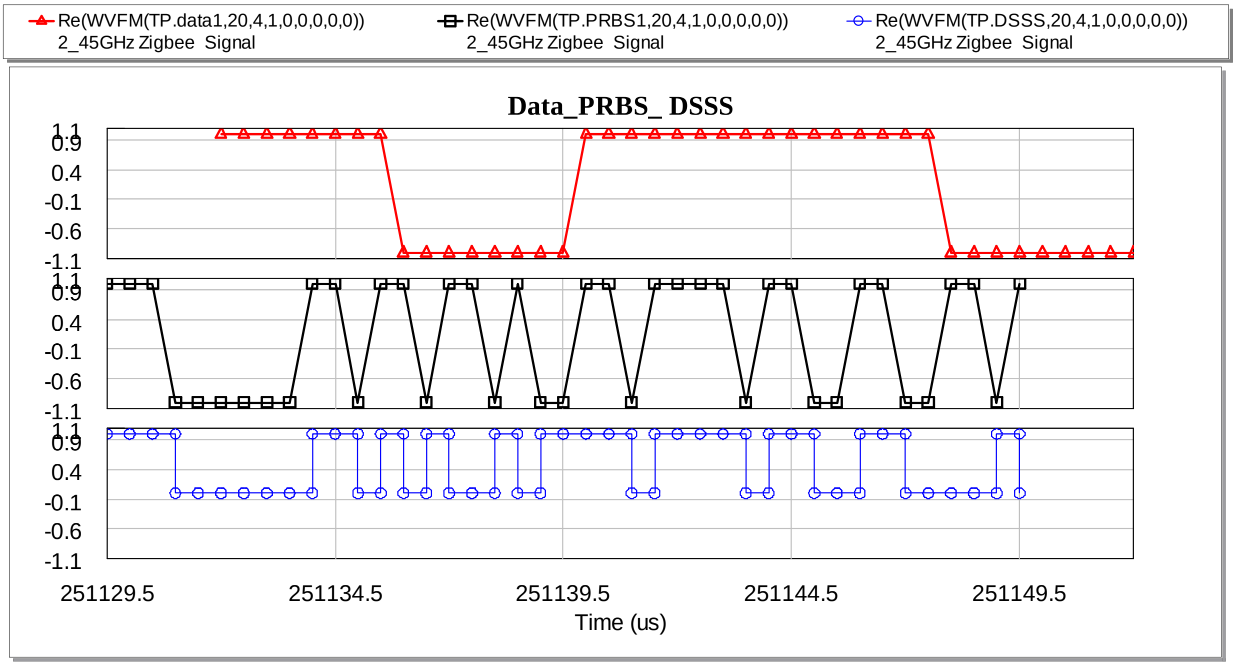


Figure 3.17: The Signal Spreading in DSSS

Chapter 4

ZigBee Hardware Comparison

Contents:

In this chapter, the main circuit’s parts of Zigbee are compared. The ZigBee system comprises the following components: transceivers, antenna, memory module and MCU as hardware and a firmware for the programming means. The companies whose circuits are implemented in ZigBee’s are: Texas instrument, Ember, Freescale, Microchip, Atmel, Meshnetics, RFM (circonet), Jennic, MaxStream (Digi).

4.1 Transceivers, Chips, SoCs, SiPs

4.1.1 Texas Instruments

Texas Instruments [19] is one of the largest producers of ZigBee-compliant and also one of the most promising. During this work, Texas has developed a 2nd generation of ZigBee SoC solutions and released two new chips (CC2520 and CC2480A1). There is a total of 5 chips in their ZigBee portfolio now (2009).

4.1.1.1 Transceiver “only generation”

A single product was released in this generation, however, with huge success - the CC2420 radio [20]. It was one of the first 802.15.4-2003 compliant chips available on the market (early in 2006), enabling several other manufacturers developing their modules and ZigBee solutions.

Since, it was only an 802.15.4 radio, thus an external computing unit is required to run the ZigBee stack to create a complete ZigBee compliant solution. The development kit offered by TI includes the MSP430 processor.

Reference boards designs and sample applications are freely available at Texas Instruments site web [19].

4.1.1.2 First generation of ZigBee SoC

There are two products that were released in the first generation of SoCs-CC2430 [21] and CC2431 [22]. CC2431 is exactly the same as CC2430 (pin compatible), but features a location engine.

Both of the SoCs use an 8051 core (running at 32MHz) as MCU and CC2420 radio core and run the Texas Instruments Z-Stack. They have up to 128 KB of programmable flash and 8KB of RAM (4KB with sleep-mode retention). Their official current usage in power down mode is excellent 0.3µA.

They are an all-in-one solution for ZigBee application, with minimum external components required. With Texas Instruments ZigBee Stack (Z-Stack) [23] they can be used for ZigBee Applications, but can be as well serving any other purposes in 2.4GHz, e.g. using Texas Instruments MAC Stack (TIMAC) [24] only CC2431’s location engine is a sophisticated feature, enabling relative position determination based on the: Using the RSSI it can measure the distance from a single node. By measuring signal strength from multiple sources (three at least), an X-Y position can be determined.

The maximum software resolution is 0.25 m, but the real resolution may be much lower, depending on local conditions. Also, the fact that it uses RSSI to determine the distance brings several issues and may lead to numerous errors in measurements. No other manufacturer released similar feature in its products.

4.1.1.3 Second generation of ZigBee SoC

CC2520 [25] is the second generation of TI’s Zigbee SoCs, released in December 2007. Compared to previous generation, it offers wider range of operating voltages (1.8V – 3.8V), smaller packaging, lower power consumption, extended operational temperature range (- 40 –125 °C) (-40 – 257 F), extended range, 728 KB of RAM and much more.

Its radio is no longer the CC2420 core, but a new 802.15.4-2006 compliant platform. Also, the processing unit is a specialized block instead of 8051 core. It no longer features any kind of programmable memory. All custom computation has to be done in an external processor.

Communication with CC2520 is done by sending instructions via the SPI interface. Instructions are summarized in product’s description and are freely available.

At the moment, CC2520 outperforms other radios in the terms of sensitivity and RF performance.

4.1.2 Freescale

Freescale [26] has both radio transceivers (MC1319x (first generation) and MC1320x (second generation) series) and SiPs (MC1321x series) in its portfolio.

4.1.2.1 Transceiver family – first generation

This is Freescale’s first family to provide complete 802.15.4 compliant transceivers. They are designed to interface HCS08 Family of MCUs, but any processor with four-wire SPI can be connected as well.

MC13191 is intended for proprietary applications in the 2.4GHz band. The MC13192 is an 802.15.4 compliant transceiver [28], which can be also used for ZigBee applications with an external MCU and ZigBee stack. MC13193 is a transceiver for ZigBee applications.

MaxStream is using Freescale chips in its products.

4.1.2.2 Transceiver family – second generation

The second generation was released in 2006. It has some features above the first generation that can lower the overall size of final product e.g. integrated transmit/receive switch. However, the improvements to the successor of MC13192 - the MC13202 transceiver are on the whole not that noticeable as the improvements between e.g. TI’s CC2420 CC2520.

4.1.2.3 SiP family

These chips came to the market in early 2006 and their capabilities are comparable to TI’s CC2430 series. They feature an integrated HCS08 family MCU (40 MHz) together with up to 60KB Flash and up to 4KB of RAM.

Due to the flash sizes MC13212 and MC13211 chips are not capable of running ZigBee stack, thus are not suitable for ZigBee applications without external MCU components.

4.1.3 Ember

Ember’s [28] portfolio is narrower than Freescale’s or TI’s. There are two products related to ZigBee ,the EM250 SoC and the EM260 Co-Processor.

4.1.3.1 EM250 SoC

EM250 is a system-on-chip. It has a Cambridge Consultants’ 12MHz XAP2b processor with 120KB flash and 5KB of RAM and offers under 1µA sleep current. However, transmit / receive currents are slightly higher than Freescale’s or TI’s. Its special feature is the boost-mode enhancement of transmit power for the cost of higher power consumption.

By default, it is loaded with Ember’s EmberZNet Zigbee Stack.

4.1.3.2 EM260 Coprocessor

EM260 is a device that enables hardware separation of the end-manufacturer application and EmberZNet Zigbee Stack. The EM260 handles the stack processing, while other MCU can fully use its power for the application. This approach bears resemblance to the TI’s CC2520 transceiver.

4.1.4 Microchip

Microchip [29] produces only a single IEEE 802.15.4-2003 transceiver – the MRF24J40 [30]. It has lower sleep consumption than either Freescale’s or TI’s radio 0.2µA. Additionally, receive/transmit currents are lower. This might be because the MCU is running at 20MHz only. External MCU can be connected via 4-wire SPI.

4.1.5 Atmel

Atmel [31] offers two generation of RF transceivers for 802.15.4 / ZigBee application. Both of them are transceivers only, thus require an external MCU with ZigBee stack, connected via 4-wire SPI. They younger generation (AT86RF231) offers 0.02µA current consumption in sleep mode and 0.4mA with the radio turned off. It claims to have better sensitivity than most of the industry chips. Atmel also offers 'bundled' packages of MCUs and theirs radios.

4.2 Modules and modems

Modules and modems are devices based on SoC, SiP, RoC or transceivers with MCU. They are usually small integrated circuits with some of the internal GPIO and communication interfaces exposed to external pins. Typically, they use UART for communication, but some of them do have also I2C or SPI.

They come with pre-flashed firmware and usually, the internal components cannot be directly accessed and reprogrammed with custom firmware.

4.2.1 Radiocrafts

Radiocrafts [32] offers two ZigBee module series-RC220x, based on CC2420 and Atmel MCU, and RC230x, based on CC2430/CC2431 with 8051 core.

Radiocrafts' modules offer more GPIO than any others, but transmit current, physical dimensions, sensitivity and range are average.

4.2.2 Meshnetics

Meshnetics [33] manufacture two versions of their basic module – one with balanced RF port only and one with dual chip antenna.

A special product of Meshnetics is the ZigBit Amp Module. Using higher output power, it is able to reach exceptional range performance (the company shows a detailed demonstration of a 4 km link). [34]

Standard modules have the best sensitivity, range and physical dimensions of all modules mentioned in this thesis. But on the contrary, they use more power while asleep.

4.2.3 RFM (Cirronet)

The Cirronet [35] modules are now manufactured under the RFM brand. There are 2 families of these modules; each family is offering 3 versions differing in transmit power strength. High power versions come either with or without on-board antenna. The difference in the two others families is at least only in the module layout, according to data sheets and web information.

4.2.4 Jennic

Jennic also features two module families in its portfolio. The older of them (JN5121) is however largely outperformed by the new family JN5139 [36].

4.2.5 Digi (MaxStream)

MaxStream [37] supplies two families of their XBee modules, both of them coming in standard and boosted (Xbee-PRO) power versions. First generation, with Freescale's MC13193 chip inside, is 802.15.4 compliant only, while the second features ZigBee Stack and is certified for ZigBee and is based on Ember's EM250.

These are the only modules that come in package with pins. Their parameters might not be exceptional, but still they are the least expensive of all the modules mentioned in this thesis.

4.3 MaxStream XBee ZNet 2.5 (Series 2)

In this project, MaxStream's XBee ZNet 2.5 OEM RF Modules are chosen for the wireless platform. (MaxStream comes now under the new name of Digi).

Initially, Texas Instruments SoCs were considered as our possible solution, but the time required for development and implementation seemed too long to suit the time devoted for this thesis, thus ready-made modules solution is adopted.

According to parameters, MaxStream's modules might not be the best performing, but since their price is lower than of any other modules, have an appropriate package, and could be provided at once (time of their purchase is small). These facts altogether made our final platform based on for these modules.

4.3.1 Hardware versions

4.3.1.1 Antennas

Two variations of modules antennas are available – whip and chip.

An external antenna can be attached as well. There are two versions of modules with antenna connectors–with U.FL connector (used e.g. in Wi-Fi Mini PCI cards) and with reversed-polarity SMA (Subminiature version A) connector.

4.3.1.2 PRO / Standard version

For applications where extended range is required, MaxStream offers a solution in the form of power-boosted modules. In absolute numbers, these modules offer 50mW (that means +17dBm) extra power for North America and 10mW (+10dBm) power boost for International use. The manufacturer claims that up to 1.6 km range can be reached with PRO modules.

The transmit current required by the PRO devices is, however, 295mA, which results in about 1W of total power required for transmissions. Additionally, the PRO modules are 8mm longer (24.3x32.9 mm) final size than the standard version, plus only 13 channels can be used.

Standard versions have the ability to dynamically enable or disable ‘boost mode’-mode of enhancing output power (by 2dBm) and receive sensitivity (by 1dBm) with a slight sacrifice of the battery life. Over more, there is another option of 4-level power regulation, enabling even more saving of battery life at end devices and higher gain at routers.

4.3.2 Firmware versions

4.3.2.1 Communication interface

Based on the type of communication with the device, the firmware can be:

- AT transparent operation, commands are entered only in a special mode
- API all communication is done via structuralized packets

In one network, devices running both AT and API can be present and communicate with each other.

4.3.2.2 Device type / purpose

Based on the propose of the device, the firmware can be

- End device/router
- Coordinator

Note that although the coordinator is basically a router with some additional features, devices with coordinator firmware cannot have parents – i.e. they cannot join an existing network and act as routers. Only one coordinator per network is allowed.

All the devices contain a ‘ZigBee End Device / Router AT’ firmware by default. To create a coordinator, a “ZigBee or (ZNet 2.5) Coordinator” firmware must be flashed to the device by the X-CTU software (as described in appendices).

4.3.2.3 Firmware capabilities

Based on the firmware capabilities

- Zigbee firmware
- ZNet firmware

These two firmwares are interchangeable and compatible for both radio and UART communication.

However, there are some minor issues and undocumented differences between the firmwares; these are discussed in 4.3.6 and 4.3.7 below. Generally, the ZigBee firmware offers a subset of the ZNet firmware features, as depicted in the following table:

	Zigbee	**ZNet 2.5**
Extended Node Discovery options	No	Yes
Enciphering	No	Yes
Periodic IO Sampling	No	Yes
Associate LED blink time speed control	No	Yes
IN wake from cyclic sleep	No	Yes

Table 4.1: Comparison of ZigBee and ZNet 2.5 firmwares for XBee

For PRO devices, there is only the ZNet firmware available.

4.3.3 Network configuration

4.3.3.1 Notice - Deny / Allow Joining

Every router can permit or deny device joining. This can be used as a security precaution, but only under certain conditions, as described below.

Based on the joining policy, networks can be open or closed.

- Open network is a network where joining is always permitted (the NJ is 0xFF on all devices). This type of network is useful for end-devices roaming and other dynamic changes in the network.

- Closed network on the other hand may but may not permit devices to join. It should be used only if the network is static -all devices have their permanent parents and will keep the same operating channel all the time.

4.3.3.2 Coordinator

The base of the network is the coordinator. To base a new ZigBee network, coordinator must choose a channel and personal area network identification (PAN ID).

Prior to choosing a channel, the coordinator performs an energy scan on all available channels and PAN IDs discovery. It allows the coordinator to avoid starting on channels with high energy levels or existing PAN ID.

The energy scan is passive, while the PAN ID discovery process is active. That means that during energy scan, the device only listens and measures the power levels. During the discovery, the coordinator sends a single-hop broadcast beacon request on all channels chosen by the energy scan. Any nearby routers will respond to the request by sending a beacon frame, containing source PAN ID and whether or not the device allows joining.

Possibly, a passive PAN ID scan could be performed by listening for a beacon packet on every channel for a defined period, but that would prolong the PAN creation duration.

After finishing the scans, the coordinator attempts to start on an unused PAN ID and channel. Once chosen, both the PAN ID and channel parameters are saved and kept even after reset. The 16-bit network address of the coordinator is always 0x0000.

User can not define the coordinator’s operating channel directly, but can select which channels to scan for energy levels by the SC command. PAN ID can be chosen as desired.

4.3.3.3 Router

A router device can only join an existing ZigBee network. For discovering the existing networks, routers use the active PAN ID scan. If the ID parameter is set to 0xFFFF, they will attempt joining any available network. Otherwise, the device will only join a network that has PAN ID equal to the ID parameter.

When a response to the beacon request is received from a device that belongs to the desired network and that allows joining, the router sends an association request to this device. The device will then send back an association response, indicating the router was allowed onto the network. It also receives its 16-bit network address.

When in network, the router first sends a broadcast discovery frame to retrieve the coordinator’s 64-bit address.

After that, the router can participate in routing data. If the NJ parameter is set to 0xFF, router will allow other devices to join the network using itself as the connection point. Values from 0x01 to 0x40 will allow devices to join this router for that number of seconds. Zero value denies other devices joining the router. The parameters of the network are stored in the router and retained through reset.

4.3.4 Firmware AT

The AT firmware operates the device in a similar way to modems. It offers fewer features than API firmware and has other disadvantages, but is much simpler to use for communicating. Basically, all data received at UART are packetized and sent to the radio, and vice-versa – the device acts as a replacement of a serial cable.

Communication through modules with AT firmware is always set to either a fixed, single target (means you can’t change destination on the fly or use multiple destinations) or a broadcast.

Because of the following reason: For each module to communicate, a target 64-bit and 16-bit address has to be set at the module. A change of these addresses, however, can be only done at the command mode. That means rapid changes of transmit destinations are not possible, because there is a total over 2 seconds of silence requirement before entering command mode.

AT firmware also does not provide sending option to remote device’s configuration. This is because the firmware does not handle UART data as packets, so there is no way to inform the device that the current data on UART really represent a remote command.

However, the device still can receive remote commands, thus it is possible to remotely configure an AT firmware running device by a remote device with API firmware.

So, the only way to address multiple devices is using broadcast address at the transmitter. This might, however, lead to other issues regarding the broadcast policy in ZigBee networks.

To conclude, AT firmware is crucial for applications where every device is always communicating only with a single target. These might be mainly point to point links, such as remote controls for TV or RC-Models, or any other applications where Xbee can act as a replacement of serial cable. But in addition to that, it is possible to use it for sensors that are reporting their status to a single master.

Some applications for the broadcast-only network could be also found – e.g. networks with distributed leadership, where any device can send command to any other device.

Many difficulties with end devices can also be solved using AT firmware running end devices and API firmware running backbone. This would create the possibility of remote configuration of the end devices which consequently means an extended robustness in network addressing while keeping the software of end devices simple.

4.3.4.1 Operation Modes

In AT mode, the device is in the idle mode after power up by default. The following states transition is represented in figure 4.1.

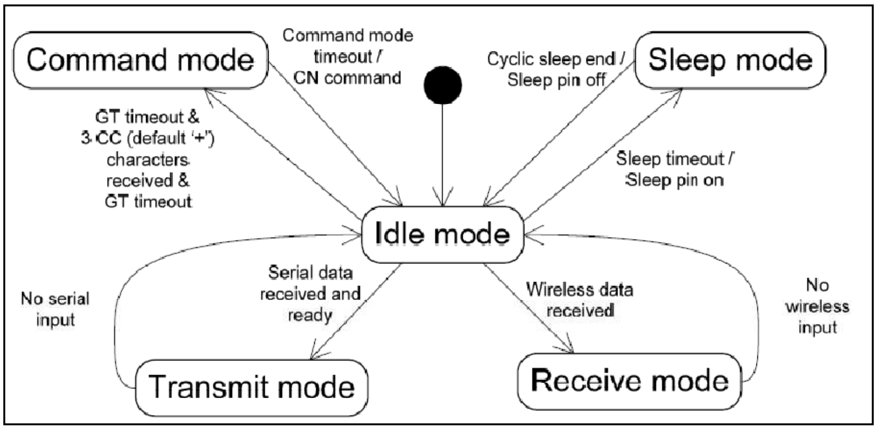

Command mode
Command mode timeout / CN command
Cyclic sleep end / Sleep pin off
Sleep mode
GT timeout & 3 CC (default '+') characters received & GT timeout
Sleep timeout / Sleep pin on
Idle mode
No serial input
Serial data received and ready
Wireless data received
No wireless input
Transmit mode
Receive mode

Figure 4.1: AT Firmware modes state chart

a) Idle mode

In idle mode, the device is waiting for either RF or serial input. If the device is an end device and no data is received for the sleep timeout period (the ST parameter) or a sleep pin is asserted (high), the device will enter the idle mode.

b) Transmit mode

When the device enters transmit mode, data is already prepared for sending. The system Checks for existence of route to the target and performs a network address discovery if needed prior to transmitting data. If any of the previous actions fails, data is silently discarded.

c) Receive Mode

Data received on the radio are buffered and sent out thru UART.

d) Command mode

In command mode, command can be issued to configure the device, retrieve the device configuration or run a network discovery.

List of AT commands is included in the Xbee manual [37]

To access command mode, there has to be one second of silence (no data input on UART), then sequence of three plus (by default) characters (‘+++’) has to be sent within one second, and then another second of silence is required. On entering the command mode, the module sends ‘OK\r’ to UART.

d) Command syntax

The syntax is well illustrated in figure 4.2.

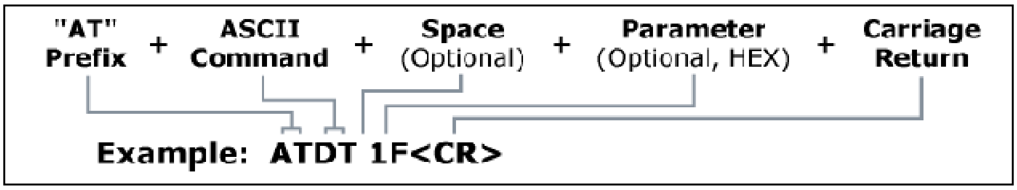


Figure 4.2: AT command structure in AT mode

Every command has to start with the “AT” characters. The following two ASCII characters represent the AT command, so for e.g. reading the node ID string, the command would be “ATNI”. Every command needs to be confirmed by sending the “\r” (carriage return, hex 0x0D – that means not “\n” - line feed, hex 0x0A) character.

So the complete command for retrieving the node ID is “ATNI\r”. Upon receiving the command, the XBee device will respond by sending the node ID followed by ‘\r’. To command without a value to return, the device will respond ‘OK\r’.

The commands can also have parameters. Parameter is any character(s) sent after the first four characters. They do not need to be separated by a space. The length of the parameter is command specific, but is always interpreted as a series of hexadecimal numbers.

Multiple commands can be chained into a single command. The separator of the commands is a comma (‘,’, hex 0x2C) character. Not all of the commands can be chained.

4.3.4.2 Configuring a coordinator

In the following part, there are acronyms, AT commands and ZigBee/IEEE 802.15.4 terminology (see chapter 4 for reference).

By default, the PAN ID (ATID) is 0x0234 and all channels except 0 and 15 are enabled for scanning (SC=0x1FFE = 0111111111111110b). Therefore, when a coordinator is started for the first time, it chooses its operating channel from 1 to 14 and PAN ID 0x234.

Its destination address is 0xFFFE. That means that all data are broadcasted. (one should be advised of the broadcasting policy in ZigBee networks and its consequences).

Some of the parameters you might want to change are given in table 4.2.

Parameter	Default	Meaning
DH / DL	0xFFFF	The destination address of all data received on UART
NI		The node identifier (“master” coordinator,…)
DD	0	Custom device type identifier
PL	4	Power level – by default full power
PW	1	Power boost – by default boosted
BD	3	Baud rate of UART – 9600 by default.
SP	0x20	Sleep period – in this case means how long coordinator should store data for its sleeping children.

Table 4.2: Suggested changes in parameters for a coordinator

4.3.4.3 Configuring a router

When router is first started, it performs a PAN ID scan using channels defined by SC. If a network matching its PAN ID is found, it attempts to join it. From that point on, the only way to change the channel or PAN ID is:

- Change the ID parameters
- Change the IC parameters
- Submit NR command

The only way to remotely invoke the network parameters reset is to send a “NR 1” command. This will cause all the devices, in broadcast range of the node the command was issued on, to reset their network parameters. This means there is no hardware way to reset the network parameters of an Xbee device.

The other parameters you might want to change are similar to those of coordinator.

4.3.4.4 Configuring an End device

To create an end device, the SM parameter has to be changed to a non-zero value, i e:

- to hibernate on PIN
- to doze on PIN
- to use cyclic sleep
- to use cyclic sleep with PIN wakeup (only for ZNet 2.5 devices)

This will cause the device to quit the network (if it was joined as a router) and rejoin as an end device. The joining procedure and network parameters are the same for a router.
Here some parameters that you might want to change as illustrated in table 4.3:

Parameter	Default	Meaning
ST	0x1388	The delay (in ms) between the last data is received and entering sleep mode
SN	1	Number of sleep periods for which a wakeup is not sent to an external device (the On/Sleep – PIN 13)
P	0x20	The length (in 10ms) (quarter second resolution) of a sleep period

Table 4.3: Suggested changes in parameters for an end device

Note that when in sleep mode, the module is unresponsive to both RF and UART data. Thus, command mode cannot be entered.

4.3.4.5 Network configuration and basic tasks

i) Point to point (peer to peer) link

By default, all the devices are flashed with an ‘End Device/Router AT’ firmware. To create a basic peer to peer link, one coordinator and one Router / End device is required. So, prior to any configuration, one of the devices has to be flashed with the coordinator firmware**.**

The basic configuration of respective devices is described in points 4.3.4.2 and 4.3.4.2.

For a point to point link, the coordinator (master) should be connected to a PC that will provide data processing and power. The other device (slave) can be either a router or an end device, based on the requirements on responsiveness, duty cycle and power saving.

For better orientation in which device you can run:

- “ATNI” master\rATWR\r” on the master device
- “ATNI” slave\ r ATWR\r on the slave device

To set up the coordinator you need to acquire the 64-bit address of the slave. The address can be retrieved by

- Looking at the bottom side of the XBee module (it is the number next to the 2D barcode
- Issuing ND command on the coordinator(the second number)
- Issuing DN command on the coordinator with the parameter “slave”, if you have run the ATNI, this will set the address automatically

For the first two points, you have to run the DH and DL commands with the address parameters to set the destination address.

Then repeat the same procedure with the slave device. If you use the third point, you have to type “master” instead of “slave”.

From now on, any data received on the UART of master are transmitted to slave and vice versa. To be sure, you should try if the data is really received, by e.g. using HyperTerminal, Putty, and X-CTU (terminal tab).

ii) Network

To create a network, the initial procedure is the same as described in point to point link. You have to start with coordinator and another device. Then you can repeat the ‘slave device’ configuration for each new node of your network. You only have to change the destination addresses to fit your needs.

However, creating a network using AT firmware might lack the configurability and dynamics of an API network, as described in the introduction of this chapter (4).

4.3.5 Firmware API

This is the more sophisticated option of communication with the modules. The API firmware uses a structured packets interface to communicate via UART.

All the packets have a common base structure – starting with delimiter, followed by two bytes of length, API frame type identifier, n bytes of frame-specific data and a checksum byte. The hierarchy of packet structure is shown in appendices.

There are two modes of API operation – standard and escaped. In standard version, all data is sent as it is.

In escaped mode, all special bytes except the frame start delimiter have to be escaped. That means 0x7E has to be escaped only when occurring in data, and then 0x7D (escape), 0x11(XON), 0x13 (XOFF) have to be escaped as well. Escaping is done by inserting 0x7D in front of the escaped byte and XOR-ing it with 0x20.

Length and checksum in escaped mode is calculated as if the data were sent non-escaped. That means escape characters are not included in the length, and the checksum is calculated from the non-escaped data without escape characters. There are several types of API frames. Their complete list is available in the XBee Manual.

4.3.5.1 Remote configuration

Any device running Xbee API firmware is capable of configuring any remote device within its current PAN – even remote devices running AT firmware. It is done by sending a "Remote AT command request". The command is executed immediately. If acknowledge is requested, the remote node will reply with 'Remote Command Response' frame. Remote commands can be broadcasted to configure all devices within broadcast range.

4.3.5.2 Data transfers

Data transfers can be done by either network layer addressing (16-bit address) or application layer addressing (endpoints and clusters).

The "ZigBee Transmit Request" frame is used for network layer addressing. It resulting action is the same as if an "Explicit Addressing ZigBee Command Frame" was issued with source and destination endpoint 0xE8 (data endpoint) and cluster ID 0x11 (transparent).

Receiving device will send an appropriate receive frame when receiving data out of its UART.

If loopback testing is required, data should be sent using with 0xE8 as endpoints and 0x12 for cluster ID. The loopback target device does not need to be specially configured; it will return any packet with Cluster ID 0x12 to that packet source.

4.3.5.3 Data sample requests

Both ZigBee and ZNet firmwares are able to collect data from their analog / digital inputs and send them via air on-demand. For this purpose, 'ZigBee IO Data Sample Rx Indicator' and 'XBee Sensor Read Indicator' frames are used. The request is issued by sending remote AT command "S".

ZNet 2.5 firmware devices are capable of periodical data samples sending or sending on change. See Xbee Manual [37] for details.

4.3.6 Considerations

4.3.6.1 Networks discovery

Unlike Wi-Fi, the XBee modules are not able to obtain the network discovery results, except for matching PAN ID which allows joining. However, the procedure of finding the proper PAN ID consist also of all nearby networks discovery, thus it should be possible to get the list of networks by modifying the firmware.

4.3.6.2 Transmissions to sleeping devices

When using cyclic sleep, following precautions have to be considered and carried out:

- The SP parameter has to be the same for sender, the sleeping device’s parent and the sleeping device. This is because end device sleeps for the time defined in SP, while routers store the packet for the time defined by own SP and then discard it.

  Possibly, the scenario where the SP parameter is shortest for the end device and longer for other devices should also work.
- Only a small data portion should be sent. The data which destination is sleeping device are stored in the sleeping device’s parent data queue. The size of this queue is, however, limited. Therefore when a router with full queue receives a packet for its sleeping child, the packet will not be delivered, and in AT mode silently discarded.

  Advanced sleep solution is required when using PIN sleep, as a device in pin sleep does receive neither serial nor RF data.

4.3.6.3 Maximum Children Issue

Contrary to other manufacturer’s modules, XBee routers only allow 8 end devices to join a single parent. This can pose a large disadvantage for networks with lower numbers of backbone routers. It also lowers the self-healing capabilities of the network.

4.3.6.4 Enciphering

The standard firmware used in the devices (ZigBee AT/API firmware) does not provide any enciphering methods. However, for these purposes the ZNet 2.5 can be used, which is interoperable to the ZigBee firmware and provides both enciphering and periodical data sampling features.

4.3.6.5 Maximum range

I have made several attempts to determine the maximum achievable outdoor range of the XBee modules. In my measurements, I used two modules with whip antennas, both running XBee ZNet 2.5 Router / End device firmware. One was configured as a router, and placed on a balcony in the height of about 20 m. The other one was sleep-enabled end device (sleep period 2 seconds). Both devices were set to Boost Enabled mode with Power Level 4 (total 2mW (+3dB power). No other devices were present in the network.

The range was measured upon the end device’s associate status – if the associate LED reported the device as orphaned, it meant Signal is lost. If the module was able to reassociate with the network, the current range was marked as achieved.

The longest distance I achieved this way is 360 m. It was in line-of-sight connection ground-to-balcony (18 m height difference). The Fresnel zone [38], however, was not entirely clear.

The condition of our whip antenna modules should be considered – refer to appendices for details and module photos.

4.3.6.6 Broadcasts

According to manual, XBee devices only retain 8 entries in their broadcast tables. Each of these entries last for 8 seconds maximum. XBee manual only says that ‘Since broadcast transmissions are retransmitted by each device in the network, broadcast messages should be used sparingly.

The observed result of using more than 8 broadcasts within 8 seconds is that 8 packets are transmitted and propagated correctly. But every new packet over 8 within the 8 seconds has to be buffered and is only sent when one of the entries in broadcast table is available.

4.3.7 Bugs

4.3.7.1 Command mode time out

When a device is running

- ZIGBEE COORDINATOR AT, firmware version 1020
- ZIGBEE ROUTER/END DEVICE AT, firmware version 1220

The maximum value of Command Mode Timeout parameter (CT) is 0x147, not 0x28F. If the value is set above 0x147, XBee quits the command mode immediately after entering it, so it is impossible to enter any command at all (so do not change CT value by something else).

4.3.7.2 Node discovery response

Despite of what is written in the manual, the ZigBee AT firmware devices return the network discovery message with SH and SL on the same line (not separated by a CR). Manual says SH and SL should be returned on separate lines. In ZNet firmware the message is returned as stated in the manual.

Chapter 5

Design process

Contents:

This work is concerned with the design a wireless network system based mainly on five nodes (may be extended to several nodes). A mesh network option is considered and standard node architecture is implemented. Recall that all WSN development platforms are relatively similar, and mainly consist of five major components as depicted in Figure 5.1.

5.1 Node architecture

A node comprises the following parts:

- Processor – The brain of the sensor node
- Wireless Communication unit–the electronic to ensure wireless link between sensor nodes
- Sensor Interface–this unit guarantees the technological Interface with sensors and other devices
- Power Supply – Power source of the sensor node.
- Operating System – Software for managing the network and resources

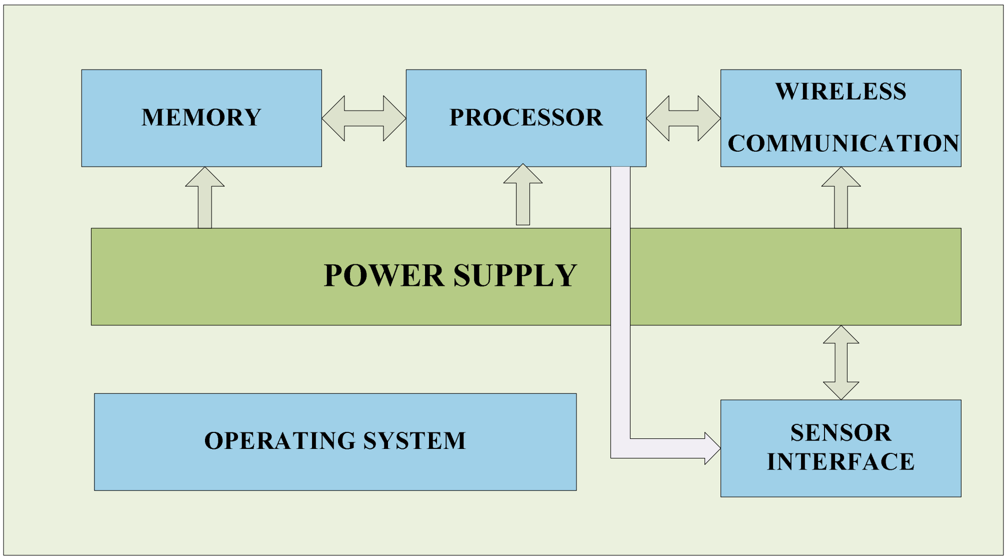


Figure 5.1: Common components of WSN nodes

5.1.1 Processor

The processor is the heart of the node. It samples the sensor data and handles the radio communication. The PIC 18LF452 from MICROCHIP is chosen for its power save functions and its large memory. It is an 8-bit microcontroller designed for embedded applications. It can deliver up to 10MIPS (*Million Instructions per Second*). It is equipped with a relatively large programmable flash memory (2Megabytes), 8-channels of 10-bit ADCs (*Analogue-to-Digital Converters*) and low operating voltage (2.7V). The synchronous serial port can be configured as either 3-wire Serial Peripheral Interface (SPI™) or 2-wire Inter-Integrated Circuit (I²C™) bus and Addressable Universal Asynchronous Receiver Transmitter (AUSART).

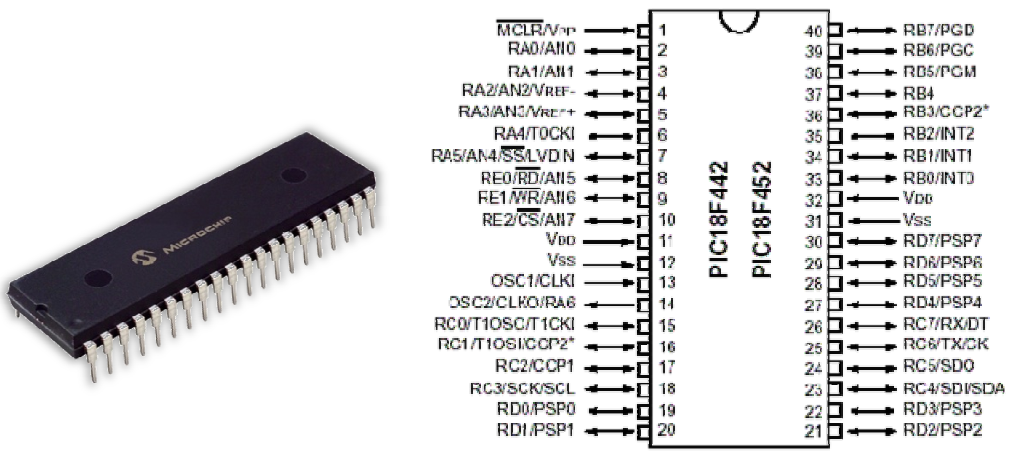


Figure 5.2: PIC18F4520 pin diagram

5.1.2 Radio transceiver

This low power radio transceiver of small size and hybrid design permits bidirectional transmission and reception. Two versions are available from MaxStream: XBee and XBee PRO. Both versions are functionally identical and pin compatible .The only difference is the transmit power, which is 1 mW maximum for the Xbee and 63 mW maximum for the Xbee PRO.

Of course, transmit power is an important factor because the range of the ultimate product depends on it, but it is certainly not the only thing you have to take into account.

Another consideration that is at least as important is that higher transmit power means higher current consumption. The Xbee included an ember 8-bit microcontroller with several antenna options integrated in a small package at a relatively low cost. The XBee device had the following specifications:

- A transmission range of up to 100 meters
- A maximum data rate of 250 kbps
- Receiver Sensitivity of -92 dBm
- Low Power Requirement (2.8- 3.4 V)
- Very small dimensions
- A ZigBee device (mesh-networking capable)
- Low Cost

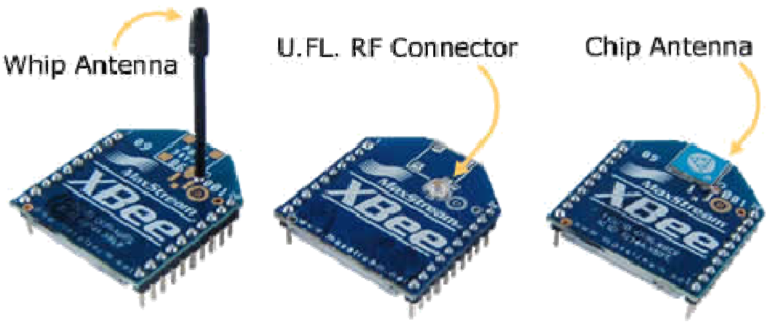
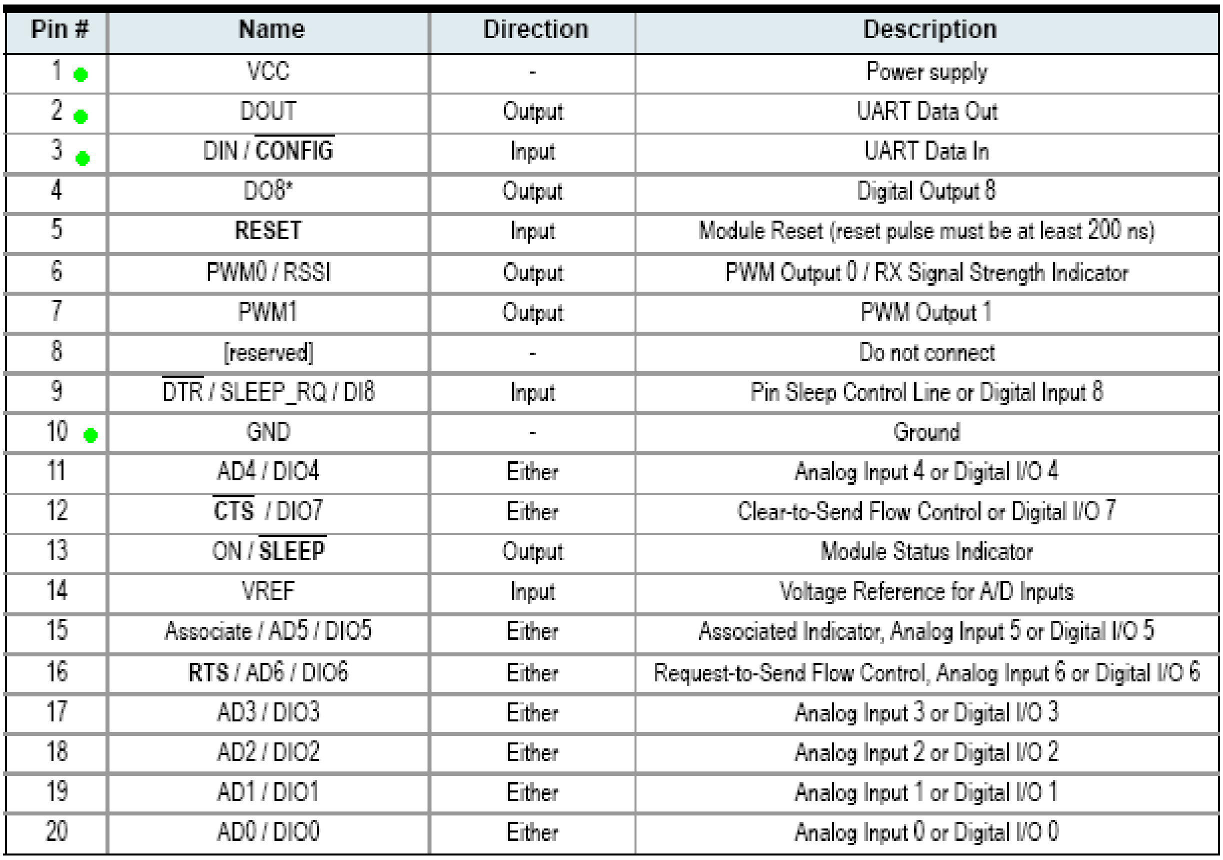


Figure 5.3: XBee in different configurations



Pin #	Name	Direction	Description
1 ●	VCC	-	Power supply
2 ●	DOUT	Output	UART Data Out
3 ●	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10 ●	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

Figure 5.4: The pin layout for the MaxStream Xbee module

5.1.3 Memory

In this project, the EEPROM of MCU PIC 18LF452 is used. Due to the non-volatile nature of the EPROM, it is used in most embedded systems for storing configuration information because it does not require power to retain the stored data. It is also used as an immediate storage for sensor readings. For instance, in order to perform feature extraction or filtering of the sampled data, the EEPROM can be used as a processing buffer for these algorithms.

5.1.4 Sensor interface

WSN platforms are equipped with ADC interfaces for data sampling and acquisition., the PIC 18LF452 MCU has an eight-channel 10-bit ADC that can sample at a rate up to 15.4ksps (*kilo-samples per second*). In my thesis I used two types of sensors: LM35DZ temperature sensor and MPX2200A pressure sensor.

5.1.4.1 LM35DZ

It is a precision integrated circuit with an output voltage that is linearly proportional to the Centigrade temperature. The advantage of LM35DZ is that it is not necessary to subtract a large constant voltage from the output to obtain the Centigrade scale. The chip does not require any external calibration to achieve accuracies of ±4/10 °C at room temperature and ±8/10°C over full temperature range. The device draws only 56µA from voltage supplies in the range 4V to 30V so it has very low self-heating <0.1°C in still air. The LM35DZ operates in the range 0 C to 100 C.

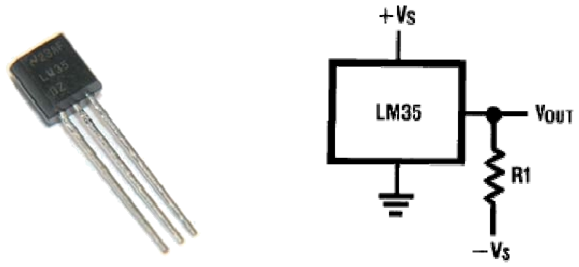


Figure 5.5: Temperature sensor (LM35DZ)

5.1.4.2 MPX 2200

The MPX2200 series device is a silicon piezoresistive pressure sensor providing a highly accurate and linear voltage output directly proportional to the applied pressure. The sensor is a single monolithic silicon diaphragm with the strain gauge and a thin–film resistor network integrated on–chip. The chip is laser trimmed for precise span and offset calibration and temperature compensation (figurr5.6).They are designed for use in applications such as pump/motor controllers, robotics, level indicators, medical diagnostics, pressure switching, barometers, altimeters, etc.

Figure 5.6: Pressure sensor

5.1.5 Power supply

Power supply is the main determining factor for the size and lifetime of the WSN hardware. The battery or alternative power source is often the largest single component of WSN nodes. Other power sources, such as the scavenging of power from temperature gradients or movement have been proposed. Due to the relatively high power requirement for radio transmission, batteries still remain the main source of power for current WSN platforms. Among different battery technologies, Li-ion battery is the most popular choice for WSN hardware because of its high power density. Although zinc-air batteries have a higher energy capacity than that of Li-ion batteries, the high rate of power drains from the current radio transceivers limits the direct use of zinc-air battery for WSN. In my nodes I used the of alkaline batteries

5.2 Node board design

CAD program Proteus VSM Co-Simulation Software from Labcentre Electronic is used to design our node board. This CAD Combines the ISIS schematic capture and ARES PCB layout programs to provide a powerful, integrated and easy to use suite of tools for professional PCB. Proteus VSM Co-Simulation provides schematics and layout designs in the same program environment (figure 5.7). The design process is basically carried out in two steps. Firstly, the schematic is designed with ISIS. Secondly, the schematic is transferred to a layout using ARES. Note that it is possible to manufacture Printed Circuit Boards using the layout.

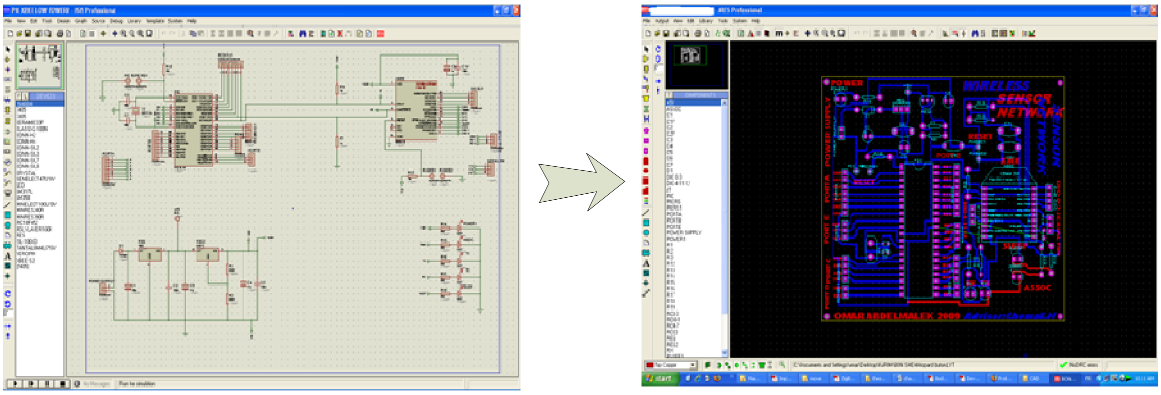


Figure 5.7: Proteus VSM Co-Simulation software screen

The main task of the project is to develop fully functional WSN modules. It was required to develop five different modules. One dedicated module to contain a Radio Frequency part (RF) and should be connected to a PC and the others constitute real nodes which comprise Radio Frequency part (RF), a Microcontroller part (MCU) and power unit as illustrated in figure 5.8.

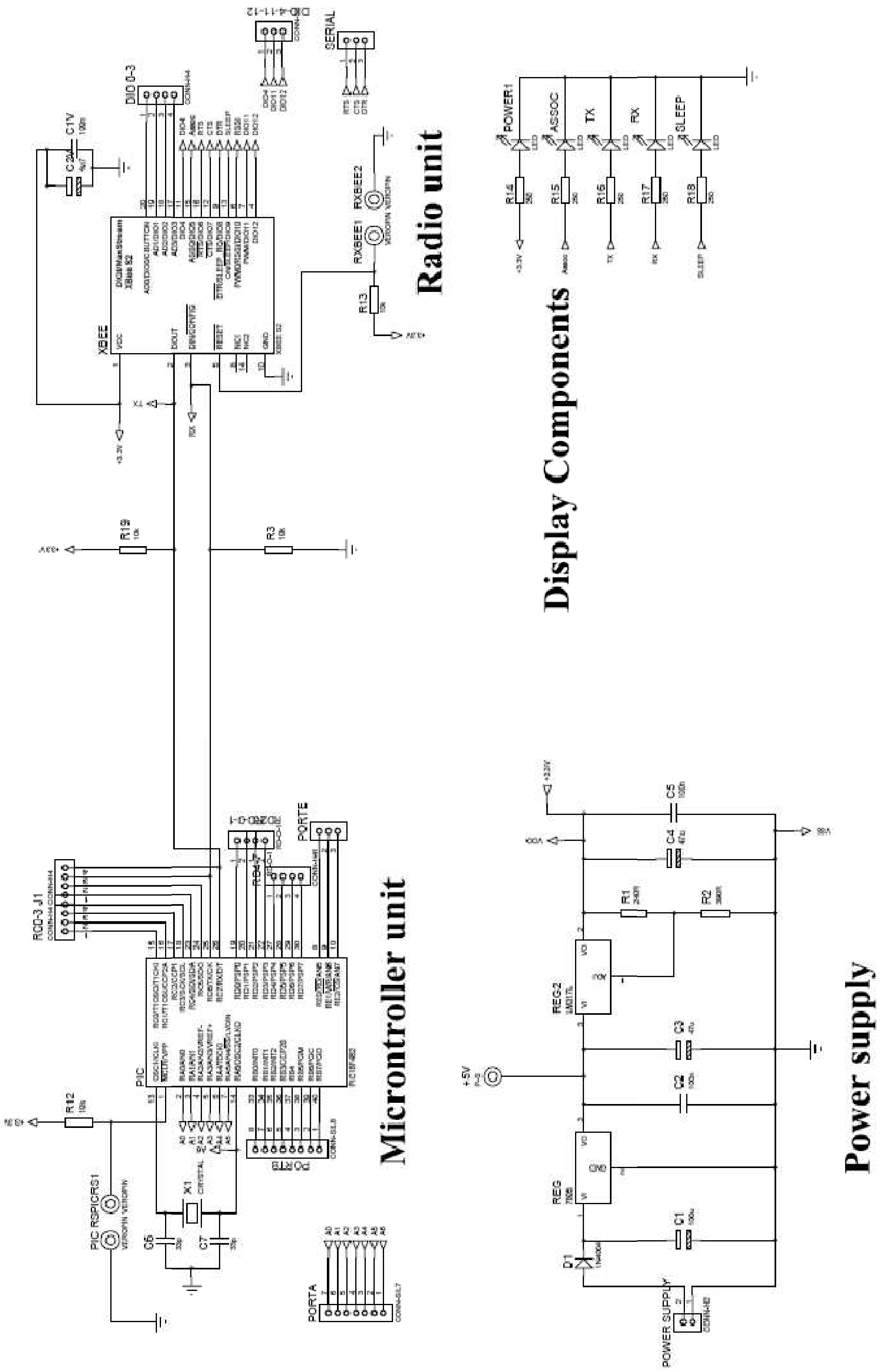


Figure 5.8: Schematic of node

5.2.1Power unit

A classical power unit is used to adjust voltages as illustrated in figure5.9. It serves to adjust tow particular voltage ranges. The first deals with 9V to 5V voltage adjustment, the second covers reduction of voltage 5V to 3.3V. Two regulators are used 7805 and LM317. A resistive voltage divider (R10/R12) ensures the right 3.3V Xbee power supply. Capacitive decoupling and filtering means improve circuit dynamic functionality.

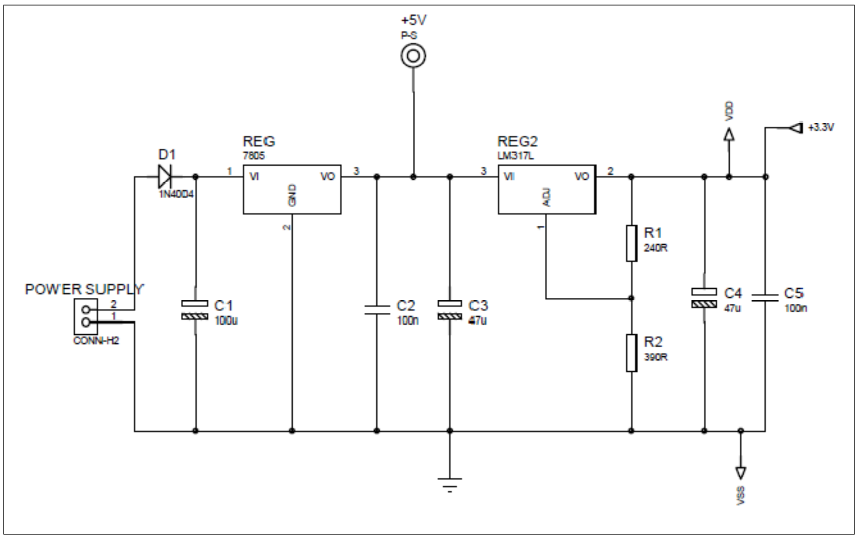


Figure 5.9: Power unit

5.2.2 Microntroller unit

This unit consists of a PIC18LF452, an 8 MHz quartz oscillator, 3.3V power supply, reset and port connectors as shown in figure 5.10.

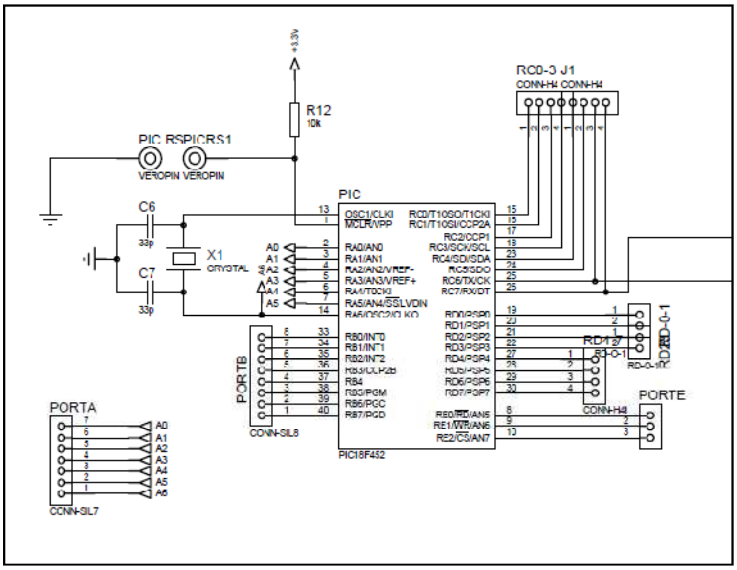


Figure 5.10: Microntroller unit

5.2.3 Radio unit

It consists of an XBee module, decoupling and filtering capacitors. The interface of Xbee transceiver with the PIC18LF452 microcontroller is achieved by connecting the right transmit and receive pins accordingly. Some considerations should be taken into account to accommodate the needs of the Xbee. The XBee package is a particular standard with a 2.0 mm inter pin distance therefore does not in general purpose PC Boards sockets.

The PIC18LF452 and the XBee communicate easily since both use serial UART interface mode. Figure 5.11 shows Xbee communicating pins: Din (3), Dout (2), VDD (1) and GND (10). Pin 2 from the XBee needs to be connected to the Rx (26) pin of the microcontroller. Pin 3 from the XBee needs to be connected to the Tx (27) pin of the microcontroller. Recall that the PIC18LF452 microcontroller should be configured to transmit and receive serial data.

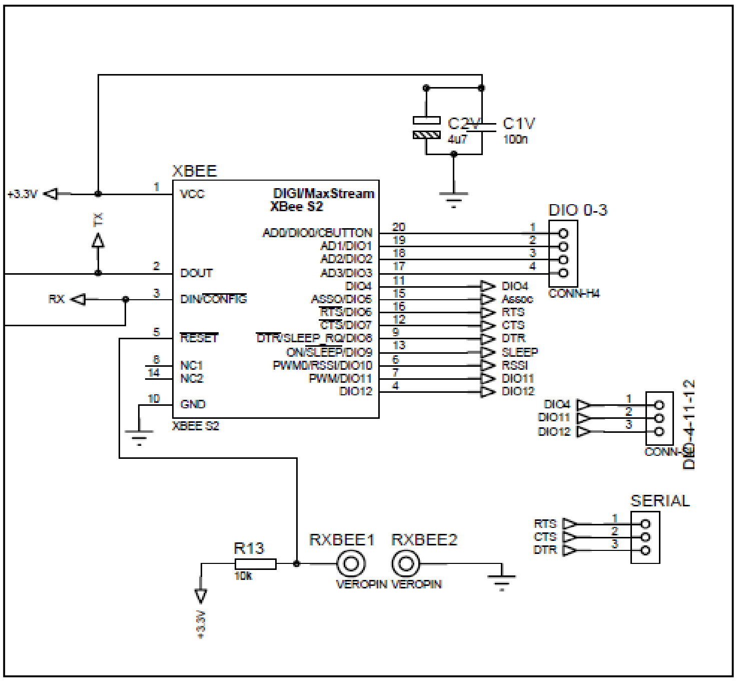


Figure 5.11: Radio unit

5.2.4 Sensor interfaces

5.2.4.1 Temperature sensor interface

Since the voltage sensor is very low it should be increased to meet design interface. The developed circuit is based on the LM324 amplifier. In order to get rapid functioning, calibrating means are added and consist of connecting through a jumper the circuit reference (D1diode and R3) to get predefined output (figure5.12).

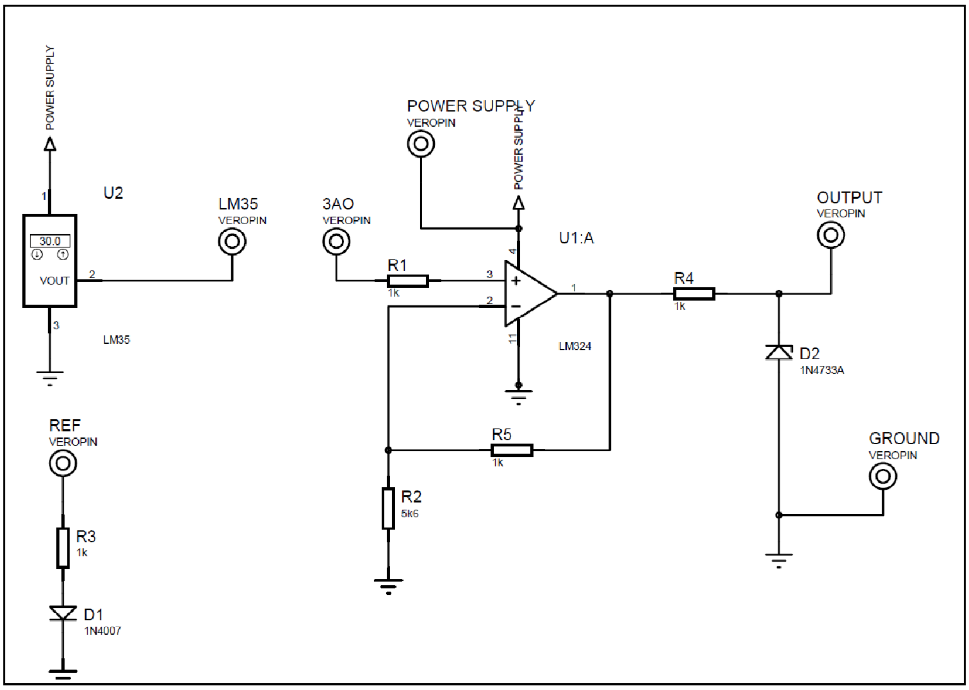


Figure 5.12: Temperature sensor interface

5.2.4.2 Pressure sensor interface

This circuit consists of a differential amplifier and a power-adapter amplifier. The input signal is firstly acquired through an LM324 stage and then amplified with a second LM324 stage. The OP AMP circuits are parts of the 4 that counts a LM324 IC package. The output voltage range is 3V (figure 5.13).

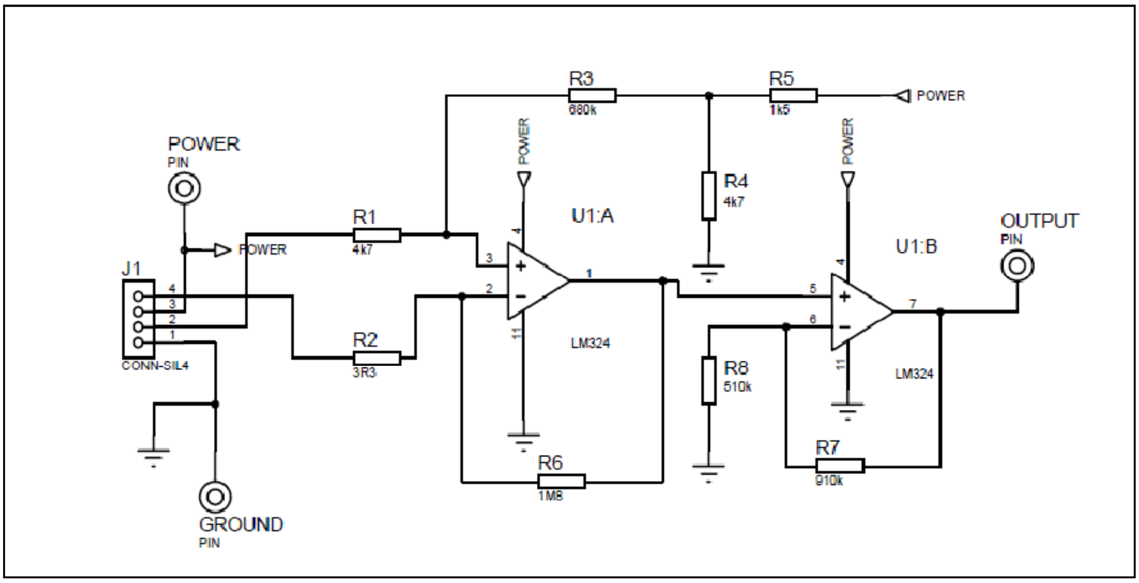


Figure 5.13: Pressure sensor interface

5.2.5 Display components

For testing and debugging, the board is equipped with multi-color Light Emitting Diodes (LED).The states of the following signals are displayed: power, Tx, Rx, ASSOC and Sleep mode (figure 5.14).

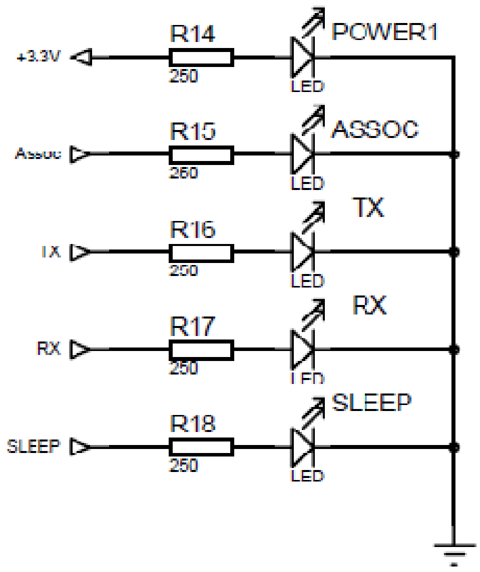


Figure 5.14: Display components

5.3 Hardware circuit Implementation

After successful simulation design of the developed system, physical implementations are the next strait forward tasks. We should emphasis that this operation goes through multiple steps.

Five nodes PCB circuits and five sensor interfaces circuits are implemented. Two PCB boards are produced.

5.3.1 Node PCB board

Each of the five nodes PCB board is produced in an 11x10 cm^2 double sided printed circuit. The width of the printed connecting lines is 2 mm. It comprises in addition to the PIC and Xbee sockets, 6 pin connectors: PIC ports (A, B, and C, D) and Xbee ports. A battery connector is also provided for easy connection. This PCB board is then mounted in a transparent Plexiglas support (figure 5.15).

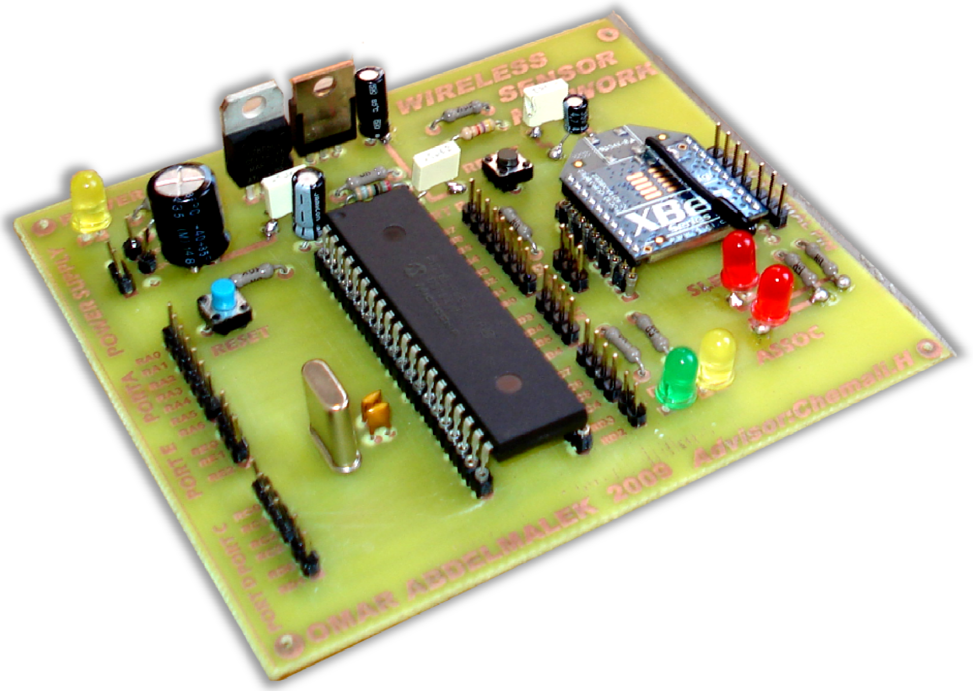


Figure 5.15: Xbee node circuit

5.3.2 Sensor PCB board

The sensor interface is implemented in a 5x5 cm^2 single sided board and comprises the pin socket sensor, battery supply pins and the sensor amplified output to connect to node board (figure 5.16).

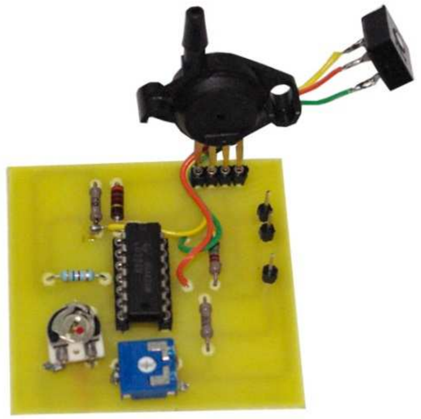
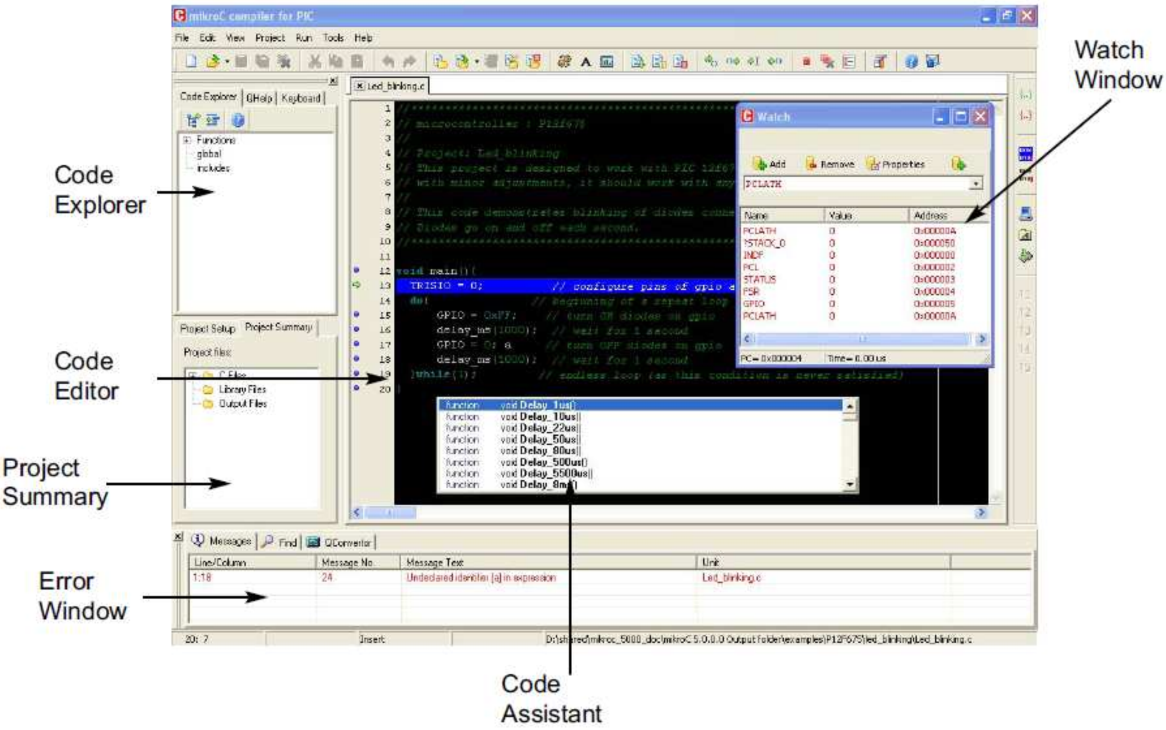


Figure 5.16: Sensor interface circuit

5.4 PIC programming

We used MikroC of MikroElektronika (web site: www.microe.com) which comes with rich resources: library functions, integrated development environment with a built-insimulator and an in-circuit debugger (e.g., mikroICD). Figure 5.17 illustrates the environment of this tool.



```
//*************************************************************
// microcontroller : P12f675
//
// Project: Led_blinking
// This project is designed to work with PIC 12f67
// with minor adjustments, it should work with any
//
// This code demonstrates blinking of diodes conne
// Diodes go on and off each second.
//*************************************************************

void main(){
  TRISIO = 0;           // configure pins of gpio a
  do{                   // beginning of a repeat loop
     GPIO = 0xFF;       // turn ON diodes on gpio
     delay_ms(1000);    // wait for 1 second
     GPIO = 0; a        // turn OFF diodes on gpio
     delay_ms(1000);    // wait for 1 second
  }while(1);            // endless loop (as this condition is never satisfied)
}
```

```
function    void Delay_1us()
function    void Delay_10us()
function    void Delay_22us()
function    void Delay_50us()
function    void Delay_80us()
function    void Delay_500us()
function    void Delay_5500us()
function    void Delay_8ms()
```

Name	Value	Address
PCLATH	0	0x00000A
?STACK_0	0	0x000050
INDF	0	0x000000
PCL	0	0x000002
STATUS	0	0x000003
FSR	0	0x000004
GPIO	0	0x000005
PCLATH	0	0x00000A

PC= 0x000004   Time= 0.00 us

Line/Column	Message No.	Message Text	Unit
1:18	24	Undeclared identifier [a] in expression	Led_blinking.c

20: 7   Insert   D:\shared\mikroc_5000_doc\mikroC 5.0.0.0 Output folder\examples\P12F675\led_blinking\Led_blinking.c

Figure 5.17: MikroC IDE screen

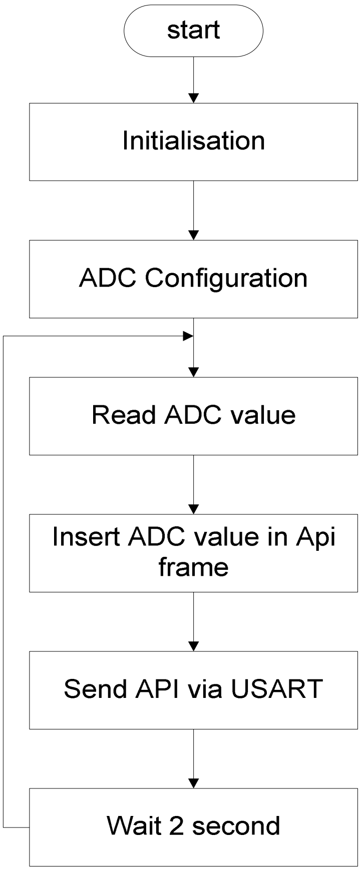


Figure 5.18: PIC Program data flow

The temperature sensor device, in the test application phase, connects itself wirelessly to the coordinator of the network. When the node device is connected to the network the MCU PIC 18LF452 is initialized. A 10-bit value from the temperature sensor is then read and sent to the coordinator device. Once the node device is in rest for about two seconds, a new value is read again from the temperature sensor, the same step using MPX220 pressure sensor. Figure 5.18 shows the complete MCU program flow chart.

5.5 Developing user interfaces

In this section, we developed programs to configure and acquire data from nodes.

By using the platform “LabVIEW”, which is designed to the data acquisition and the control system, we develop some user interfaces to communicate with XBee devices utilizing the two communication modes, AT and API.

First, an abstract which describe and summarize the most features of LabVIEW.

5.5.1 Introduction to LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical programming language that uses icons instead of lines of text to create applications. LabVIEW contains a comprehensive set of tools for acquiring, analyzing, displaying, and storing data, as well as tools to help you troubleshoot code you write.

How does LabVIEW work?

In contrast to text-based programming languages, where instructions determine the order of program execution, LabVIEW uses dataflow programming, where the flow of data through the nodes on the block diagram determines the execution order of the VIs and functions. Every VI uses functions that manipulate input from the user interface or other sources and display that information or move it to other files or other computers.

What is Virtual Instruments?

LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation imitate physical instruments, such as oscilloscopes and millimeters.

A VI within another VI is called a subVI. A subVI corresponds to a subroutine in text-based programming languages.

A VI contains the following three components:

- **Front panel** Serves as the user interface.
- **Block diagram** Contains the graphical source code that defines the functionality of the VI.
- **Icon and connector pane** Identifies the interface to the VI so that you can use the VI in another VI.

➢ **Front panel**

The **front panel** is the user interface of the VI. The figure 5.19 shows an example of a front panel (figure 5.19).

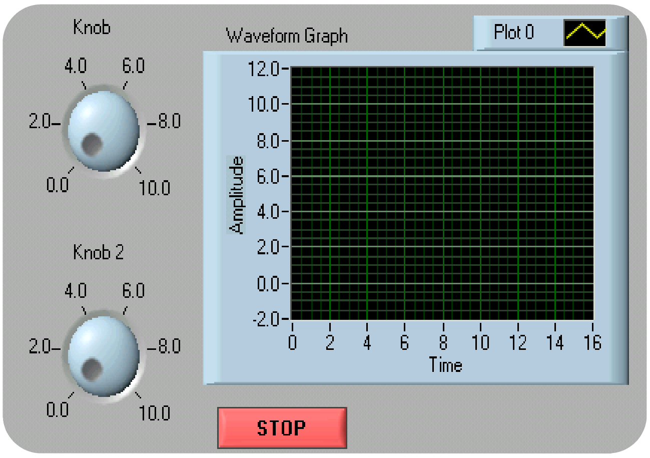


Figure 5.19: Example of front panel

You build the front panel using controls and indicators, which are the interactive input and output terminals of the VI, respectively. Controls are knobs, push buttons, dials, and other input devices. Indicators are graphs, LEDs, and other output displays. Controls simulate instrument input mechanisms and supply data to the block diagram of the VI. Indicators simulate instrument output mechanisms and display data the block diagram acquires or generates.

➢ **Block diagram**

After you build the front panel, you add code using graphical representations of functions to control the front panel objects.

The block diagram contains this graphical source code. Front panel objects appear as terminals on the block diagram. Every control or indicator on the front panel has a corresponding terminal on the block diagram. Additionally, the block diagram contains functions and structures from built-in LabVIEW VI libraries. Wires connect each of the nodes on the block diagram, including control and indicator terminals, functions, and structures (figure 5.20).

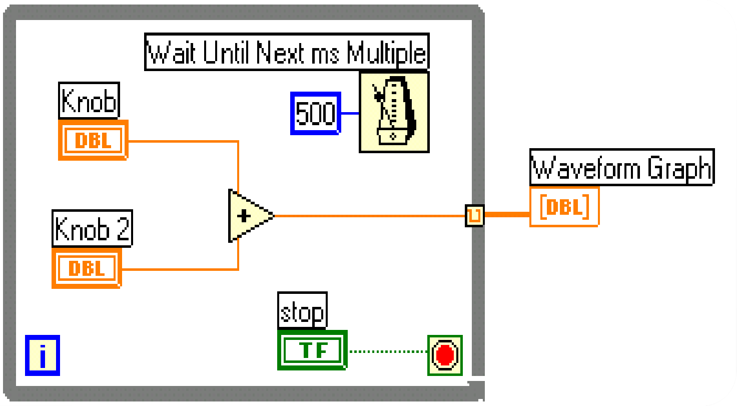


Figure 5.20: Example on block diagram

Why should use LabVIEW?

- LabVIEW empowers you to build your own solutions for scientific and engineering systems.
- LabVIEW gives you the flexibility and performance of a powerful programming language without the associated difficulty and complexity.
- LabVIEW gives thousands of successful users a faster way to program instrumentation, data acquisition, and control systems.
- By using LabVIEW to prototype, design, test, and implement your instrument systems, you can reduce system development time and increase productivity by a factor of 4 to 10.
- You can use LabVIEW to easily communicate with several hardware interfaces such as data acquisition, vision, and motion control devices, as well as GPIB, PXI, VXI, USB, Ethernet, RS232, and RS485 instruments. and plug-in data acquisition devices.
- LabVIEW provides us with an excellent set of tools for examining all sorts of DSP (digital signal processing) and digital communication topics. Its graphical nature allows us to quickly and efficiently get to the core of a communication problem, without all the overhead that generally accompanies a digital communication system
- LabVIEW also has built-in features for connecting your application to the Internet using the LabVIEW web server and software standards such as TCP/IP networking and ActiveX.
- Using LabVIEW, you can create 32-bit compiled applications that give you the fast execution speeds needed for custom data acquisition, test, measurement, and control solutions. You can set breakpoints, animate program execution, and single-step through the program to make debugging and development easier.
- LabVIEW also provides numerous mechanisms for connecting to external code or software through DLLs, shared libraries, ActiveX, and more.

- You also can create stand-alone executables and shared libraries, like DLLs, because LabVIEW is a true 32-bit compiler.

After this abstract on LabVIEW, I try to get a few details on front panel, flow chart and a functional data diagram for my user interfaces.

5.5.2 User interfaces

5.5.2 .1 AT user interface

a) AT command user interface

This user interface provides AT firmware supported in the AT mode. It supports command mode initialization, sending AT commands with parameters, and retrieving responses. The follow chart of figure 5.21 gives an idea how this interface works.

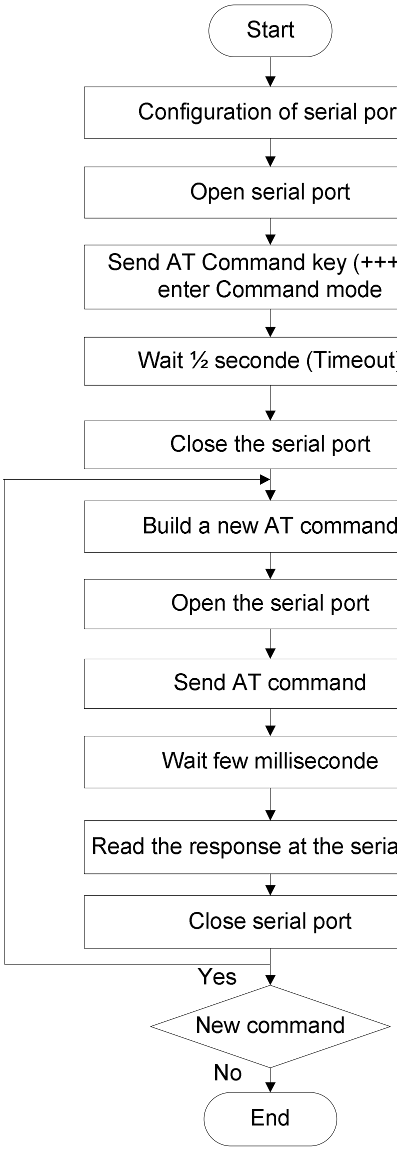


Figure 5.21: AT command flow chart

Then, the resulting bloc diagram is shown in figure 5.22

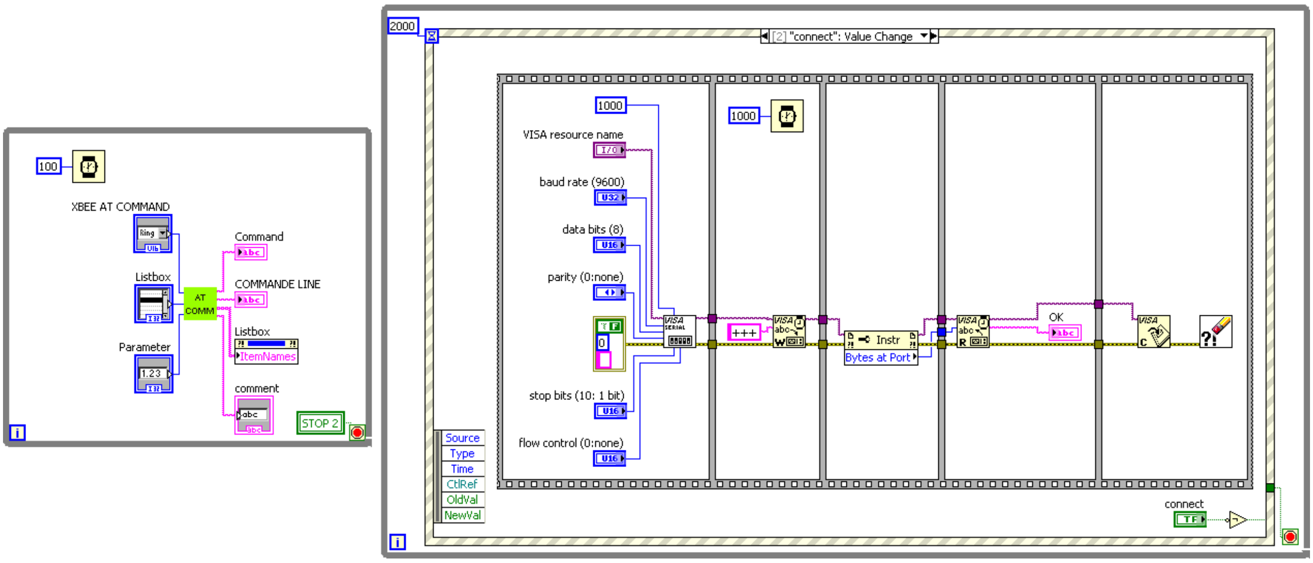
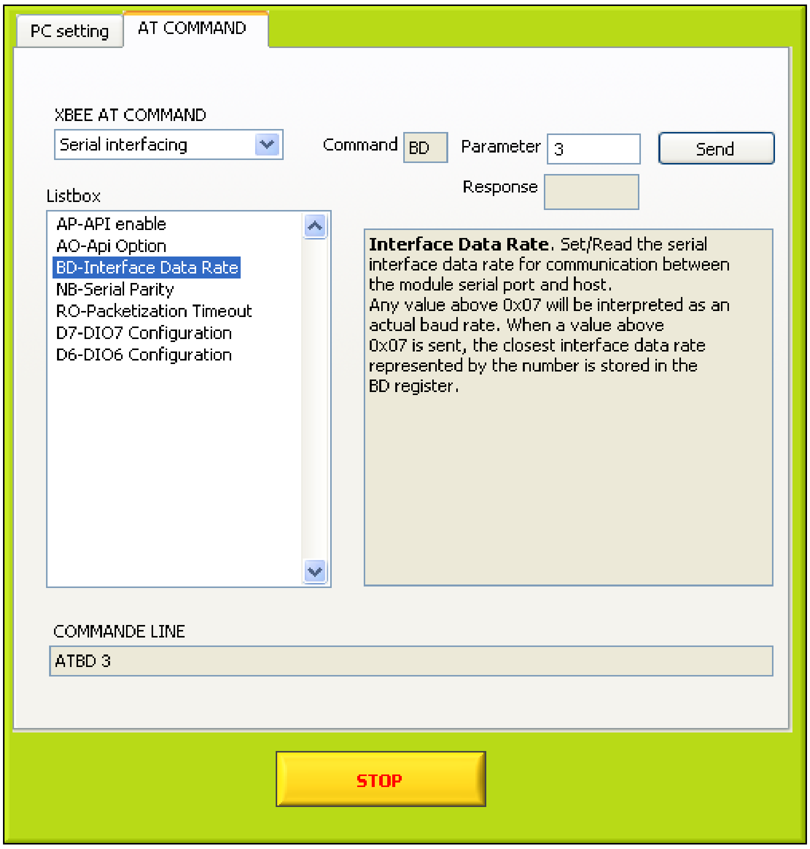


Figure 5.22: AT command bloc diagram

The corresponding front panel of figures 5.22 diagram is illustrated by figure 5.23.



XBEE AT COMMAND

Serial interfacing

Command BD

Parameter 3

Send

Response

Listbox

- AP-API enable
- AO-Api Option
- BD-Interface Data Rate
- NB-Serial Parity
- RO-Packetization Timeout
- D7-DIO7 Configuration
- D6-DIO6 Configuration

Interface Data Rate. Set/Read the serial interface data rate for communication between the module serial port and host.
Any value above 0x07 will be interpreted as an actual baud rate. When a value above 0x07 is sent, the closest interface data rate represented by the number is stored in the BD register.

COMMANDE LINE

ATBD 3

Figure 5.23: AT command front panel

b) Transferring DATA using AT mode

The AT firmware, as described in previous chapter, works in a transparent mode, where all data received are instantly transmitted over the radio.

For this case, a simple interface to communicate with Xbee is programmed as illustrated in the flowchart, and the front panel figure 5.24 and 5.25 respectively.

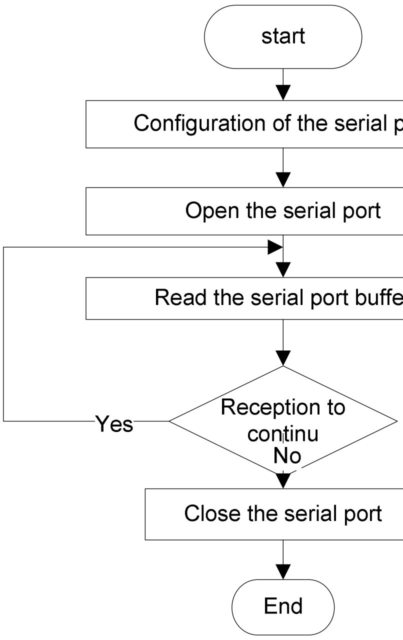
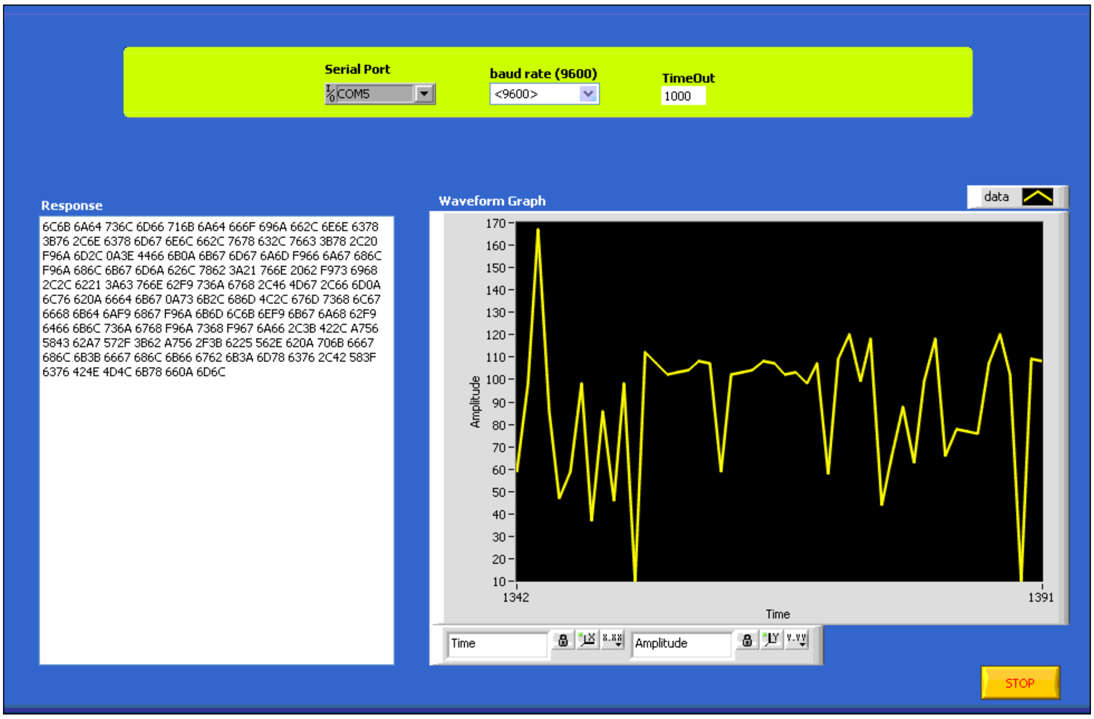


Figure 5.24: Transferring data using AT mode flowcharts



Serial Port: COM5

baud rate (9600): <9600>

TimeOut: 1000

Response

```
6C6B 6A64 736C 6D66 716B 6A64 666F 696A 662C 6E6E 6378
3B76 2C6E 6378 6D67 6E6C 662C 7678 632C 7663 3B78 2C20
F96A 6D2C 0A3E 4466 6B0A 6B67 6D67 6A6D F966 6A67 686C
F96A 686C 6B67 6D6A 626C 7862 3A21 766E 2062 F973 6968
2C2C 6221 3A63 766E 62F9 736A 6768 2C46 4D67 2C66 6D0A
6C76 620A 6664 6B67 0A73 6B2C 686D 4C2C 676D 7368 6C67
6668 6B64 6AF9 6867 F96A 6B6D 6C6B 6EF9 6B67 6A68 62F9
6466 6B6C 736A 6768 F96A 7368 F967 6A66 2C3B 422C A756
5843 62A7 572F 3B62 A756 2F3B 6225 562E 620A 706B 6667
686C 6B3B 6667 686C 6B66 6762 6B3A 6D78 6376 2C42 583F
6376 424E 4D4C 6B78 660A 6D6C
```

Figure 5.25: Transferring data using AT mode front panel

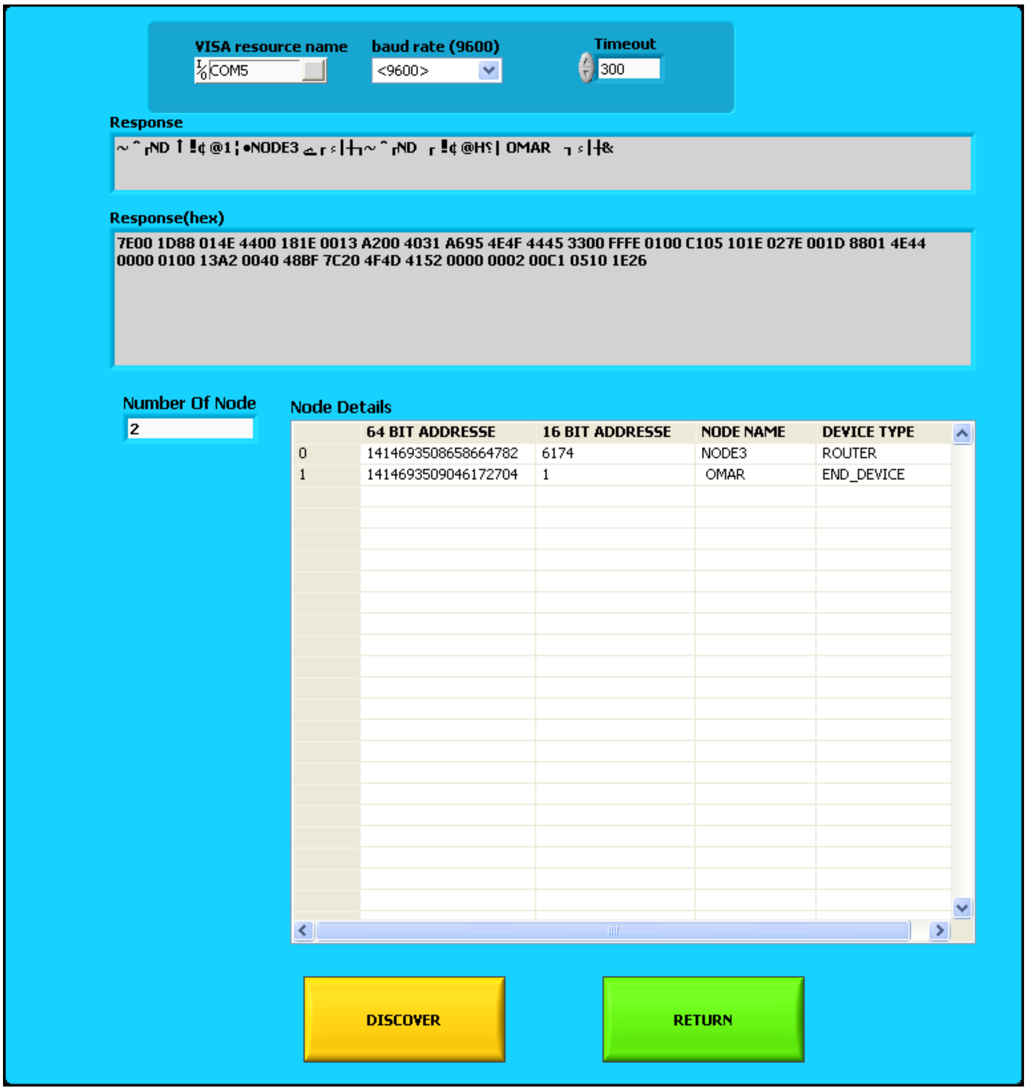
5.5.2.2 API user interface

The API support program provides a set of features enabling basic communication with devices using API firmware.

The XBee API user interfaces is an up build of a serial port, which provides functions for basic node information retrieval, local AT commands issuing, network discovery, sensor read and general API frame packet retrieval.

a) Network Discovery

For this case, a user interface, after sending a network discovery frame (type 0x08) using the coordinator we receive for each node a 64bit address, 16 bit address, Node identifier string, Device Type (Coordinator, Router and End Device), The result is shown in figure 5.26. The manager front panel is building according to 5.27 flow chart.



VISA resource name: COM5

baud rate (9600): <9600>

Timeout: 300

Response

```
~ˆ┎ND ↑‼¢ @1¦•NODE3 ⏏┎ˢ|┼┐~ˆ┎ND  ┎‼¢ @Hˀ| OMAR  ┐ˢ|┼&
```

Response(hex)

```
7E00 1D88 014E 4400 181E 0013 A200 4031 A695 4E4F 4445 3300 FFFE 0100 C105 101E 027E 001D 8801 4E44
0000 0100 13A2 0040 48BF 7C20 4F4D 4152 0000 0002 00C1 0510 1E26
```

Number Of Node: 2

Node Details

	64 BIT ADDRESSE	16 BIT ADDRESSE	NODE NAME	DEVICE TYPE
0	1414693508658664782	6174	NODE3	ROUTER
1	1414693509046172704	1	OMAR	END_DEVICE

Figure 5.26: Network discovery front panel.

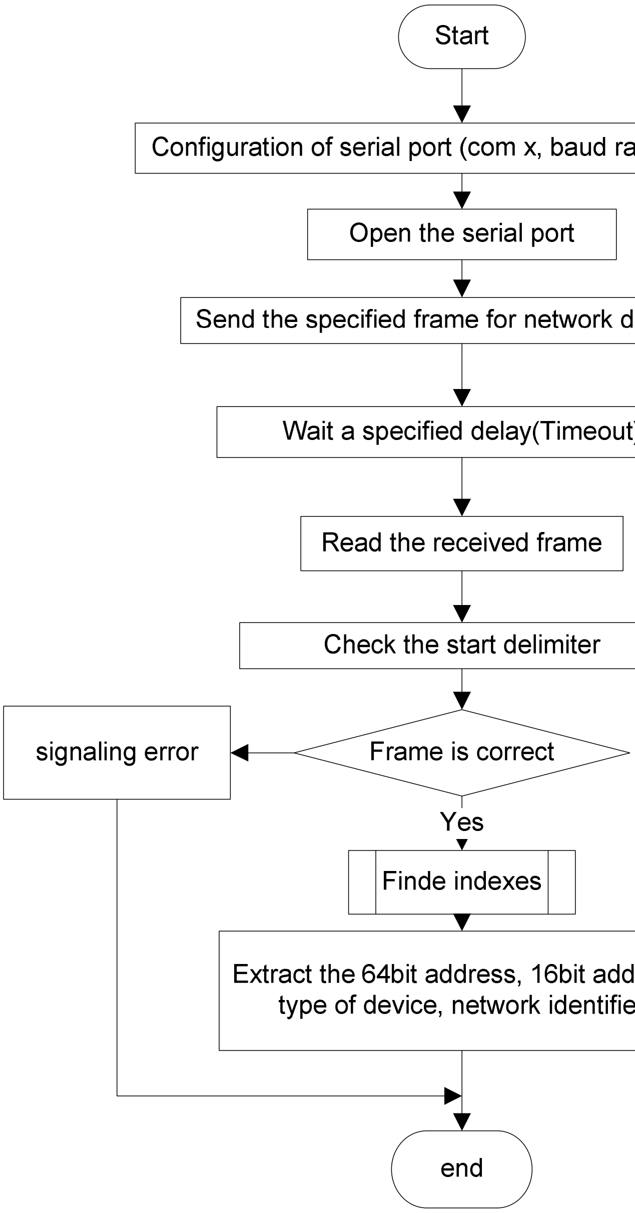

Start
Configuration of serial port (com x, baud rate,timeout..)
Open the serial port
Send the specified frame for network discovery
Wait a specified delay(Timeout)
Read the received frame
Check the start delimiter
Frame is correct
signaling error
Yes
Finde indexes
Extract the 64bit address, 16bit address, type of device, network identifier
end

Figure 5.27: Network discovery flow chart

b) Network Data monitoring

This user interface is designed for a general usage of wireless sensor network; you can send any data using either PIC MCU or Xbee. For example: we send the ADC value read from PIC ADC and we send it through the **ZigBee Transmit Request (**0x10). Then we receive it at **ZigBee Explicit Rx Indicator** (0x91). (See details in figure 5.28 and figure 5.29 and 5.30).

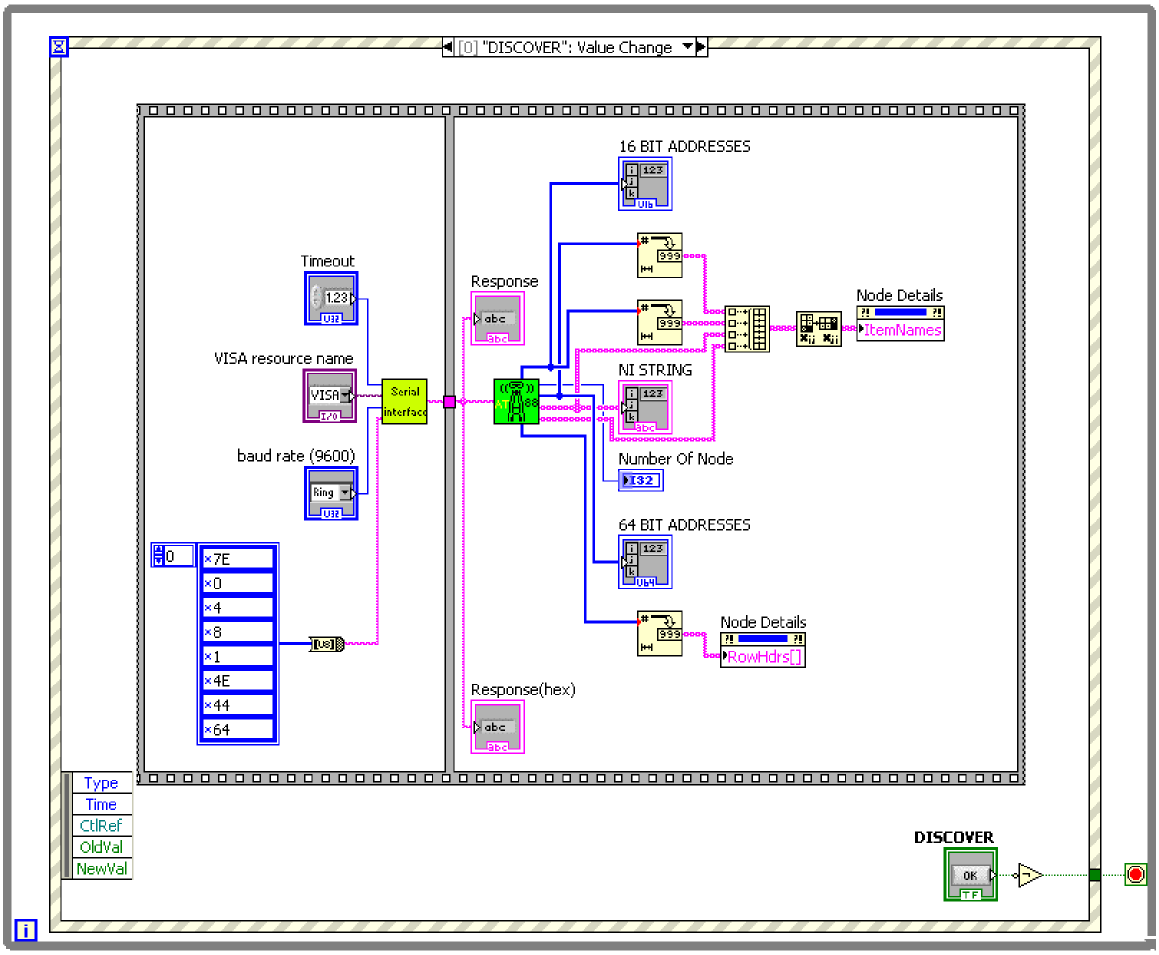
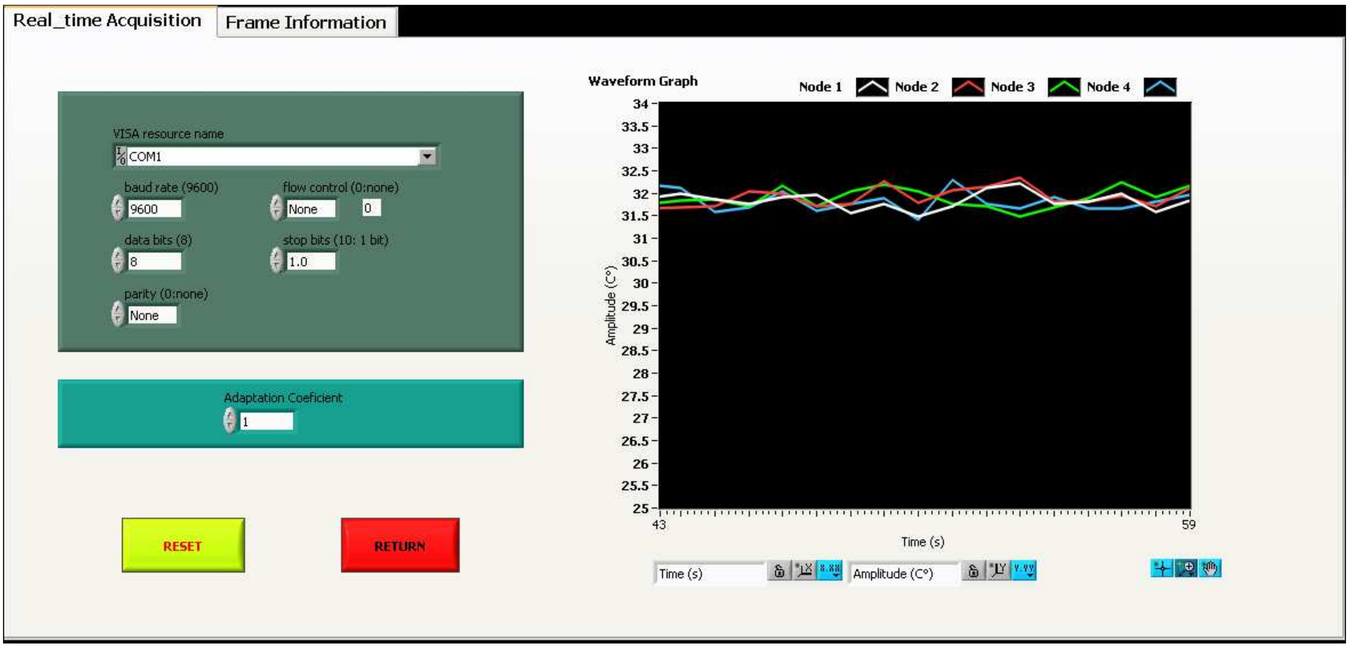


Figure 5.28: Network discovery bloc diagram


Real_time Acquisition | Frame Information

VISA resource name: COM1

baud rate (9600): 9600

flow control (0:none): None 0

data bits (8): 8

stop bits (10: 1 bit): 1.0

parity (0:none): None

Adaptation Coeficient: 1

Figure 5.29: Network data monitoring front panel

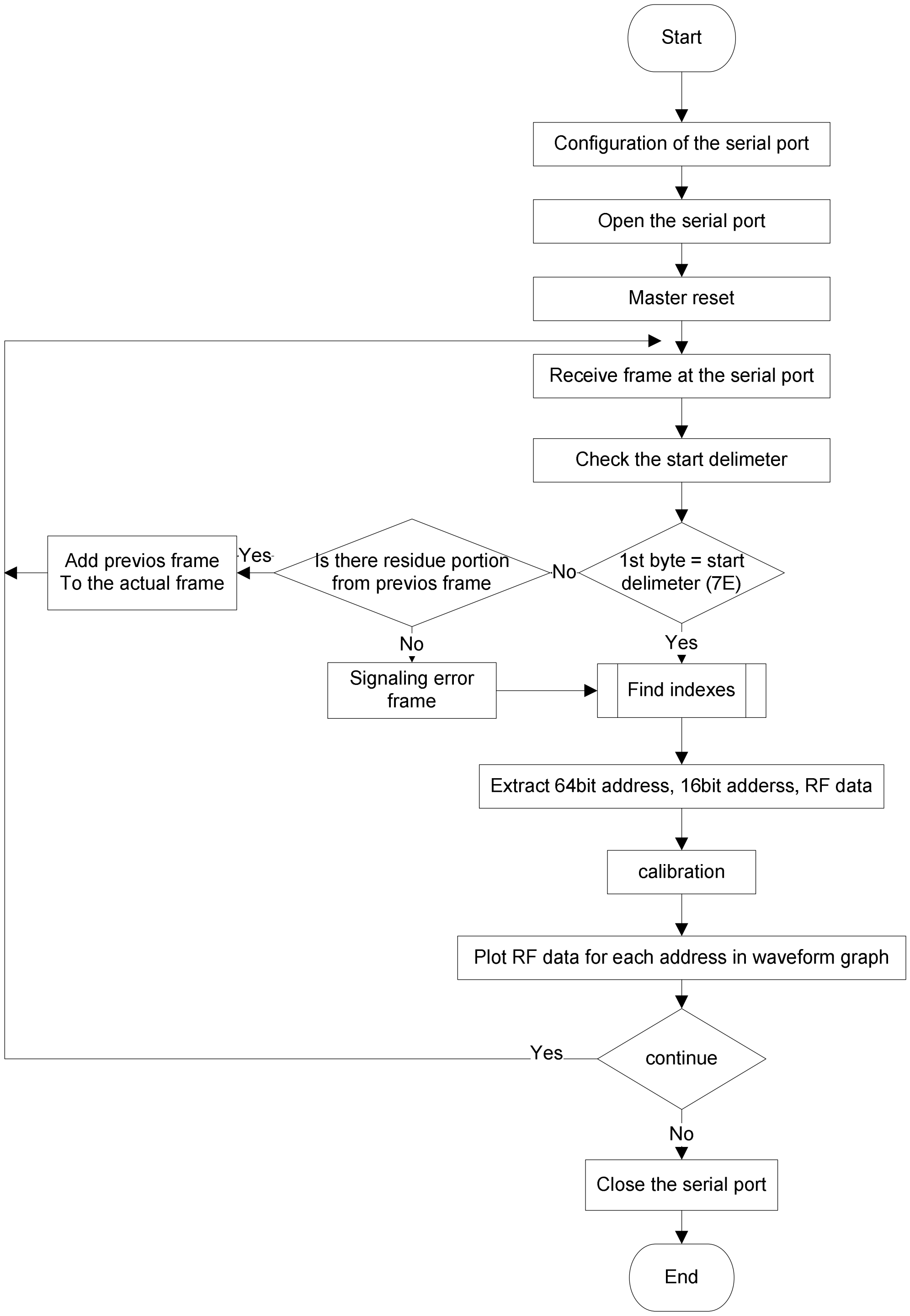
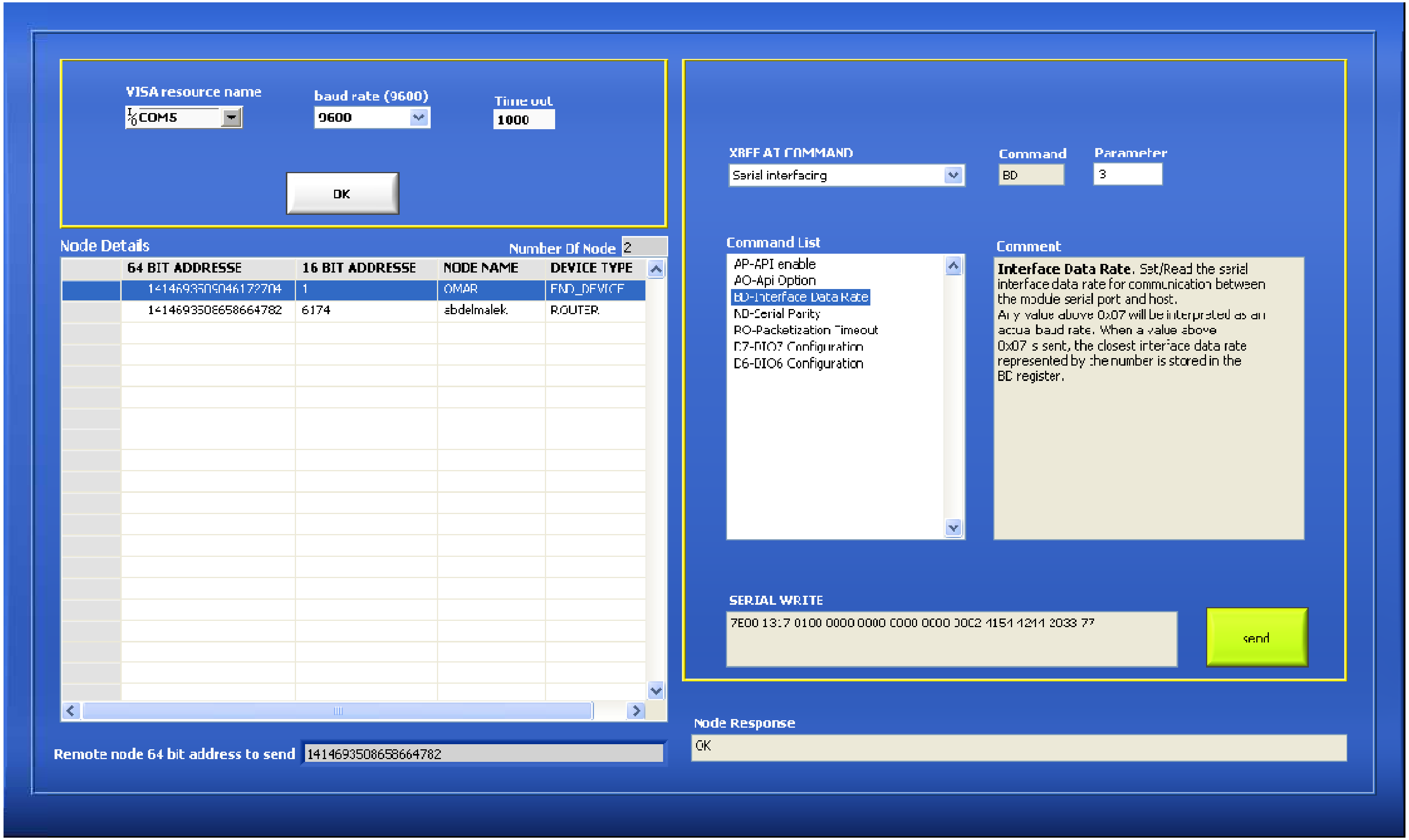

Start
Configuration of the serial port
Open the serial port
Master reset
Receive frame at the serial port
Check the start delimeter
1st byte = start delimeter (7E)
No
Is there residue portion from previos frame
Yes
Add previos frame To the actual frame
No
Signaling error frame
Yes
Find indexes
Extract 64bit address, 16bit adderss, RF data
calibration
Plot RF data for each address in waveform graph
continue
Yes
No
Close the serial port
End

Figure 5.30: Network data monitoring flow chart

c) Remote configuration

To query or set module parameters on a remote device after network discovery, we can configure all nodes in network wirelessly, this is done if we know the 64-bit address and 16-bit address of node, the user can choose any node to configure it using a list of commands. The following figures 5.31 - 5.33 show the front panel of user interface, the flow chart and the bloc diagram respectively.



Node Details

Number Of Node 2

	64 BIT ADDRESSE	16 BIT ADDRESSE	NODE NAME	DEVICE TYPE
	1414693505046177704	1	OMAR	END_DEVICE
	1414693508658664782	6174	abdelmalek	ROUTER

Remote node 64 bit address to send 1414693508658664782

XBEE AT COMMAND: Serial interfacing

Command: BD

Parameter: 3

Command List

AP-API enable
AO-Api Option
BD-Interface Data Rate
ND-Serial Parity
RO-Packetization Timeout
D7-DIO7 Configuration
D6-DIO6 Configuration

Comment

Interface Data Rate. Set/Read the serial interface data rate for communication between the module serial port and host. Any value above 0x07 will be interpreted as an actual baud rate. When a value above 0x07 is sent, the closest interface data rate represented by the number is stored in the BD register.

SERIAL WRITE

```
7E00 13:7 0100 0000 0000 C000 0C00 JOC2 4151 4214 2033 77
```

Node Response: OK

Figure 5.31: Network discovery front panel

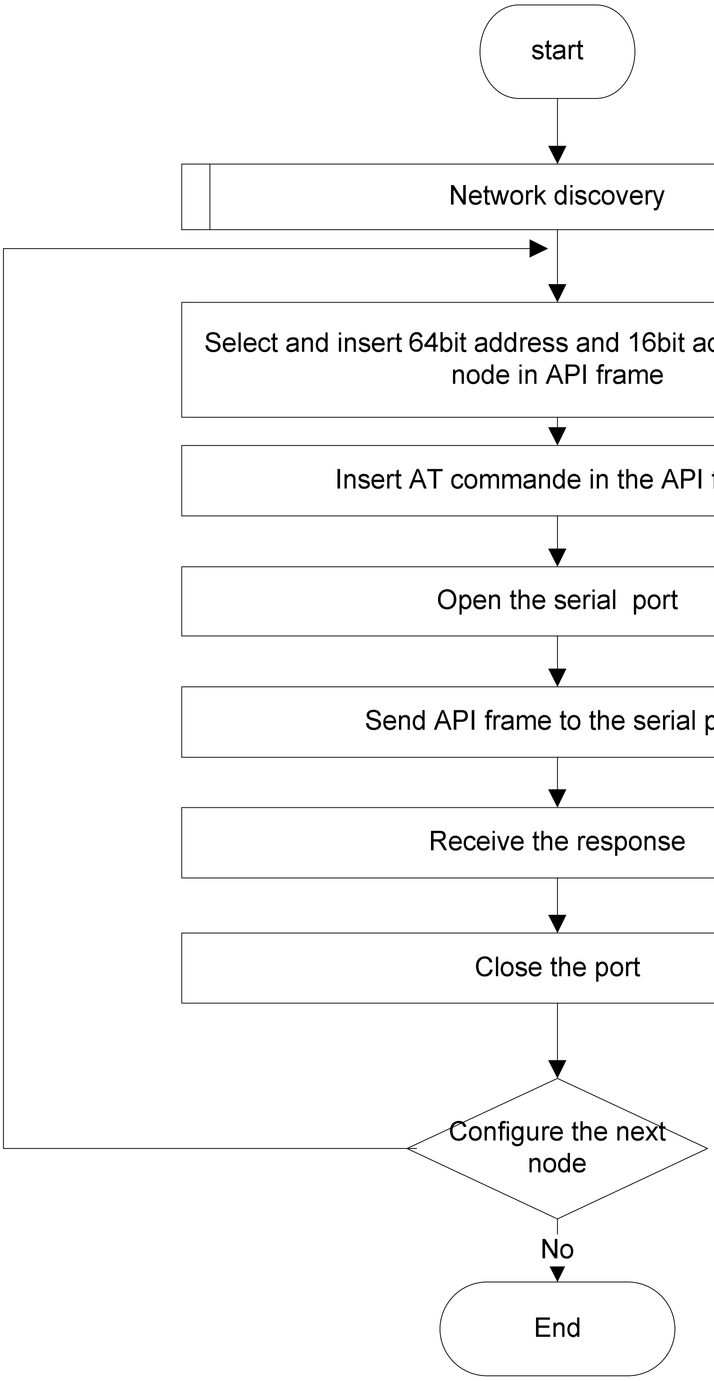

start
Network discovery
Select and insert 64bit address and 16bit address of desired node in API frame
Insert AT commande in the API frame
Open the serial port
Send API frame to the serial port
Receive the response
Close the port
Configure the next node
No
End

Figure 5.32: Remote configuration flow chart

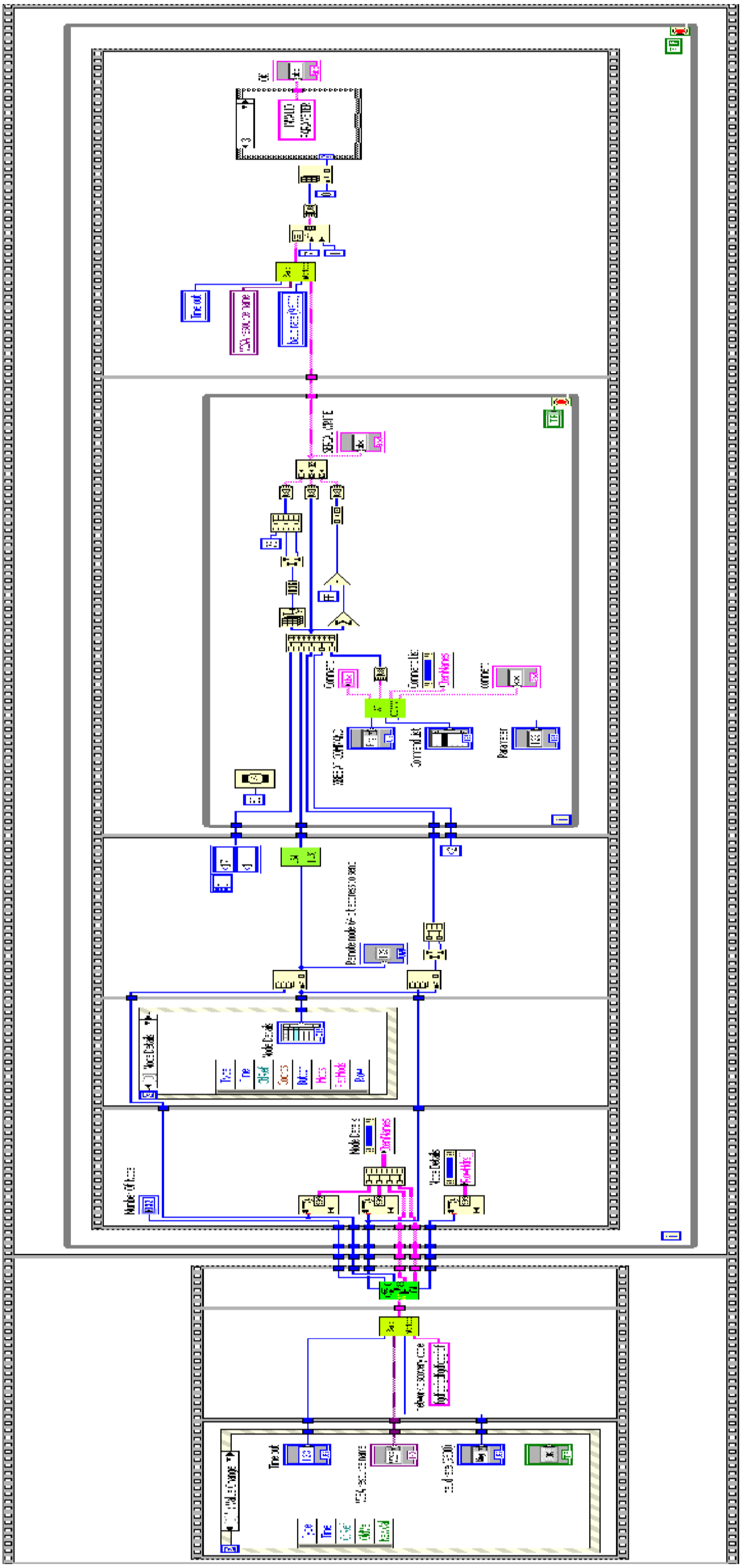
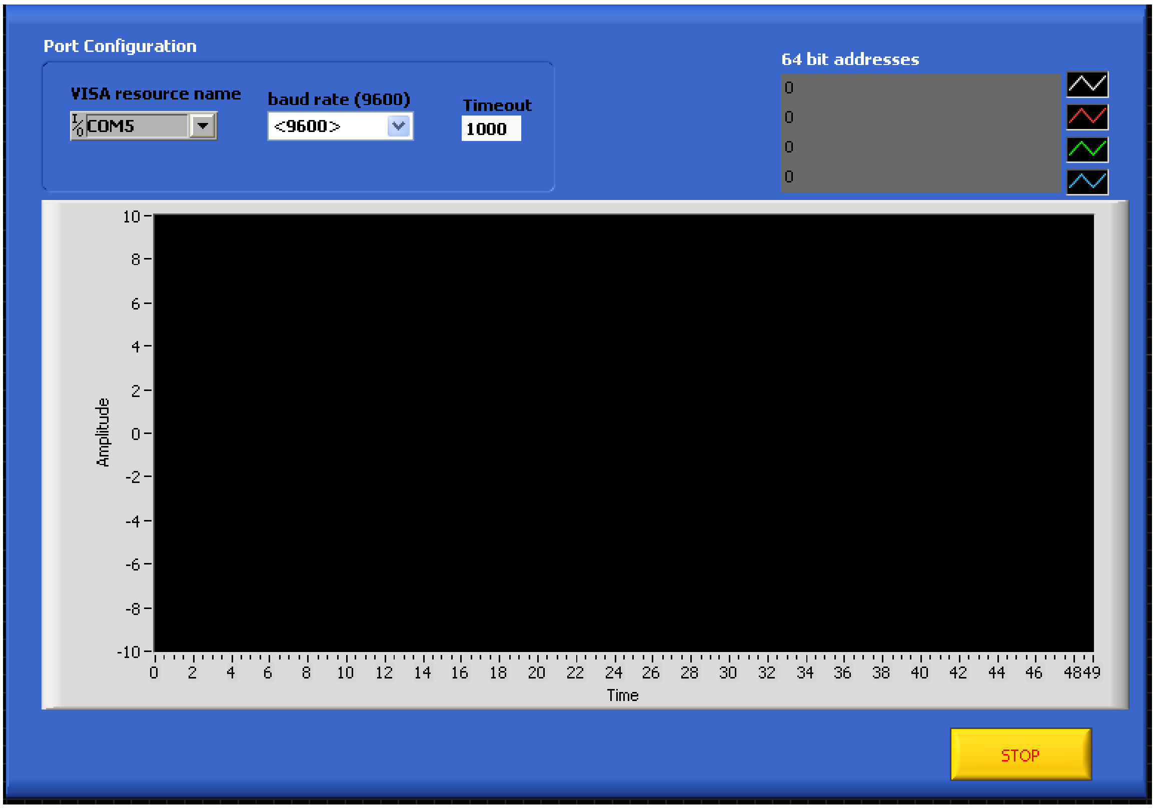


Figure 5.33: Remote configuration bloc diagram

d) Xbee sensor read indicator

XBee modules have the ability to monitor and sample the analog and digital IO lines. IO samples can be read locally or transmitted to a remote device to provide indication of the current IO line states. Analog samples are returned as 10-bit values. (The analog inputs on the Xbee module cannot read more than 1.2V.), the **user interface shown in figure 5.34 displays the value of** sensor sample. The manager front panel is building according to 5.35 flow chart.

(Note one should configure Xbee using 1641firmware version ZNET2.5 ROUTER/END DEVICE ANALOG I/O before operation).



Port Configuration

VISA resource name: COM5

baud rate (9600): <9600>

Timeout: 1000

64 bit addresses:
0
0
0
0

Figure 5.34: Sensor read indicator front panel

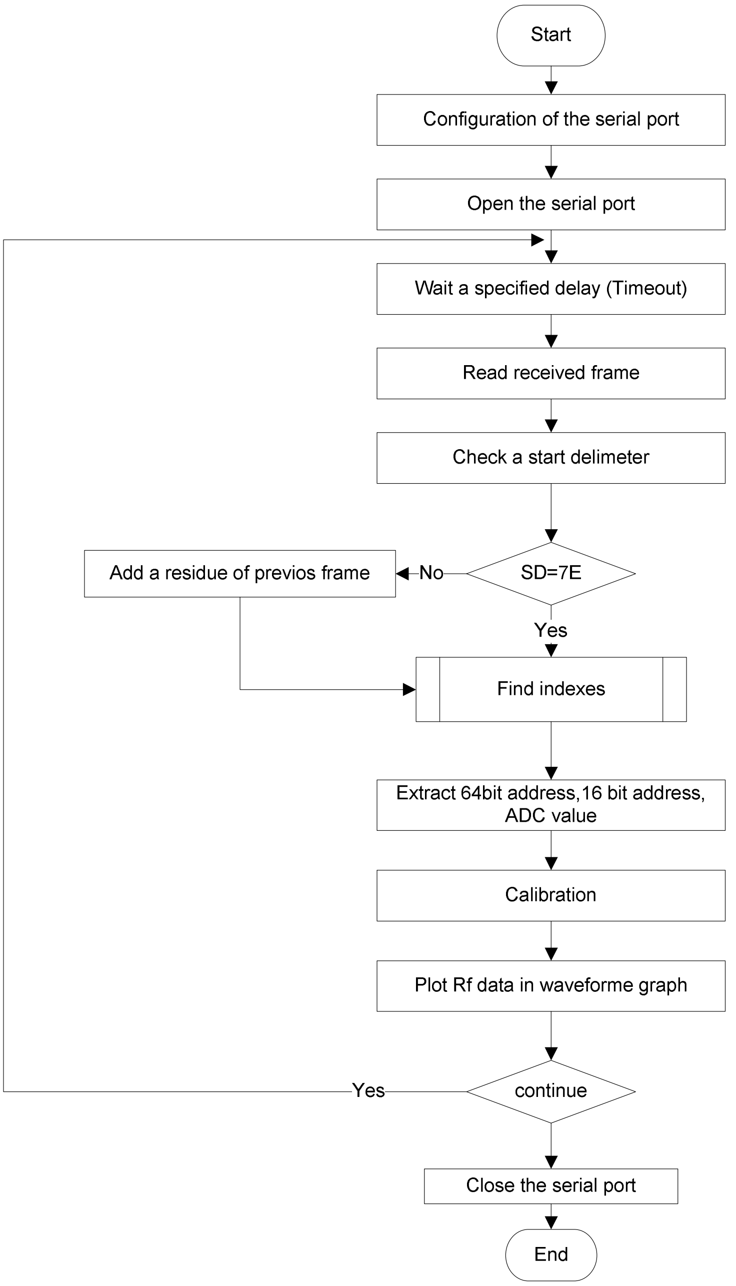

Start
Configuration of the serial port
Open the serial port
Wait a specified delay (Timeout)
Read received frame
Check a start delimeter
Add a residue of previos frame
No
SD=7E
Yes
Find indexes
Extract 64bit address,16 bit address,
ADC value
Calibration
Plot Rf data in waveforme graph
Yes
continue
Close the serial port
End

Figure 5.35: Sensor read indicator flow chart

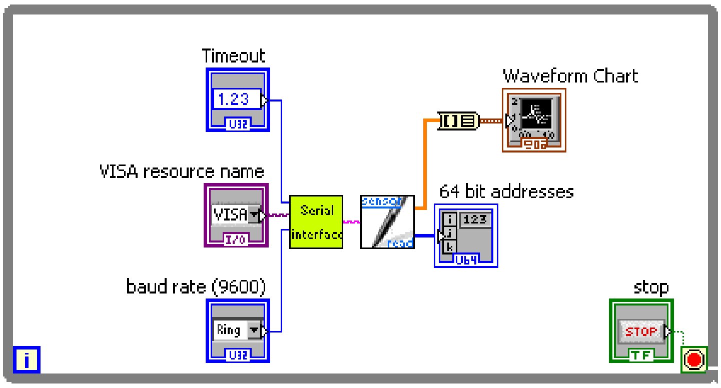
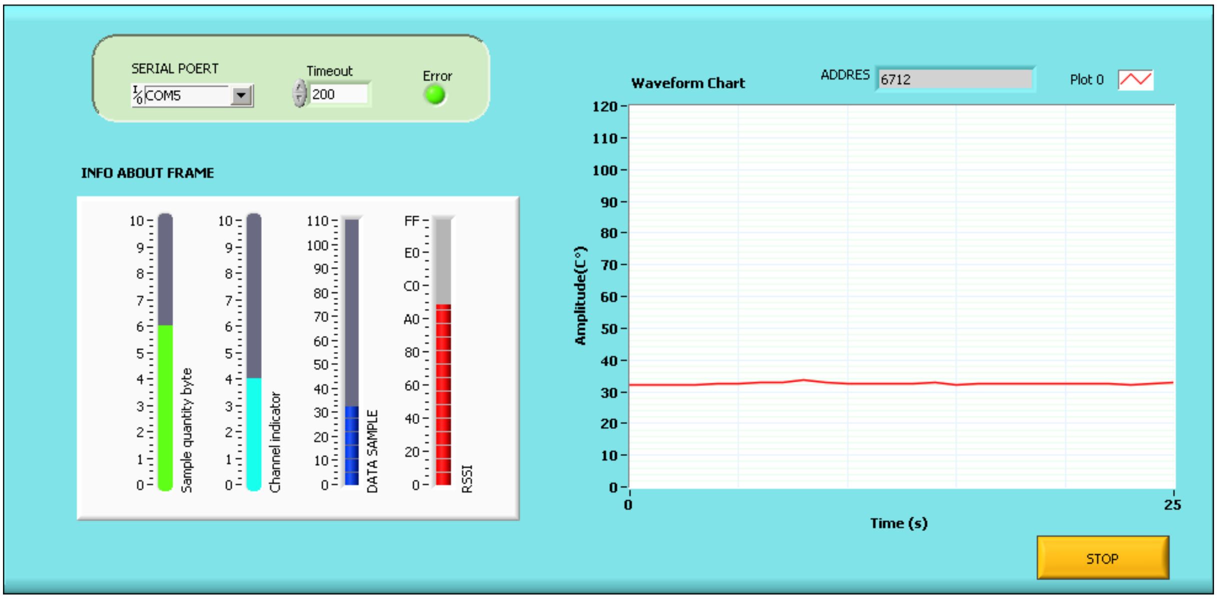


Figure 5.36: sensor read indicator Bloc diagram

The same application with of sensor read in AT mode is developed in figure 5.37.



SERIAL POERT COM5

Timeout 200

Error

ADDRES 6712

Plot 0

Figure 5.37: sensor read indicator in AT mode front panel

e) Serial port parameters

All the previously user interfaces defined above, are equipped with serial port parameters, in order to easily setup the configuration of serial ports.

5.6 Conclusion

Both hardware and software were extensively tested. Due to lack of specialized equipment, only limited range of hardware features are tested.

The different options and situations which appear during exploiting Xbee modules are detailed through the development of our Lab view designed system.

Programming and monitoring real data sensors according to users' needs and recommendations are the only procedures which have to be added for achieving a complete network system.

The boards of our designed system and accessories are shown in figures 5.38 - 5.39.

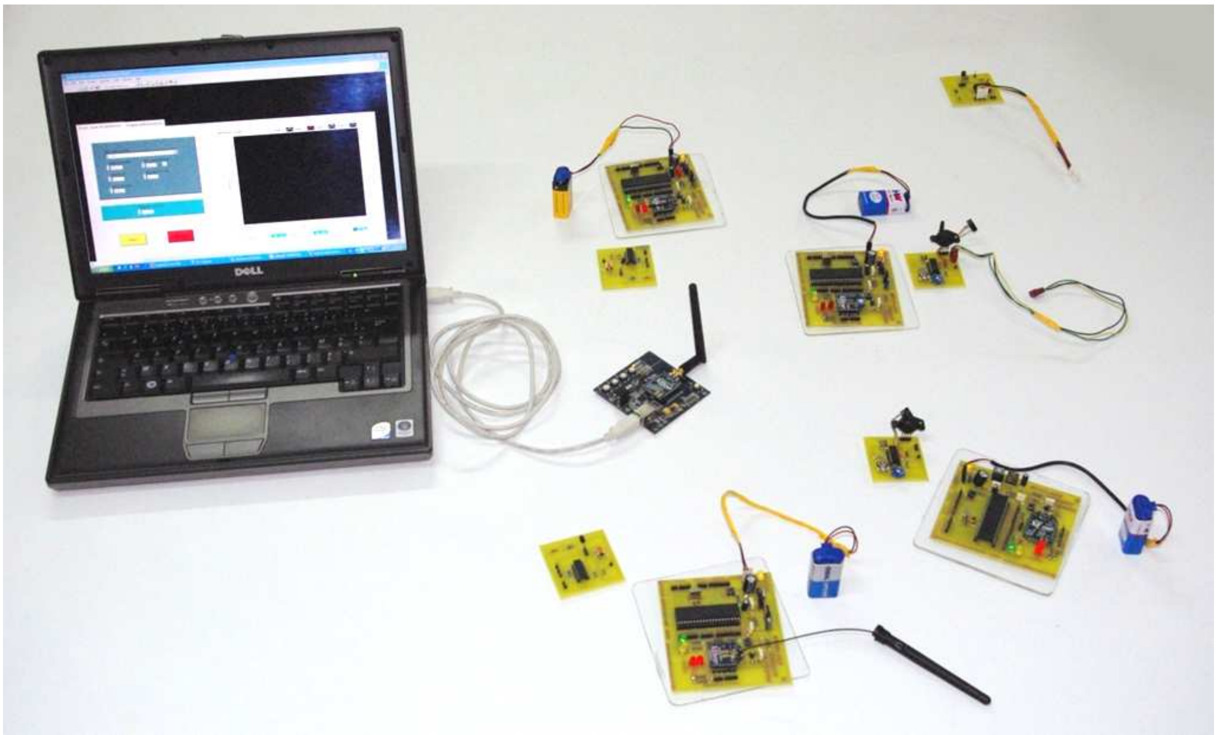


Figure 5.38: The designed system

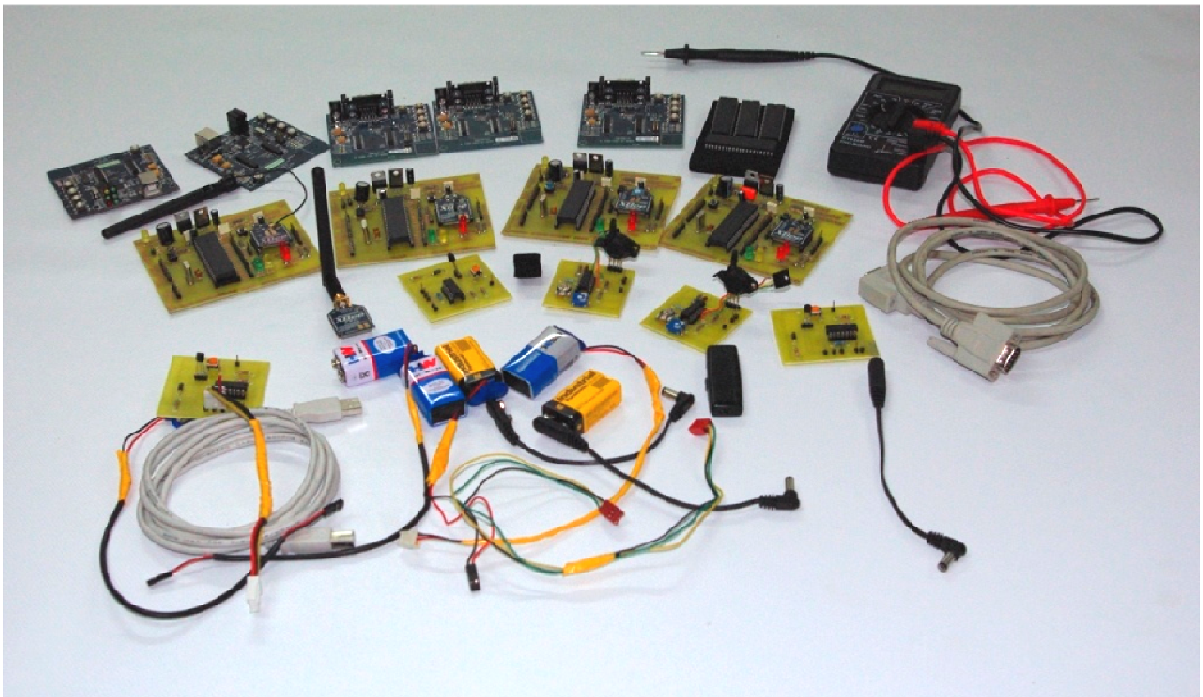


Figure 5.39: Material used in project

Conclusion and future work

The design of a successful Wireless Sensor Network using ZigBee MaxStream XBee Series 2 Modules, MICROCHIP MCU PIC18LF452 Microcontroller under a LabVIEW 8.6 platform and a series of pressure and temperature sensors (MPX2200A, LM35DZ) is achieved.

The fundamental of network specifications are investigated. The hardware and software contributions to build a WSN are analyzed.

The obtained results, after completing this project, should open multiple horizons to undertake a novel view for tackling and searching solutions through WSN uses.

We may conclude that in near future, WSN field will be expanding rapidly so its design phases appearing now to be highly complex and involving interdisciplinary approaches will see substantial reductions. The WSN state-of-the-art technology and its impact on human life are the focus of advanced industrial poles and scientific communities so we will see widespread applications entering in everyday activities.

This WSN project is the first implemented in the department, it may constitute the right impulsion to vulgarize the smart network implementation.

To get a veritable impact on most targeted applications, energy node should be thoroughly studied. Energy scavenging circuits like solar cell are expected.

In order to get extra facilities with WSN modules, user interface programming all API frames should be developed.

For network optimal reconfiguration, programming procedures of MCUs should be carried out following certain exploitation criteria and mainly based on simples actions.

Bibliographie

[1]H. LABIOD, H.AFIFI and C. DE SANTIS:Wi-Fi, Bluetooth, ZigBee and WiMAX**,** edition 2007 Springer.

[2] KAZEM SOHRABY, DANIEL MINOLI,TAIEB ZNATI WIRELESS SENSOR NETWORKS Technology, Protocols, and Applications 2007 by John Wiley.
[3] Anna Ha´c, Wireless Sensor Network Designs, John Wiley & Sons 2003
[4] Rajeev Shorey A.Ananda, Mun Choon Chan, Wei Tsang Ooi MOBILE, wireless, and sensor networks technology, applications,and future directions.
[4]Ian F. Akyildiz, W. Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: 2002.
[5]MOHAMMAD ILYAS ANDIMAD MAHGOUB **Handbook of Sensor Networks:Compact Wireless and Wired Sensing Systems**, 2005 by CRC Press
[6] Ivan Stojmenovic,**HANDBOOK OF SENSOR NETWORKS ALGORITHMS AND ARCHITECTURES**, 2005 by John Wiley & Sons.
[7]2008 ieee International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing June 2008Taichung, Taiwan.
[8]Guang-Zhong Yang,**Body Sensor Networks,** Springer-Verlag London 2006.
[9]Shahin Farahani,ZigBee Wireless Networks and Transceivers, 2008, Elsevier.
[10] IEEE 802.15.1-2005 standard of information technology
http://standards.ieee.org/getieee802/download/802.15.1-2005.pdf
[11] Bluetooth Special Interest Group Web hhttp://www.bluetooth.org

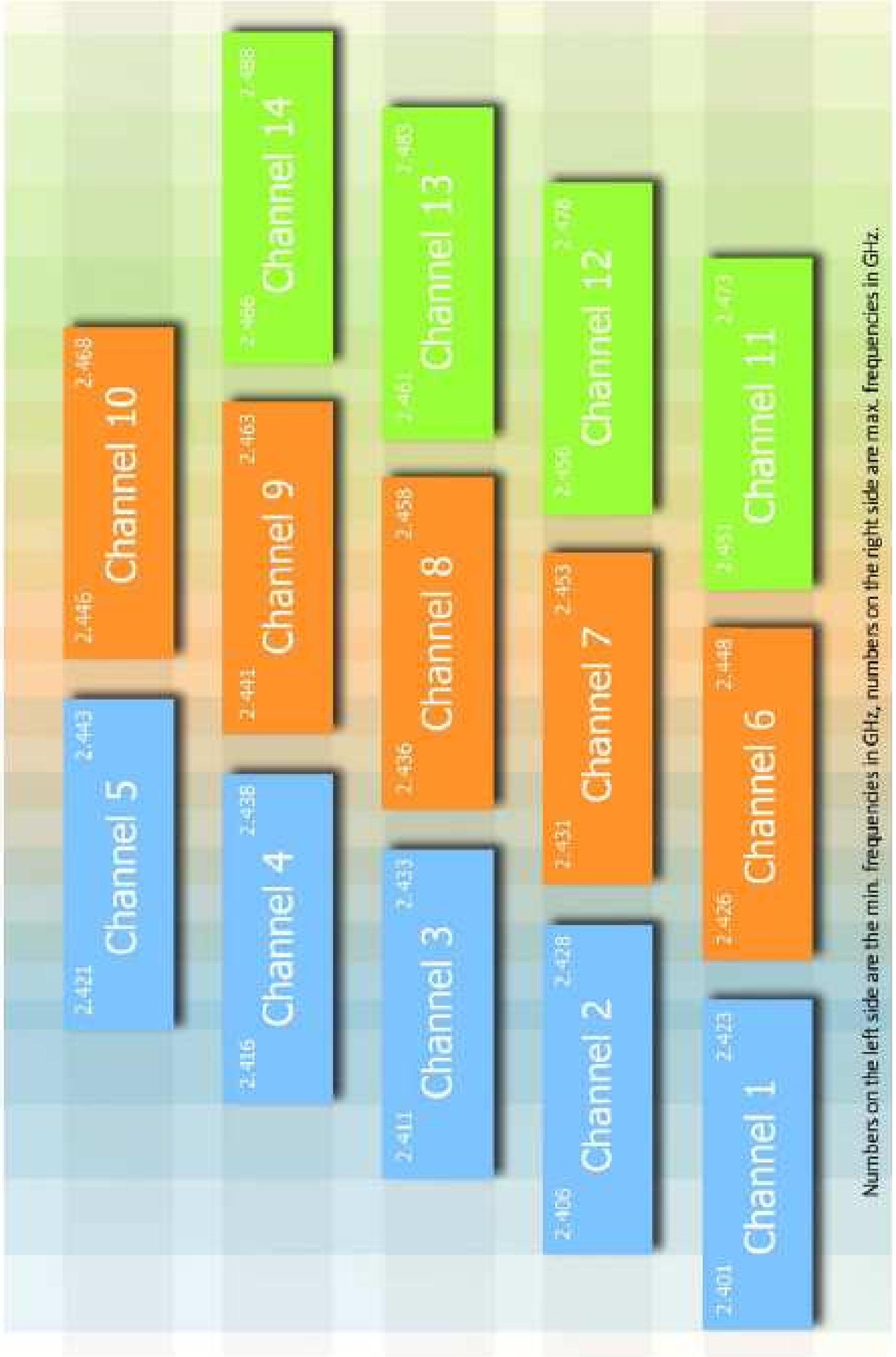
[12] Bluetooth profile specification
http://www.bluetooth.com/Bluetooth/Technology/Building/Specifications/Default.htm
[13] CSR Blue Core
www.csr.com/products/bcrange.htm
[14] Broadcom Bluetooth SoC
http://www.broadcom.com/products/Bluetooth/Bluetooth-RF-Silicon-and-Software-Solutions
[15] CSR BlueCore price list
http://www.csr.com/products/bcrange.htm
[16] Gainspan’s low power Wi-Fi solution product brief
http://standards.ieee.org/getieee802/download/802.15.3-2003.pdf
[17] ZigBee Specification 2007 by ZigBee Alliance
http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf
[18] Fresnel Zone explanation
[19] Texas Instruments Web
www.ti.com
[20] Texas Instruments CC2420 development kit
http://focus.ti.com/docs/prod/folders/print/cc2430.html
[21] Texas Instruments CC2430
http://focus.ti.com/docs/prod/folders/print/cc2430.html

[22] Texas Instruments CC2431
http://focus.ti.com/docs/prod/folders/print/cc2431.html
[23] Texas Instruments Z-Stack
http://focus.ti.com/docs/toolsw/folders/print/timac.html
[24] Texas Instruments TIMAC
http://focus.ti.com/docs/prod/folders/print/cc2430.html#toolssoftware
[25] Texas Instruments CC2520
http://focus.ti.com/docs/prod/folders/print/cc2431.html
[26] Freescale site Web
http://www.freescale.com
[27] Freescale MC13192
http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MC13192
[28] Ember site web
www.ember.com
[29]Microchip site web
www.microchip.com
[30]Microchip MRF24J40
[31] Atmel site web
www.atmel.com
[32]radicraft site web
www.radiocraft.com
[33] Meshnetics site web
www.Meshnetics.com
[34] Meshnetics ZigBee range test
http://www.meshnetics.com/dl.php?id=10
[35] RFM (Cirronet) site web
www.rfm.com
[35] Jennic Web
www.**jennic.com**
[36] MaxStream
www.digi.com
[37] MaxStream XBee Series 2 Manual
Available on Digi site web
[38] Fresnel Zone explanation
http://www.digi.com/support/kbase/kbaseresultdetl.jsp?kb=90
[39] Nitaigour P. Mahalik Sensor Networks and Configuration Fundamentals, Standards, Platforms, and Applications. Springer-Verlag Berlin Heidelberg 2007.
[40]Creed Huddleston, Intelligent Sensor Design Using the Microchip dPIC, 2007, Elsevier.
[41]Dogan Ibrahim. Advanced PIC Microcontroller Projects in C, 2008, Elsevier.
[42] Holger Karl, Andreas Willig, PROTOCOLSAND ARCHITECTURES FOR WIRELESS SENSOR NETWORKS,John Wiley,2005
[43]Feng Zhao, Leonidas J. Guibas, Wireless Sensor Networks2005, Elsevier.

[44]Matthew Gast. 802.11 Wireless Networks: The Definitive Guide. Number ISBN: 0596100523. O’Reilly, 2005.

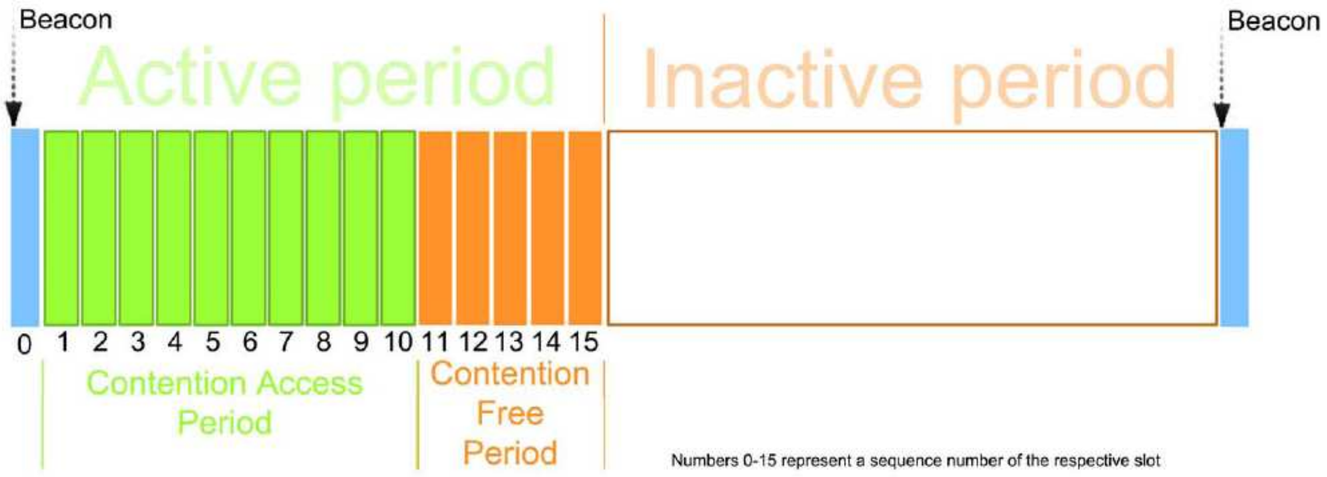
Appendices

The following pages contain images and summary tables referenced from the main text's they depict in details the issues discussed in this thesis and illustrate results of comparisons.

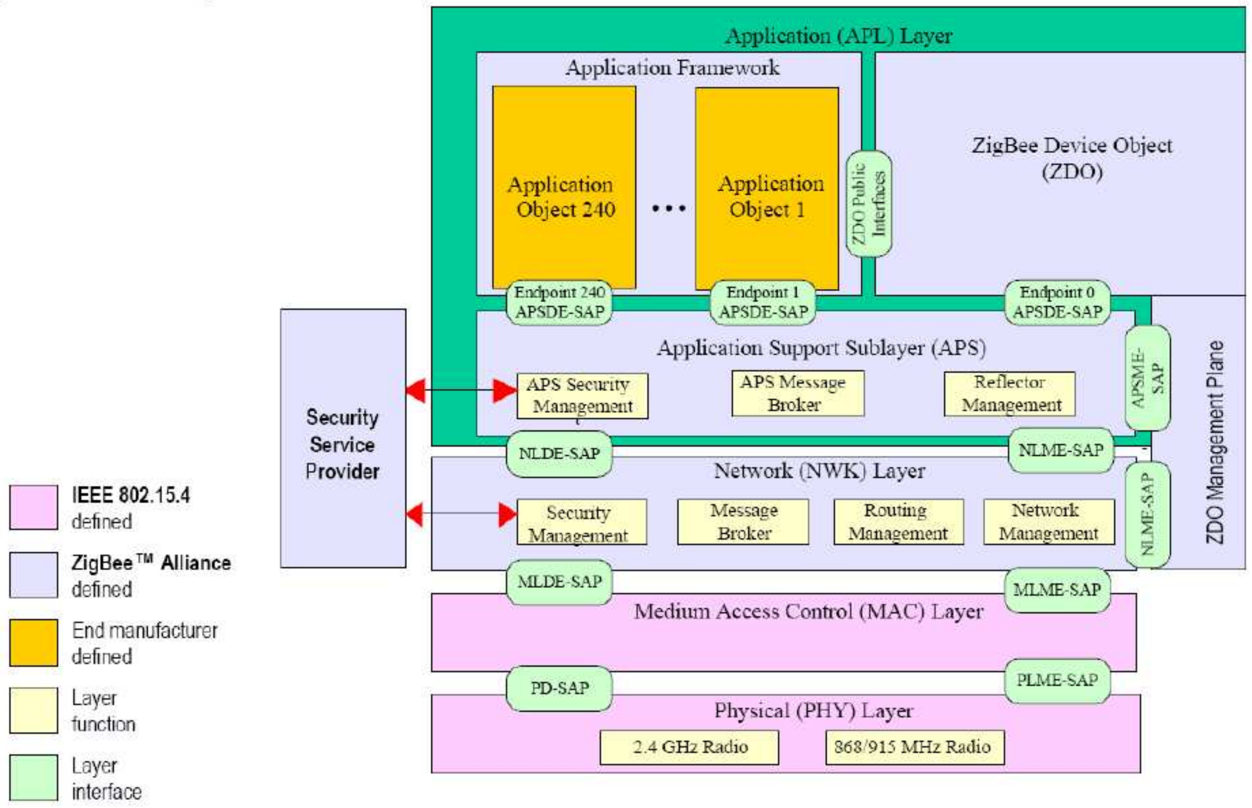


Wi-Fi channels spectrum and its overlay

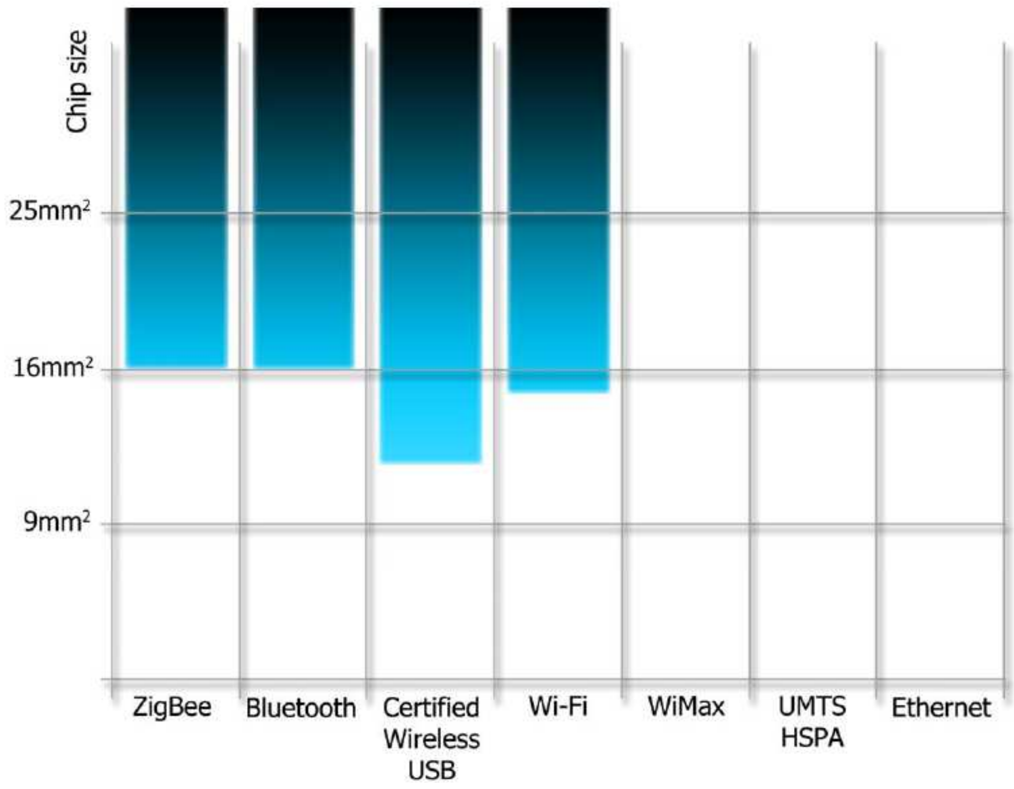
ZigBee Superframe Structure



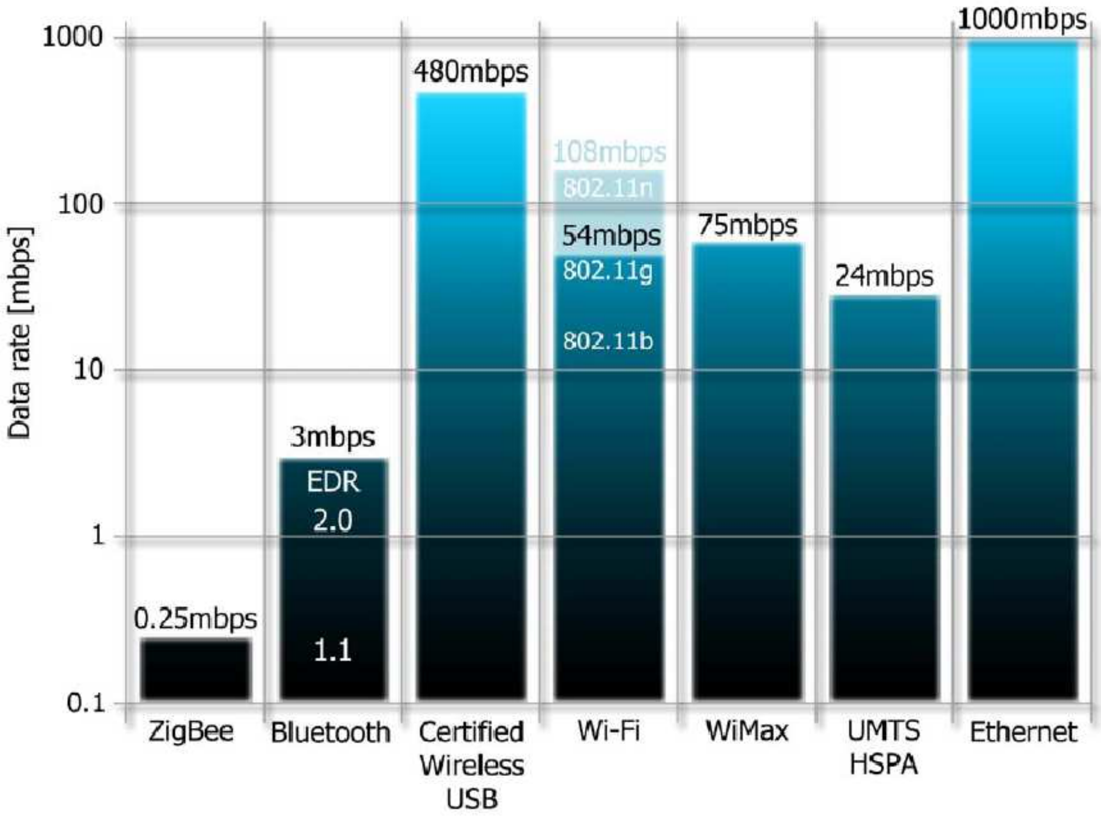
ZigBee Stack Layered Structure



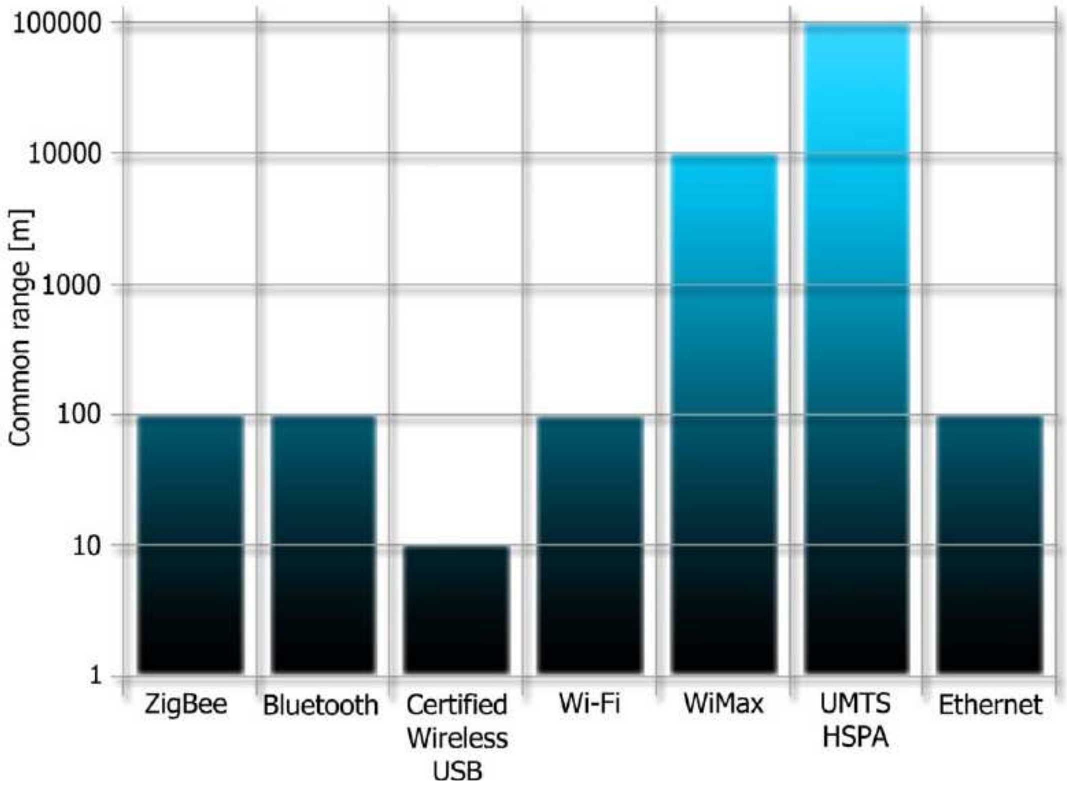
Wireless Technologies Comparison – Dimensions



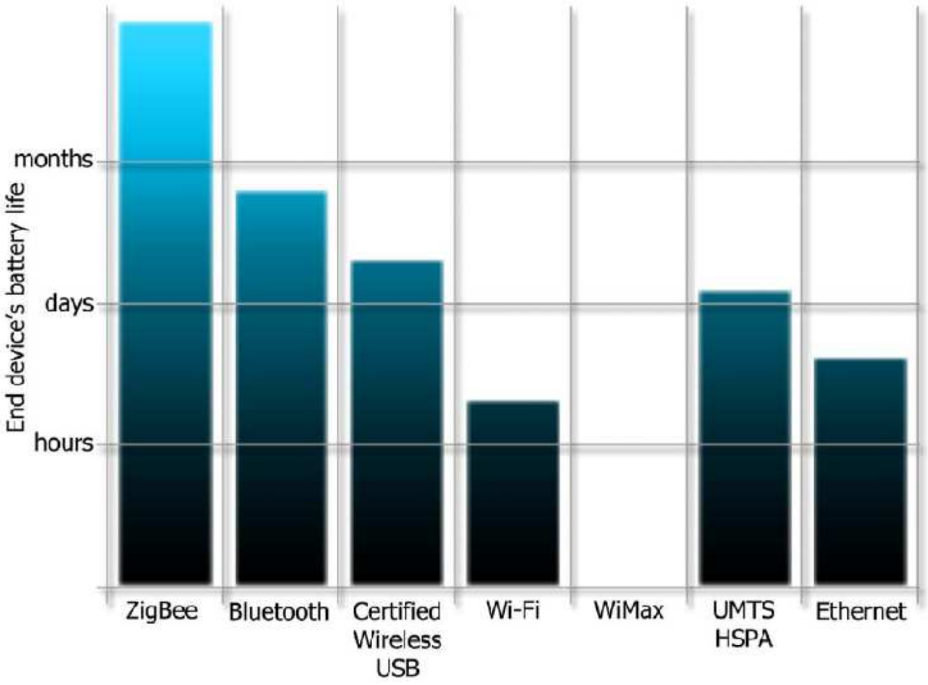
Wireless Technologies Comparison - Data Rates



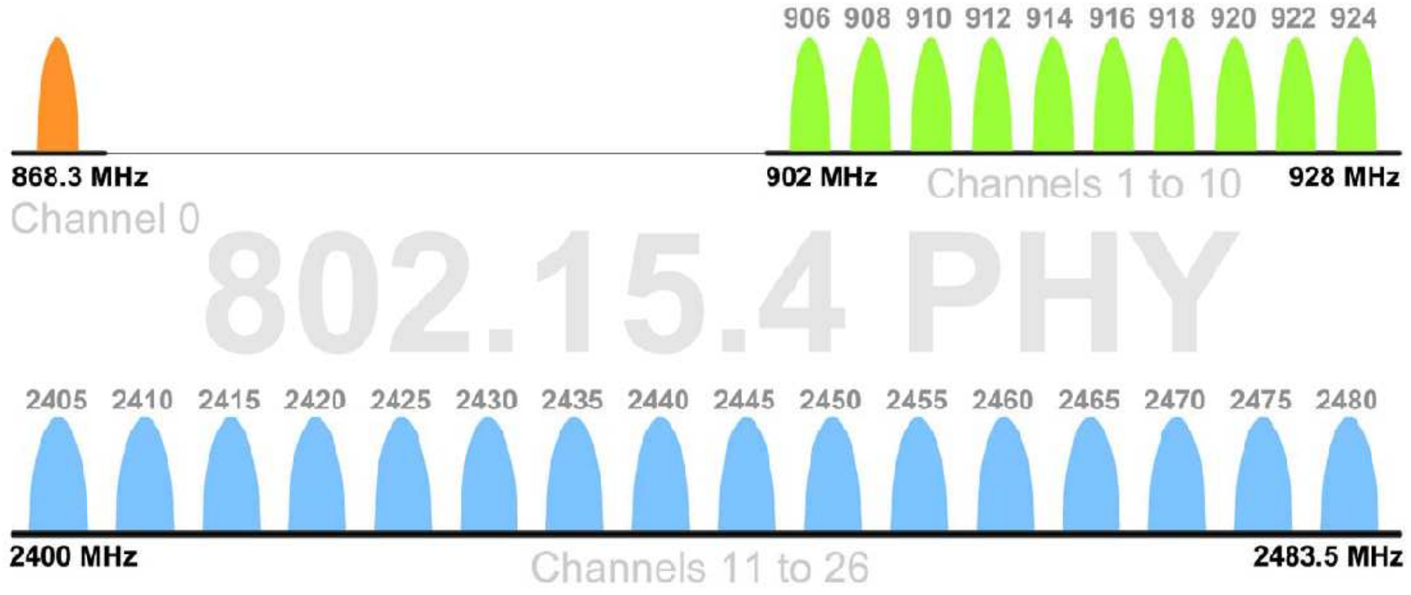
Wireless Technologies Comparison – Ranges



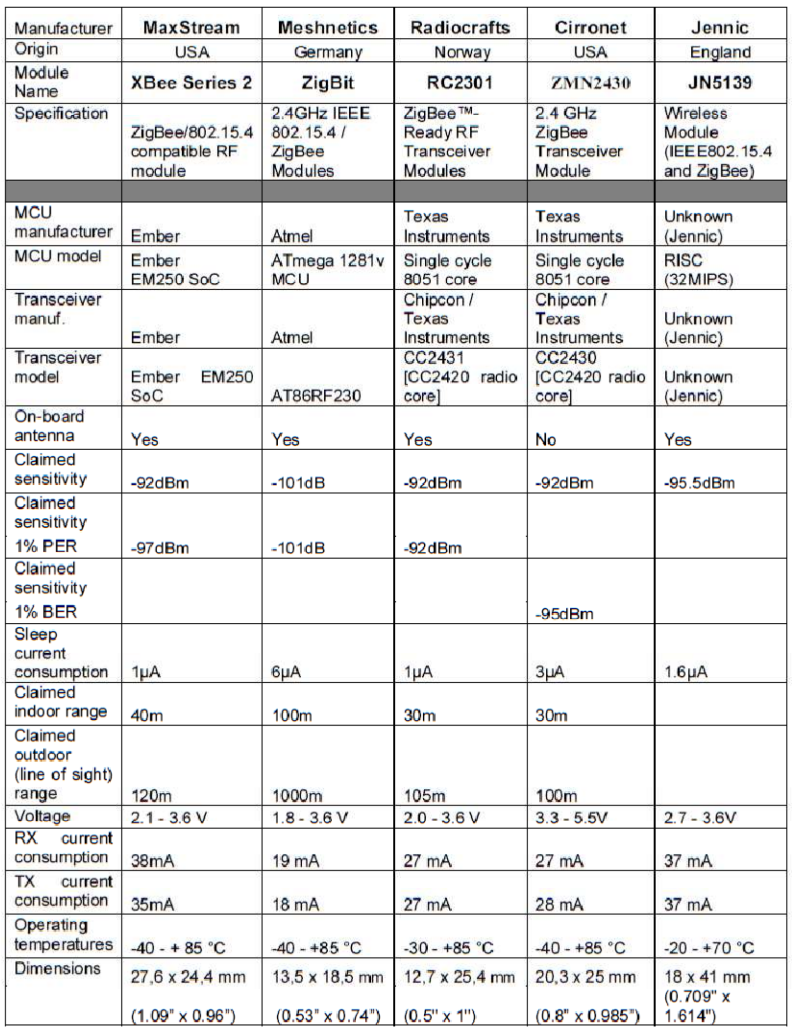
Wireless Technologies Comparison - Power Consumption



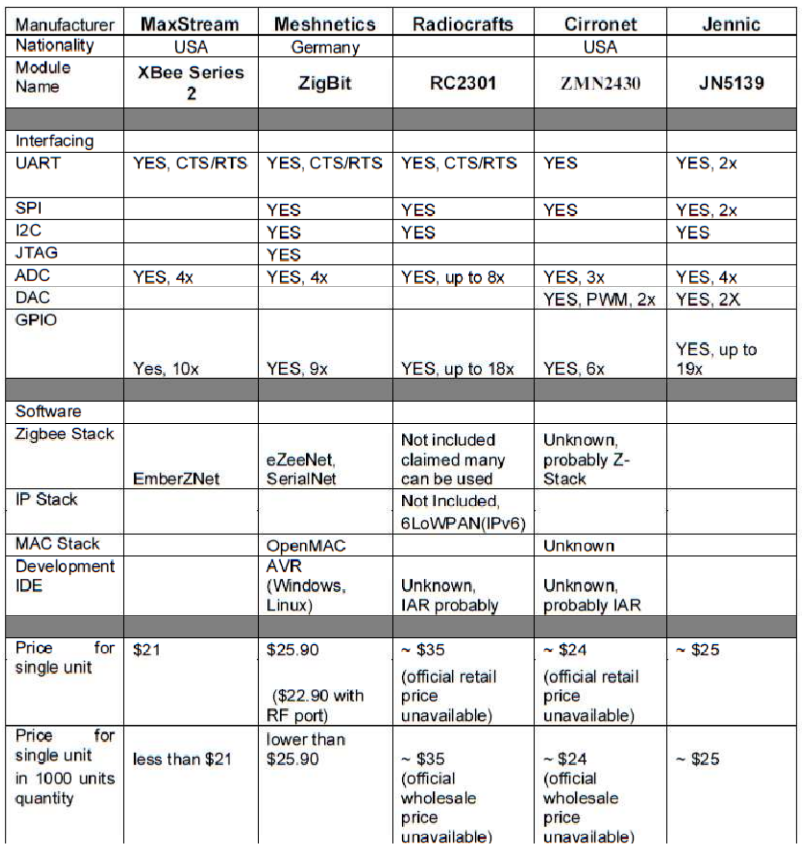
IEEE 802.15.4 Channels Spectrum



ZigBee Modules Comparison Table



Manufacturer	MaxStream	Meshnetics	Radiocrafts	Cirronet	Jennic
Origin	USA	Germany	Norway	USA	England
Module Name	**XBee Series 2**	**ZigBit**	**RC2301**	**ZMN2430**	**JN5139**
Specification	ZigBee/802.15.4 compatible RF module	2.4GHz IEEE 802.15.4 / ZigBee Modules	ZigBee™-Ready RF Transceiver Modules	2.4 GHz ZigBee Transceiver Module	Wireless Module (IEEE802.15.4 and ZigBee)
MCU manufacturer	Ember	Atmel	Texas Instruments	Texas Instruments	Unknown (Jennic)
MCU model	Ember EM250 SoC	ATmega 1281v MCU	Single cycle 8051 core	Single cycle 8051 core	RISC (32MIPS)
Transceiver manuf.	Ember	Atmel	Chipcon / Texas Instruments	Chipcon / Texas Instruments	Unknown (Jennic)
Transceiver model	Ember EM250 SoC	AT86RF230	CC2431 [CC2420 radio core]	CC2430 [CC2420 radio core]	Unknown (Jennic)
On-board antenna	Yes	Yes	Yes	No	Yes
Claimed sensitivity	-92dBm	-101dB	-92dBm	-92dBm	-95.5dBm
Claimed sensitivity 1% PER	-97dBm	-101dB	-92dBm		
Claimed sensitivity 1% BER				-95dBm	
Sleep current consumption	1µA	6µA	1µA	3µA	1.6µA
Claimed indoor range	40m	100m	30m	30m	
Claimed outdoor (line of sight) range	120m	1000m	105m	100m	
Voltage	2.1 - 3.6 V	1.8 - 3.6 V	2.0 - 3.6 V	3.3 - 5.5V	2.7 - 3.6V
RX current consumption	38mA	19 mA	27 mA	27 mA	37 mA
TX current consumption	35mA	18 mA	27 mA	28 mA	37 mA
Operating temperatures	-40 - + 85 °C	-40 - +85 °C	-30 - +85 °C	-40 - +85 °C	-20 - +70 °C
Dimensions	27,6 x 24,4 mm (1.09" x 0.96")	13,5 x 18,5 mm (0.53" x 0.74")	12,7 x 25,4 mm (0.5" x 1")	20,3 x 25 mm (0.8" x 0.985")	18 x 41 mm (0.709" x 1.614")



Manufacturer	**MaxStream**	**Meshnetics**	**Radiocrafts**	**Cirronet**	**Jennic**
Nationality	USA	Germany		USA	
Module Name	**XBee Series 2**	**ZigBit**	**RC2301**	ZMN2430	**JN5139**
Interfacing					
UART	YES, CTS/RTS	YES, CTS/RTS	YES, CTS/RTS	YES	YES, 2x
SPI		YES	YES	YES	YES, 2x
I2C		YES	YES		YES
JTAG		YES			
ADC	YES, 4x	YES, 4x	YES, up to 8x	YES, 3x	YES, 4x
DAC				YES, PWM, 2x	YES, 2X
GPIO	Yes, 10x	YES, 9x	YES, up to 18x	YES, 6x	YES, up to 19x
Software					
Zigbee Stack	EmberZNet	eZeeNet, SerialNet	Not included claimed many can be used	Unknown, probably Z-Stack	
IP Stack			Not Included, 6LoWPAN(IPv6)		
MAC Stack		OpenMAC		Unknown	
Development IDE		AVR (Windows, Linux)	Unknown, IAR probably	Unknown, probably IAR	
Price for single unit	$21	$25.90 ($22.90 with RF port)	~ $35 (official retail price unavailable)	~ $24 (official retail price unavailable)	~ $25
Price for single unit in 1000 units quantity	less than $21	lower than $25.90	~ $35 (official wholesale price unavailable)	~ $24 (official wholesale price unavailable)	~ $25

MaxStream XBee Modules

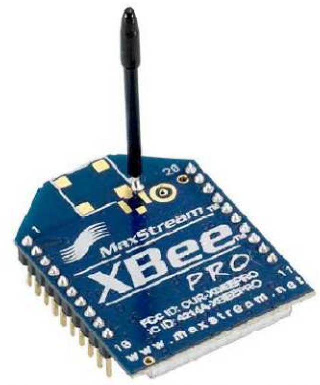
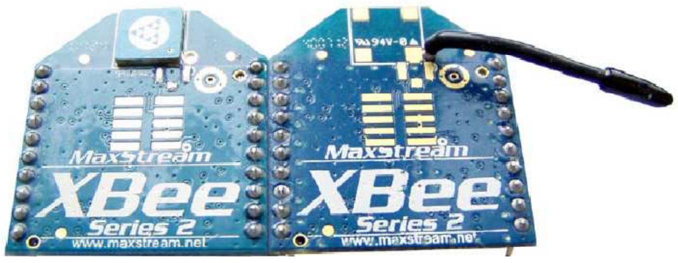
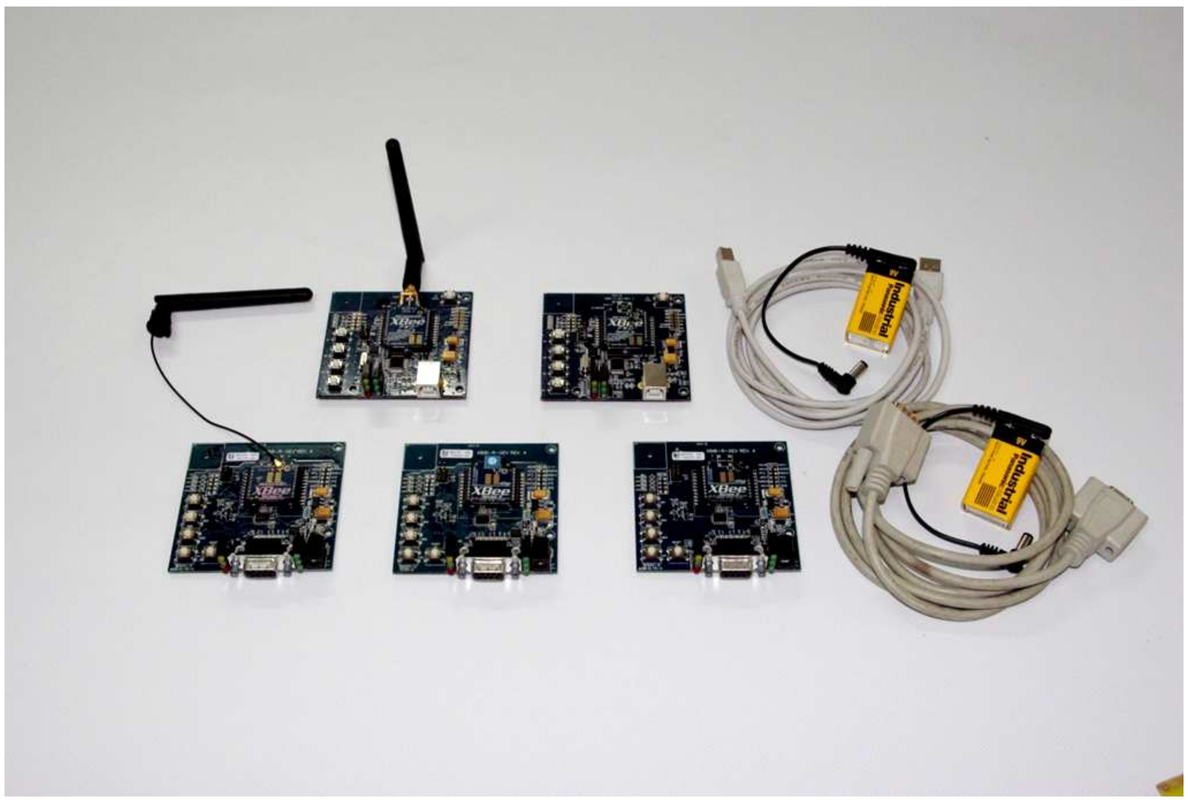


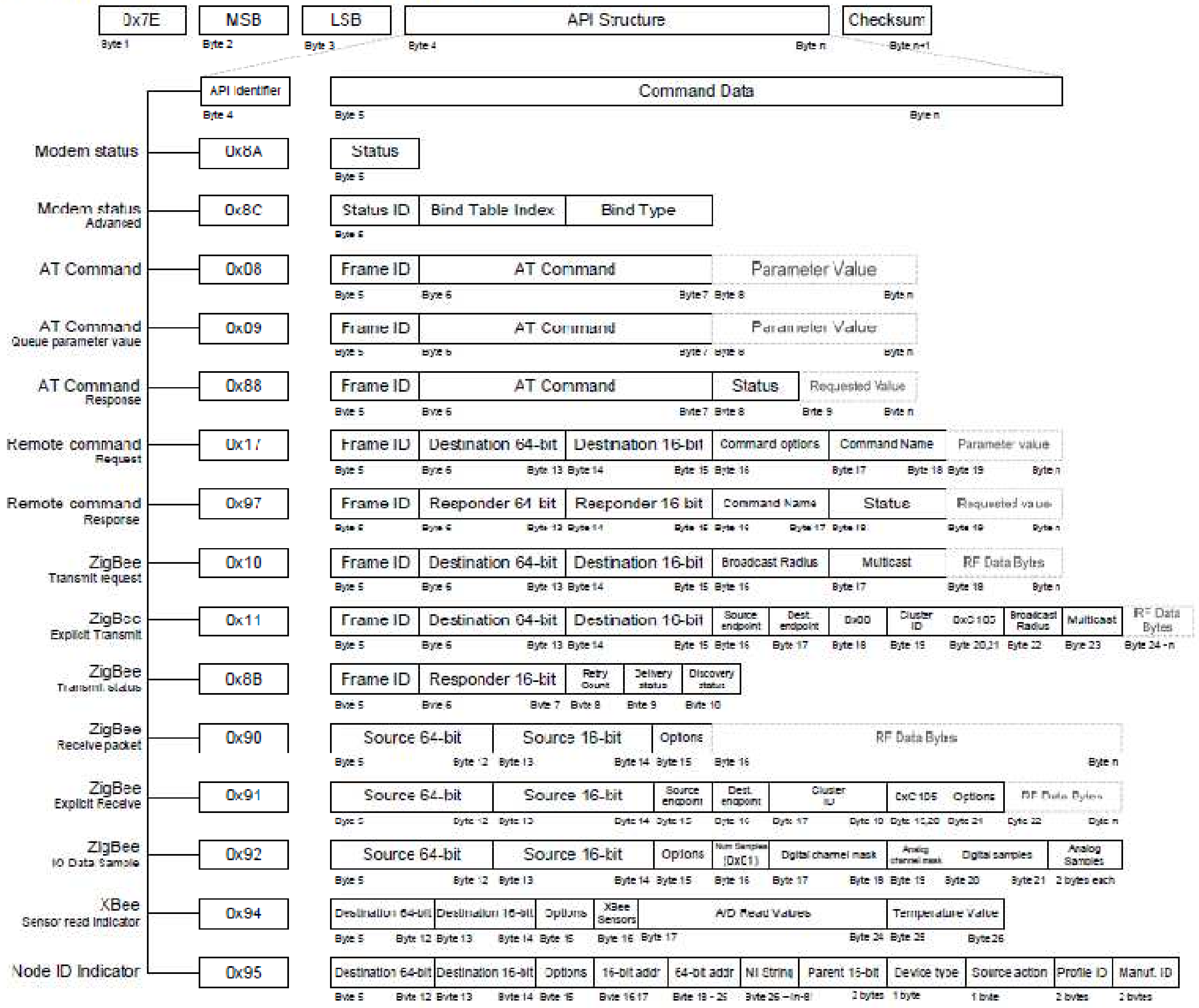
Figure C.11 – XBee Pro Module with whip antenna

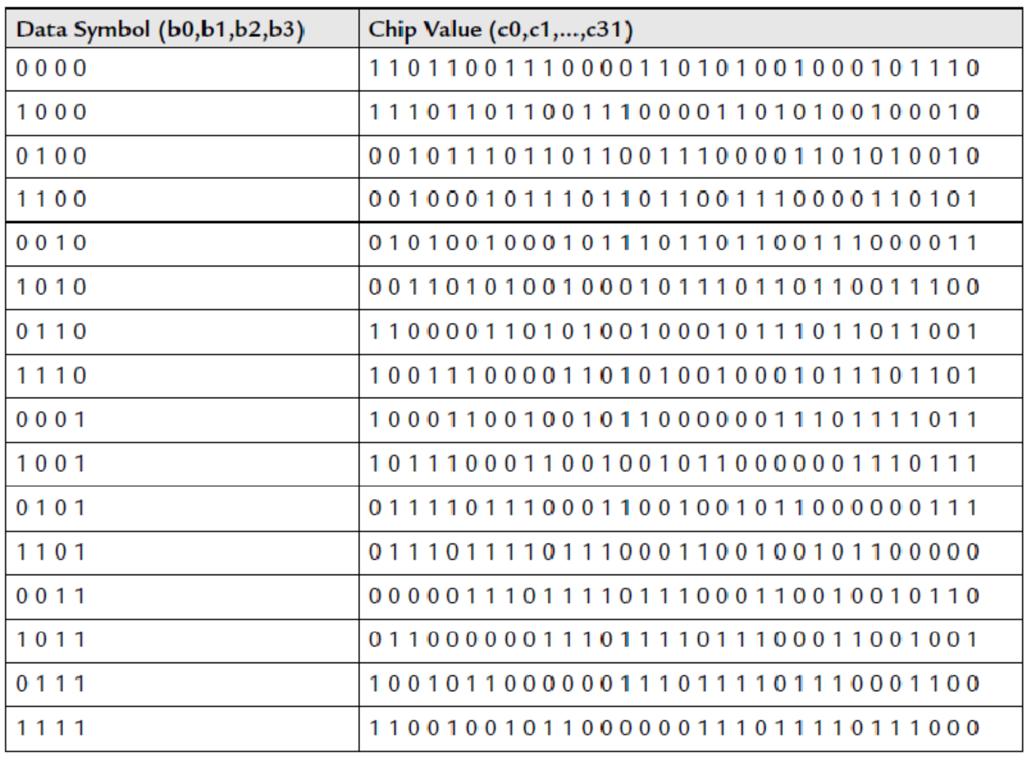




MaxStream Zigbee KIT

XBee ZNet 2.5 API Frames Structure

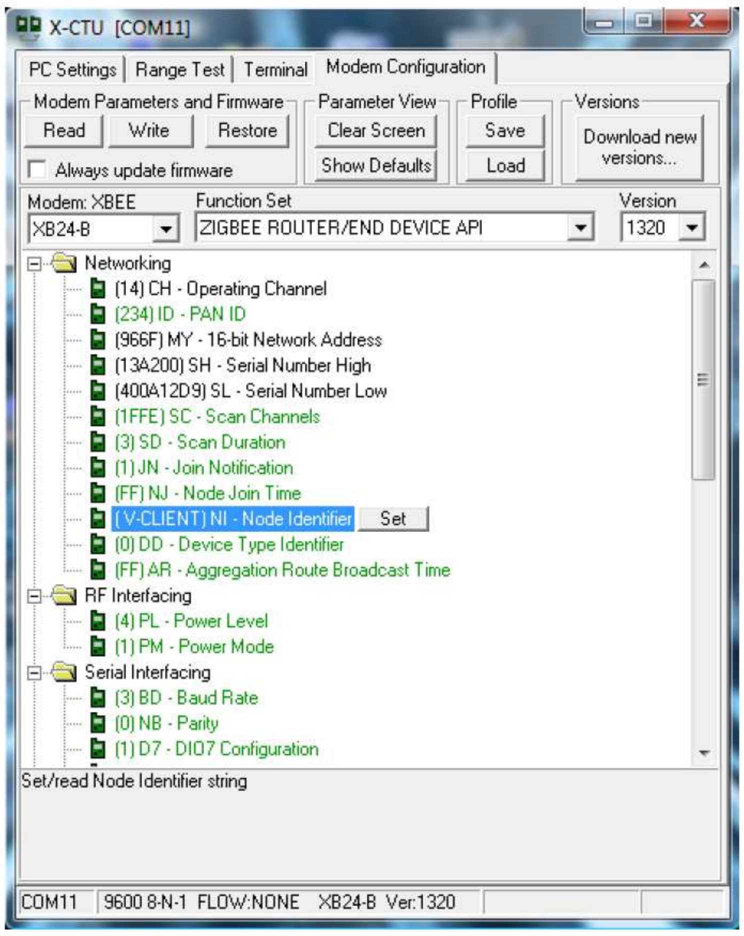




Data Symbol (b0,b1,b2,b3)	Chip Value (c0,c1,...,c31)
0 0 0 0	1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0
1 0 0 0	1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0
0 1 0 0	0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0
1 1 0 0	0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1
0 0 1 0	0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0 0 0 1 1
1 0 1 0	0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1 1 1 0 0
0 1 1 0	1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1 1 0 0 1
1 1 1 0	1 0 0 1 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 0 0 0 1 0 1 1 1 0 1 1 0 1
0 0 0 1	1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1
1 0 0 1	1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1
0 1 0 1	0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1
1 1 0 1	0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0
0 0 1 1	0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1 0 1 1 0
1 0 1 1	0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0 1 0 0 1
0 1 1 1	1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 0 0
1 1 1 1	1 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0

DSSS Symbol-to-Chip Mapping TablesX-CTU software

X-CTU software for configuring and testing MaxStream radio modem.



□ Always update firmware

Modem: XBEE: XB24-B

Function Set: ZIGBEE ROUTER/END DEVICE API

Version: 1320

Networking
- (14) CH - Operating Channel
- (234) ID - PAN ID
- (966F) MY - 16-bit Network Address
- (13A200) SH - Serial Number High
- (400A12D9) SL - Serial Number Low
- (1FFE) SC - Scan Channels
- (3) SD - Scan Duration
- (1) JN - Join Notification
- (FF) NJ - Node Join Time
- (V-CLIENT) NI - Node Identifier
- (0) DD - Device Type Identifier
- (FF) AR - Aggregation Route Broadcast Time

RF Interfacing
- (4) PL - Power Level
- (1) PM - Power Mode

Serial Interfacing
- (3) BD - Baud Rate
- (0) NB - Parity
- (1) D7 - DIO7 Configuration

Set/read Node Identifier string

COM11 9600 8-N-1 FLOW:NONE XB24-B Ver:1320

Wireless Technologies Comparison Summary Table

	ZigBee	Bluetooth	WUSB	WHDI	WiFi	WiMAX	HSPA	GPRS/ EDGE
Classification	PAN	PAN	PAN	PAN	LAN	WAN	Cellular	Cellular
IEEE	802.15.4	802.15.1	-	-	802.11	802.16	-	-
Primary field of use	Serial cable replacement, low power	Cable replace-ment for mobile devices	Wireless alternative for USB	High definition video and audio	Local wireless networks	Metro-politan networks	3G cellular networks	Cellular networks
Downstream data rate (maximum)	250 kb/s	3 Mb/s	480 Mb/s	3 Gb/s	108 Mb/s	70 Mb/s	14 Mb/s	2 Mb/s
Upstream data rate (maximum)	250 kb/s	3 Mb/s	480 Mb/s	-	108 Mb/s	70 Mb/s	5 Mb/s	1 Mb/s
Outdoor range (high gain antennas and high output power)	1 km	100 m	10 m	10 m	1 km	10 km	-	-
Common range	30-100 m	10-100 m	3-10 m	1-10 m	10-100 m	1-10 km	-	-
Common speed at 100m	250 kb/s	~1 Mb/s	-	-	11 Mb/s	50 Mb/s	14 Mb/s	2 Mb/s
Network topology	Mesh network	Peer to Peer	Peer to Peer	Peer to Peer	IP / Peer to Peer	IP	IP	IP
Frequency band	868/915 MHz, 2,4GHz	2,4GHz	3,1-10,6GHz	5GHz	2,4GHz, 5,8GHz	2-11GHz	869-894MHz	900/1800/1900MHz
RF Modulation	DSSS	FHSS	DSSS	DSSS	DSSS	FHSS	-	-
Common end device battery life	Years	up to a month or more	days	-	few days	-	days	days
Maximum children of single parent	theoretically 65535	7	127	1	tens	-	-	-
Chip price	$5 - $10	$3-$10	N/A	-	N/A	-	-	-
Chip dimensions (smallest found) [mm]	5x5	4x4	3.36x3.36	-	5x3	-	-	-

Updating Device Firmware

Devices bought from MaxStream (unless specially requested) come with a ‘ZigBee Router / End Device AT’ firmware.

To create a coordinator or API device, a proper firmware has to be programmed to the device. X-CTU offers all the required functionality.

To update the firmware, install and run the X-CTU software. In the first tab (‘PC Settings’), select the COM port of the device and go to the ‘Modem Configuration’ tab. Press the Download new version to get the most recent firmware release. By this time, no hardware operations are done yet.

Upon successful downloading, press the ‘Read’ button to obtain device settings and firmware information. This way, a proper modem type should be selected for you (XB24□B or XBee Series 2, XB24 for XBee Series 1 and XBP… for PRO versions), as well as ‘Function Set’ and ‘Version’.

If you have not done any previous firmware flashing, the values you see should be ‘XB24□B’ ‘ZIGBEE ROUTER/END DEVICE AT’ ‘1220’.

Now, change the Function Set to ‘ZIGBEE COORDINATOR AT’ (or ‘ZNET 2.5 COORDINATOR AT’), depending on the required functionality (discussed in chapter 5.2.3). Press the Write button and wait for the programming to finish.

When the programming is done, an exception might emerge because of inability to communicate with the device (i.e. to access the command mode). This is only due to the recent firmware flashing and probably because of not keeping the one second data silence on the UART, You can press the reset button or only wait for a few seconds and try to read the parameters again.

PCB layer OF all board:

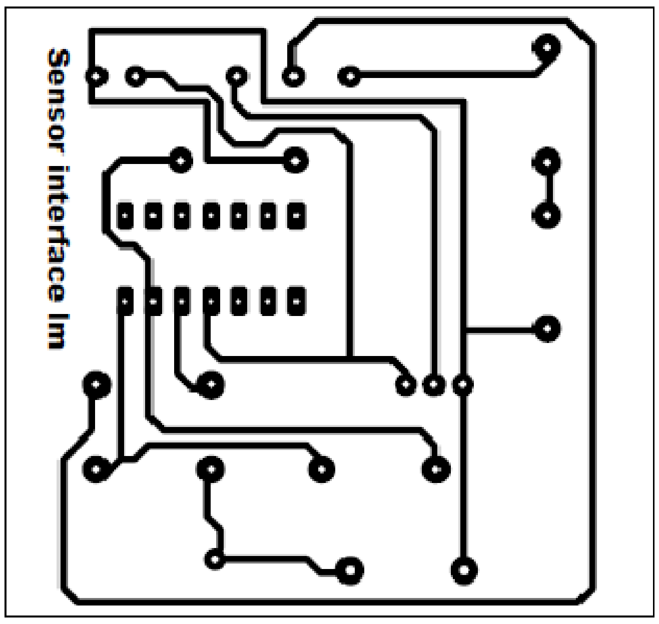


Figure A.1 Temperature Sensor interfaces

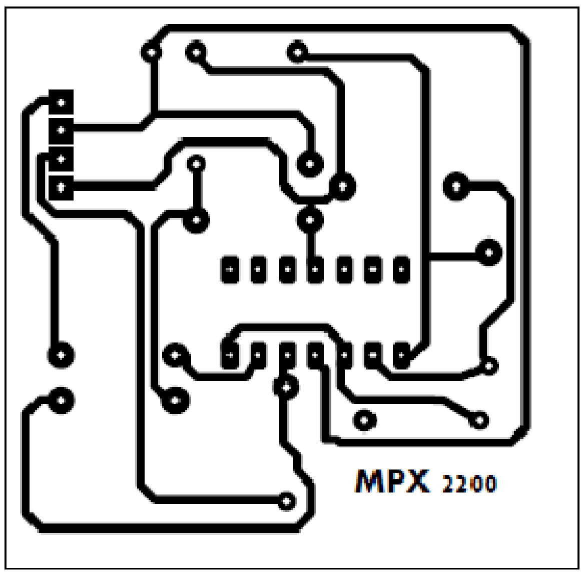


Figure A.2 pressure Sensor interfaces

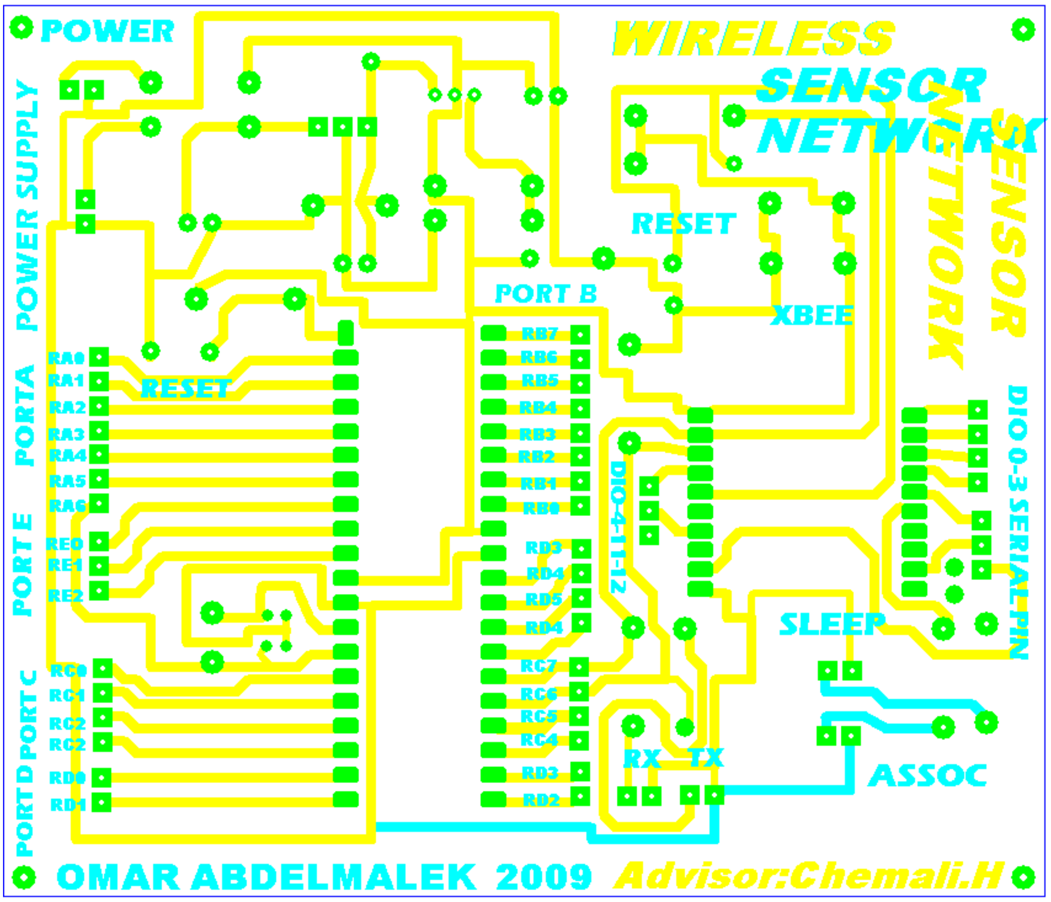


Figure A.3: NODE **Board PCB layer**

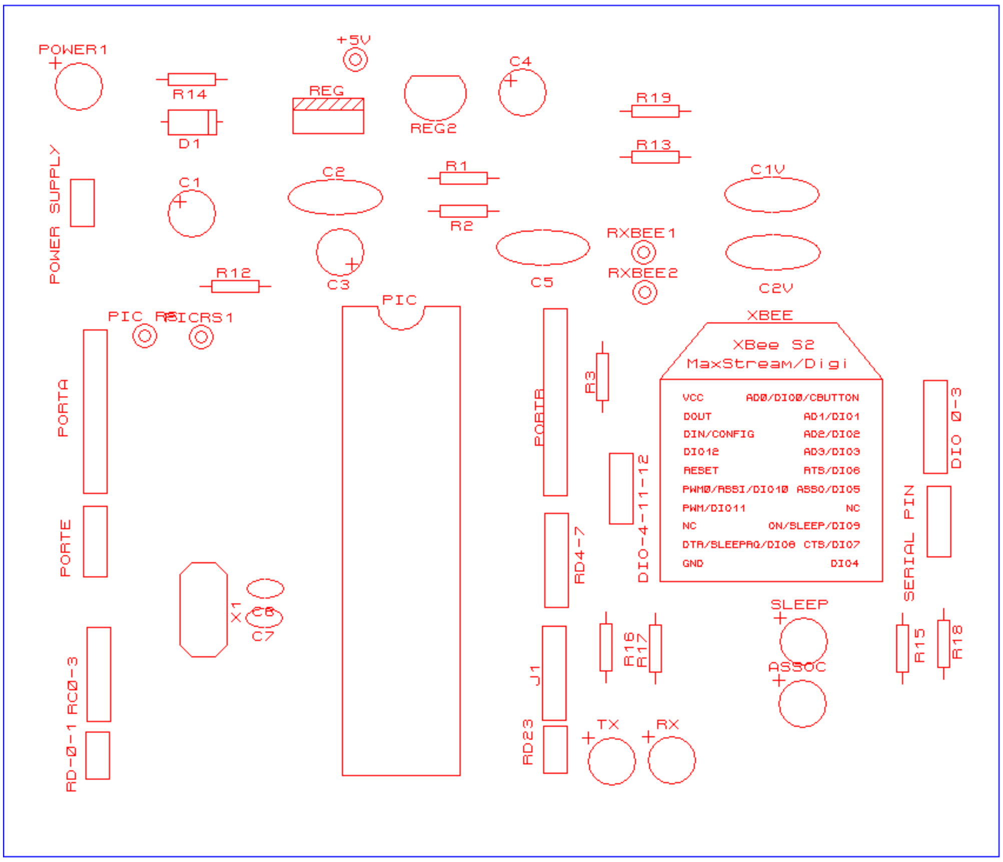


Figure A.4: NODE Components layer

```

/*******************************************************/
/******* OMAR ABDELMALEK 2009*************** ***********/
/****ESIGN AND IMPLEMENTATION OF SMART ****************/
/*********WIRELESS SENSOR NETWORK**********************/
/******* program of NETWORK DATA MONOTORING ************/
/*************************JUNE 2009*********************/
#define Lo(param) ((char *)&param)[0]
#define Hi(param) ((char *)&param)[1]
#define Higher(param) ((char *)&param)[2]
#define Highest(param) ((char *)&param)[3]

#define lo(param) ((char *)&param)[0]
#define hi(param) ((char *)&param)[1]
#define higher(param) ((char *)&param)[2]
#define highest(param) ((char *)&param)[3]


#define START_DELIMITER 0x7E // START DELIMITER
#define TX_REQUEST 0x10 //ZIGBEE Transmit Request
#define FRAME_ID 0x01 //
#define TX_REQUEST_OPTIONS 0x00 //Broadcast Raduiss


// variable uses in my programme
long temp ; // Temperature in Celcius * 10
//-----------------------------------------------------------

void ADC_setup(){
//---------- Config <AN3:AN0> as analog input -------
TRISA |= 0x0F;

//---------- Config input channel 0 -------
//---------- RC internal clock and enable ADC -------
ADCON0 = 0xC1;
//---------- Right justified, Vref+ = Vref+ pin -------
//---------- Vref- = GND -------
ADCON1 = 0x90;
}
//------------------------------------------------------------

/* offset reference of the sensor : 0°C is 500 mV => 102.4
* since the sensor is factory calibrated, there is no need for adjustment*/
int ref = 1024 ; // offset is multiplied by 10 to get tenth of degrees

unsigned char i;
unsigned char j;
unsigned char TX_REQUEST_DESTINATION_ADDRESS_64[8]; //64 bit Destination Adresse
unsigned char TX_REQUEST_DESTINATION_ADDRESS_16[2]; //16 bit Destination Adresse
unsigned char TX_REQUEST_RF_DATA[8];
unsigned char Packet_checksum = 0xFF;

//begin of my programme
void main()
{
// USART initialization
Usart_Init(9600);

// Port C initialization
TRISC = 0x80;
PORTC = 0x00;
LATC = 0x00;
```

```
//-----------------
TRISA |= 0x0F;

//---------- Config input channel 0 -------
//---------- RC internal clock and enable ADC -------
ADCON0 = 0xC1;
//---------- Right justified, Vref+ = Vref+ pin -------
//---------- Vref- = GND -------
ADCON1 = 0x90;
temp = ADRESH;
temp <<= 8;
temp |= ADRESL;

//-------------------------------------------------------------------------------------
Lo(temp) = TX_REQUEST_RF_DATA[0];
Hi(temp) = TX_REQUEST_RF_DATA[1];
Higher(temp) = TX_REQUEST_RF_DATA[2];
Highest(temp) = TX_REQUEST_RF_DATA[3];

//----------------------------------------------------------------------------------------------------
// Adresse of destination 000000000000FFFF for broadcast
TX_REQUEST_DESTINATION_ADDRESS_64[0] = 0x00;
TX_REQUEST_DESTINATION_ADDRESS_64[1] = 0x00;
TX_REQUEST_DESTINATION_ADDRESS_64[2] = 0x00;
TX_REQUEST_DESTINATION_ADDRESS_64[3] = 0x00;
TX_REQUEST_DESTINATION_ADDRESS_64[4] = 0x00;
TX_REQUEST_DESTINATION_ADDRESS_64[5] = 0x00;
TX_REQUEST_DESTINATION_ADDRESS_64[6] = 0xFF;
TX_REQUEST_DESTINATION_ADDRESS_64[7] = 0xFF;
// Adresse of destination 000000000000FFFE for broadcast
TX_REQUEST_DESTINATION_ADDRESS_16[0] = 0xFF;
TX_REQUEST_DESTINATION_ADDRESS_16[1] = 0xFE;
//---------------------------------------------------------------------------------------------------
// intialise packet check sum
Packet_checksum -= FRAME_ID;
Packet_checksum -=TX_REQUEST;
for (i = 0 ; i < 8 ; i++)
Packet_checksum -=TX_REQUEST_DESTINATION_ADDRESS_64[i];
Packet_checksum -=0xFF; // mins 16bit Adress
Packet_checksum -=0xFE;// mins 16bit Adress
for (i = 0 ; i < 8 ; i++)
Packet_checksum -=TX_REQUEST_RF_DATA[i];// mins data frame
//-------------------------------------------------------------------------------------------------------------------
// packet of data request to send in network

Usart_Write(START_DELIMITER);
Usart_Write(0x00);
Usart_Write(0X16);
Usart_Write(TX_REQUEST);
Usart_Write(FRAME_ID);

for (i = 0 ; i < 8 ; i++)
Usart_Write(TX_REQUEST_DESTINATION_ADDRESS_64[i]);

for (i = 0 ; i < 2 ; i++)
Usart_Write(TX_REQUEST_DESTINATION_ADDRESS_16[i]);

Usart_Write(0x00);
Usart_Write(0x00);
for (i = 0 ; i < 8 ; i++)
Usart_Write(TX_REQUEST_RF_DATA[i]);
Usart_Write(Packet_checksum);

delay_ms(3000);

}
```