POLITECNICO DI MILANO
Polo Regionale di Como
Facoltà di Ingegneria dell'Informazione
Corso di Laurea Specialistica in Ingegneria Informatica

Final Thesis for MSc. Computer Engineering

# RSSI Based Localization

## for Indoor Application by using 868 MHz Radio Signal

Supervisor: Prof. Fabio Salice

Shahnewaz, Ali (737700)

Tabibi, Mohammad Ali (765781)

December, 2012

# RSSI Based Localization
# for Indoor Application by Using
# 868 MHz Radio Signal

a Thesis
Submitted to the Department of
Computer Engineering
at Politecnico di Milano
in Partial Fulfillment of
the Requirements
for the Degree of
Master of Science in Computer Engineering

Supervisor: Fabio Salice, Ph.D.

Mohammad Ali Tabibi (765781)
Ali Shahnewaz (737700)

DECEMBER 2012

# Abstract in English

A wireless sensor network is a collection of nodes which are in together establish a network to do a certain work. Each node of the network consists of processing capability and have a RF transceiver and a power supply. Now a days, wireless sensor networks has become a key technology for different types of smart environments, and an intense research effort is currently underway to enable the application of wireless sensor networks for a wide range of industrial problems. Wireless networks are of particular importance when a large number of sensor nodes have to be deployed, and/or in hazardous situations.

Localization in wireless sensor network means to locate node which may be in of some fixed location or mobile devices. One example has been in the supervision of humidity and temperature in forests and/or fields, where thousands of sensors are deployed by a plane, giving the operator little or no possibility to influence the precise location of each node. An effective localization algorithm can then use all the available information from the wireless sensor nodes to infer the position of the individual devices. WSN localization techniques are used to estimate the locations of the sensors with initially unknown positions in a network using the available a priori knowledge of positions of a few specific sensors in the network and inter-sensor measurements such as distance, time difference of arrival, angle of arrival , connectivity, RSSI etc.

This thesis work is carried out on localization of Wireless Sensor Network based on Receive Signal Strength (RSSI). Focusing on localization processes, we will first give an overview of the state of the art in this area. From the various techniques, one idea was

found to have significant bearing for the development of a new algorithm. We present analysis and simulations of the algorithms, demonstrating improved accuracy compared to other schemes . A third aspect of the work concerns the feasibility of approaches based on received signal strength indication (RSSI). Data are collected from different indoor environments and also in outdoor.

# Abstract in Italian

Una rete di sensori wireless è puo' essere definita come collezione di nodi che operano insieme per svolgere un certo lavoro. Ogni nodo della rete ha capacità di elaborazione e di ricevere e trasmettere dati (tipicamente in RF). Le reti di sensori wireless appartengono alla famiglia delle tecnologie per la realizzazione di ambienti intelligenti (nell'ambito dei sistemi pervasivi) e un intenso sforzo intenso lavoro di ricerca è attualmente in corso per consentire l'applicazione di reti di sensori wireless per una vasta gamma di problemi sia civili che industriali. Un esempio è il controllo di umidità e temperatura nelle foreste e campi, dove migliaia di sensori, lanciati da un aereo, convogliano i dati verso un operatore che, tra le varie cose, ha poca possibilità di influenzare la posizioneprecisa di ogni nodo.

Localizzare utilizzando di una rete di sensori wireless significa identificare la posizione di uno o più elementi, fissi o mobili, attraverso una rete di rilevazione. Un algoritmo di localizzazione efficace utilizza tutte le informazionidisponibili dai sensori wireless per dedurre la posizione dei singoli dispositivi. Tecniche di localizzazione basate su reti di sensori sono utilizzate per stimare le posizioni dei sensori attraverso le posizioni inizialmente sconosciute e utilizzando la conoscenza a priori della posizioni di qualche sensore della rete (sensori di calibrazione) e di misure inter-sensore come la distanza, la differenza di tempo di arrivo, l'angolo di arrivo, la connettività, e l'RSSI.

In questo lavoro di tesi è implementato un sistema di localizzazione basato sulla ricezione della intensità del segnale (RSSI). Il lavoro si sviluppa nei seguenti passi. Dopo aver presentato il processo di localizzazione, viene sviluppata una panoramica sullo stato dell'arte in questo campo. Nel seguito viene descritta la tecnica utilizzata; questa riunisce degli aspetti e delle tecniche che sono statiritenuti i più promettenti per la risoluzione del problema in esame: localizzazione wireless in ambienti chiusi attraverso l'uso di RF in sub GHz; a supporto del lavoro svolto vengono presentate sia l'analisi e che la simulazione degli algoritmi dimostrando, cosi', che il metodo proposto e' caratterizzato da una maggiore precisione rispetto ad altri metodi. Un terzo aspetto del lavoro riguarda la fattibilità di approcci basati su indicazione di intensità del segnale ricevuto (RSSI). I dati sono raccolti da differenti ambienti interni ed anche in esterno.

# Acknowledgment

We would like to thank our admirable advisor *Prof. Fabio Salice* deeply, for his valuable and helpful advice, support and encouragements during this project. Also we have to mention the name of *Mr. Roberto Galli* who really helped us in hardware implementation part. Special thanks to *Prof. Matteo Cesana* for his useful and important advice and guidance at the beginning of the project. In addition we would like thank to all staff and professors of *Politecnico di Milano University*, especially *Como Campus*, for all their admirable and worthy work and dedication.

And finally we want to express our love and gratitude to our dears families for their endless love, supporting and understanding during our studies.

# Contents

# Table of Figures

# List of Tables

# ***Chapter 1:*** Wireless Sensor Network Localization in Indoor Environments

## 1.1- Introduction

Distributed sensor networks have been discussed for more than 30 years, but the vision of wireless sensor networks (WSNs) has been brought into reality only by the recent advances in wireless communications and electronics, which have enabled the development of low-cost, low-power and multi-functional sensors that are small in size and communicate over short distances. Today cheap and smart sensors, are networked through wireless links and deployed in large numbers, provide unprecedented opportunities for monitoring and controlling homes, cities, and the environment. In addition, networked sensors have a broad spectrum of applications in the defense area, generating new capabilities for reconnaissance and surveillance as well as other tactical applications

Localization is an important aspect in the field of wireless sensor networks that has attracted significant research interest recently. The interest in wireless sensor network localization is expected to grow further with the advances in the wireless communication techniques and the sensing techniques, and the consequent proliferation of wireless sensor network applications. Wireless sensor networks are particularly interesting in hazardous or remote environments, or when a large number of sensor nodes have to be deployed.

The localization issue is important where there is an uncertainty about some positioning. If the sensor network is used for monitoring the temperature in a building, it is likely that we can know the exact position of each node. On the contrary, if the sensor network is used for monitoring the temperature in a remote forest, nodes may be deployed from an airplane and the precise location of most sensor may be unknown. An effective localization algorithm can then use all the available information from the motes to compute all the positions.

## 1.2- Problem Formulation

Localization is the problem of determining the geographical location of each node in the system. Localization is one of the most fundamental problems that must be solved for WSN. The aim of this thesis is to find the position of a mobile object which is equipped with a microcontroller that send RSSI information iteratively over 868 MHz channel. There are some other nodes which are used to receive signals from object and send to server over the master node. As the network is based on tree hierarchical manner where root node is the server which mainly perform the localization, *Master* is the node which control the *Sensor Nodes*. These sensor nodes are connected with one shared bus (RS-485) when a specific sensor is connected with master it will send the RSSI packet to the master which is received from the *Moving Object*.

The goal of this thesis is, using RSSI information from the sensor it locates the position of the moving object. To be more accurate and precise there are several steps involving with that procedure, such as RSSI optimization which involves for each sensor. Instead of using one sample RSSI that receive by a sensor, It uses a fixed sample size of RSSI to optimize its value. On the other hand *Channel Identification* procedure stands for identification of RSSI channels namely *Non Line of Sight* and *Line of Sight*. By using *Log Normal Path Loss Model*, *Server* calculate the distance of the moving object from the sensor i.e. d1,d2, ,d3 for the sensors [1, 2, 3]. These distance is used to locate the object such as trilateration method.

This work involves some components which are as follows: (These modules will be introduced in detail further.)

- ***Moving Object***

A small microprocessor transceiver with 868 MHz wireless module. This mobile object which is equipped with a microcontroller will send RSSI information iteratively through 868 MHz channel and has to be localized.

- ***Sensors***

A small microprocessor transceiver with 868 MHz wireless module. Sensor are placed in a different location in a specific indoor environment and after placing the sensor node , the location of the sensor node is known. This node is used to receive the signals from the moving object through the 868 MHz channel and calculate RSSI. This RSSI then send to the server over the master.

- ***Master***

A small microprocessor. Master are places in one position and it is connected with the main RS485 Bus and also connected with server using RS-232 serial communication. Master performs two important jobs, one is to maintain the network by broadcasting different data packet through RS485 data bus and establish connection one at a time to the each sensor to receive its message. That message then send to server using one predefined packet protocol over RS232 .

- ***Server***

This is a Windows Form application developed by *Microsoft Visual Studio 2008* framework and used language is C#. It has one communication layer which involve to establish connection between server and the master through RS-232. The mathematical tools used in this thesis work is also developed by C#, and graph are developed by Microsoft charting library. Server performs localization algorithm and estimate distance of the moving object from each receive sensors. These distances are used calculate the co-ordinate of the moving object.

- ***RS-485 Communication***

RS-485 are serial communication methods for computers and devices. RS485 is the most versatile communication standard and network topology is probably the reason why RS-485 is now the favorite of the four mentioned interfaces in data acquisition and control applications. RS-485 is the only of the interfaces capable of internetworking multiple transmitters and receivers in the same network

- ***RS-232 Communication***

Communication as defined in the RS-232 standard is an asynchronous serial communication method. The word serial means, that the information is sent one bit at a time. Asynchronous tells us that the information is not sent in predefined time slots. Data transfer can start at any given time and it is the task of the receiver to detect when a message starts and ends. In this thesis work, it used start bit 1 , parity bit none , and baud rate 9600.

- ***868 MHz Wireless Channels***

This spectrum was recommended and adopted by the *Frequency Management, Regulatory Affairs and Spectrum Engineering Working Groups* as well as the *European Conference of Postal and Telecommunications Administration*, as a long range solution for the European market. The respective radio spectrum is specific to the European market and falls within the 868.000 - 870.000 MHz frequencies.

- ***Network Topology***

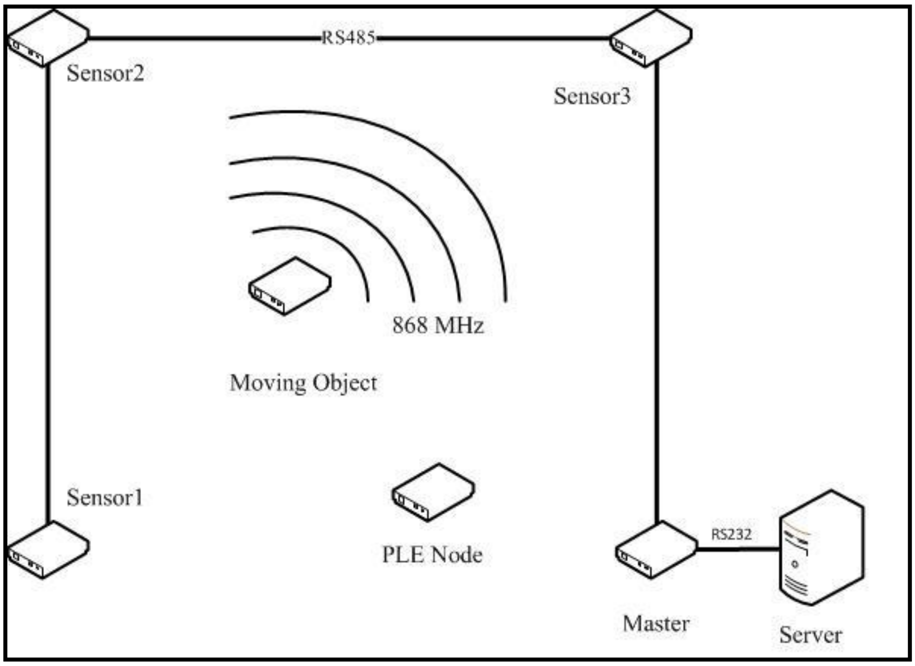


Figure 1. Schematic of WSN Localization Components

## 1.3- Research Approach

Politecnico di Milano involves many different areas of research regarding the wireless sensor localization. The previous project regarding Localization was LAUREA project which performs localization, tracking and monitoring of WSN based in IEEE 802.15.4 (Zigbee) standard. Its method is based on RSS measurement. In this project the location of anchor nodes or sensors are fix and clear and RSS measurement are collected between pairs of nodes since they are related to the inter-node distances. In case of moving client nodes the RSSI fluctuations are too high and indeed the accuracy of the system will be too low which they by using the knowledge about the moving node and the geometry of the environment improved this localization accuracy in three steps: Prediction, Update and Initialization. Finally with RSSI and computing distance, it is possible to estimate the position of the client node with below 2.5 meter error according to the zero-configuration indoor localization.

Localization in indoor application there many approaches is used for several years. With literary survey we found most of them are using 2.4 GHz bandwidth. In our approach we use 868 MHz bandwidth which is narrower than previous approach. Due to indoor application contains a lot of uncertain obstacles with different types. In our approach , we found that 868 MHz channels gives better result considering the presence of different obstacles.

Other expect of the work involves Error minimization which deals one polynomial function. To optimize the result, this work also involves one more step called RSSI optimization. Mainly a sample of RSSI for each Sensor is used to define one optimize RSSI by using filters.

The evaluation of various algorithms and methods is done by extensive simulations, studying the accuracy and repeatability of the results. An extra study about different parameters that could affect the measurements accuracy could describe their different effects as for example to illustrate the case of a sensor mote running in low battery.

## 1.4– Thesis Contribution

The contribution of thesis can be summarized as follows which all of them have been described completely in next chapters.

- Considering the difference microcontroller and transceiver and choosing the suitable package for the implementing the system
- Finding and considering the different methods and algorithms for localization and choosing the best one and defining the best and suitable system for implementing it
- Implementing the Hardware and firmware of each parts of the system and define theirs instructions
- Defining an improved procedure of the localization based on RSSI

- Implementing the software of server side for localization procedure
- Doing the several indoor and outdoor tests to finding the accuracy and error of the system in different environments by different situations

Generally according to main goal of the project, our work tried to find that the 868 MHz frequency is suitable and accurate for localization in field of wireless sensor network or not? Also by considering the effects of changing in temperature, humidity and etc. we have tried to find the accuracy of the system.

## 1.5 – Thesis Book Organization

As mentioned before in this chapter we tried to propose the problem and our contribution to localized the position of an unknown moving object according to power of its signals in 868 MHz frequency which are received by some sensors. The rest of this thesis has been organized as follow:

- Chapter 2 is a brief introduction to the Wireless Sensor Network localization methods.
- Chapter 3 considers the Received Signal Strength Indicator briefly and introduces the signal power issues and finally proposes the Radio Propagation mechanism.
- Chapter 4 explains the background and related works in this field.
- Chapter 5 is for presenting the localization procedure in field of Wireless Sensor Network. Also in this chapter we have brief definition of each component of the system.
- Chapter 6 is about hardware implementation which introduces the microcontroller, transceiver and programmer of the embedded system which we used. Also the algorithm of firmware for each component has been presented.
- Chapter 7 is about the software implementation which is related to the duty of the server for localization.
- Chapter 8 is the measurement and result of the system in different situation which has been presented by several tables and graphs.
- Chapter 9 is application of the system. Where this system would be useful?
- Chapter 10 concludes by summarizing the contributions and the direction of future research based on this thesis is also discussed.

# ***Chapter 2:*** Introduction to the Wireless Sensor Network Localization Methods

## 2.1 – Introduction

Localization can be defined as estimation of location or position estimation of a substance. Reliability matters and due to the difficulties for the presence of many factors, it is challenging to implement in real world and with availability sensor wireless devices, it is becoming an interesting and important research filed in wireless sensor networks. It has a significant application in area of smart home application as well as pervasive computing systems. Development in the wireless communications technology and the electronics it is now possible to have a low-cost, low-power and multi-functional sensors that are small in size and can be communicate over short distances. With the availability of the low cost smart sensor nodes with the networked through wireless links, it is possible to provide opportunities for monitoring and controlling indoor such as smart home application, cities ,even though networked sensors have a broad spectrum of applications in the defenses area and surveillance system.

The algorithms for localization are divided in two categories, Range-base and range-free.

- ***Ranged based localization :***

Estimating the node position in this kind of algorithms depends on vicinity or closeness sensing or connectivity information. CPE, APIT, Centroid and Distributed algorithms are used in this category.

- ***Range free localization :***

In this category algorithm by measuring some factors such as Time of Arrival (TOA), Time Difference of Arrival (TDoA), Received Signal Strength (RSS) and also Angle of Arrival (AoA), estimate the distance between nodes.

## 2.2 - Underneath Technology

Localization of the Wireless sensor network techniques typically organized in to such way that there are some sensors are distributed over a region whose position are not known. The system knows position of the some of the sensors are often called ANCHOR node. The anchor is used (basically the position of the anchor node) to recognize the unknown position of the node. To estimate the position (localize) of sensors information such as distance, time difference of arrival, angle of arrival and RSSI.

Position of the anchor is obtained manually typically placed the anchor nodes into known position as priori information. The measurement technique is essential for the accuracy of the localization .

## 2.3 - Measurement Techniques

Measurement is an important steps in the area of Wireless sensor network localization. To measure sensors in a given network, sensor given information (basically signals) is measure with angle-of-arrival (AOA) measurements, time-difference-of arrival (TDOA) and received signal strength (RSS) indicator.

Localization algorithm has close relation to the measurement techniques. On the basis of the measurement technique is used , the localization algorithm varies.

Most of the existing location discovery methods consist of two basic phases:

- ***Distance (or angle) estimation***
- ***Distance (or angle) combining***

The most popular methods for **estimating** the distance between two nodes are:

### *2.3.1 - Angle-of-Arrival Measurements: (AOA)*

The Angle of Arrival measurements or the direction of arrival measurements obtained by using mainly two methods, by Using the ***1)*** receiver antenna's amplitude response and ***2)*** the receiver antenna's phase response.

Angle of arrival measurement techniques using phase interferometry is based on the phase differences in the arrival of a wave front. An array of antenna or a big receiver antenna is needed to implement this method.

Amplitude is measured in the receiver antenna. Anisotropy in the reception pattern has impact on the amplitude response of the antenna .The size of the measurement unit may be small with regards to the wavelength of the signals. To implement this method typically anisotropic antenna and beam pattern is used. When the beam of the receiver antenna is rotated ,the direction corresponding to the maximum signal strength is taken as the direction of the transmitter .The sensitivity , beam width are responsible to get the accuracy of the measurements

Anisotropy in the reception pattern involve some potential problem regarding signal strength variation. Another secondary non-rotating and omnidirectional antenna at the receiver is placed to deal with this problem. The impact of varying signal strength is largely removed by normalizing the signal strength received by the rotating anisotropic antenna with respect to the signal strength received by the non-rotating omnidirectional antenna.

Angle of arrival measurements rely on a direct line-of-sight (LOS) path between the transmitter and the receiver. The other factors are also influenced the measurements are shadowing and multipath. A multipath component from the transmitter signal may appear as a signal coming from an entirely different direction and consequently causes a very large error in the Angle of arrival measurement. Maximum likelihood algorithms is use to deal Multipath problems. Depending on the statistical characteristics of the transmitter signals, deterministic maximum likelihood algorithms and stochastic deterministic maximum likelihood algorithms are introduced.

AOA is defined as the angle between the propagation direction of an incident wave and some reference direction, which is known as orientation. Orientation, defined as a fixed direction against which the AOAs are measured, is represented in degrees in a clockwise direction from the North. The AOA is absolute when the orientation is 0 degree or pointing to the north. By using the antenna array on each sensor node we can measure the AOA.

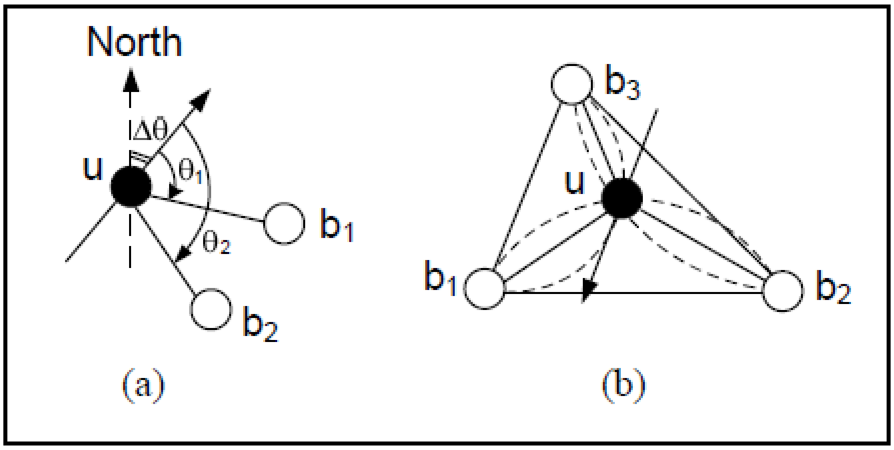


Figure 2. Triangulation in AOA localization: (a) Localization with orientation information; (b) Localization without orientation information

- If the orientation of the unknown node is known, we have:

$\theta_1$: *AOA of the signal sent from node $b_1$ to node U*

$\theta_2$: *AOA of the signal sent from node $b_2$ to node U*

$\Delta\theta$: *The orientation of the U*

The absolute AOAs from $b_1$ and $b_2$ = $(\theta_i + \Delta\theta)(mod 2\pi)$ i = 1, 2

The location of the unknown node is located at the intersection of all rays when two or more non-collinear nodes are available.

- When the orientation of the unknown node is not available, we cannot obtain the absolute AOAs and instead of that we can use the difference of the AOAs. For example in Fig. 1(b) we can calculate the angles $b_1ub_2$, $b_1ub_3$ and $b_2ub_3$ by using the knowledge of the relative AOAs. In this case all angles subtended by the same chord are equal. So by each two nodes and a third point from which the chord subtends a fixed angle is constrained to an arc of a circle. Since each chord determines one arc, he location of the unknown node is at the intersection of all arc when three or more non-collinear nodes are available.

### *2.3.2 - Time-Difference-of-Arrival Measurements: (TDOA)*

Time-Difference-Of-Arrival, short form is TDOA is another way between the receivers to measure the difference of the arrival times of signal. This method based on the time difference .The receivers has synchronized time and the method is done by comparing the time of the receiver. The estimation techniques used to measure the time are the general method which subtracting the time of a signal arrival between receiver measurements from two base stations to produce a relative TDOA, or through the use of cross-correlation techniques, in which the received signal at one base station is correlated with the received signal at another base station.

Other is pattern marching algorithm named as cross-correlation of the two versions of the signal and determine the time lag between them. The cross correlation has the maximum value with 0 lag time. If receiver receive identical signal but at different time lag the cross co relation index high at that lag time when it is measure with different lag time.

By measuring the TDOA of signals which have been emitted from three or more synchronized transmitter with known location helps to a single receiver to find its position by multilateration method.

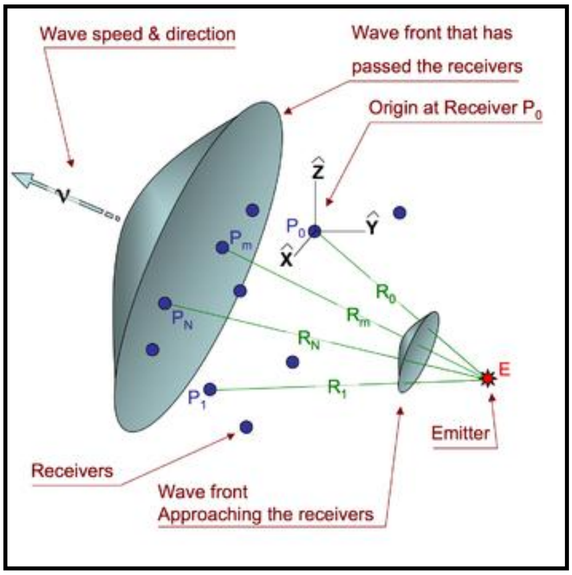


**Figure 3. TDOA Geometry**

The principle of time difference measurements in hyperbolic mode is presented in below figures:

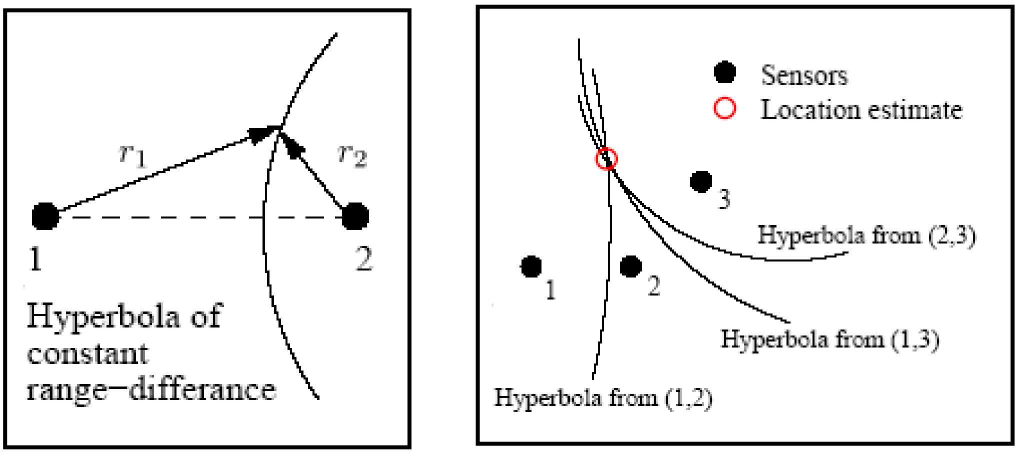


Figure 4. Hyperbolic Location Theory

The hyperbolic is the set of points at a constant range-difference ($C_0\Delta t$) from two sensors and each sensor pair gives a hyperbola on which the emitter lies. Location estimation is intersection of all hyperbolas.

Imagine unknown transmitter (Emitter) at the point E and N+1 known receiver in $P_i$, where i = 0 … N .

$P_m = (x_m, y_m, z_m)$ $0 \le m \le N$

R is the distance from the transmitter to one of the receiver:

$$R_m = |\vec{P}_m - \vec{E}| = \sqrt{(x_m - x)^2 + (y_m - y)^2 + (z_m - z)^2}$$

$$R_0 = \sqrt{x^2 + y^2 + z^2}$$

v is wave speed, $T_m$ is transit time and $t_m$ time is the time difference of a wave front touching each receiver, so:

$$vt_m = vT_m + vT_0 = R_m - R_0$$

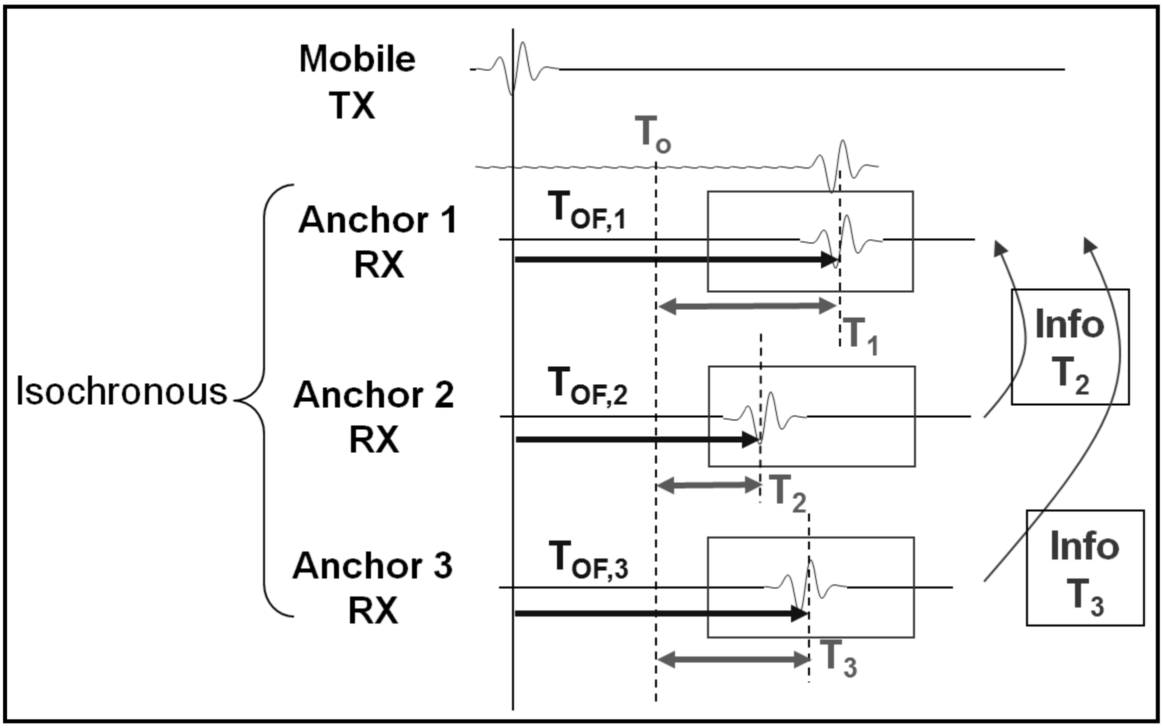


Figure 5. TDOA

In case of positioning from TFOA at least we need 3 anchors (sensors) with known positions to find a 2D-position from a couple of TDOAs.

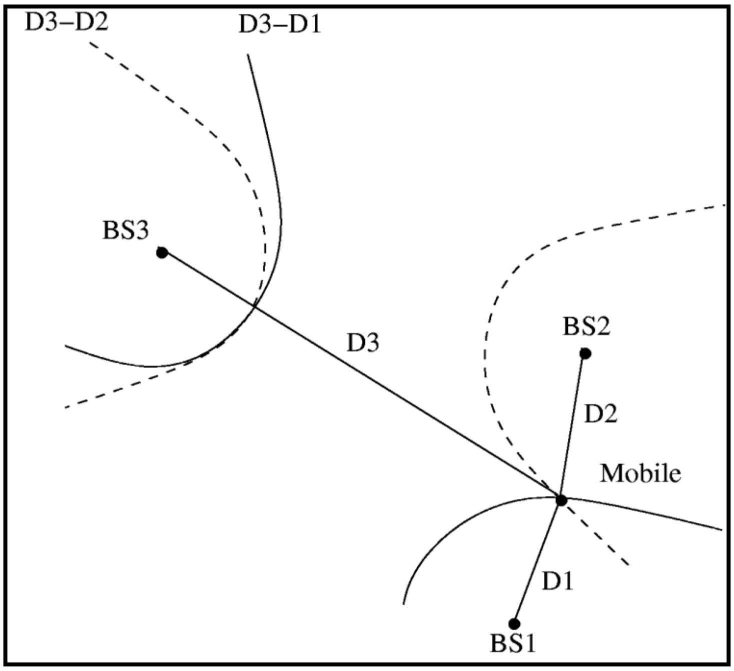


Figure 6. Positioning by TDOA

BS1, BS2 and BS3 are known anchors (sensors) and mobile is unknown moving object which its position should be found.

$$d_{32} = \sqrt{(x_3 - x_M)^2 + (y_3 - y_M)^2} - \sqrt{(x_2 - x_M)^2 + (y_2 - y_M)^2}$$
$$d_{31} = \sqrt{(x_3 - x_M)^2 + (y_3 - y_M)^2} - \sqrt{(x_1 - x_M)^2 + (y_1 - y_M)^2}$$

We have to remember that TDOA can operate in one of two below modes:

- The moving object receives multiple reference pulses and calculates the TDOA
- The moving object transmits a reference pulse which is received by multiple fixed nodes, then the fixed nodes must forward the TDOA information to a workstation which then runs the hyperbolic location algorithms

### *2.3.3 - Time Of Arrival Measurements (TOA)*

Another technique name Time Of Arrival which measure the time that the signal takes from the source to the receiver through a link. This may be done by measuring the time in which the mobile responds to an inquiry or an instruction transmitted to the mobile from the base station. The total time elapsed from the instant the command is transmitted to the instant the mobile response is detected, is composed of the sum of the round trip signal delay and any processing and response delay within the mobile unit**.** If the processing delay for the desired response within the mobile is known with sufficient accuracy, it can be subtracted from total measured time, which would give us the total round trip delay.

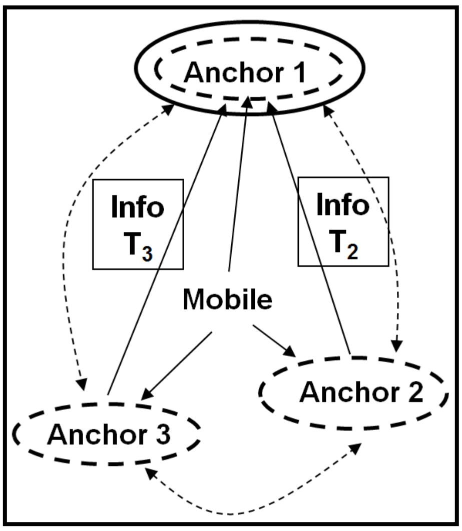


Figure 7. Isochronous TOA Estimation -- and TDOA Estimation –

If $T_1$, $T_2$ and $T_3$ be the TOA estimations, $t_{21}$ and $t_{23}$ are the TDOA estimations according to below equations:

$$T_{21} = T_1 - T_2 \rightarrow d_{21} = t_{21}.v$$

$$T_{23} = T_3 - T_2 \rightarrow d_{23} = t_{23}.v$$

In TOA method, three or more sensors measure the TOAs of the transmission from the target which each of them makes a circle and the intersections of the circles give the target location. Finding the a location that best fits the measurement is necessary to decrease the error in situation which circles do not intersect at a unique point. Also arriving signal at sensors from reflections in case of no Line-Of-Sight (LOS) path, leads the estimation to large errors.

In two dimensional sensor field where n sensors and one target exist, for each sensor the TOA measurement determines a circle centered at the sensor and the radius is equal to the TOA measurement multiplied by the light speed. The target must locate on each circle if there is no Non-Line-Of-Sight (NLOS) errors and noises. Then the intersection of three such circles is the position of target.

$$R_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad , \quad i = 1,2, \dots , n$$

- (x, y): *Real position of the target*
- $(x_i, y_i)$: *Position of the sensor i*

The TOA measurement for the sensor i is:

$$t_i = \frac{R_i}{v_c} + w_d + u_i$$

- $W_d$: *TOA measurement noise*
- $U_i$: *NLOS range error*

$w_d$ is additive white Gaussian noise, whose distribution is the normal probability distribution function $N(0,\delta_d^2)$ with zero-mean and the variance of the TOA measurement noise $\delta_d^2$.

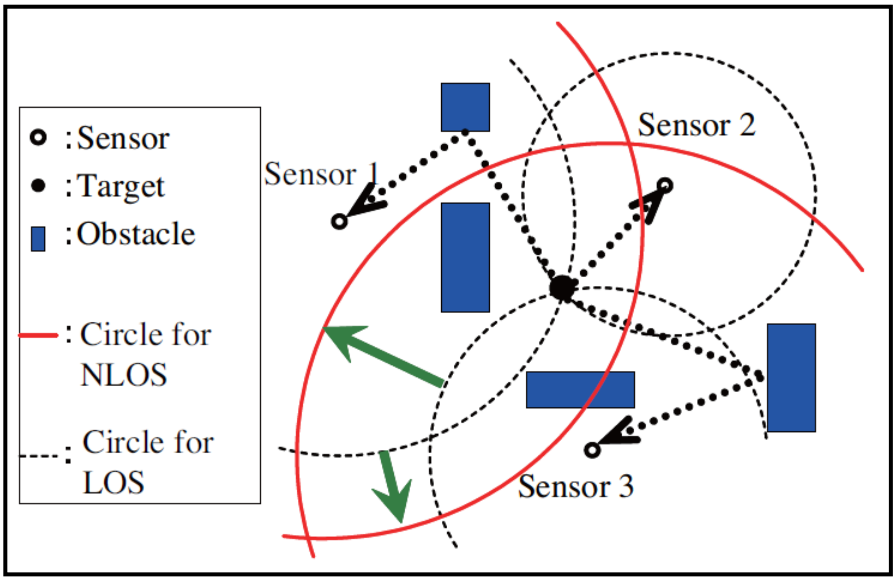


Figure 8. Sample of TOA measurement under LOS and NLOS situation

### *2.3.4 - Received Signal Strength Indicator (RSSI)*

This technique measure the power of the signal at the receiver. Based on the known transmit power, the effective propagation loss can be calculated. Theoretical and empirical models are used to translate this loss into a distance estimate. This method has been used mainly for RF signals.

Received signal strength is based on the Power observations. According to this method the sensor estimate the received signal power by means of integrating signal power of a certain frequency band as a estimation to the received signal energy. Evaluating the transmitting power of a antenna RSS gives a wide range of information. So RSSI is a measurement of the power present in a received radio signal and also we can say is an indication of the power level being received by the antenna.

RSSI is use in a range based application having a predefined frequency band. The frequency of the emitting Radio signal of a transmitter and the gain of the antenna is consider to achieved a proper RSSI information. No methods are without limitation. Dealing RSSI in a indoor application is a complicated matter. There are too many factor can influence the RSSI.

The principal of RSSI ranging describes the relationship between the transmitted power and received power of wireless signal and the distance among nodes.

(Received Power) $P_r$ = (Transmitted Power) $P_t . (1/d)^n$ → $10logP_r = 10logP_t – 10nlogd$

d: *distance between sending and receiving nodes*

10logP: *expression of power converted to dBm*

$$\rightarrow \quad P_R(dBm) = A - 10nlogd$$

So the relationship between the RSSI and d can be determine by A and n parameters.

For the combining phase, the most popular methods are:

### 2.3.5 – Hyperbolic Trilateration

The most basic and intuitive method is called hyperbolic trilateration. It locates a node by calculating the intersection of 3 circles.

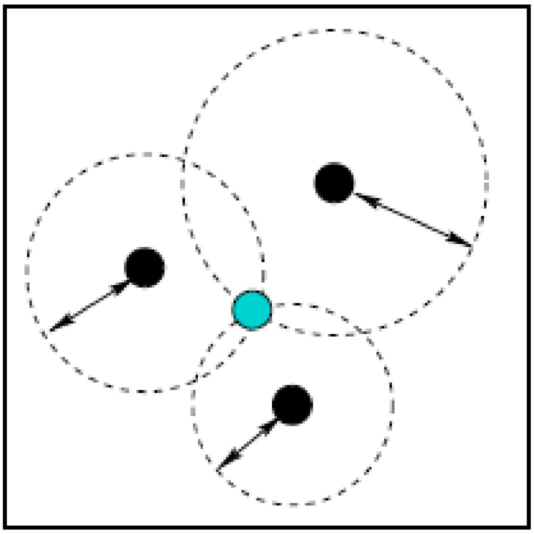

Figure 9. Hyperbolic Trilateration

### 2.3.6 – Triangulation

*Triangulation* is used when the direction of the node instead of the distance is estimated, as in AoA systems. The node positions are calculated in this case by using the trigonometry laws of sines and cosines.

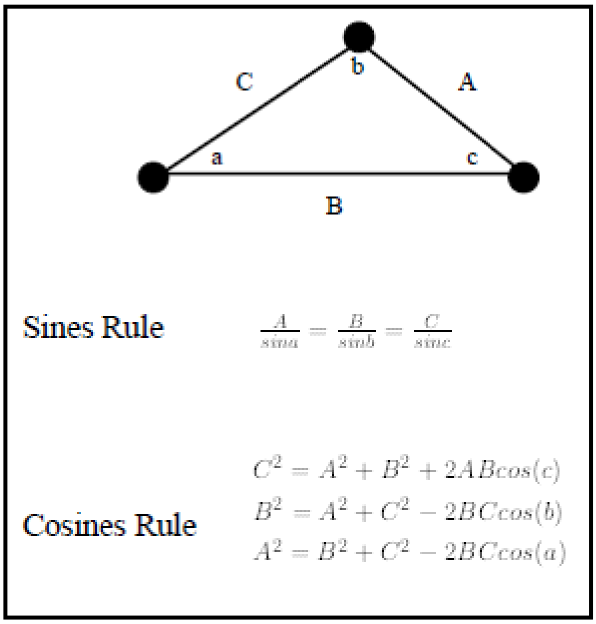


Figure 10. Triangulation

### 2.3.7 – Maximum Likelihood (ML) Estimation (Multilateration)

The third method is *Maximum Likelihood (ML) estimation (Multilateration)*. It estimates the position of a node by minimizing the differences between the measured distances and estimated distances.

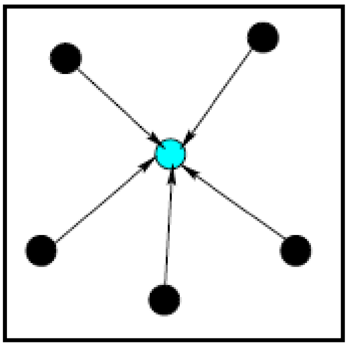

Figure 11. Multilateration

Hear the nodes with known positions are named as beacon and those with unknown positions as unknown nodes. The goal is to estimate the positions of as many unknown nodes as possible in a fully distributed fashion. The beacon nodes broadcast their locations to their neighbors. Neighboring unknown nodes measure their separation from their neighbors and use the broadcasted beacon positions to estimate their own positions. Once an unknown node estimates its position, it becomes a beacon and broadcasts its estimated position to other nearby unknown nodes, enabling them to estimate their positions. This process repeats until all the unknown nodes that satisfy the requirements for multilateration obtain an estimate of their position. This process is defined as *iterative multilateration* which uses *atomic multilateration* as its main primitive.

- *Atomic Multilateration:*

This method makes up the basic case where an unknown node can estimate its location if it is within range of at least three beacons.

- *Iterative Multilateration:*

The iterative multilateration algorithm uses atomic multilateration as its main primitive to estimate node locations in an ad-hoc network. The algorithm starts by estimating the position of the unknown node with the maximum number of beacons using atomic multilateration. Since at a central location all the entire network topology is known so starts from the unknown node with the maximum number of beacons to obtain better accuracy and faster convergence (in the distributed version an unknown will perform a multilateration as soon as information from three beacons). When an unknown node estimates its location, it becomes a beacon. This process repeats until the positions of all the nodes that eventually can have three or more beacons are estimated.

- *Collaborative Multilateration:*

In an ad-hoc deployment with random distribution of beacons, it is highly possible that at some nodes, the conditions for atomic multilateration will not be met; i.e. an unknown node may never have three neighboring beacon nodes therefore it will not be able to estimate its position using atomic multilateration. When this occurs, a node may attempt to estimate its position by considering use of location information over multiple hops in a process we refer to as collaborative multilateration. If sufficient information is available to form an over-determined system of equations with a unique solution set, a node can estimate its position and the position of one or more additional unknown nodes by solving a set of simultaneous quadratic equations.

# *Chapter 3:* Received Signal Strength Indicator (RSSI)

## 3.1 – Introduction

Wireless network is a system that transmits and receives radio signals over the air. Radio signal is used to transmit and receive information, in words the radio signal makes the communication channel .

Wireless communications make use of electromagnetic waves to send signals across long distances

## 3.2 - Radio Wave Basics

Radio wave are electromagnetic wave which travel through medium such air. Electromagnetic wave have electric field and magnetic field which oscillating time by time. The amplitude of the propagating wave decrease according to the increase of the distance from the source when it travel through medium.

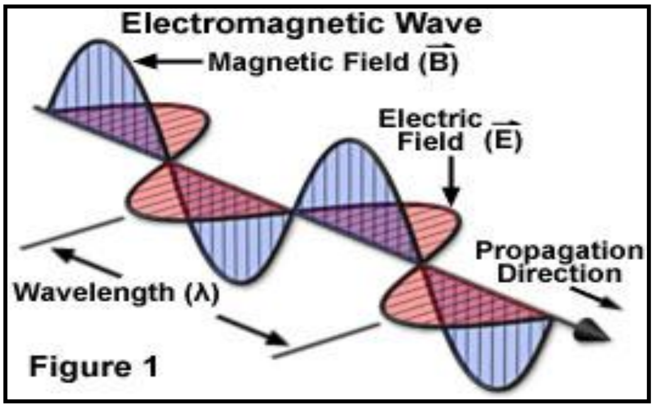


Figure 12. Electromagnetic Wave

Fig.12 illustrated the basic of the electromagnetic wave, where magnetic field and the electric field are perpendicular to each other.

Radio engineering often use some terms which are defined as follows :

- ***Period:*** Period of the wave can be define as a time required to finish one wave cycle or other words time elapsed between the start of the wave and finish of the wave.
- ***Frequency:*** Frequency can be define as the number of cycles can be completed in a second.
- ***Polarization:*** Polarization describes the direction of the electrical field vector.
- ***Bandwidth:*** Bandwidth is measured for communication channel as the difference between the lowest frequency and the highest frequency. It has another meaning that the amount of data it can transmit or receive.

## 3.3 - ISM band

ISM or Industrial Scientific and Medical radio bands or radio spectrum for use of radio frequency energy for industrial, scientific and medical reasons other than communications have been reserved. Radio-Frequency process heating, Microwave ovens and medical diathermy machines are some examples of applications in these bands. In general, communications equipment operating in these bands must tolerate any interference generated by ISM equipment, and users have no regulatory protection from ISM device operation. Despite the intent of the original allocations, in recent years the fastest-growing uses of these bands have been for short-range, low power communications systems.

The ISM bands defined by the ITU-R are:

| Frequency range | | Bandwidth | Center Frequency | Availability |
|---|---|---|---|---|
| 6.765 MHz | 6.795 MHz | 30 KHz | 6.780 MHz | Subject to local acceptance |
| 13.553 MHz | 13.567 MHz | 14 KHz | 13.560 MHz | |

| | | | | |
|---|---|---|---|---|
| 26.957 MHz | 27.283 MHz | 326 KHz | 27.120 MHz | |
| 40.660 MHz | 40.700 MHz | 40 KHz | 40.680 MHz | |
| 433.050 MHz | 434.790 MHz | 1.84 MHz | 433.920 MHz | Region 1 only and subject to local acceptance |
| 902.000 MHz | 928.000 MHz | 26 MHz | 915.000 MHz | Region 2 only |
| 2.400 GHz | 2.500 GHz | 100 MHz | 2.450 GHz | |
| 5.725 GHz | 5.875 GHz | 150 MHz | 5.800 GHz | |
| 24.000 GHz | 24.250 GHz | 250 MHz | 24.125 GHz | |
| 61.000 GHz | 61.500 GHz | 500 MHz | 61.250 GHz | Subject to local acceptance |
| 122.000 GHz | 123.000 GHz | 1 GHz | 122.500 GHz | Subject to local acceptance |
| 244.000 GHz | 246.000 GHz | 2 GHz | 245.000 GHz | Subject to local acceptance |

**Table 1. The ISM bands**

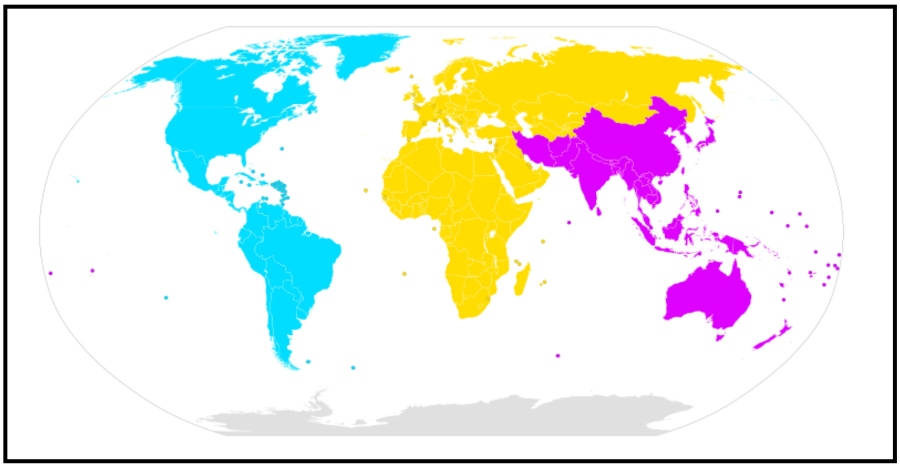

**Figure 13. ITU regions. Region 1 Region 2 Region 3**

## 3.4 - Radio Wave Transmission model

Basically, Radio signal transmission model consist of a Antenna (Transmitter antenna, Receiver antenna or Transceiver antenna) and transmitter/receiver module.

The original data which has to be send through the transmitter antenna need to amplified first to due to the loss during propagation through the medium. The same thing is also need to the receiver part as well.

There are wide range of transmitter and receiver wireless module exist in the present market which can be used as it is with the external hardware such as microcontroller who needs to send data packet to the medium or vice versa. These module has already implemented antenna gain, pre amplifier etc.

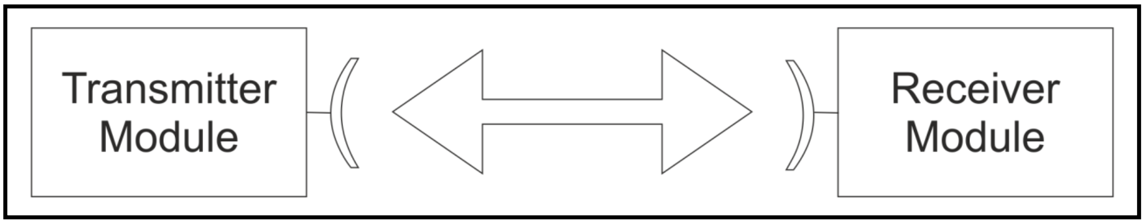


Figure 14. RF transmission link

Propagation model can be defined as the relation between the signal radiated and signal received as a function of distance and other variables. There are can be different propagation model such as indoor propagation model, free space propagation model and etc.

Based on the propagation model it varies how to model data, because of the different environment model  and different transmission characteristics. Apart form the underneath algorithm used in the system or the system model, communication between the transmitter and the receiver follows  some basics steps .

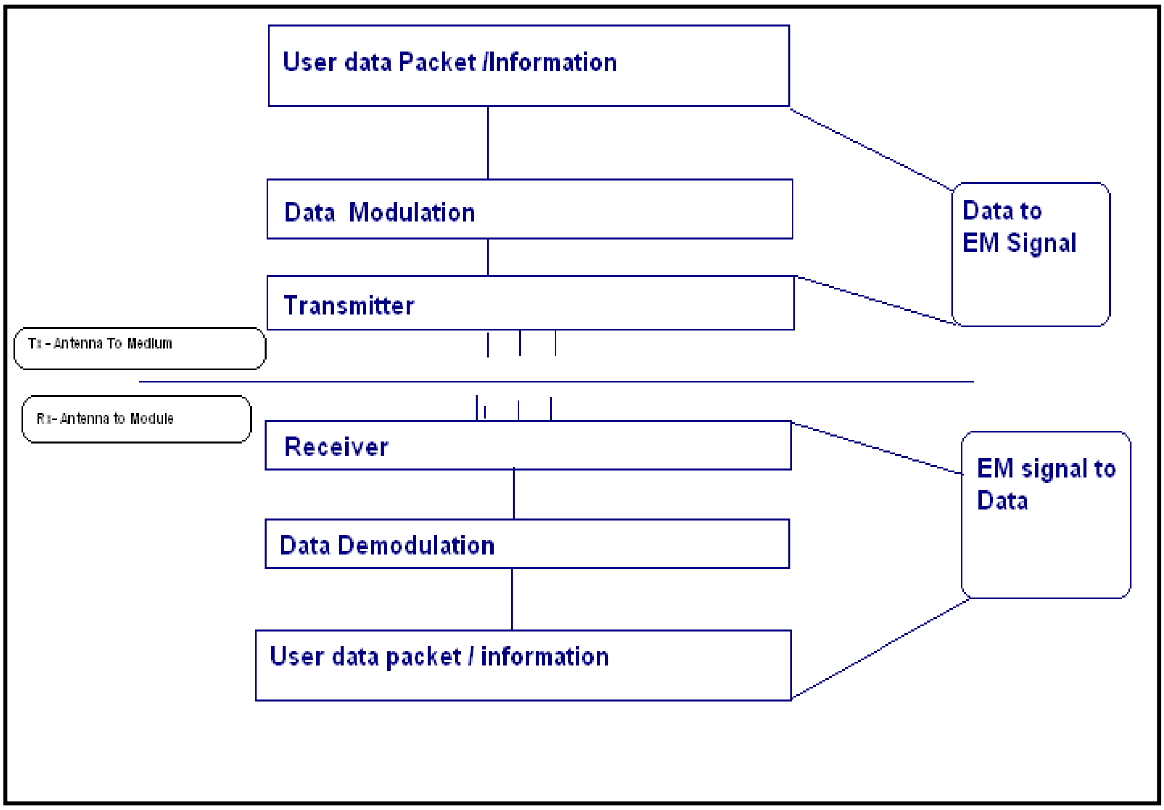


Figure 15. RF Transmission Model

A radio system will consist of a transmitter which is responsible to send the radio signal as EM wave to medium. It consist of an Tx-antenna through which it send the information [Data packet]. Beside that user data typically considering the digital system means the digital data is modulated and pre amplified before it send to the transmitter.

On the other side the, receiver part contain the Rx- antenna that receive the information of course modulated information through EM signal. Before the data is demodulated it usually pre amplified due to signal become weaker when it travel a distance.

## 3.5 - Signal Power Issue

There are various obstacle and also due to propagation, signal will fade or loss its actual transmitting power. Mobile radio channel has serious limitation on wireless system performance like changing the transmission path from Line-Of-Sight (LOS) to fully obstructed by obstacles like buildings, hills etc. which we know it as Non-Line-Of-Sight (NLOS). In free space propagation consider the signal strength when the transmitter and receiver gave a clear line-Of-Sight path and it is a large scale model which power decays exponentially with distance:

$$P_r(d) = P_t G_t G_r \lambda^2/(4\pi)^2 d^2 L^2$$

where:

$P_t$: *transmitted power*

$P_r$: *received power*

$G_t$: *transmitted antenna gain*

$G_r$: *received antenna gain*

d: *Tx-Rx separation*

λ: *c/f – wavelength*

L: *other hardware losses (L ≥ 1)*

Antenna gain related to effective aperture $A_e$:

$$G = 4\,\pi A_e/\,\lambda^2$$

There are various things that effect the power of the signal usually known as attenuation. We can say that in large scale propagation the **Path Loss** and ***Shadow Fading*** models can be effective in field wireless channels and in small scale, ***Fading*** model.

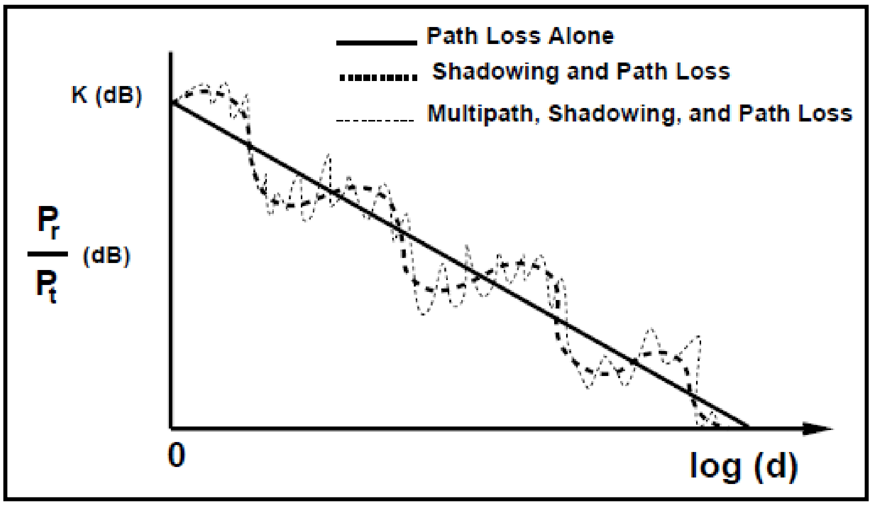


Figure 16. Path Loss, Shadowing and Multipath

### 3.5.1 - Path Loss

Signal is emitted from the transmitter with a transmitting power and traverse a distance to reach the receiver antenna. Power decrease with the increasing of the distance. The loss due to the propagation through the space is known as Path Loss or Path Attenuation.

Path loss is the reduction in power density of an electromagnetic wave as it propagates through the environment. It happens because of many causes such as free-space loss, refraction, diffraction, aperture-medium coupling loss and absorption. Also we have another factors which effect on path loss like the distance between the transmitter and the receiver or the height and location of antenna.

It is possible that the signal radiated by a transmitter travel along many and different paths to a receiver. This effect is called *multipath*.

Normally path loss can be represented by the path loss exponent which its value is in the range of 2 to 4. 2 is for propagation in free space and 4 is for lossy environment and for full specular reflection from the earth surface. Generally path loss represents the signal attenuation as a positive quantity measured in dB and it is the difference in dB between the effective transmit power and receive power.

Path loss in free space defines as:

$$PL(dB) = 10Log(P_t/P_r) = 10Log[G_tG_r\lambda^2/(4\pi)^2d^2]$$

This model is valid for values of d which are in the *far-field* or the *Fraunhoffer Region* of the transmit antenna. The far-field is the distance beyond $d_f$, related to the largest linear dimension D of the antenna aperture and wavelength $\lambda$:

$$D_f = 2D^2/\lambda$$

$$( D_f >> D \ \& \ d_f >> \lambda )$$

It is clear that this model is not valid for d = 0 so instead of that a reference distance $d_0$ with known received power reference point is used:

$$P_r(d) = P_r(d_0)\ (d_0/d)^2 \qquad d \geq d_0 \geq d_f$$

In indoor spaces the path loss exponent can have the value in the range of 4 to 6. Path loss usually expressed in dB.

$$PL(dB) = 10nLog_{10}(d) + C$$

PL: *path loss in decibels*

n: *path loss exponent*

d: *distance between the transmitter and the receiver in meter*

C: *constant which accounts for system losses*

Most propagation models using the combination of analytical and empirical techniques together which the empirical approaches are based on measured data to which characteristics are then fitted. Taking into account all known and unknown propagation phenomena is the most important advantage of empirical methods. In the other hand the validity of this kind of model is restricted to the operating environment and parameters used in the measurements.

As we know RSSI ranging needs less communication overhead, lower implementation complexity and lower cost with comparing to TOA, TDOA and AOA, so it is more suitable for nodes in WSN which have limited power.

According to the principle of the RSSI, we have below relationship between power of transmitted signal, received signal and distance:

$$P_r = P_t.(\frac{1}{d})^n \rightarrow 10\log P_r = 10\log P_t - 10\,\text{n}\log d$$

- $P_r$: *is the received power of wireless signal*
- $P_t$: *is the transmitted power of wireless signal*
- d: *the distance between sending node and receiving node*
- n: *is the transmission factor whose value depends on the propagation environment.*

$10\log P_r$, is the expression of the power converted to dBm, so for above equation we have:

$$P_R(dBm) = A - 10nlogd$$

The values of parameter *A* and parameter *n* determine the relationship between the strength of received signals and the distance of signal transmission.

### *3.5.2 – Shadow Fading*

Path loss is a function of parameters such as antenna height, environment and distance, which means that an obstacle like a tree or a building along a path at a given distance will be different for every path. Some of these paths suffer increased loss and some of them less obstructed and have an increased signal strength. This is shadowing or slow fading.

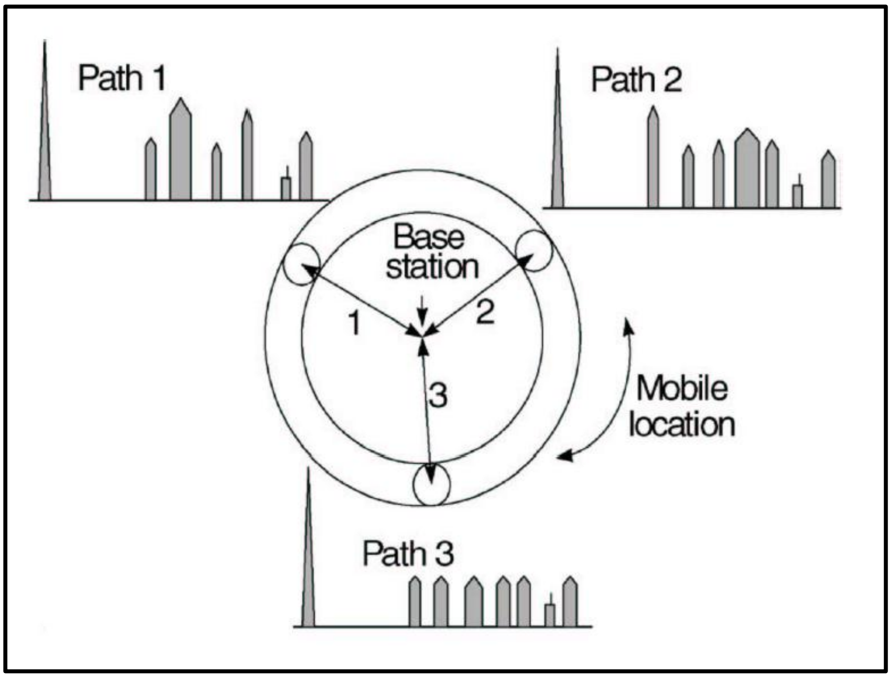


**Figure 17. Variation of path profiles encountered at a fixed range from a base station**

According to the ***Log-distance*** path loss model, the average received signal power decrease logarithmically with distance in indoor and outdoor spaces:

$$\overline{PL}(d) \propto (\frac{d}{d_0})^n$$

$$\overline{PL}(dB) = \overline{PL}(d_0) + 10n\log(\frac{d}{d_0})$$

As mentioned before n is the path loss exponent which in different environments has different value:

| Environment | Path Loss Exponent (n) |
|---|---|
| Free Space | 2 |
| Urban area cellular radio | 2.7 to 3.5 |
| Shadowed urban cellular radio | 3 to 5 |
| In building line-of-sight | 1.6 to 1.8 |
| Obstructed in building | 4 to 6 |
| Obstructed in factories | 2 to 3 |

Table 2. Values of Path Loss Exponent

The problem with Log-distance model is not considering the existence of vastly differ locations in equal situation of Tx-Rx according to different surrounding environment, so the measured signals can greatly differ from the average values predicted by this path loss model. To solve this problem actual value have been shown to be random and distributed ***Log-normal*** in dB about the mean distance dependent value:

$$PL(d) = \overline{PL}(d) + X_\sigma = \overline{PL}(d_0) + 10n\log\left(\frac{d}{d_0}\right) + X_\sigma \quad (*) \qquad \text{PL(d) in dB}$$

*$X_\sigma$: zero-mean Gaussian Random Variable in dB with standard deviation $\sigma$ in dB*

The Log-normal variable $X_\sigma$, describes the shadowing effect which occur over many measured signal strength at location with equal Tx-Rx separation. So by considering that the antenna gains included in PL(d) we will have:

$$P_r(d) = P_t - PL(d) \qquad P_r \text{ and } P_t \text{ in dBm}$$

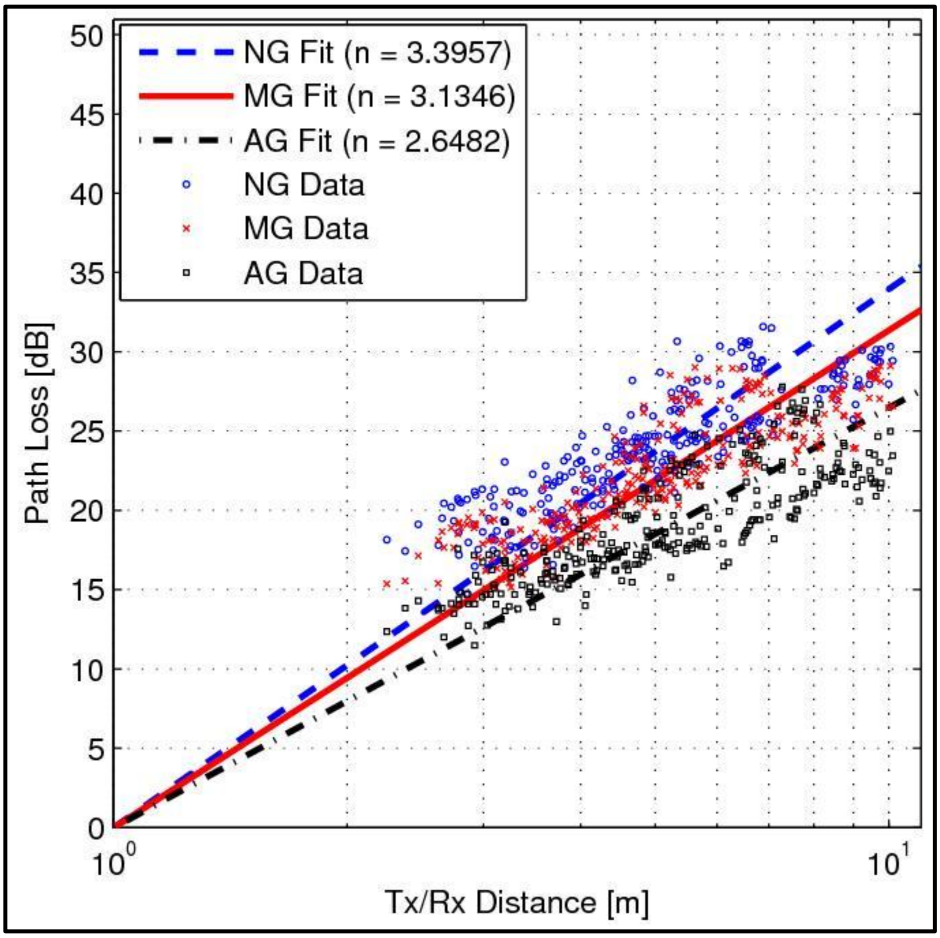


Figure 18. Log-normal Shadowing

In ***Log-normal Shadowing*** we will have below equation for the probability that the received signal level exceed a particular level if PL(d) is a normally distributed random variable:

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_{z}^{\infty} e^{-\frac{x^2}{2}}\, dx = \frac{1}{2}[1 - \mathrm{erf}\left(\frac{z}{\sqrt{2}}\right)]$$

This probability for received signal in dB for exceed a value γ:

$$\Pr[P_r(d) > \gamma] = Q(\frac{\gamma - \overline{P_r}(d)}{\sigma})$$

This probability for received signal in dB for fall below a value γ:

$$\Pr[P_r(d) < \gamma] = Q(\frac{\overline{P_r}(d) - \gamma}{\sigma})$$

Log-normal shadow model is a more general propagation model. It is suitable for both indoor and outdoor environments. The model provides a number of parameters which can be configured according to different environments.

The log-normal probability density function is given by:

$$f_{LN}(x) = \frac{1}{\sqrt{2\pi}\sigma x}\exp\left(\frac{-(\ln x - \mu)^2}{2\sigma^2}\right)$$

μ: *The mean received signal strength*

σ: *Standard deviation*

By experiment over some of signal strength group (d, RSSI) the below function is defined between variance and distance:

$$\sigma(d) = ad^3 + bd^2 + cd + e \quad (**)$$

The symbols a, b, c and e are undetermined coefficients, the values of which are dynamically adjusted according to different environments.

By putting the (**) into (*) we have:

$$PL(d)(dB) = \overline{PL}(d) + X_\sigma = \overline{PL}(d_0) + 10\eta\log\left(\frac{d}{d_0}\right) + \sigma(d)X$$

The function $\sigma(d)$ makes Log-normal shadowing model be able to dynamically control the error function according to difference distances.

Imagine a mobile object is driven around a sensor at a constant distance, so the local mean signal level is typically similar to below figure.

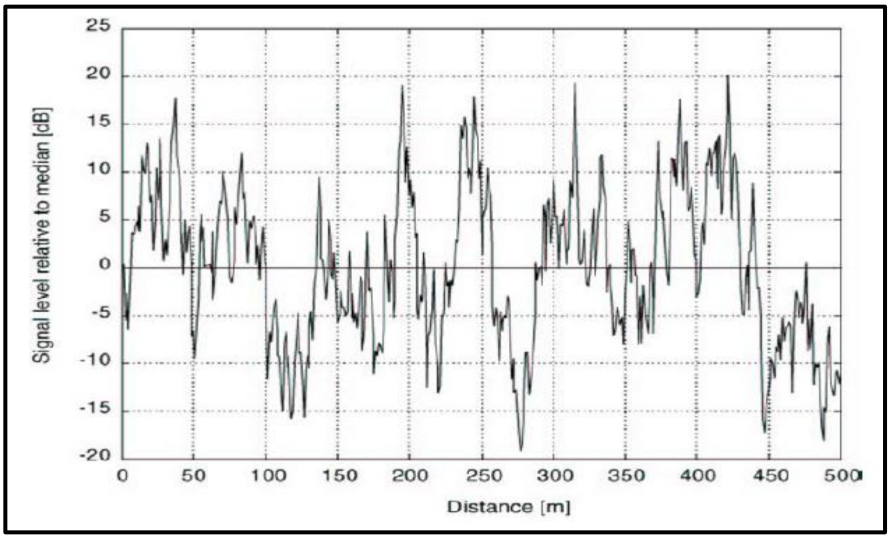


**Figure 19. Typical variation of shadowing with fixed distance between Mobile object and Sensor**

The distribution of the signal powers is log-normal which is the signal measured in decibels has a normal distribution. A typical probability density function of the signal is shown in below figure.

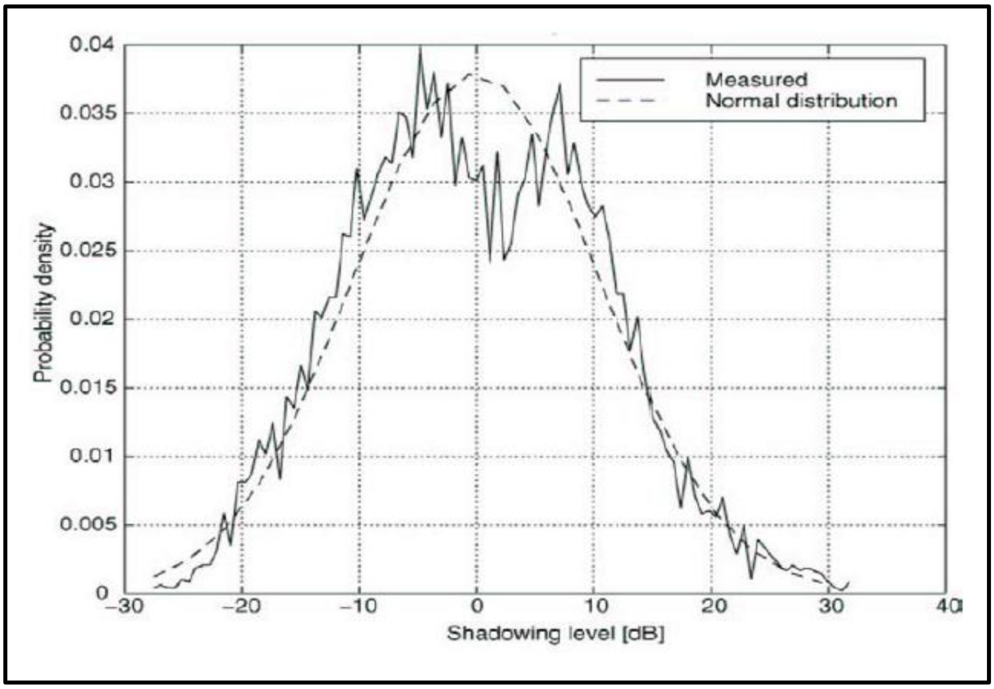


**Figure 20. Probability density function of shadowing**

### 3.5.3 – Fading

In small scale fading is rapid fluctuation of amplitude, phase and multipath delays of a signal over a short time and/or distance. Because of analyzing the propagation over such short time or distance, the large scale effects are ignored. The constructive and destructive interference between multiple versions of the signal reaching the receiver at different times is the cause of fading. Combining these multipath waves at the antenna will have resulting signal varies in amplitude and phase.

Fading effect happens because of multipath channel and has different result such as changing in signal strength over small distance of time, having random frequency modulation due to Doppler for each multipath and time dispersion or echoes due to propagation delay.

According to the Doppler, difference in path length traveled by wave is:

$$\Delta l = d \cos\Theta = v\ \Delta t\ \cos\Theta$$

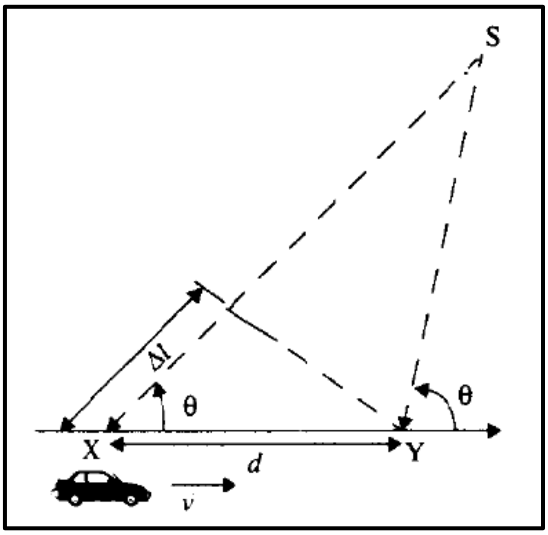


Figure 21. Doppler

and phase change in received signal due to path length difference is:

$$\Delta\phi = 2\pi\Delta l/\lambda = 2\pi v\Delta t/\lambda \times \cos\Theta$$

and apparent change in frequency – Doppler shift is:

$$fd = \Delta\phi/2\pi\Delta t = (v/\lambda) \cos\Theta$$

According to multipath time delay spread we can divide small-scale fading in two types:

- ***Flat Fading***
    - Bandwidth of signal < Bandwidth of channel
    - Delay spread < Symbol period
- ***Frequency Selective Fading***
    - Bandwidth of signal > Bandwidth of channel
    - Delay spread > Symbol period

But according to Doppler spread the types would be:

- ***Fast Fading***
    - High Doppler spread
    - Coherence time < Symbol period
    - Channel variation faster than baseband signal variations
- ***Slow Fading***
    - Low Doppler spread
    - Coherence time > Symbol period
    - Channel variations slower than baseband signal variations

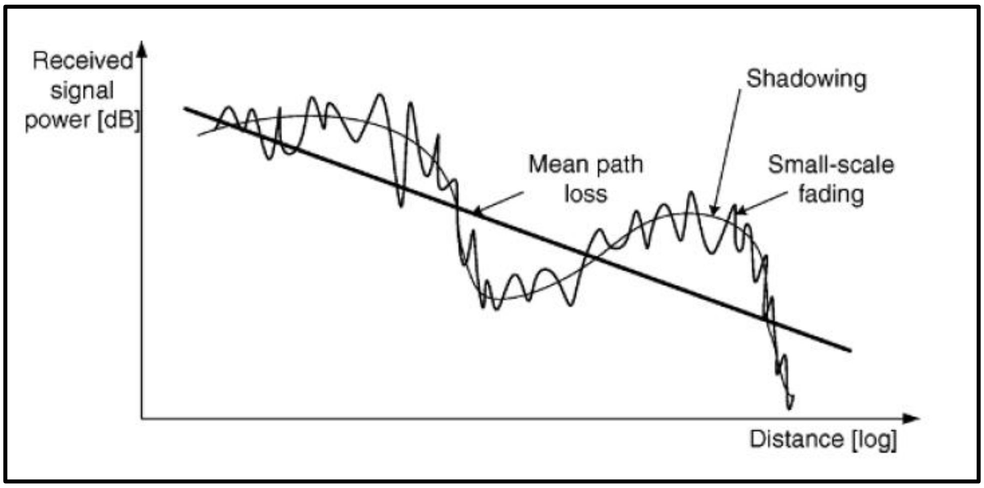


Figure 22. Large-scale fading vs. Small-scale fading

## 3.6 – Radio Propagation Mechanisms

Three main propagation mechanisms are exist for indoor and outdoor areas.

### *3.6.1 – Reflection*

When radio wave comes in contact with a material, it will reflected as it is for light. The angle will preserve means, at which angle the signal will hits the surface the same angle it will reflected. Reflection cause a big issue in indoor application due to a lot of and different types of substance can present in the application environment.

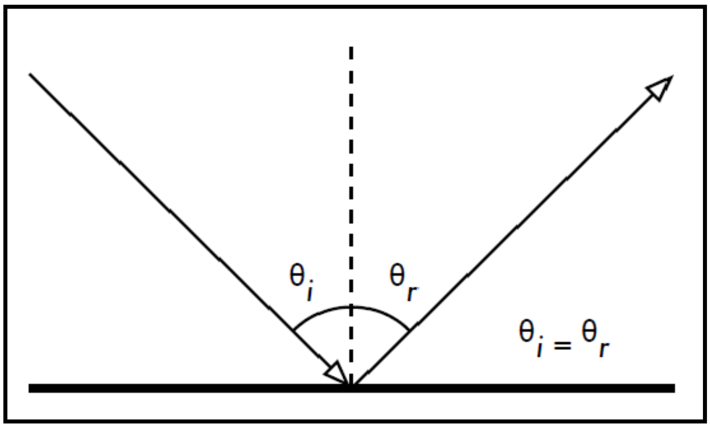


Figure 23. Reflection of EM signal

### 3.6.2 - Diffraction

Due to presence of the sharpness of the object, such as corner, the original signal wave is becoming bend . This effect is known as diffraction.

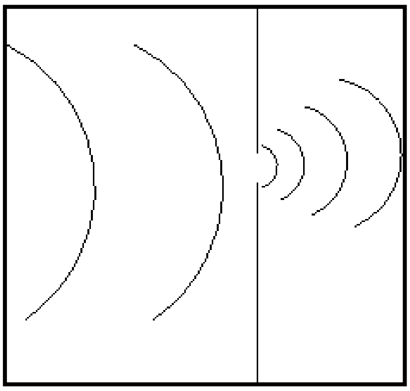

Figure 24. Diffraction of EM signal

The effect of diffraction, waves will "bend" around corners or through an opening in a barrier. It refers to various phenomena which occur when a wave encounters an obstacle and we can say that the diffraction is the apparent of waves around small obstacle and the spreading out of waves past small opening.

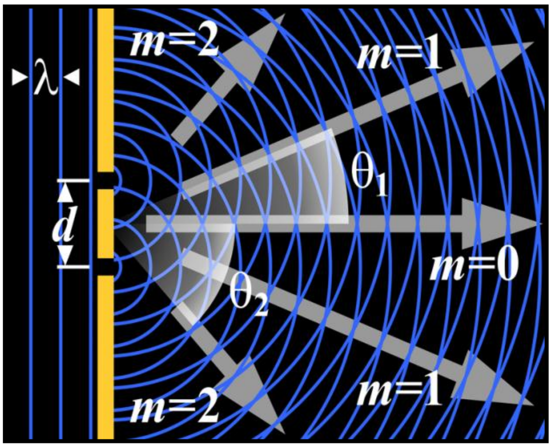


Figure 25. Generation of an interference pattern from two-slit diffraction

### *3.6.3 – Scattering*

When objects like walls with rough surfaces and furniture in indoor area or vehicles in outdoor area scatter rays in all directions in the form of spherical waves, scattering happens. Propagation in many directions results in reduced power levels, especially far from the cause of scattering.

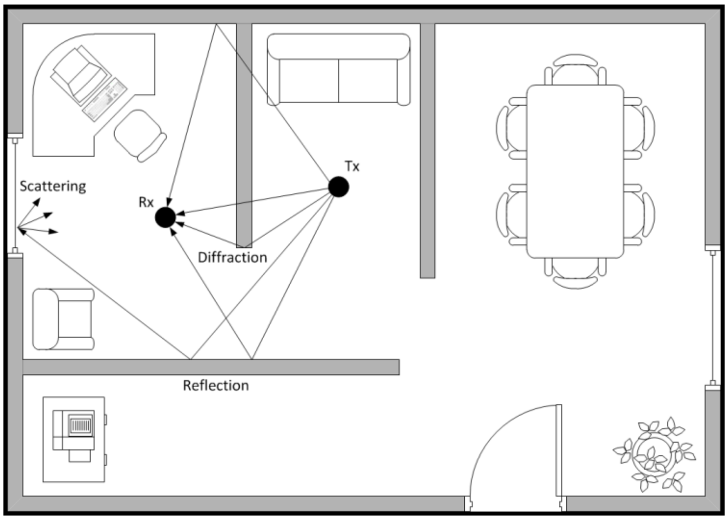


Figure 26. Radio propagation mechanisms in an Indoor area

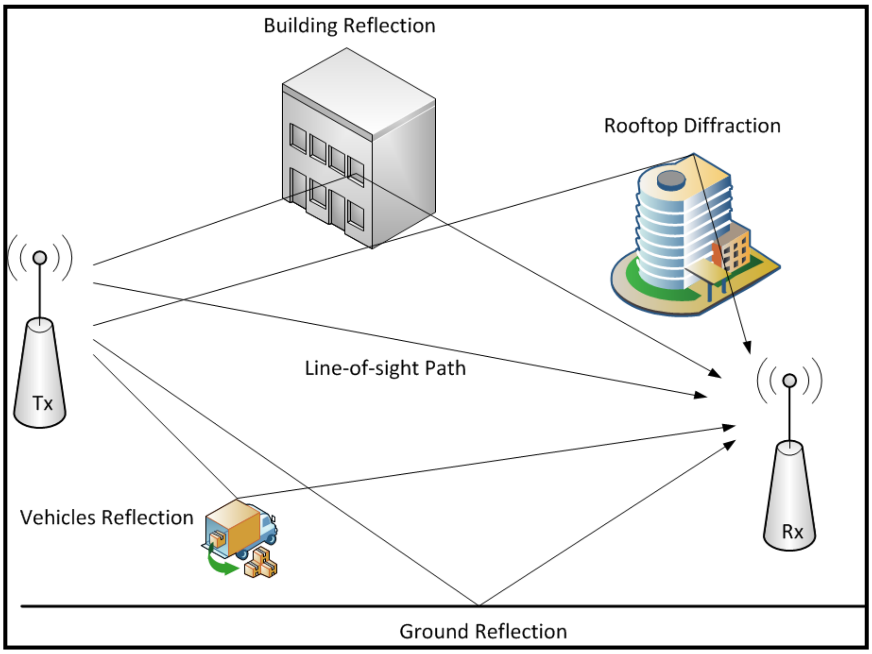


Figure 27. Radio propagation mechanisms in an Outdoor area

## 3.7 - Interference

Specially in Indoor application, there may have than other RF signal. Due to wave properties of the signal, addition of the other signal such as amplitude and phase may change the original signal. This phenomena is known as Interference.

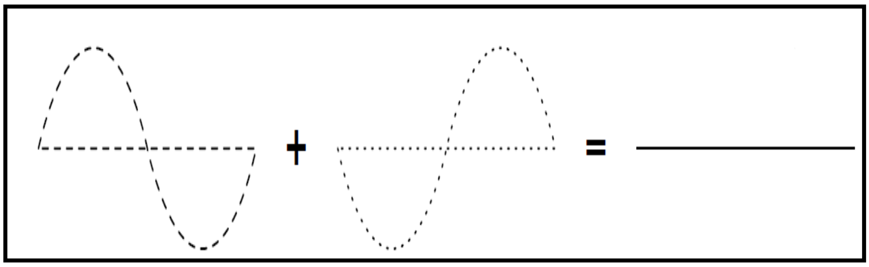


Figure 28. Interference of EM signal

Generally interference is anything which alters, modifies or disrupts a signal as it travels along a channel between a source and a receiver and it may be broadly categorized into two types; *narrowband* and *broadband*.

International transmission such as radio TV stations, cell phones and etc. are causes of the narrowband broadband comes from incidental radio frequency emitter.

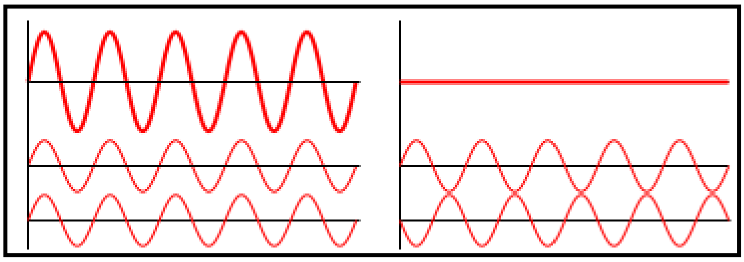

**Figure 29. Constructive and Destructive Interference**

Spread-spectrum and frequency-hopping techniques can be used with both analogue and digital signaling to improve resistance to interference. Also for excluding the other signals and select one of them in space, a highly directional receiver such as a parabolic antenna or a diversity receiver can be used.

# *Chapter 4:* Related Works and Literature review

## 4.1 – Introduction

Wireless signal propagate through an environment. Environment consist of a lot issues that causes attenuation of the signal. These various interfaces sources such as temperature, multipath propagation ,diffraction ,presence of obstacle and many such factors have effect on the signal. It is not possible to get a value without considering these issue. Many researchers used RSSI for localization and discover some method to deals with these problems. The average statistical model, Gaussian model and the correction model based on the distance between the fixed nodes are commonly used to optimize the RSSI value.[2]

## 4.2 - Literature over view

Wu Chengdong, Chen Shifeng, Zhang Yunzhou and Cheng Long Wu Hao proposed a probabilistic localization method. As it is stated on the paper [1] according to their method RSSI value between node i and j is considered with taking into account that power is decreased due to distance and also zero-mean normally distributed random variable with standard deviation corresponding to the fading effect. The probability density function of the distance between node i and j is measured for the case line of sight (LOS) and non-line of sight (NLOS). Propagation model is assumed before to measure initial value of RSSI.

Some researcher and scientist are introduce attenuation constant of weather with to measure the RSSI signal. [3] Preliminary received signal power is measured using logarithm distribution and consider the variance of the shadowing effect of radio signal propagation model consider path loss, antenna gain, receive antenna gain and transmission power.

Received signal strength attenuates in proportion to the negative weather attenuation. The location estimation using RSSI which is power of the distance between nodes [3]. The attenuation constant is derived from a function of the distance of the nodes.

According to Jungang ZHENG, Chengdong and WU Hao and CHU Peng JI [4] for their RSSI based localization is based on path loss model for free space propagation and logarithmic path loss for the radio signal. Considering this value, antenna again, emission power is used to measure the distance. To correct the errors of RSSI measurement author introduces the geometric constraints of two-dimensional space of Cayley-Menger determinant. Centroid triangle method is used to localize the position.

As it is did by Qingxin Zhang [2] in his localization method RSSI localization method, the raw RSSI value with considering the path is used keep into an array for a sequence of RSSI of the nodes. To overcome the environmental problem such as multipath diffraction etc. there are some statistically method can be used with certain limitation. Considering this factor and accuracy author proposed a Gaussian distribution function. RSSI value is filtered with Gaussian distribution model and compare with threshold value. If it can satisfy the rules it is keep to another array and then average of these RSSI value is to be consider for the application. Reference node is use for error minimization and further a penalty function is introduced which can be solved by sequential quadratic programming.

The author [5] suggested RSSI-based localization based on fingerprint and on signal propagation modeling. They used anchors (sensor node) whose position is known. Their suggested method has two phases one is training phase another is localization phase.

In the training phase the anchors broadcast packets containing the identifier. This information used by the server to calibrate propagation model by configuring the parameter. Walls and Floor attenuation is consider with path loss model. They introduced the global virtual calibration that produce the same parameter for every walls where per-wall virtual calibration provides an attenuation factor for the walls that directly affect the communication between specific pairs of anchors. Heuristic method is used for the comparison purpose between the two method which estimate the propagation model parameters by using the RSSI measurements on a grid of points in the environment, as it is done with fingerprinting.

In [6] Sheng-Po Kuo and his colleague introduced a new problem in localization system which in several views, articles and systems have been not considered. In most localization systems, beacons or sensors are being placed as references to determine the positions of moving objects and consider that are fixed in their position. But we will have inaccurate results if the positions of the beacons change during the process and localization procedure. In this situation according to their idea, *Beacon Movement Detection (BMD)* would be a good solution to detect this problem. To have more accurate and deep considering in this problem they divided the detection methods of movement in four types.

The first one is *Location-Based (LB)* scheme which tries to calculate each beacon's current location, and compares the result with its predefined location to decide if it has been moved or not. This method has excellent detection result if there are many beacons. In the second method, *Neighbor-Based (NB)* scheme, beacons will keep track of their neighbor beacons and report their observations to the BMD engine to determine if some beacons have been moved or not which this method is easy to implement. In the third one, *Signal Strength Binary (SSB)* scheme, the change of signal strengths of beacons will be exploited and this method performs well under most situation. And finally in the last method *Signal Strength Real (SSR)* scheme, the BMD engine will collects the sum of reported signal strength changes of each beacon to make decisions. Also this one has good performance under the most of situations.

Just the first methods assumes that the original locations of beacons are known for the system. The other three have no pre knowledge about the original position of the beacons. Also the BMD has another issue about the covering the effect of the noise.

They presented an algorithm which allows beacons to monitor each other and identify this unexpected movement and also proposed some heuristic methods by mapping the Beacon Movement Detection problem to the vertex-cover problem and this method improve the error ratio more than 70 percent in most cases for Signal Strength Binary and Signal Strength Real situation.

AHLoS in [7] enables nodes to dynamically discover their own location through a two-phase process:

- ***Ranging:***

  During the ranging phase, each node estimates its distance from its neighbors.
- ***Estimation:***

  In the estimation phase, nodes with unknown locations use the ranging information and known beacon node locations in their neighborhood to estimate their positions. Once a node estimates its position it becomes a beacon and can assist other nodes in estimating their positions by propagating its own location estimate through the network. This process iterates to estimate the locations of as many nodes as possible.

To justify the ranging choice the authors performed a detailed comparison of two promising ranging techniques: one based on received RF signal strength and the other based on the Time of Arrival (ToA) of RF and ultrasonic signals. They found that ToA is a good method for find-grained localization and is less sensitive to physical effects but RF signal strength is not good for fine-grained localization. They improve the accurate the location estimation with centralized implementation which increase the system robustness. Also they mentioned that the accuracy of iterative multilateration is satisfactory for small networks but needs to be improved for large scale network.

In the article [8] Kuo-Feng and his colleagues introduce a new method for localization according to mobile anchors. They have introduced different components for their procedure:

- ***Sink Node:*** A particular node is responsible for collecting sensing data reported from all the sensors. The sink finally transmits the data to a task manager.
- ***Sensor Node:*** The mobile beacons which its position is unknown and the result of the localization method will declare its position.
- ***Mobile Anchor Node:*** A mobile transceiver which was equipped with GPS can help to its position be clear in real time.

They divided the schemes which have been proposed for dealing with the localization in two categories:

- First, the *range-based* schemes need either *node-to-node distances* or *angles* for estimating locations. The information can be obtained using *time of arrival (TOA)*, *time difference of arrival (TDOA)*, *angle of arrival (AOA)*, and received signal strength indicator (RSSI) technologies.
- Second, the *range-free* schemes do not need the distance or angle information for localization.

In their paper they have developed a localization mechanism using the geometry conjecture (Perpendicular bisector of chord) which states that the perpendicular bisector of chord passes through the center of the circle. If we consider that the transmission center of a sensor node is a circle and this sensor node is in the center of this circle, by finding two chords according to the mobile anchor nodes, the position of the sensor node will be found.

In detail if three different mobile anchor nodes pass the circle of sensor node coverage, by their cross points (endpoints) we will have two different chords in same circle which their perpendicular bisectors cross each other in the center of the circle which is the position of the sensor node and because of knowing the positions of the mobile anchor nodes, the position of the sensor node will be find easily. Also it is possible one of the mobile anchor nodes pass the circle in two endpoints and just with another mobile anchor node instead two others, have two different chords on the circle. It is necessary to collect at least three endpoints on the circle to have two different chords.

For enhancing the performance of the method they introduced and considered three different subjects, *Beacon Scheduling*, *Chord Selection* and *Obstacle Tolerance*. The enhancement of their method improve performance and can tolerate radio irregularity due to obstacle. Also according to their methods the execution time for the localization procedure can be shortened if the moving speed, the radio range or the number of mobile anchor point is increased. They results show that their mechanism outperformed two range-free localization scheme and the average location error is also competitive to other range-based methods.

Author [9] has designed and implemented RSSI based localization according to logarithmic path loss propagation model and statistical average. The desired distance is measured by considering the average RSSI value on the logarithmic path loss model and later implementation of the triangulation method for the localization purposes. According to their paper, the average value of the RSSI is also follow the path loss model of the radio wave propagation.

Radio propagation model for indoor application is becoming a challenge now days due to presence of various obstacle such as furniture, movement of people etc. which are unpredictable. Various learning algorithm is introduce to define an appropriate model of the propagation such as neural network [10] and cluster analysis as well as various power level of transmission for a single anchor [11].

Chen Zhong and his colleagues in [12] worked on performance of RSSI based localization method according to clustering the body sensors for heterogeneous WSN in indoor area. The main target of their research was improving the safety of works in hazardous environments and the most important one is to locate them in those area in case of emergency situation. They used the min-Max as their localization algorithm because it is not so complicated. In their algorithm for each anchor a bounding square box will be constructed which anchor itself is in center of that and edges are double of estimated distance of anchor with the target. If exist three of these boxes the center of their intersection of them will be the location of the target. According to their the accuracy of the RSSI based localization method is too low but by analyzing a specific application scenario and adjusting a defined method it can be improved.

In their scenario a person is equipped with several different sensors to collect different types of information and these sensors can communicate wirelessly with each other and send their packets from a gateway. They named this small wireless network as Body Area Network (BAN) or a cluster. This BAN or cluster has different sensors, one gateway and one cluster head. All sensors estimate the distance between themselves and others clusters according to receiving the RSSI signal. But because of degradation of the signal propagation, the receiving RSSI would be different for different sensor node even if they be in same position. In this situation the highest RSSI will be chosen out of all RSSI which have been received by all sensor nodes in a cluster.

For measuring the distance characterization they did several measurement for the RSSI which have been received by the sensor nodes in different heights and introduced logarithmic formula according to the different heights.

They evaluated the performance by some estimating error means, like *estimation error-RSS-distance model* which found that the performance of localization not rely on the model. The other estimation error was about *localization request packet used amount* which is the influence of the number of RSSI measurement packets used for localization. According to this estimating the target need to broadcast three to five localization request packets. The last estimation error was about *the number of involved anchor nodes in each cluster (Reference Point)* which the result showed that positioning is slightly improved by adding anchor nodes into each reference point (cluster) and the position estimation is better when the target is set on head comparing with on knee and belly.

Jiuqiang Xu and his colleagues used Log-Normal Shadowing Model (LNSM) in [13], as a more general signal propagation model, can better describe the relationship between the RSSI value and distance, but the parameter of variance in LNSM is depended on experiences without self-adaptability. In this paper, it is found that the variance of RSSI value changes along with distance regularly by analyzing a large number of experimental data. Based on the result of analysis, they proposed the relationship function of the variance of RSSI and distance, and established the log-normal shadowing model with dynamic variance (LNSM-DV). Also they used the Least Squares (LS) method to estimate the coefficients in the model. Finally LNSM-DV model which is a self-adaptable method in different environment reduces errors.

The establishment of LNSM-DV has great practical significance for improving the accuracy of ranging, the accuracy of positioning and the self-adaptability of ranging models in wireless sensor networks.

Redondi and his colleagues in [14] worked on performing localization, tracking and monitoring of WSN based in IEEE 801.15.4 (Zigbee) standard. Their method based on RSS measurement. In this project the location of anchor nodes are fix and clear and RSS measurement are collected between pairs of node since they are related to the inter-node distances. According to the parametric method the relationship between the RSSI and distance would be:

$$S = S_0 - \frac{10\alpha log_{10} d}{d_0} + v$$

Where S is value of RSSI measured in dBm units and d is the inter-node distance. $S_0$ is the RSSI metrics measured between two nodes $d_0$ meters apart. The parameter α represents the power decay index (also known as path loss exponent) and is in the range [2,4] for indoor environments. The noise term v is typically modeled as a Gaussian random variable $N(0, \sigma_v^2)$ representing shadow-fading effects in complex multipath environments, whereas the value of standard deviation $\sigma_v$ depends on the characteristics of the specific environment, and the localization problem is solved by computing the position that maximizes the likelihood with respect to the above model in. In some other methods, we have above model but the distance between a client and an anchor node is modeled as a linear combination of the RSSI measurements between the client and all the anchors.

In this project the overall RSSI measurement among all anchor node pairs can be represented by an NXN matrix $S = [S_1, S_2, \ldots, S_N]$. The corresponding matrix for the Euclidean distance between anchors is $D = [D_1, D_2, \ldots, D_N]$ which is symmetric and has zero diagonal entries. In "zero Configuration Localization" method the linear relationship between the RSSI measurements and the logarithm of the inter-node distances is:

$$Log(D) = TS$$

Which T is NXN matrix defining the signal-to-distance mapping. Given the measurements between pairs of anchor nodes, the matrix T is estimated by means of least squares as:

$$T = Log(D)S^T(SS^T)^{-1}$$

Finally:

$$\hat{d} = \exp(T\hat{s})$$

Which, $\hat{s}$ is RSSI measurements between client node and its neighboring anchor nodes vector and $\hat{d}$ is its corresponding distance vector. At last a gradient descent algorithm is employed to estimate the location based on the obtained distance vector $\hat{d}$.

In case of moving client nodes the RSSI fluctuations are too high and indeed the accuracy of the system will be too low which they by using the knowledge about the moving node and the geometry of the environment improved this localization accuracy in three steps: Prediction, Update and Initialization.

The composition of LAURA system is anchor nodes which are statically deployed in the areas to be monitored, and the client nodes which are movable objects. The Hierarchical Addressing Tree routing protocol, operates the network by creating a tree-like routing/forwarding topology among the network nodes.

In initializing phase, a node starts to scanning states and collects beacons sent by the neighbor nodes and tries to choose the parent node to be associated to among the all elements. This association proceed depends on three factors:

- Received Signal Strength Indicator
- Current number of children of the $i^{th}$ candidate parent ($N_i$)
- distance of the candidate parent from the PAN coordinator (Hop Count, $HC_i$)

According to below equation a node chooses its parent node:

$$i^* = arg.\max_{i \in P}(\alpha RSSI_i + \beta N_i + \gamma HC_i)$$

Now there is two functions for localization of the node:

- dynamic collection of RSSI samples at the sensor nodes
- effective delivery of such collected data to the PAN coordinator, where the localization algorithm is executed.

Finally with RSSI and computing d it is possible to estimate the position of the client node with below 2.5 meter error according to the zero-configuration indoor localization.

In [15], Mogi and Ohtsuki presented a new method for localization by combining the TOA and RSS weight with path loss exponents estimation in NLOS environment. The believable factor algorithm or (BFA) is a kind of weighting algorithm to minimize the effects of the NLOS conditions and has good performance in this case but needs some assumptions which is not favorable for the WSN, like need of three sensor around the mobile object and using just these three sensors for localization. But in the Mogi and his colleague method no prior knowledge need about the path loss exponent for NLOS environment and used both TOA and RSS measurements.

In their algorithm the distances estimated with TOA measurements are weighted by the believable factor (BF) derived from the difference between the estimated distance with TOA measurements and that with RSS ones. In addition the path loss exponents are estimated for each node in a maximum likelihood (ML) manner.

Finally this algorithm has high localization accuracy without the knowledge of path loss exponent.

# *Chapter 5:* Methodology for Localization

## 5.1 - Introduction:

In Wireless Sensor filed, are several essential and important goals, which are related to each other. One of the most important of them which other issues depend on that, is *Localization*. If the results of localization step be accurate, there will be less problems in other steps in wireless sensor network field like deployment, coverage and etc.

According to the problem which had been described in first chapter and different methods and algorithms which had been presented in third chapter of this book, the goal of this chapter is to define, present and implement localization methodology for the problem mentioned in chapter one. In general, finding the position of an unknown mobile object which has been equipped with a microcontroller and transceiver that sends RSSI information iteratively through 868 MHz channel is the main goal of this project.

## 5.2 – Components and Instructions:

As mentioned basically above, there are some critical and different components in this procedure:

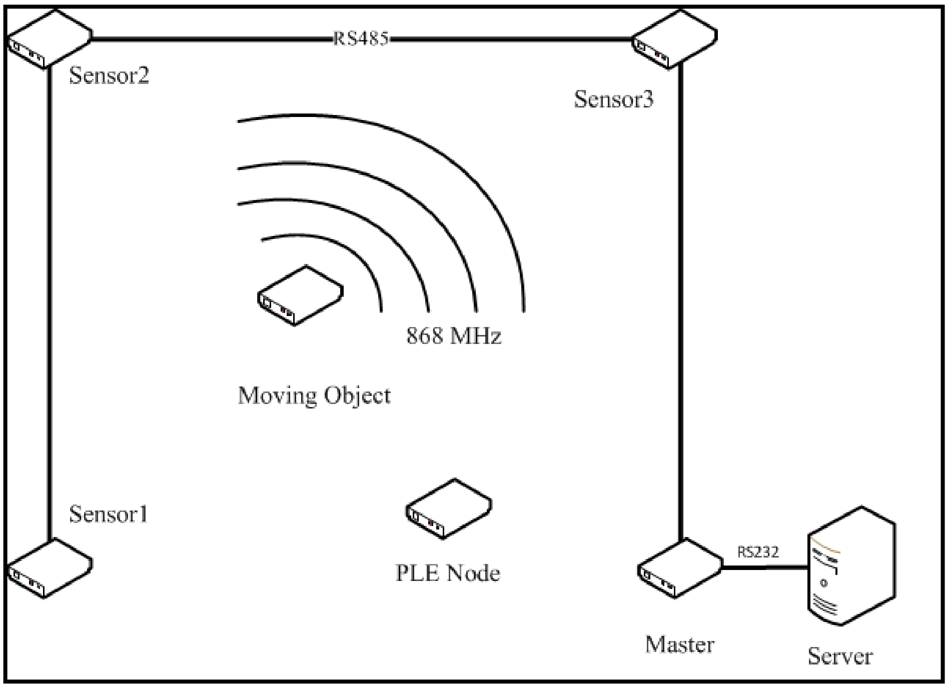


Figure 30. Schematic of WSN Localization Components

### *5.2.1 - Mobile Object (MB)*

"A small microprocessor transceiver with 868 MHz wireless module."

According to different scientific literatures, it can be known as *Unknown Beacon* too. In general the beacon is an object which attracts and absorbs the attention of other objects. But we have to consider that the position of this object should be known, or this feature is not so important for the procedure. Usually the position of beacons are known and these objects help unknown objects to find their positions, and also in some articles and project beacons are unknown themself.

In our project according to sending signals from mobile objects which theirs positions are unknown to the known objects, we can name them "Unknown Beacons" too.

### *5.2.2 - Sensor (S)*

"A microcontroller which is work as a receiver to get the transmitted wireless signals from mobile objects, packages and transfers them to the server."

Sensor works as a transmitter too, because not only get the wireless signal but also send them through the wire RS 485 to the server. The position of sensors are fixed and known, and also wait for call from server to start working. The number of sensors and theirs IDs have been known for the master. Sensor waits to receive RSSI signal from the moving object and in this case the *High Priority Interrupt* activate and sensor tries to send the packet to the master.

Master sends a token packet sequentially to the sensors and the first sensor receives this packet and if it has available RSSI packet to transfer to the master, keeps the token and sends and time slot request to the master. Master by receiving this request send back a ACK packet to get the permission for sending the data. Sensor after receiving the ACK packet, sends its RSSI packet to the master and if the connection was without problem receives another ACK and the connection will be closed by master. Then next sensor received the token packet from master and tries to send its RSSI packet to it and this procedure continues sequentially for all sensors.

After sending the RSSI packet of a sensor to the master the high interrupt priority deactivates and sensor can receive another RSSI signal from moving object.

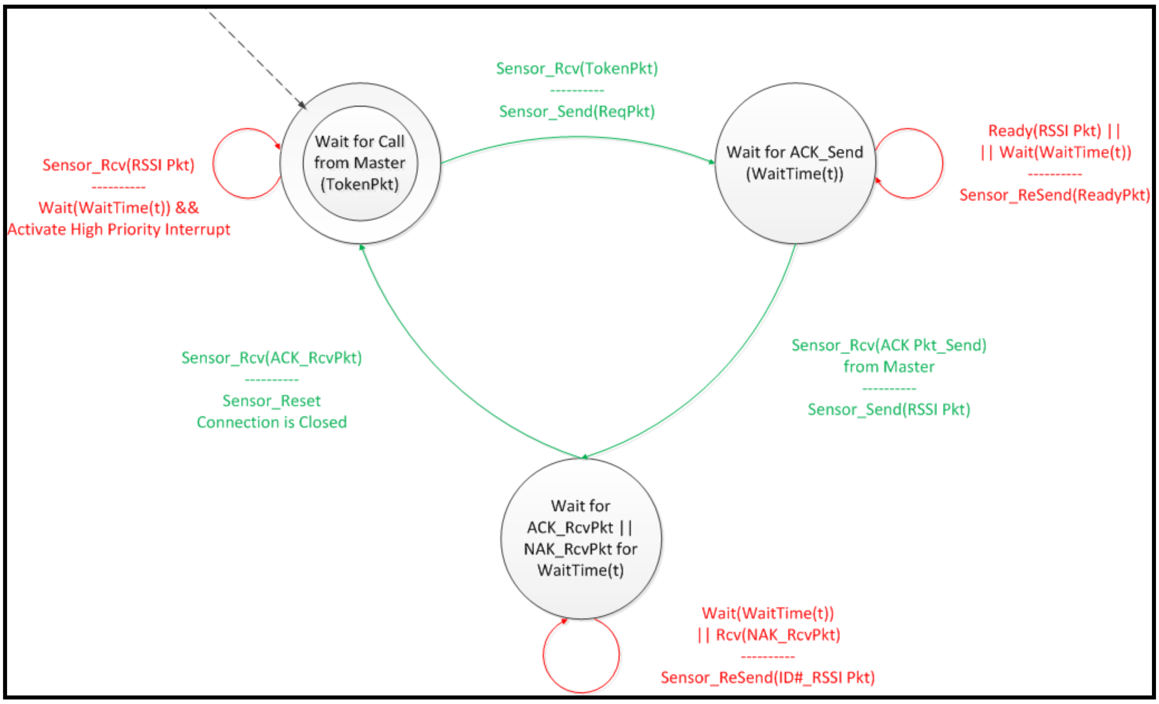


**Figure 31. State Diagram of Sensor Instruction**

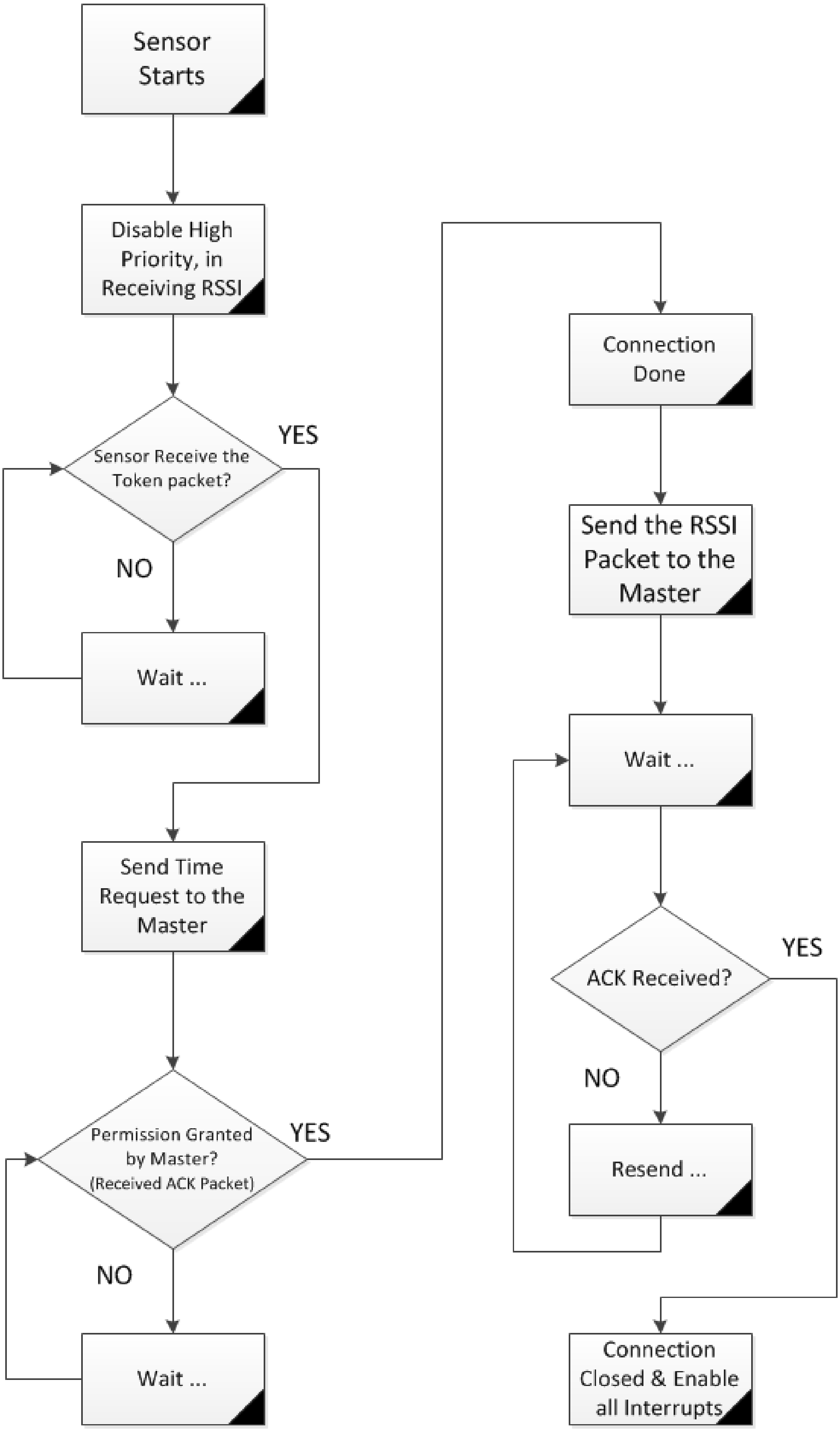


Figure 32. Diagram of the Sensor Instruction

### 5.2.3 – Path Loss Exponent Node (PLE Node)

"A particular node which is responsible for initializing phase to obtain the path loss exponent and average power values."

In initialization phase, we are trying to find the path loss exponent to use that in localization phase. For this reason before starting the localization phase the PLE node receives the RSSI value from a sensor in different distances. In each distance the average of received RSSI value calculate and finally the path loss exponent is estimated.

At any distance (d) the path loss (PL(d)) at a particular location is random and distributed log-normally in dB about the mean distance dependent value. According to different situation in existence of clutters or changing the humidity in surrounding environment, the path loss will be different than the average value as predicted in the first experiment.

This reason of this variation is because of refraction and diffraction of interfering objects in the path of the travelling signal and is an additive term to the path loss with random value and as we mentioned before in chapter this is named shadowing.

$$\mathrm{PL(d)_{in\ dB}} = \overline{\mathrm{PL}}(\mathrm{d_0}) + 10\mathrm{n}\log\left(\frac{\mathrm{d}}{\mathrm{d_0}}\right) + \mathrm{X_\sigma}$$

$X_\sigma$, is a zero-mean Gaussian distributed random variable in dB with standard deviation σ in dB too. As mentioned before over large number of measurement which have same Tx-Rx separation with different levels of clutter on the propagation path, random shadowing effects occur and this is log-normal shadowing.

The value of n and σ are computed according to the measured data by minimizing the mean square error of the measured and estimated path loss values for wide range of measurement locations. Also the value of $\overline{PL}(d_0)$ is depends on either close-in measurements or on a free space assumption from the transmitter to $d_0$, which can be fined in initializing phase by PLE node.

Thus for calculating the received power at a distance d from the transmitter we have:

$$\mathrm{P_r(d)} = \mathrm{P_r}(d_0) - 10n\log(\frac{d}{d_0})$$

Then according to the below formula the minimum mean square error (MMSE) for the path loss exponent (n) would be:

$$J(n) = \sum_{i=1}^{k}((P_r - \widehat{P}_r))^2$$

Where $P_r$ is computed value and $\widehat{P}_r$ is the estimated value.

For instance according to one of our measurement data list we have below table:

| # | Distance (meter) | RSSI<br>Rx power (dBm) |
|---|---|---|
| **1** | 2.00 | 112 |
| **2** | 4.00 | 93 |
| **3** | 6.00 | 82 |
| **4** | 8.00 | 64 |
| **5** | 10.00 | 60 |
| **6** | 12.00 | 57 |

Table 3. Measured values of the example test

In this test the received power at close-in-reference from transmitter is 130 dBm in distance 1 meter. Now we can calculate the received power in these 6 location of the mobile object:

$P_r(d_1) = 130 – 10n\log(2/1) \approx 130 – 3n$

$P_r(d_2) = 130 – 10n\log(4/1) \approx 130 – 6n$

$P_r(d_3) = 130 – 10n\log(6/1) \approx 130 – 8n$

$P_r(d_4) = 130 – 10n\log(8/1) \approx 130 – 9n$

$P_r(d_5) = 130 – 10n\log(10/1) = 130 – 10n$

$P_r(d_6) = 130 – 10n\log(12/1) \approx 130 – 11n$

In this case the J(n) is the sum of squared errors between measured and estimated values:

$J(n) = (112 – 130 + 3n)^2 + (93 – 130 + 6n)^2 + (82 – 130 +8n)^2 + (64 – 130 + 9n)^2 + (60 – 130 + 10n)^2 + (57 – 130 + 11n)^2$ (*)

$$\rightarrow J(n) = 411n^2 – 5514n + 18582$$

By derivative of J(n) equal it to zero, the n which is the minimum mean square error can be obtained. Thus, $822n – 5514 = 0 \rightarrow n = 6.71$ which is the path loss exponent of this measurement.

So for calculating the sigma, we have:

$\sigma^2 = J(n)/6$ and J(n) at n = 6.71 can be obtained by putting this value in formula (*):

$$\sigma^2 = J(6.71)/6 \approx 87.97 / 6 \approx 14.66 \rightarrow \sigma \approx 3.83 \text{ dB}$$

### *5.2.4 - Master (M)*

"A fixed sensor which its position is known

and is connection between sensors and server."

The master is connected to sensors through RS 485 cable and is connected to the server through RS 232 cable. This device get the RSSI packets from sensors via RS 485 cable which have been collected from mobile objects. Then sends them via RS 232 to the server.

Master sends a token packet to the sensors sequentially to get the permission to them for sending theirs available RSSI packets. The sensor who is the owner of the token packet, keep this packet and send back a request time slot to the sensor to reserve the connection for itself. Master after receiving it, send back and ACK packet to allow it for sending the RSSI packet. Then after receiving its RSSI packet, sends ACK packet to complete the connection and closes it. So in next phase, sends this packet to the server according to the normal handshaking protocol through RS 232 cable.

After that sends another token packet to the next sensor and this procedure continues repeatedly.

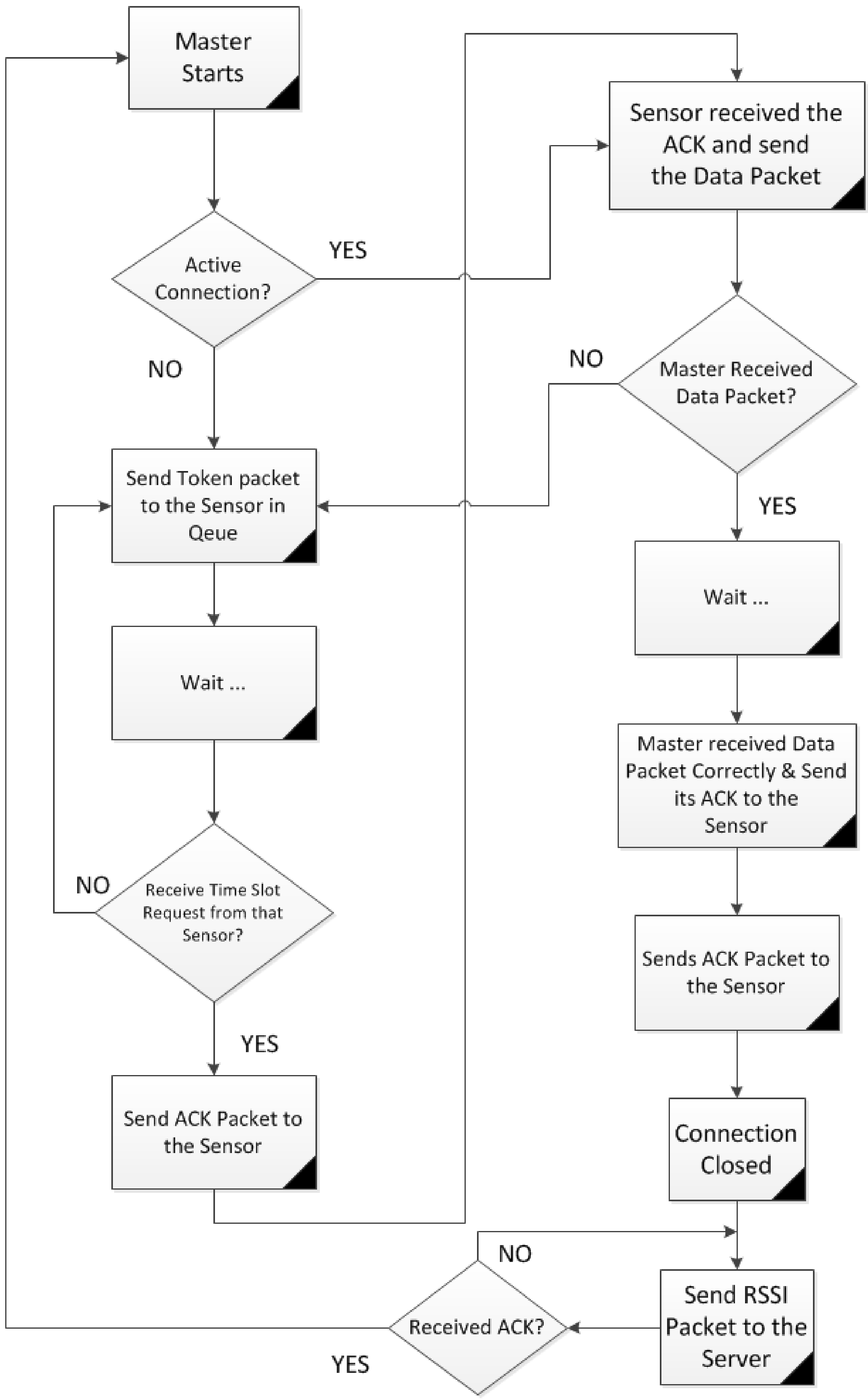


**Figure 33. Diagram of the Master Instruction**

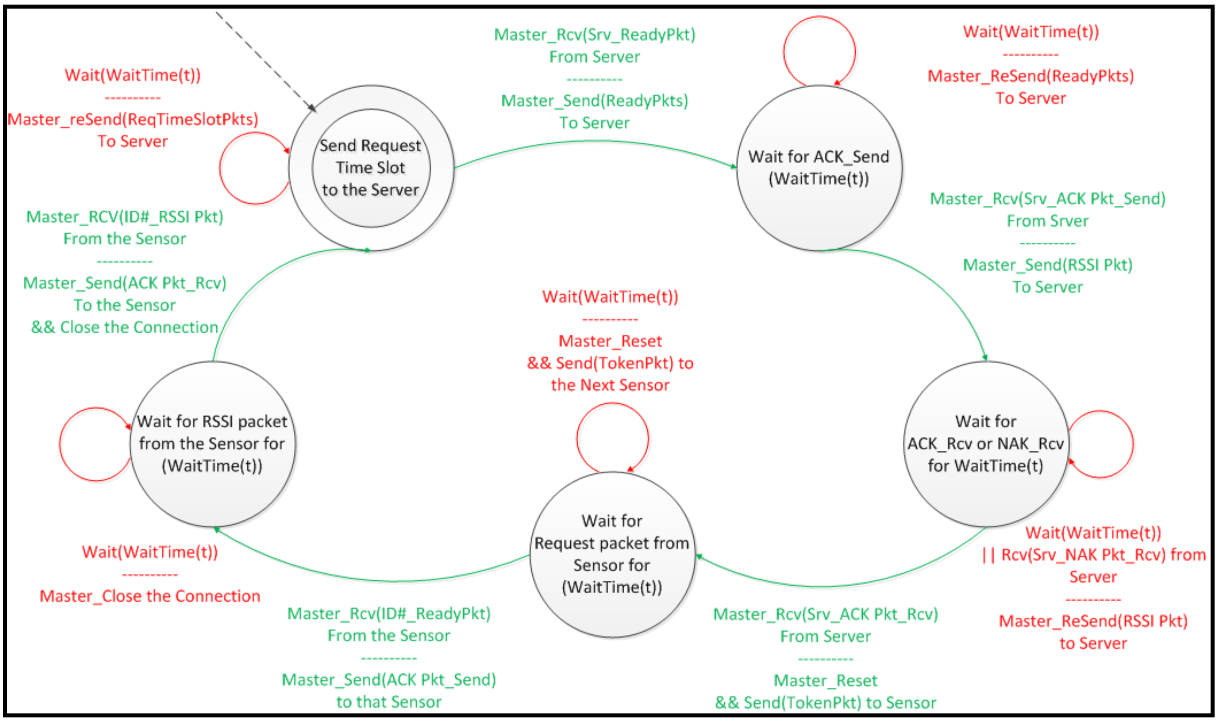


**Figure 34. State Diagram of the Master Instruction**

### *5.2.5 - Server (Srv)*

"Final destination of the RSSI information for localization procedure

and also the task manager of sensors and master."

The final procedure of localization is done in server. We used C# with visual studio 2008 framework to write localization application or LOC-RSSI. After receiving the RSSI packets from master to the server, some instructions like adding some filters on data, calculating the average and etc. are done which in specific chapter will be presented completely.

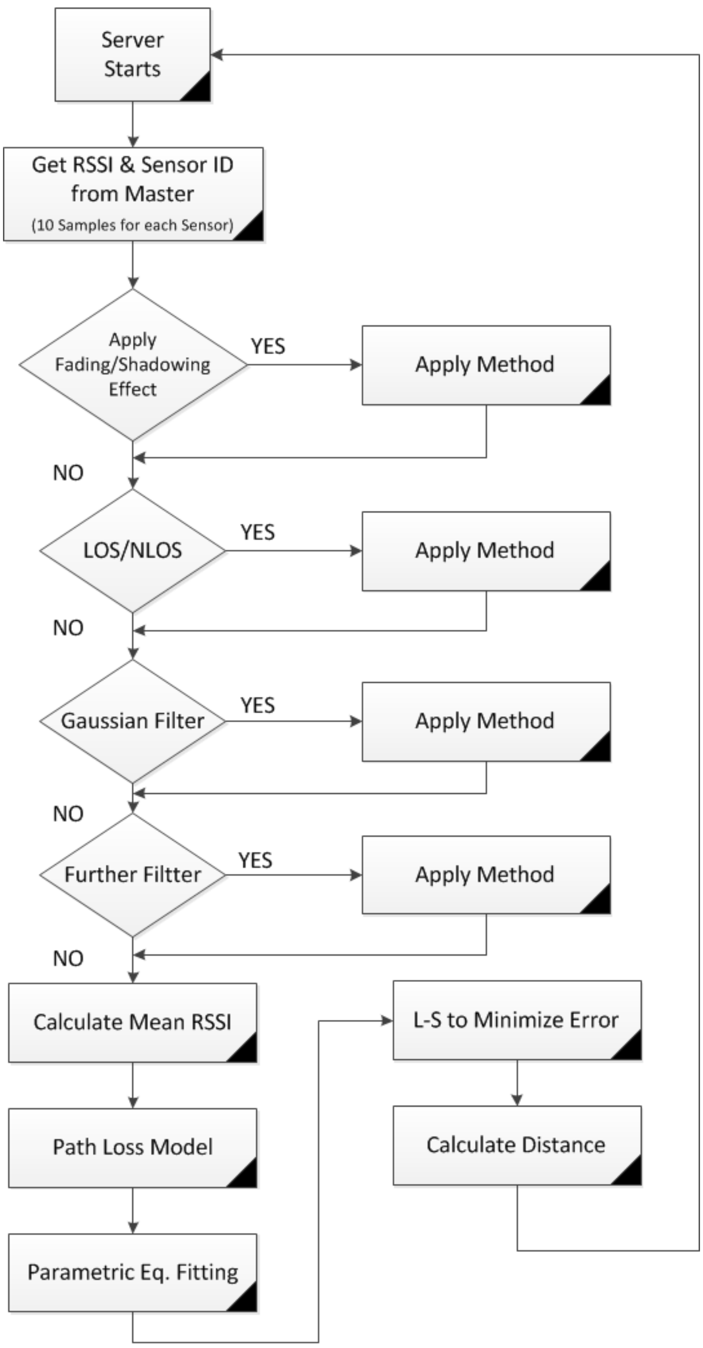


Figure 35. Diagram of the Server Instruction

### *5.2.6 - 868 MHz Radio Spectrum*

This spectrum was recommended and adopted by the Frequency Management, Regulatory Affairs and Spectrum Engineering Working Groups as well as the European Conference of Postal and Telecommunications Administration, as a long range solution for the European market. The respective radio spectrum is specific to the European market and falls within the 868.000 - 870.000 MHz frequencies and is separated into 4 sections, G1 toG4:

| | **Frequency Band** | **Power Magnetic Field** | **Duty Cycle** | **Channel Spacing** | **Notes** |
|---|---|---|---|---|---|
| **G1** | 868.000 To 868.600 MHz | ≤ 25 mW ERP (Effective Radiated Power) | ≤ 1% or LBT | No spacing, for 1 or more channels | Narrow / wide-band modulation. No channel spacing, however the whole stated frequency band may be used. |
| **G2** | 868.700 To 869.200 MHz | ≤ 25 mW ERP (Effective Radiated Power) | ≤ 1% or LBT | No spacing, for 1 or more channels | Narrow / wide-band modulation. No channel spacing, however the whole stated frequency band may be used. |
| **G3** | 869.400 To 869.650 MHz | ≤ 500 mW ERP (Effective Radiated Power) | ≤ 10% or LBT | 25 kHz (for 1 or more channels) | Narrow / wide-band modulation. The whole stated frequency band may be used as 1 channel for high-speed data transmission. |
| **G4** | 869.700 To 870.000 MHz | ≤ 5 mW ERP (Effective Radiated Power) | Up to 100% | No spacing, for 1 or more channels | Narrow / wide-band modulation. No channel spacing, however the whole stated frequency band may be used. Audio applications excluded. Voice applications allowed with LBT together with 1 minute carrier time-out timer. |

Table 4. Four sections of 868 MHz Radio spectrum

In selecting the applied radio standard some points are important which force consumers to choose suitable frequency according to their needs and purposes:

- The distance between sender and receiver
- The battery life time (Low Energy Consumption)
- The amount of data to transmit per second
- The reliability of transmission

According to about points and in the license free bands there several possibilities:

- The 2.4 GHz band using WiFi, Bluetooth or Zigbee
- The 868 MHz band with optional use of Zigbee
- The 434 MHz band
- The RFID frequencies (13.56 MHz, 125 KHz and etc.)

For using in office space and transmitting large amount of data over a short distance like 100 meters free line-of-sight or 10 meters typically in buildings, WiFi or wireless LAN is used. But the energy consumption of that is high and relatedly we will have low battery life time. Also congestion may happen in the extensive use. In similar characteristics as WiFi with lower energy consumption and good data security, Bluetooth would be the choice. New standard of 2.4 GHz band which especially designed for transmitting small amounts of data is Zigbee. The energy consumption of Zigbee is low and has reliable transmission.

For smart sensors because of superior characteristics with respect to range, 868 MHz standard is used. Its range in free line-of-sight is 2 to 3 times more than the previous standards and in building is 2 times. And also its implementation is relatively at low cost and its energy consumption is low leading to battery lifetimes from 3 to 5 years according to the transmission intervals. Notice that the 868 MHz is not a worldwide standard and in US requires the 915 MHz band instead of that.

The 434 MHz band is better in term of range but reliable transmission is not guaranteed through the use of the large amount of device in this band. Finally the range of RFID frequencies is too limited (less than 1 meter).

According advantage and disadvantage of above standards we decided to choose 868 MHz according to its power consumption and is less sensitive to the change of environment like humidity with compare to the WiFi and Zigbee.

### *5.2.7 – RS-485 Cable*

RS485 are serial communication methods for computers and devices. RS485 is the most versatile communication standard in the standard series defined by the EIA, as it performs well on all below points:

Connect DTE's (Data Terminal Equipment) directly without the need of modems

Connect several DTE's in a network structure

Ability to communicate over longer distances

Ability to communicate at faster communication rates

That is why RS485 is currently a widely used communication interface in data acquisition and control applications where multiple nodes communicate with each other.

| **Characteristics of EIA RS-485** | | | |
|---|---|---|---|
| **Physical media** | Twisted pair | **Network topology** | Point-to-point, Multi-dropped, Multi-point |
| **Maximum devices** | 32 - 256 devices | **Maximum distance** | 1200 meters (4000 feet) |
| **Mode of operation** | Differential Signaling | **Maximum baud rate** | 100 kbit/s - 10 Mbit/s |
| **Voltage levels** | -7 V to +12 V | **Mark (1)** | Positive Voltages (B-A > +200 mV) |
| **Space (0)** | Negative voltages (B-A < -200 mV) | **Available signals** | Tx+/Rx+, Tx-/Rx- (Half Duplex) Tx+, Tx-, Rx+, Rx- (Full Duplex) |

Table 5. Characteristic of RS-485

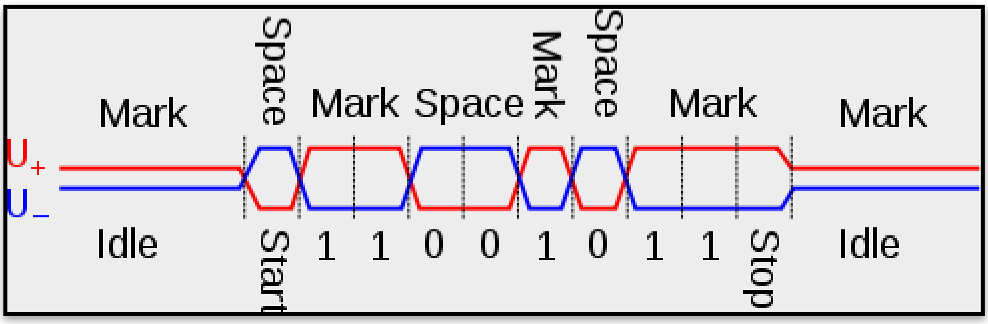


Figure 36. Waveform of RS-485

In this project the RS485 is the wire to connect the sensors to the master as you can see in Fig. 1. The model which we used is RS485 2 pair cable that 1 pair is used for power and another pair is used for communicating between master and sensors.

### *5.2.8 - RS-232 Cable*

To connect the master to the server the RS-232 wire cable has been used. RS-232 is the traditional name for a series of standards for serial binary-ended data and control signal connecting between a Data Terminal Equipment (DTE) and a Data Circuit-terminating Equipment (DCE) and it is commonly used in computer serial ports. In RS-232, data is sent as a time-series of bits in synchronous and asynchronous transmission.

In our project we used 3-wire RS-232 consisting only of transmit data, receive data and ground.

| **Signal** | | | **Origin** | | **DB-25 pin** |
|---|---|---|---|---|---|
| **Name** | **Typical purpose** | **Abbreviation** | **DTE** | **DCE** | |
| Data Terminal Ready | Indicates presence of DTE to DCE. | DTR | ● | | 20 |
| Data Carrier Detect | DCE is connected to the telephone line. | DCD | | ● | 8 |
| Data Set Ready | DCE is ready to receive commands or data. | DSR | | ● | 6 |
| Ring Indicator | DCE has detected an incoming ring signal on the telephone line. | RI | | ● | 22 |
| Request To Send | DTE requests the DCE prepare to receive data. | RTS | ● | | 4 |
| Clear To Send | Indicates DCE is ready to accept data. | CTS | | ● | 5 |
| Transmitted Data | Carries data from DTE to DCE. | TxD | ● | | 2 |
| Received Data | Carries data from DCE to DTE. | RxD | | ● | 3 |
| Common Ground | | GND | common | | 7 |
| Protective Ground | | PG | common | | 1 |

Table 6. RS-232 signals and pins assignments

## 5.3 – Localization

Up to know the procedure of receiving and transferring RSSI packets from sensors to the master through the master, and then sending them to the server has been proposed. So in final step server should estimate the location of the moving object according to the received RSSI packets. This step has several phases which are done sequentially and will be presented in this chapter.

LOC Methodology contains:

1. Receive RSSI and Optimize it, by using Gaussian filter inequality check and statistical mean
2. Path Loss Exponential calculation with STD deviation for channel LOS and NLOS
3. Calculating Gaussian random variable with zero mean and sigma for each channel for fading and shadow effects
4. Calculating linear system parameter estimation using L-S when path loss is in dB
5. Channel identification using Probability Density Function (PDF)
6. Calculate distance for specific channel (log normal path loss model)
7. Polynomial (Degree3) curve fitting for error minimization

$$F(\log(error)) = A(RSSI)^3 + B(RSSI)^2 + C(RSSI) + D$$

These parameters are estimated using known distance and Least Square (LS) at calibration phase.

### *5.3.1 – Received Signal Strength*

The propagation of the wireless signal is extremely random according to the situation of the environment so server received different numbers of RSSI packets from different sensors. Several causes like humidity, existence of the obstacle, multipath signal, diffraction, distraction and also temperature will have effect on the RSSI and surely estimating the location according to these kind of data would not be accurate. Thus in first phase

***Optimization*** on RSSI value would be necessary. Using statistical mean and Gaussian filter inequality based on the distance between the sensors are commonly used.

Moving object as describe above continuously transmit a packet . Sensor reside in known position and receive each packet transmitted form the Moving object. When it receive the packet , sensor calculate the signal strength of the packet and when it is connected with the master it send the RSSI information of the moving object associated with it ID. RSSI varies time by time, in a same position in our experiment show that it is not fixed at a value in a certain position. And also at real time many environment obstacle have impact on the RSSI. The propagation of wireless signal is extremely random. There are various interference sources and unstable factors which would affect the propagation of wireless signal in real environment, such as, temperature, multipath signals, diffraction, obstacle and so on. It cannot calculate the precise distance by the values with interference, so the RSSI values must be optimized first. The average statistical model, Gaussian model and the correction model based on the distance between the fixed nodes are commonly used to optimize the RSSI value. The results optimized by the average statistical model had a poor effect in the case of large disturbance. The values of the high probability areas can be selected by Gaussian model which can improve the precision of the location results For this reason in this work involve one Gaussian filter on the Received RSSI sample set . Gaussian filter reduces the noises that resides inside the RSSI in a given Sample set .

Gaussian filter used in this work as follows:

$$u = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n} (X_i - m)^2$$

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-u)^2}{2\sigma^2}}$$

In this work inequality is checked between 0.6 to 1.00 as follows:

```
For each RSSI sample in the Set
        gRssi = GaussianFilter();
                if(gRssi >0.6 and gRsssi <1.0)
                stackRssi ← gRssi
```

Finally when every sample is evaluated from the sample set, and new sample set is constructed by comparing the inequality, statistical mean is calculated. This mean value is used in the next phase for the localization.

### *5.3.2 – Path Loss Exponential*

The principle of RSSI ranging describes the relationship between transmitted power and received power of wireless signals and the distance among nodes. This relationship $P_r$ is the receive power wireless signals. $P_t$ is the transmitted power of wireless signal. d is the distance between the sending nodes and receiving nodes and n is the transmission factor whose value depends on the propagation environment.

$$P_r = P_t \times (\frac{1}{d})^n$$

Currently, RSSI propagation models in wireless sensor networks include free-space model, ground bidirectional reflectance model and log-normal shadow model.

Free-space model is applicable to the following occasions: 1) the transmission distance is much larger than the antenna size and the carrier wavelength λ; 2) there are no obstacles between the transmitters and the receivers. Suppose the transmission power of wireless signal is the power of received signals of nodes located in the distance of d can be determined by the following formulas:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$$

$$PL(dB) = 10\log\frac{P_t}{P_r} = -10\log[\frac{\lambda^2}{(4\pi)^2 d^2}]$$

$G_t$ , $G_r$ are antenna gain, and L is system loss factor which has nothing to do with the transmission. Surface bidirectional reflectance model is applicable to the following occasions: 1) transmission distance d is in a few kilometers or so; 2) the height of antenna of transmitter and receiver is more than 50 meters or more. The model is very accurate when it is used in the urban micro-cellular environment. The received power is determined by the following formulas:

$$P_r(dB) = P_t G_t G_r \frac{{h_t}^2 {h_r}^2}{d^4}$$

$$PL(dB) = 40\log d - (10\log G_t + 10\log G_r + 20\log h_t + 20\log h_r)$$

Where $h_t$ is the height of sending antenna and $h_r$ is the height of receiving antenna.

Log-normal shadow model is a more general propagation model. It is suitable for both indoor and outdoor environments. The model provides a number of parameters which can be configured according to different environments. The calculation formula is as follows:

$$PL(d) = \overline{PL}(d) + X_\sigma = \overline{PL}(d_0) + 10\beta\log\left(\frac{d}{d_0}\right) + X_\sigma$$

The parameter $d_0$ is the near-earth reference distance, which depends on the experiential value; the parameter β is a path loss index, which depends on specific propagation environment, and its value will become larger when there are obstacles; the parameter $X_\sigma$ is zero-mean Gaussian random variable. The parameter $d_0$, β and $X_\sigma$ describe the path loss model which has a specific receiving and sending distance. The model can be used for general wireless systems design and analysis.

LNSM or Log Normal Shadowing Model is a more general propagation model. In practical applications, it is indispensable to adjust $\overline{PL}(d_0)$, β and $X_\sigma$ in the model according to specific environment. In general, the model's parameters are set based on experiences, so it does not have the self-adaptability.

### *5.3.3 – Gaussian Random Variable*

As mentioned before the problem with Log-distance model is not considering the existence of vastly differ locations in equal situation of Tx-Rx according to different surrounding environment, so the measured signals can greatly differ from the average values predicted by this path loss model. To solve this problem actual value have been shown to be random and distributed ***Log-normal*** in dB about the mean distance dependent value:

$$PL(d) = \overline{PL}(d) + X_\sigma = \overline{PL}(d_0) + 10\beta \log\left(\frac{d}{d_0}\right) + X_\sigma \quad (*) \qquad \text{PL(d) in dB}$$

$X_\sigma$: *zero-mean Gaussian Random Variable in dB with standard deviation σ in dB*

The Log-normal variable $X_\sigma$, describes the shadowing effect which occur over many measured signal strength at location with equal Tx-Rx separation. So by considering that the antenna gains included in PL(d) we will have:

$$P_r(d) = P_t - PL(d) \qquad P_r \text{ and } P_t \text{ in dBm}$$

In ***Log-normal Shadowing*** we will have below equation for the probability that the received signal level exceed a particular level if PL(d) is a normally distributed random variable:

$$Q(z) = \frac{1}{\sqrt{2\pi}} \int_z^\infty e^{-\frac{x^2}{2}}\, dx = \frac{1}{2}[1 - \operatorname{erf}\left(\frac{z}{\sqrt{2}}\right)]$$

This probability for received signal in dB for exceed a value γ:

$$\Pr[P_r(d) > \gamma] = Q(\frac{\gamma - \overline{P_r}(d)}{\sigma})$$

This probability for received signal in dB for fall below a value γ:

$$\Pr[P_r(d) < \gamma] = Q(\frac{\overline{P_r}(d)-\gamma}{\sigma})$$

To calculate the value γ we use the measured value in initializing phase and by using it in above equation we can decrease the effect of shadowing in the real time situation. Log-normal shadow model is a more general propagation model. It is suitable for both indoor and outdoor environments. The model provides a number of parameters which can be configured according to different environments.

The log-normal probability density function is given by:

$$f_{LN}(x) = \frac{1}{\sqrt{2\pi}\sigma x}\exp(\frac{-(\ln x - \mu)^2}{2\sigma^2})$$

μ: *The mean received signal strength*

σ: *Standard deviation*

By experiment over some of signal strength group (d, RSSI) the below function is defined between variance and distance:

$$\sigma(d) = ad^3 + bd^2 + cd + e \quad (**)$$

The symbols a, b, c and e are undetermined coefficients, the values of which are dynamically adjusted according to different environments. Choosing the model of the σ(d) is related to the form of the graph between variance and distance. For instance according to the measured value the polynomial form would be suitable or maybe in some other situation the linear model is the best one.

By putting the (**) into (*) we have:

$$PL(d)(dB) = \overline{PL}(d) + X_\sigma = \overline{PL}(d_0) + 10\beta\log\left(\frac{d}{d_0}\right) + \sigma(d)X$$

The function $\sigma(d)$ makes Log-normal shadowing model be able to dynamically control the error function according to difference distances.

### *5.3.4 – Empirical Distance Estimation*

When path loss is measured in dB, the equation becomes a linear equation. Many author represent it as R = Kd + S, where r is the output in this work it is considered as Received Signal Strength. K is the system parameter and S is noise associated terms. In the calibration phase the distance D and RSSI or R is known, which can establish a satisfactory linear equation. For each sample point in the calibration phase each equation is established. The calibration phase parameter K, S is estimated by using Least Square method.

When these parameters of the linear system are estimated, this linear system is used to measure the distance by using the RSSI, which is called empirical distance. Channel identification involve calculating probability density function, which require to the actual distance. In real time it is not possible to provide the actual distance to channel identification process.

### *5.3.5 – Channel Identification*

Considering position of the Transmitter antenna and Receiver antenna, channel can be defined as both Line of Sight and Non – Line of Sight. The Channel Identification Involves to determine whether the Receive data come from the Line of Sight or Non Line of Sight. Non-line-of-sight (NLOS) propagation can severely degrade the reliability of communication and localization accuracy. Link adaptation and NLOS bias mitigation techniques have respectively been proposed to alleviate these effects, but implicitly rely on the ability to accurately distinguish between LOS and NLOS propagation scenarios. A statistical NLOS identification technique based on the hypothesis-testing of received signal parameters.

The formulation of the NLOS identification problem is as follows: to identify the state H of the channel between the transmit and receive nodes, separated physically by a distance d, given estimates of RSS observed at the receive node. Here, the channel state H =H0 corresponds to LOS propagation and H = H1 corresponds to NLOS propagation. NLOS propagation has popularly been classified as 'soft' ('obstructed') NLOS, where the LOS multi path component is present albeit attenuated, and 'hard' NLOS, where the LOS path is severely attenuated or absent. From a localization standpoint, the soft-NLOS cases are classified as LOS scenarios, since the LOS multi path component can still be estimated for ranging purposes, that is, range estimates are not necessarily biased in soft-NLOS propagation environments. In the following development, it present statistical models for the RSS estimates in an indoor environment , considering both LOS and NLOS scenarios.

The normalized RSS (in dB) is defined as:

$$S_{dB} = 10\log_{10}(\frac{1}{P_0 T}\int_T |r(\tau)|^2 d\tau$$

Where $P_0$ is the received power measured at the reference distance $d_0$, and T is the measurement interval for the received signal. Because of the absence of significant multipath fading, the estimated RSS has been modeled as a lognormal random variable .

$$\hat{S}_{dB} = 10\beta\log_{10}(\frac{d}{d_0}) + X$$

Where β is the path loss exponent, and X is a zero-mean Gaussian random variable with different variances in LOS and NLOS scenarios.

$$f_X(x) = \begin{cases} (2\pi\sigma_{SL}^2)^{-\frac{1}{2}}\exp\left(-\frac{x^2}{2\sigma_{SL}^2}\right), & H = H_0 \\ (2\pi\sigma_{SN}^2)^{-\frac{1}{2}}\exp\left(-\frac{x^2}{2\sigma_{SN}^2}\right), & H = H_1 \end{cases}$$

Therefore in LOS and NLOS scenarios, the conditional density functions of $\hat{S}_{dB}$ can be succinctly represented by

$$f_{\hat{S}_{dB}}(S|d,H) = \begin{cases} (2\pi\sigma_{SL}^2)^{-\frac{1}{2}}\exp\left(-\frac{(S-10\beta_L\log_{10}d)^2}{2\sigma_{SL}^2}\right), & H = H_0 \\ (2\pi\sigma_{SN}^2)^{-\frac{1}{2}}\exp\left(-\frac{(S-10\beta_L\log_{10}d)^2}{2\sigma_{SN}^2}\right), & H = H_1 \end{cases}$$

The conditional distributions of the TOA, RSS and RDS estimates are functions of the distance d and the channel state H, Provided the physical distance d between the transmit and receive nodes is known exactly, the state of the channel can be identified by comparing the likelihood values for each of the estimates $x_i$, conditioned on the distance d, for both H =$H_0$ and H = $H_1$.

### *5.3.6 – Calculate Distance for a specific Channel*

For the known value of d, the likelihood that the available estimates occurred because of NLOS propagation rather than LOS propagation is higher, and the maximum likelihood (ML) channel state. would be $\widehat{H} = H_1$. Evidently, if d is known, based on the likelihood values of any of the given estimates, the ML channel state can be directly obtained from the known conditional distributions of the estimates. In reality however, we do not know the physical distance d between the nodes. Indeed, for localization applications, estimating the distance d accurately is a requirement. For this reason we use Empirical Distance Estimation by fitting a linear model at calibration phase to estimate the linear system model parameter.

The formulation of the joint channel state identification is as follows: our initial hypothesis is that the state of the channel is LOS, that is, $\widehat{H} = H_0$, and that d is estimated from the linear model. For each of the given estimates, it is needed to determine the conditional probabilities.

$$p_i = Pr\{X_i = x_i | \hat{d}, H_0\}$$

$$q_i = Pr\{X_i = x_i | \hat{d}, H_1\}$$

Then need to compute $D_L = \prod_i p_i$ and $D_N = \prod_i q_i$. It must be pointed out that if the estimates are assumed to be independent for a fixed distance and channel state, $D_L$ and $D_N$ are the joint conditional probabilities of the estimates. The values of $D_L$ and $D_N$ can then serve as the decision metrics for estimating the channel state. In particular, if $D_N$, $D_L$ we change our hypothesis of the channel state from LOS to NLOS

$$\widehat{H} = \begin{cases} H_0, & D_L > D_N \\ H_1, & D_L < D_N \end{cases}$$

The total work follow is as follows:

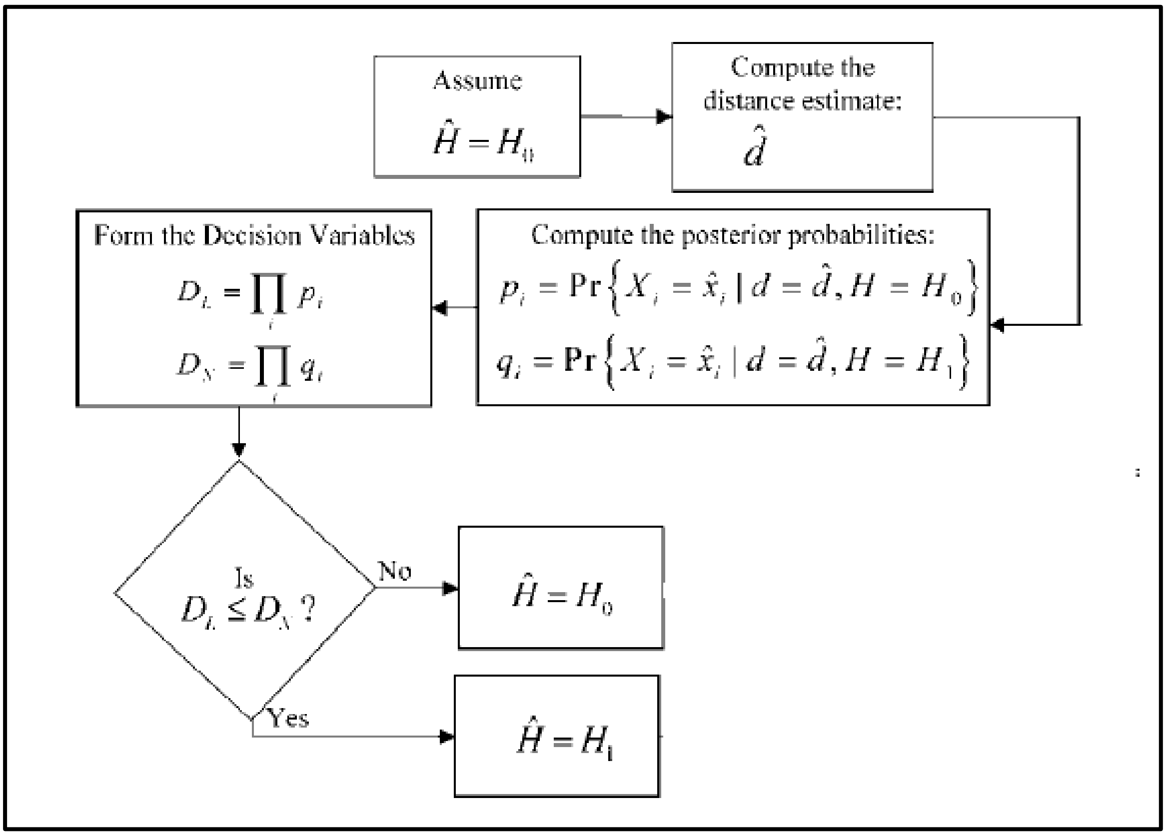


Figure 37. Algorithm of Calculating Distance for a specific Channel

# *Chapter 6:* Hardware Implementation

## 6.1 – Introduction

In this chapter we have tried to present the communication architecture between Moving Object, Sensors, PLE Node and Master. For implementing methodology we used the Concentrator V.1.0 as Mobile Object, Sensors, PLE Node and Master with different firmware. To program the concentrator the MPLAB has been used and the language of codes is in C18 and MPLAB ICD2 was the programmer which will be presented one by one in different section. The codes of each module have been proposed in appendix part.

## 6.2 – Hardware

We need four different configurations for four different connections:

- ***Moving Object to Sensors***
- ***Moving Object to PLE Node***
- ***Sensors to Master and vice versa***
- ***Master to the Server and vice versa***

For these different connections we used Concentrator V.1.0 with PIC18F47J13 high performance microcontroller with nano Watt technology from MICROCHIP company with MRF89XA Ultra-Low power and Integrated ISM band Sub-GHz transceiver from the MICROCHIP company too.

### 6.2.1 – PIC18F47J13 Microcontroller

PIC18F47J13 family 28/44-Pin is a high performance microcontrollers with nano Watt XLP technology. In details about the power management features with nano Watt XLP for extreme low power we can say that:

- Deep Sleep mode: CPU Off, Peripherals Off, SRAM Off, Currents Down to 9 nA and 700 nA with RTCC:
  - Able to wake-up on external triggers, programmable WDT or RTCC alarm
  - Ultra Low-Power Wake-up (ULPWU)
- Sleep mode: CPU Off, Peripherals Off, SRAM On, Fast Wake-up, Currents Down to 0.2 μA, 2V Typical
- Idle: CPU Off, SRAM On, Currents Down to 1.7 μA Typical
- Run: CPU On, SRAM On, Currents Down to 5.8 μA Typical
- Timer1 Oscillator w/RTCC: 0.7 μA, 32 kHz Typical
- Watchdog Timer: 0.33 μA, 2V Typical

| Features | PIC18F46J13 | PIC18F47J13 |
|---|---|---|
| Operating Frequency | DC – 48 MHz | DC – 48 MHz |
| Program Memory (Kbytes) | 64 | 128 |
| Program Memory (Instructions) | 32,768 | 65,536 |
| Data Memory (Kbytes) | 3.8 | |
| Interrupt Sources | 30 | |
| I/O Ports | Ports A, B, C, D, E | |
| Timers | 8 | |
| Enhanced Capture/Compare/PWM Modules | 3 ECCP and 7 CCP | |

| | |
|---|---|
| Serial Communications | MSSP (2), Enhanced USART (2) |
| Parallel Communications (PMP/PSP) | Yes |
| 10/12-Bit Analog-to-Digital Module | 13 Input Channels |
| Resets (and Delays) | POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT |
| Instruction Set | 75 Instructions, 83 with Extended Instruction Set Enabled |
| Packages | 44-Pin QFN and TQFP |

Table 7. Device Features for the PIC18F4XJ13

There are some peripheral highlights for this microcontrollers as:

- Hardware Real-Time Clock/Calendar or RTCC
- Four Programmable External Interrupts
- Four Input Change Interrupts
- 8-Bit Parallel Master Port/Enhanced Parallel Slave Port
- Enhanced USART modules which supports RS-485, RS-232 and LIN/J2602

Also it has some special features like:

- 5.5V Tolerant Inputs only for digital only pins
- C Compiler Optimized Architecture for Re-Entrant Code
- Priority Levels for Interrupts
- Flash Program Memory of 10,000 Erase/Write Cycles Minimum and 20-Year Data Retention

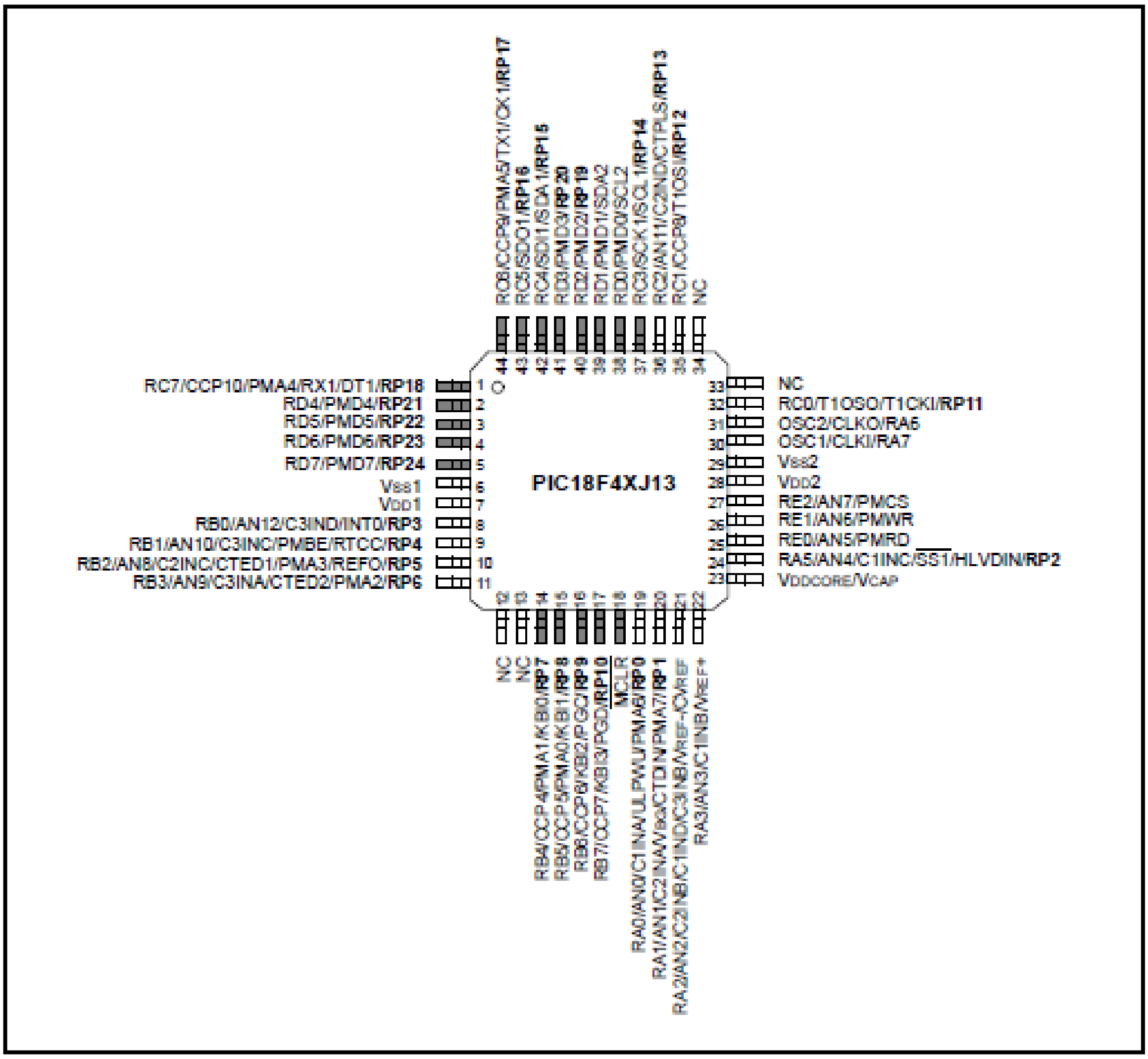


Figure 38. Pin Diagram of PIC18F4XJ13

### *6.2.2 – MRF89XA Transceiver*

MRF89XA us is an ultra-low power fully integrated ISM band which supports proprietary Sub-GHz wireless protocols with wide-band half-duplex transceiver. It is simple 4-wire SPI compatible interface with on-chip oscillator circuit that has dedicated clock output and supports power-saving modes.

- Operating voltage: 2.1V-3.6V
- Low-current consumption, typically:
  - 3 mA in RX mode
  - 25 mA @ +10 dBm in TX mode
  - 0.1µA (Typical) and 2 µA (Maximum) in Sleep mode

In field of radio frequency and analog features supports ISM band sub-GHz frequency ranges 863-870, 902-928 and 950-960 which we use 868 MHz in our project. And also support the FSK and OOK modulation techniques and high data rates up to 200 kbps. Its reception sensitivity down to -107 dBm at 25 kbps in FSK and -113 dBm at 2 kbps in OOK and RF output power is +12.5 dBm programmable in eight steps. Its wide received signal strength indicator (RSSI) dynamic range is 70 dB from Rx noise floor. We can mention to other features like:

- Signal-ended RF input/output
- On-chip frequency synthesizer
- Supports PLL loop filter with lock detect
- Integrated Power Amplifier (PA) and Low Noise Amplifiers (LNA)
- Channel filters
- On-chip IF gain and mixers
- Integrated low-phase noise VCO

This transceiver has 64 byte transmit/receive FIFO with preload in stand-by mode and supports Manchester encoding/decoding techniques. It is a low cost transceiver which is optimized for very low power consumption, 3 mA in receiver mode. Its highly integrated architecture allows for minimum external component count while still maintaining design flexibility. In this module the RF communication parameters are made programmable and most of them may be dynamically set.

Typical applications of this module are home, industrial and building automation, remote wireless control wireless PC peripherals, wireless sensor networks, vehicle sensor monitoring, telemetry data logging systems for home or industrial environments and medical applications.

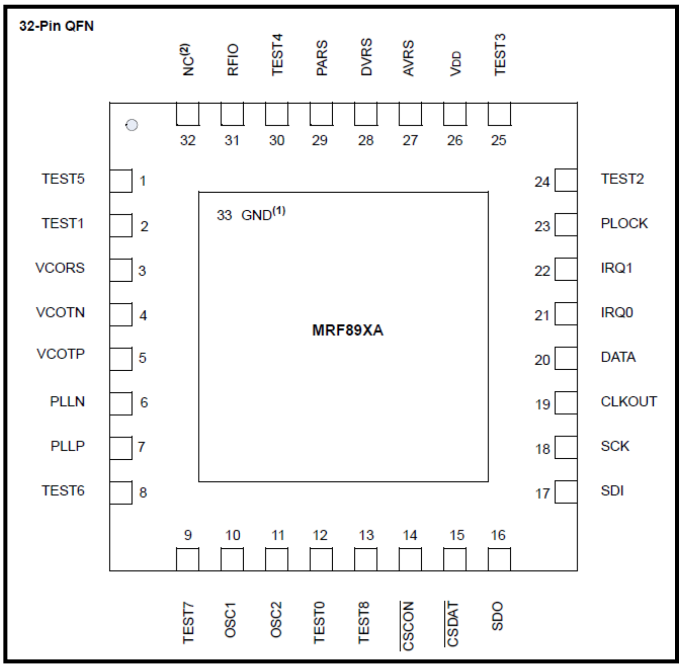


**Figure 39. MRF89XA 32-Pin QFN Pin Diagram**

In field of digital data processing the MRF89XA supports the Received Signal Strength Indicator (RSSI), sync word recognition, packet handling, interrupt and flags, different operating modes as continuous, buffer and packet, data filtering, whitening and encoding, base band power amplifier and 64-byte Tx-Rx FIFO.

In our project we used this module to interface the data to/from the microcontroller access point and controls all of the configuration registers and in general as a digital processing unit.

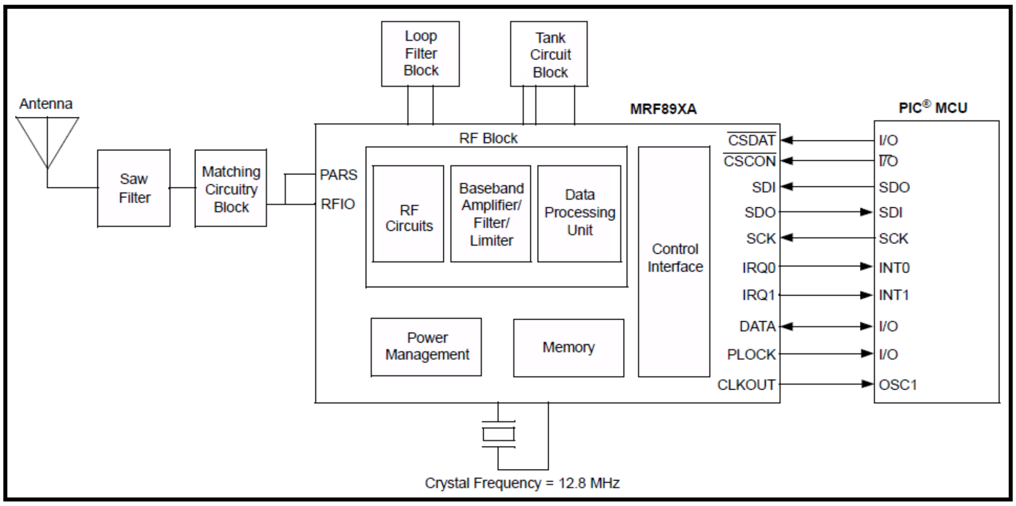


**Figure 40. MRF89XA to Microcontroller interface block diagram**

### 6.2.3 – Concentrator V1.0

In this project we used PIC18F47J13 microcontroller and MRF89XA transceiver together in a complete package as Concentrator V1.0. This module has been used as Moving Object, PLE Node, Sensors and Master by different programs.

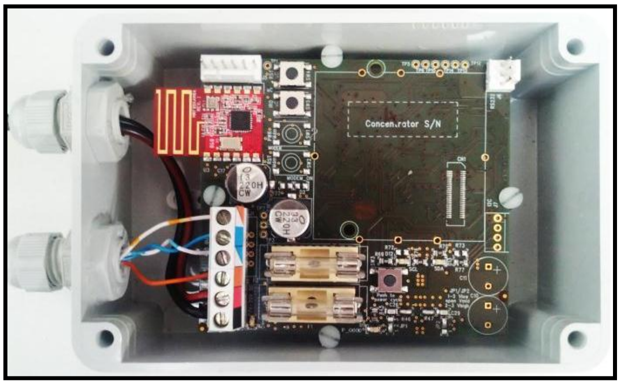

Figure 41. Image of Concentrator V1.0

This module has three different layers of the protocol:

- Physical Layer (Whitening without Manchester Encoding)
- MAC Layer (Managed by the Wireless Module)
- Application Layer

According to possibility of assigning serial number for each concentrator which is beginning with any value between "0xF0" and "0xFE", in our methodology master can send *Token Packet* to all sensors repeatedly and sequentially.

| Block | Position | Field length | Field meaning |
|---|---|---|---|
| Protocol | 1 | 1 | STX |
| | 2 | 1 | Packet Counter (Numerator) |
| | 3 | 1 | Packet Total (Denominator) |
| | 4 | 1 | Type |
| Error status | 5 | 1 | Last Sensor – Concentrator communication error |
| | 6 | 1 | Sensor – Concentrator communication failures |

| | | | |
|---|---|---|---|
| | 7 | 1 | Last Concentrator – Gateway communication error |
| | 8 | 1 | Concentrator – Gateway communication failures |
| | 9 | 4 | Error Flags |
| | 10 | | |
| | 11 | | |
| | 12 | | |
| **Serial numbers** | 13 | 4 | Sensors / Master |
| | 14 | | |
| | 15 | | |
| | 16 | | |
| | 17 | 4 | Sensor SN |
| | 18 | | |
| | 19 | | |
| | 20 | | |
| **Measurement data** | 21 | 4 | Data 1 |
| | 22 | | |
| | 23 | | |
| | 24 | | |
| | 25 | 4 | Data 2 |
| | 26 | | |
| | 27 | | |
| | 28 | | |
| | 29 | 4 | Data 3 |
| | 30 | | |
| | 31 | | |
| | 32 | | |
| | 33 | 4 | Data 4 |
| | 34 | | |
| | 35 | | |
| | 36 | | |

| | | | |
|---|---|---|---|
| | 37 | 4 | Future Data 1 |
| | 38 | | |
| | 39 | | |
| | 40 | | |
| | 41 | 4 | Future Data 2 |
| | 42 | | |
| | 43 | | |
| | 44 | | |
| | 45 | 4 | Future Data 3 |
| | 46 | | |
| | 47 | | |
| | 48 | | |
| | 49 | 2 | Sensor Temperature |
| | 50 | | |
| **Radio channel details** | 51 | 1 | RF Band |
| | 52 | 1 | RF Channel |
| | 53 | 2 | RF Measures |
| | 54 | | |
| **HW/FW revisions** | 55 | 1 | Future service info |
| | 56 | 1 | Future service info |
| | 57 | 1 | Concentrator FW Version |
| | 58 | 1 | Concentrator HW Version |
| | 59 | 1 | Sensor FW Version |
| | 60 | 1 | Sensor HW Version |
| **Protocol** | 61 | 2 | Sensor measure counter |
| | 62 | | |
| | 63 | 2 | Checksum |
| | 64 | | |

Table 8. Packet Description

The message packet is obtained by the union of 64 bytes with data exchanged using the big-endian methodology with the most significant bit of each byte send first. The important packet which are possible to send from sensors or master to another one, are:

- Radio Network Reset Packet: 0xE0
- Sync Packet: 0xE1
- ACK Packet: 0xE3
- Data Request Packet: 0x01
- Intro Packet: 0xE2
- Data Packet: 0x10

After starting the system, Radio Network Packet or RNR packet is used to perform for rebooting the master, no sensor respond and measuring of the channel noise too high. The master analyze all channels and selects the less noisy band. It scans all channels looking for the most suitable for the communication according to the minimum product between instantaneous and average power.

The Token packet would be a data request packet from master to the sensor which has serial number of the each sensor sequentially. So all sensors received this packet but just the sensor has same serial number value with serial number of packet keep this packet and continue the procedure of sending RSSI packet.

In next step that sensor sends the intro packet to the master to show that has available RSSI packet for transferring to the master. In other hand if there is no available RSSI packet does not send the intro packet and master after waiting for specific time sends next token packet to the next sensor. After sending the ACK packet from master to the sensor, the sensor by sending the data packet transfers the RSSI packet to the master, and in final step master send the ACK packet to the sensor again and closes the connection.

In our project we used some sensors in bus topology which is the simplest way to connect them to each other and according to token packet method there will be no collision problem.

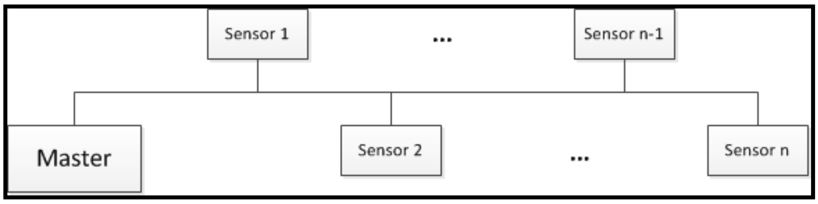


Table 9. Bus Topology for Sensors Connection

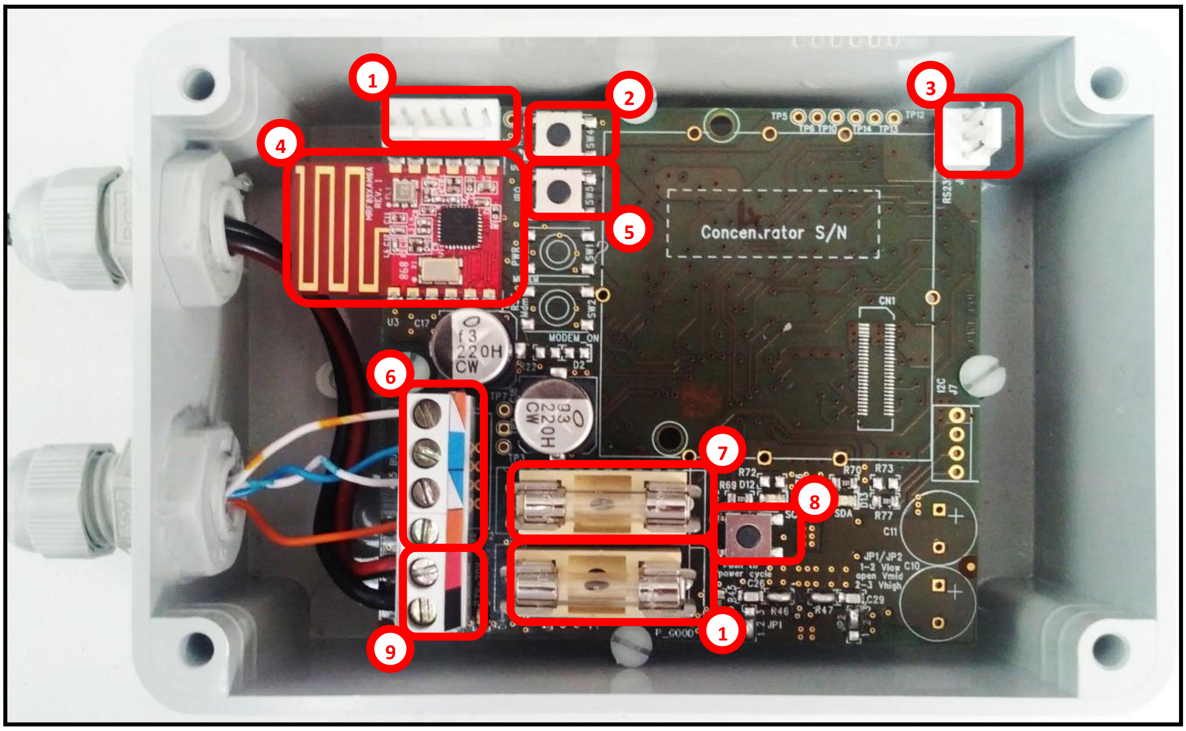

**Figure 42. Fully assembled Concentrator board –main components**

The highlighted parts are:

1. Microcontroller Programming Port (J5)
2. Microcontroller Reset Button (SW4)
3. RS-232 Interface Port (to Connect to the Server)
4. Wireless Module (MRF89XA Transceiver)
5. Concentrator Configuration Button (SW5)
6. RS-485 Interface Port for Communication to the Sensors (J2)
7. Sensor Power Rail Protection Fuse (use a 1 A fast acting fuse –F2–)
8. Power Cycle Button
9. Power Cord Connector (DC voltage between 7V and 20 V)
10. Master Power Protection Fuse (use a 1 A fast acting fuse –F1–)

## 6.3 – Communication

All sensors and master connect together by Bus Network Topology which is a network architecture in that a set of modules connected via a shared communication line called bus.

Generally systems with bus network architecture normally have collision problem because it is possible two or more modules want to transmit at the same time on the same bus. But in this project by using the token packet method we will not have this problem and also as mentioned before to overcome the negative effect of conflict between master and a specific sensor each time, Acknowledgement methodology is used to allow them to know if the communication was complete with no error.

The master can communicate with all sensors and sends and receives packets to or from them whereas each sensor can communicate with master through the RS-485 and just receives RSSI signal the moving objects.

The message packets between master and sensors have 32 bytes by below map:

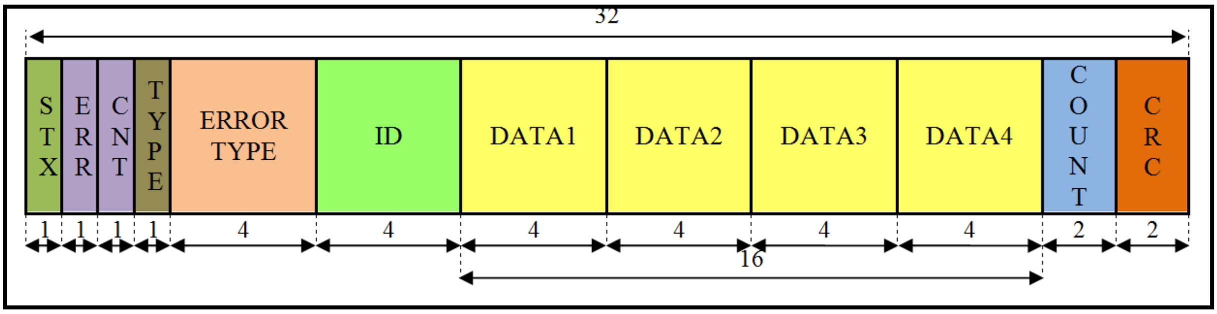


Figure 43. Transmitted Packet Map

Which each part has its mean:

- STX: Start of the frame, it represents the beginning of each packet.
- ERR: this field indicates whether a communication error has occurred between the sensor and master.
- CNT: considering the concentrator sending only the last valid collected data to master.
- TYPE: typology of the datum contained into the packet (such as a request of data or an acknowledgement)
- ERROR TYPE: typology of the error detected by the sensor.
- ID: sensor ID referenced by the message
- DATA: packet payload; according to the concentrator request, the sensor fills the necessary data fields.
- COUNT: counter of the measures. The sensor increases this internal counter by one on each new valid acquisition accomplished. The reported number can thus be used by the master to detect anomalies.

- CRC: Cyclic Redundancy Check is a non-secure hash function designed to detect accidental changes to raw computer data.

Data (sequences of bytes) are exchanged using the Little-endian methodology, with the least significant bit sent first.

Each communication can be initiated only by the master which is responsible to request data from the sensors periodically. For instance the below message would be a sample of token message from master to a specific sensor which is the request for sending available RSSI packet.

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| STX | ERR | CNT | E3 | ERROR TYPE | xx.xx.xx.xx | DATA1 | DATA2 | DATA3 | DATA4 | COUNT | CRC |

Figure 44. Sample of Token packet from master to the sensor

It is possible that some of token packets get corrupted because of contention. So to cover this problem the sending ACK packet from sensor to the master after receiving the token packet solve this problem also shows that the sensor has available RSSI packet to send. In other hand when master does not receive the ACK packet in specific time, send another token packet to the next sensor.

Each sensor has two priority interrupt. ***High*** priority interrupt which is related to the communication with moving object, and the other one is ***Low*** priority interrupt for communicating with master. By receiving the wireless signal from moving object the high priority will be activated and sensor receives the RSSI signal and after finishing this step by receiving the token packet from master the low priority will be activated and the procedure of sending RSSI packet to the master will start which has been proposed before.

## 6.4 – Firmware

To program the modules the MPLAB IDE V.8.86 from MICROCHIP company which uses Microchip C18 as tool suite for compiling. To program the concentrators the MPLAB ICD2 programmer has been used.

Figure 45. Image of MPLAB ICD

### 6.4.1 – MPLAB IDE

MPLAB IDE or Integrated Development Environment is a software program used to develop application for microcontrollers. It provides a single integrated environment to develop code for embedded microcontrollers. To write, edit, debug and program which are the intelligence of embedded system applications, into a microcontroller we need a development system for embedded controllers that the MPLAB IDE is such a system. It contains all necessary components to design and deploy embedded systems applications.

The typical tasks for developing an embedded controller application are:

1. Creating the high level design
2. Compiling
3. Testing the Code
4. Burning the code into the microcontroller

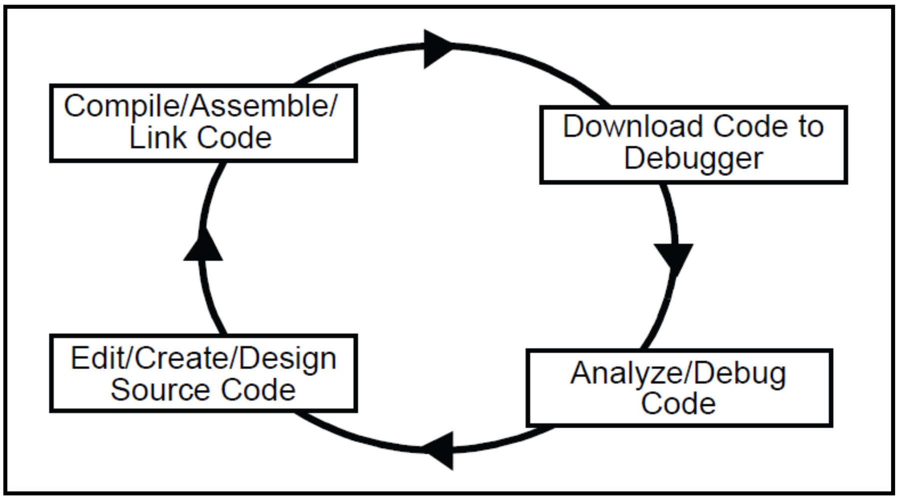


Figure 46. The Design Cycle

To organize the files to be edited and other associated files and for sending them to the language tools for assembly or compilation and ultimately to the Linker, project manager has the important role. Linker is responsible for placing the object code from assembler, compiler and libraries into the proper memory areas of the embedded controller and guarantees the correct function between each two modules. So the build phase would be the whole procedure from assembly and compilation through the link process.

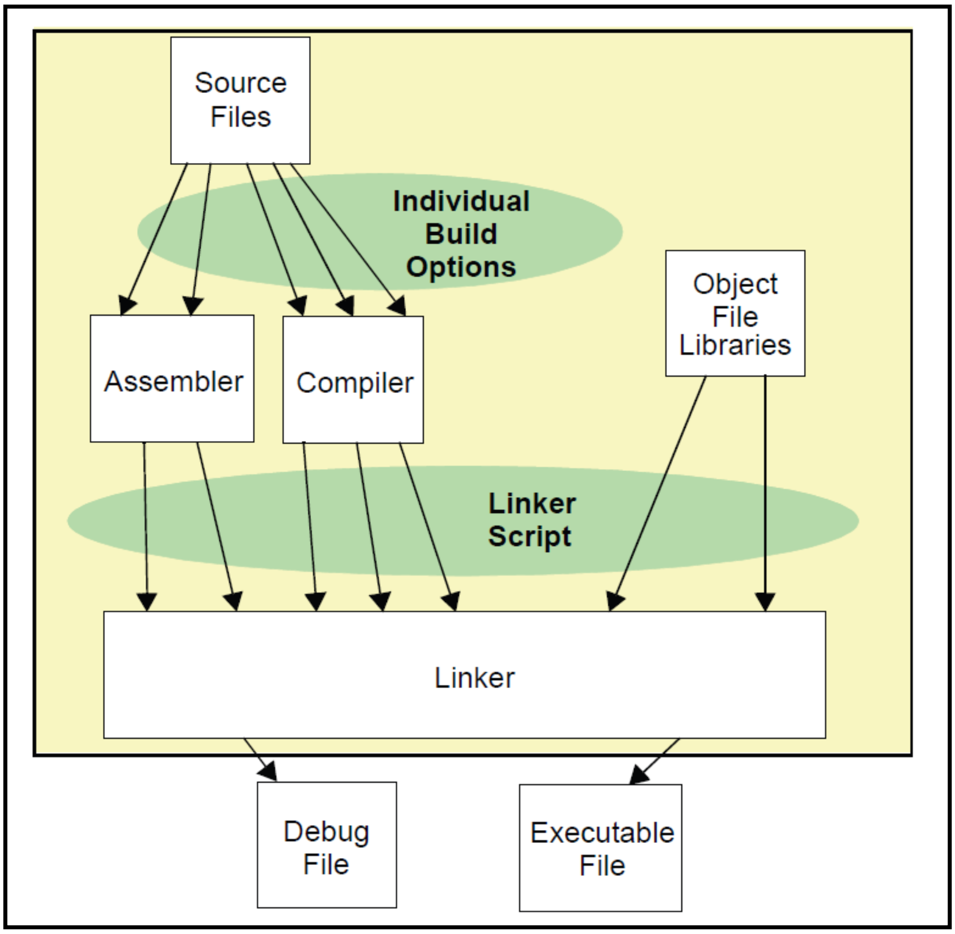


Figure 47. MPLAB IDE Project Manager

### *6.4.2 – MPLAB C18 Complier*

The MPLAB C18 C is a software tools used for embedded system programming. It examines the functions and difference between compliers and assemblers and the advantages of the C languages. He directory of this software tools provides programmers with a lot of language tools executable and execution flow. This cross-complier produces code that can be executed by microchip PIC18xxxx family of microcontrollers. Like all assemblers C18 is an interface between human and systems to translate the users states to the binary codes. The MPLAB C18 takes standard C statements and converts them into PIC18xxxx machine code.

The codes in ANSI C notation are compiled into blocks of program code and data and later have been linked to other blocks and data and then have been placed in memory region of the microcontroller, which this procedure is named “build”. C18 and its linkers and assemblers finally prepare a HEX file that can be programmed into a PIC18xxxx device. This complier makes the development of embedded systems applications easier by using the C standard language and the code of this language would be extremely efficient for the PIC18xxxx microcontrollers.

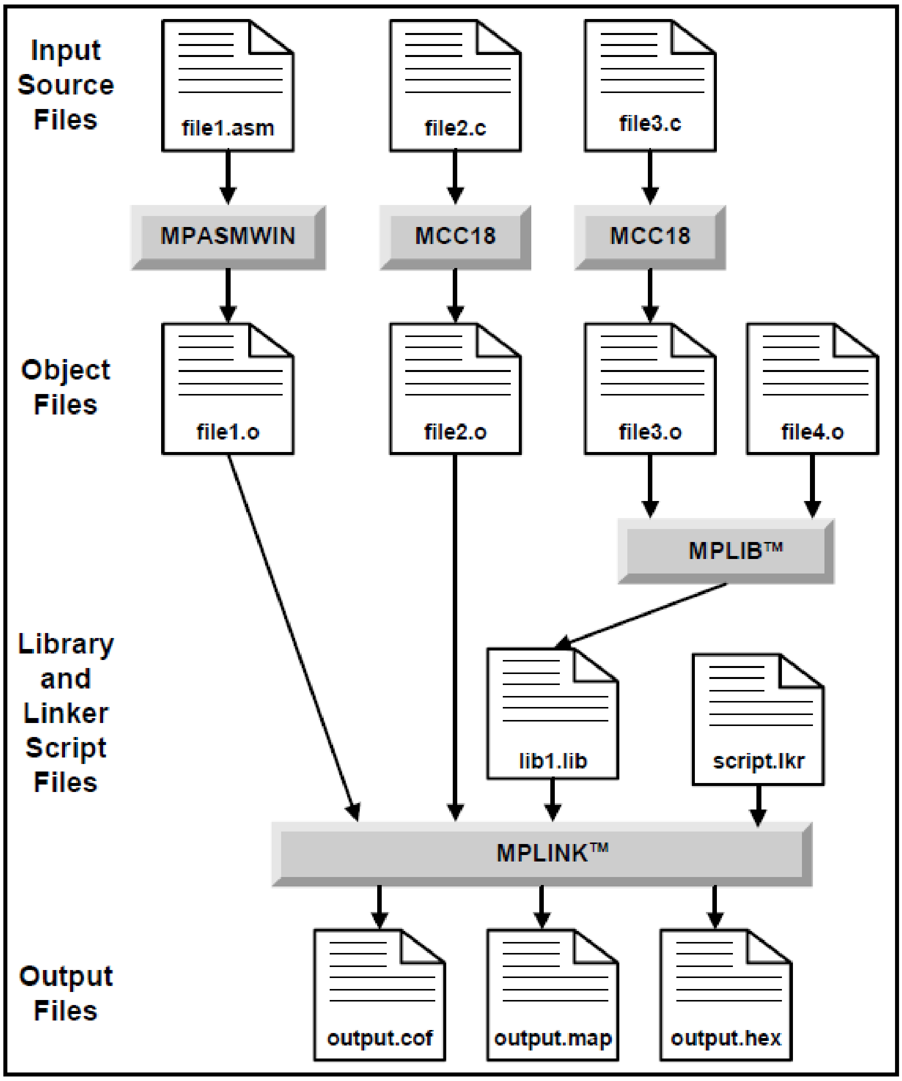


Figure 48. Language Tools Execution Flow

### 6.4.3 – MPLAB ICD2 Programmer

The MPLAB ICD2 is a low-cost In-Circuit Debugger (ICD) and In-Circuit Serial Programmer (ICSP) and is used for evaluation, debugging and programming. The below features would be the points of this programmer:

- Real-time and single-step code execution
- Breakpoints, register and variable watch/modify
- In-Circuit debugging
- Target $V_{DD}$ monitor
- Diagnostic LEDs
- MPLAB IDE user interface
- RS-232 serial or USB interface to a host PC

MPLAB ICD2 helps developers to debug their source codes in their own application and their hardware in real-time. And also helps to program a supported device using microchip's ICSP protocol.

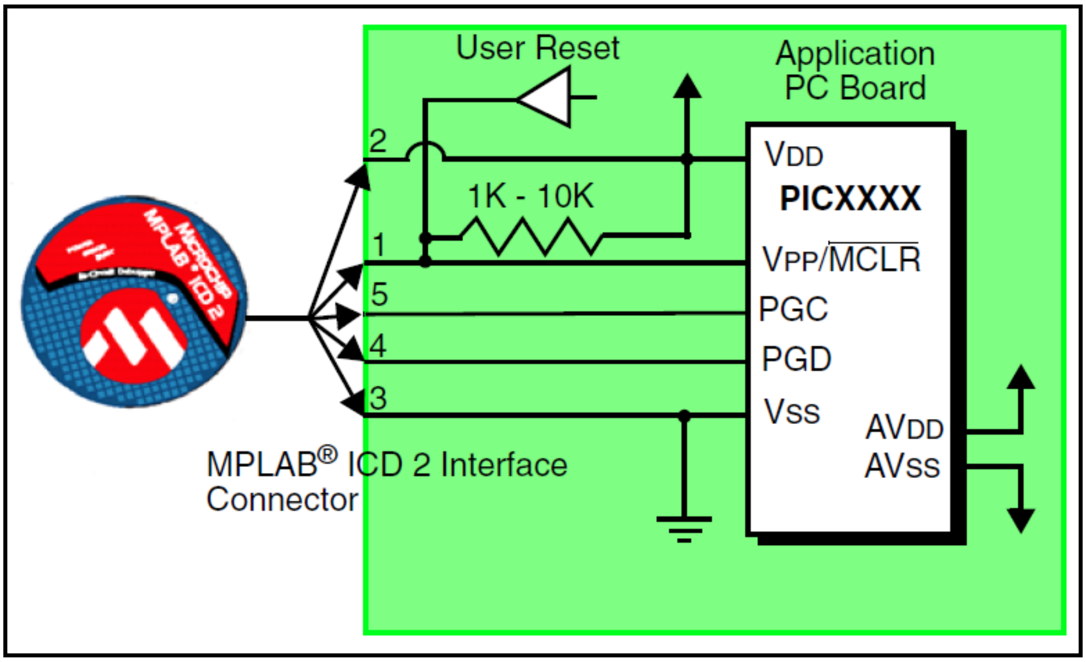


Figure 49. MPLAB ICD2 Connections to the Target Board

# *Chapter 7:* Localization Server

## 7.1 – Introduction

Localization server is developed with *Microsoft Visual Studio* using *C#*. The purpose of this application is to serve as server and perform *Localization* algorithm by using *Received Signal Strength Indicator* (RSSI) Information form the sensors.

## 7.2 - Communication Layer

Server is connected to the *Master* via RS-232. The communication Port (COM) should be defined by user from list of available communication ports in the drop down menu. When the server is able to open the desired communication port, the communication link is established. Master sends a data packet for each sensor to the server with a defined packet format sequentially.

### *7.2.1 Data Packet*

The packet format is as follows:

| **Byte** | **Contains** |
|---|---|
| 1 | 0A |
| 2 | 0D |
| 3 | I |
| 4 | D |
| 5 | = |
| 6 | Reserved for Sensor ID (MSB) |
| 7 | Reserved for Sensor ID (LSB) |

| | |
|---|---|
| 8 | # |
| 9 | R |
| 10 | S |
| 11 | S |
| 12 | I |
| 13 | = |
| 14 | Reserved for RSSI (MSB) |
| 15 | Reserve for RSSI |
| 16 | Reserve for RSSI (LSB) |
| 17 | 0A |
| 18 | 0D |

**Figure 50. Data Packet format from Master to the Server**

When the server receives the data packet from master, store it into a stack and starts parsing it until end of the packet. Parsing the packet contains retrieve the full data packet from the buffer, extract the ID information and RSSI information form the data packet and then store RSSI information into a stack for the defined sensor ID.

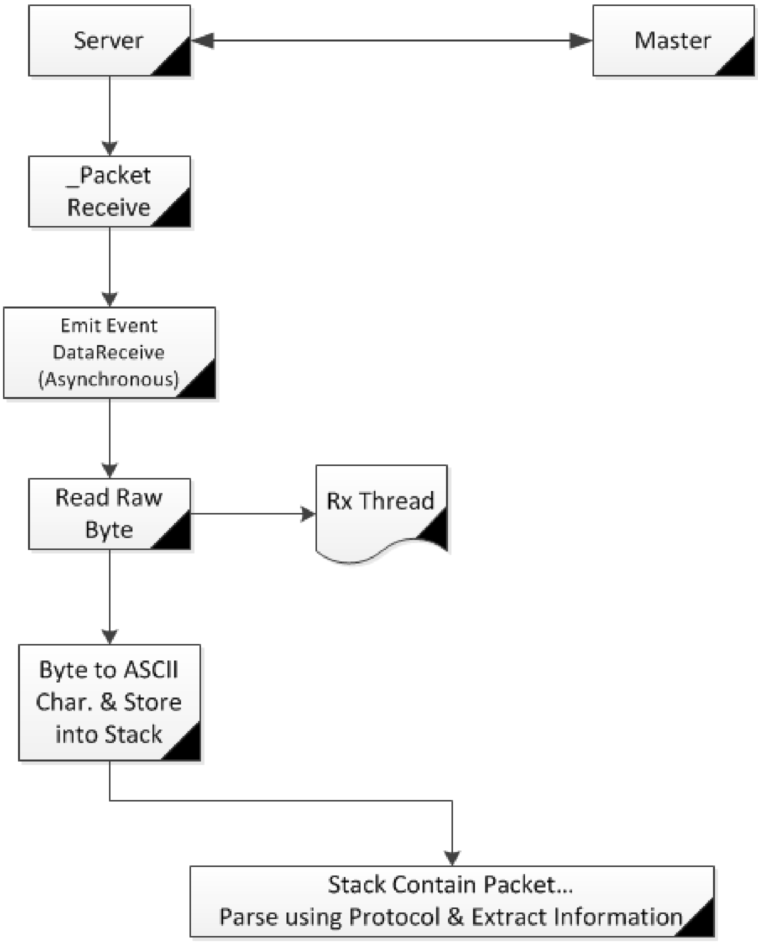


**Figure 51. RS-232 Communication**

### 7.2.2 Data Receive

In serial communication, a byte of data is transferred through a single wire one bit at a time. The packets contains a start bit, data, and stop bit. Once the start bit has been sent, the transmitter sends the actual data bits. There may either be 5, 6, 7, or 8 data bits, both receiver and the transmitter must agree on the numbers of data bits, as well as the baud rate.

Microsoft Provide one event called *DataReceived* event which is raised on a secondary thread when data is received from the serial port object. The *DataReceived* event is not guaranteed to be raised for every byte received. *BytesToRead* property is used to determine how much bytes are left or to be read. The raw bytes that are received from the serial port, are parsed as ASCII character and are saved as a character array into the *SerialPortReadBuffer* which is defined globally. It is used as an intermediate storage stack.

When this Buffer contains characters the thread starts to retrieve each char and reconstructs the data packet. Reconstruction of the message is as follows :

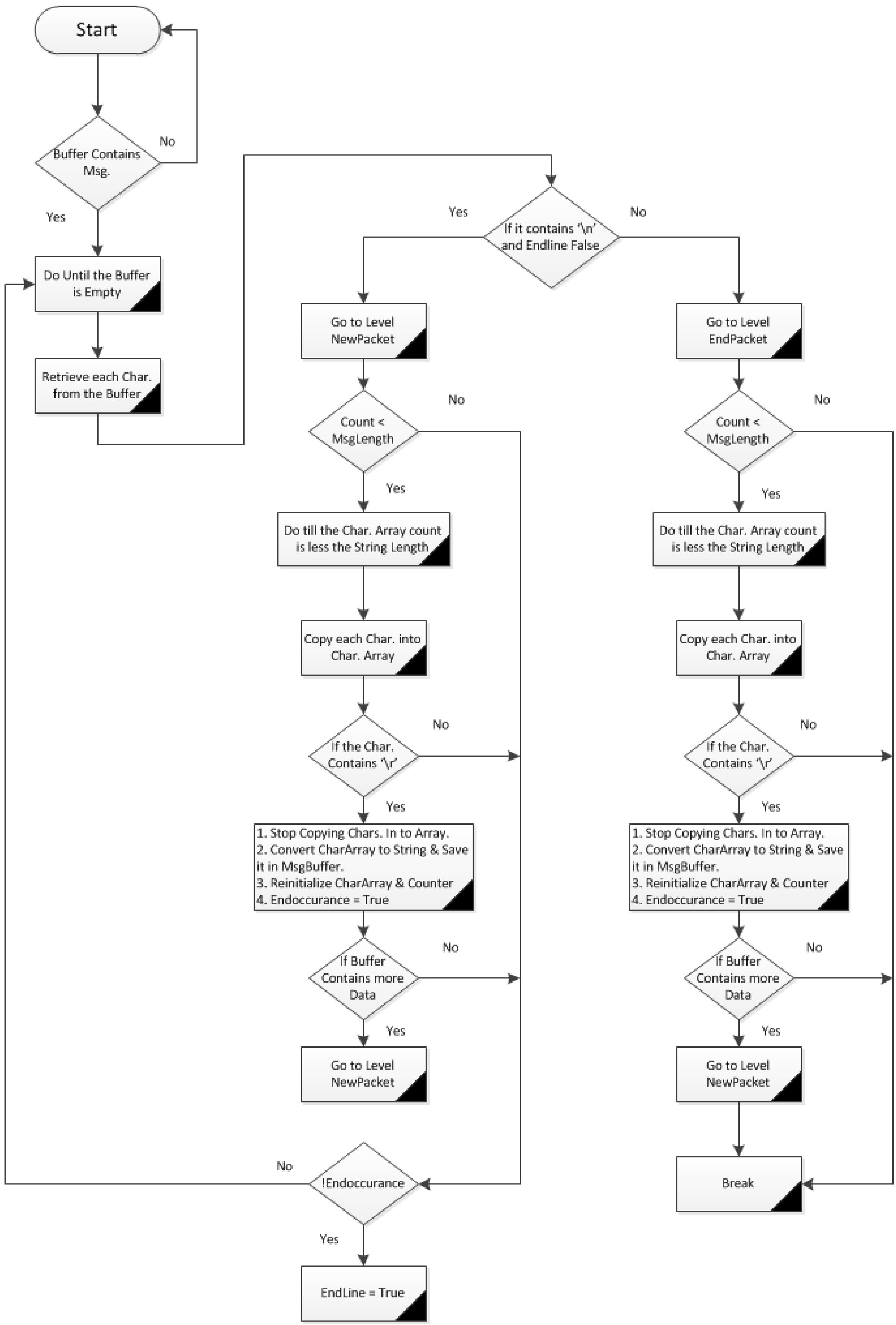


Figure 52. RS-232 Parser

## 7.3 - Parsing RSSI

When message stack contains data, it starts parsing packet. Massage comes with a predefined format contain *SensorID* and *RSSI* . Message parser parses the message into that format and extracts *SensorID* and *RSSI*. Then the RSSI is stored into the stack of the sensor according to its ID**.**

## 7.4 - Optimization of the RSSI

After having literature review, many authors used to do optimization of RSSI by taking a sample of RSSI. This work also involves such optimization, which involves performing Gaussian Filter on the sample of the RSSI. After that a new sample set of RSSI is constructed which satisfy above 0.6 value. The final RSSI is taken from the mean value of the new sample set.

## 7.5 - Least Square Estimation

Server code contains a class library which implements traditional least square estimation for a linear system. which can be done by constructing a Matrix which is Matrix A, an output vector Y which contains observed value and parameter vector X. Least Square class is implemented by a third party Matrix class library named *cMatrixLib*.

- Matrix can be declared as **IdentifierName [row, column]**
- Where vector can be declared as **IdentifierName [row, 1]**
- Row and Column value should be integer and for vector column value always 1.
- Least square library estimate the parameter for the linear system Y = A * X.

## 7.6 – Filter

Server contain one library regarding the filter. Library filter contain one function named GaussModel which perform Gaussian function as follows .

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-u)^2}{2\sigma^2}}$$

The input of the function contains a list of RSSI, which is used to calculate the parameter Standard Deviation and mean. Then returns value is also a list contain the value after using the Gaussian function. The function *MeanRSSI* returns the statistical mean from a sample set.

## 7.7 - Empirical Distance Parameter Calculation

To define a channels , whether it comes from the Line of Sight and Non Line of sight this work involves estimation of the probability density function (PDF) as many authors have explained it. To calculate PDF, it needs to know one distance that is not exact value rather than empirical value by fitting one linear equation. If we consider dB as the measuring unit, then, the relation between RSSI and distance is linear function.

$$R = kd + S$$

This class estimates the parameter of the linear equation according to the principal of the least square method. The parameter estimation is done in calibration phase.

## 7.8 - Channel Identification

Channel Identification involves defining the Antenna position whether it comes from the Line of Sight or Non Line of sight . It is done by calculating probability density function for each sensor by estimating the empirical distance for it. The Hypothesis for LOS and NLOS channel are Ho and H1 , those are calculate by using the following formula:

$$P(d_{ij}) = \begin{cases} \frac{1}{\sqrt{2\pi}\delta_L}\exp[-\frac{(d_{ij}-W_L)^2}{2\delta_L}, & H = H_0 \\ \frac{1}{\sqrt{2\pi}\delta_N}\exp[-\frac{(d_{ij}-W_N)^2}{2\delta_N}, & H = H_0 \end{cases}$$

and

$$W_L = a^{\frac{PL(d_0) - X_{\delta_L} - RSSI(d)}{b\beta_L}}, \quad H = H_0$$

$$W_N = a^{\frac{PL(d_0) - X_{\delta_N} - RSSI(d)}{b\beta_N}}, \quad H = H_1$$

## 7.9 - Distance Calculation

Distance calculation is done by Log Normal Path Loss Model . When a channel is identified the associated path loss exponential is used with related standard deviation to calculate the Distance .

$$RSSI(d) = PL(d_0) - 10n\log\left(\frac{d}{d_0}\right) + X_\sigma$$

## 7.10 - Calibration Phase

Calibration phase involves to the system calibration in terms of the system parameter estimation. Calibration is done by two phase one is Line of Sight, another is Non Line of Sight. For each phase a sample number of RSSI is taken in account , optimization phase is also involved here . After the Optimization It calculate the path loss Exponential. for each channel then it needs to repeat the process in different known position to estimate it. Finally for each channel it calculate path loss exponential and standard deviation . During this period , for each known distance value , parameter of the Linear equation is also estimated .

# *Chapter 8:* Measurements and Results

## 8.1 – Introduction

In this chapter we have presented the results of our experiments according to the defined methodology in outdoor and indoor places. As it has been mentioned before for implementation we used concentrator V.1.0 as Master, Sensors and Moving Object which have been equipped by different firmware. Also according to goal of the project, 868 MHz is the spectrum of the communication and connection between sensors and master would be RS-485 and the connection between master and server is RS-232.

## 8.2 – Outdoor Experiment

For running the system in open area we choose a wide area in the middle of the university of Politecnico in Como where there were no obstacles and sensors can communicate to each other in line of sight situation. This test was just about the measuring the RSSI values in different distances between the sensor and moving object. In each distance by different value of the RSSI the path loss exponent and standard deviation were calculated.

Figure 53. Image of outdoor area for test

The results of this test01 are in below table:

| Location: | Center of University, Via Valleggio, Como | | | |
|---|---|---|---|---|
| Weather: | Mostly Cloudy | | Wind Speed: | 4 Km/h NNE |
| Humidity: | 58% | | Sound Average: | 56 dB |
| Pressure: | 1010.16 | | Date: | 30/10/2012 |
| Time: | 15:30 | | Average RSSI (1 m): | 132 |
| # | Distance (m) | RSSI | β | LOS or NLOS |
| 1 | 1.50 | 125 | 3.41 | LOS |
| 2 | 2.30 | 112 | 5.25 | LOS |
| 3 | 3.00 | 102 | 6.29 | LOS |
| 4 | 3.60 | 97 | 6.30 | LOS |
| 5 | 4.30 | 91 | 6.79 | LOS |
| 6 | 5.10 | 85 | 6.50 | LOS |
| 7 | 5.90 | 81 | 6.62 | LOS |
| 8 | 6.70 | 82 | 5.93 | LOS |
| 9 | 7.50 | 77 | 6.40 | LOS |
| 10 | 8.30 | 67 | 7.07 | LOS |
| 11 | 9.00 | 65 | 7.23 | LOS |
| 12 | 9.90 | 56 | 7.83 | LOS |
| 13 | 10.70 | 63 | 6.70 | LOS |
| 14 | 11.90 | 56 | 7.99 | LOS |
| 15 | 13.00 | 63 | 6.10 | LOS |
| 16 | 14.20 | 57 | 6.59 | LOS |
| 17 | 15.50 | 63 | 5.71 | LOS |
| 18 | 16.30 | 61 | 5.86 | LOS |
| 19 | 17.50 | 47 | 6.68 | LOS |
| 20 | 18.40 | 48 | 6.48 | LOS |
| 21 | 19.80 | 38 | 7.17 | LOS |
| 22 | 20.90 | 53 | 5.90 | LOS |
| 23 | 22.00 | 59 | 5.43 | LOS |
| 24 | 23.20 | 52 | 5.78 | LOS |
| 25 | 24.40 | 57 | 5.48 | LOS |
| 26 | 25.50 | 60 | 5.12 | LOS |
| 27 | 26.80 | 43 | 6.23 | LOS |
| 28 | 27.90 | 48 | 5.74 | LOS |
| 29 | 28.80 | 38 | 6.44 | LOS |

Table 10. Results of the test01outdoor test

The average of β for this experiment is 6.24. Two next chart show the comparison of RSSI-Distance and β-RSSI clearly. It is really obvious that by increasing the distance between the moving object and sensor, the RSSI decreases.

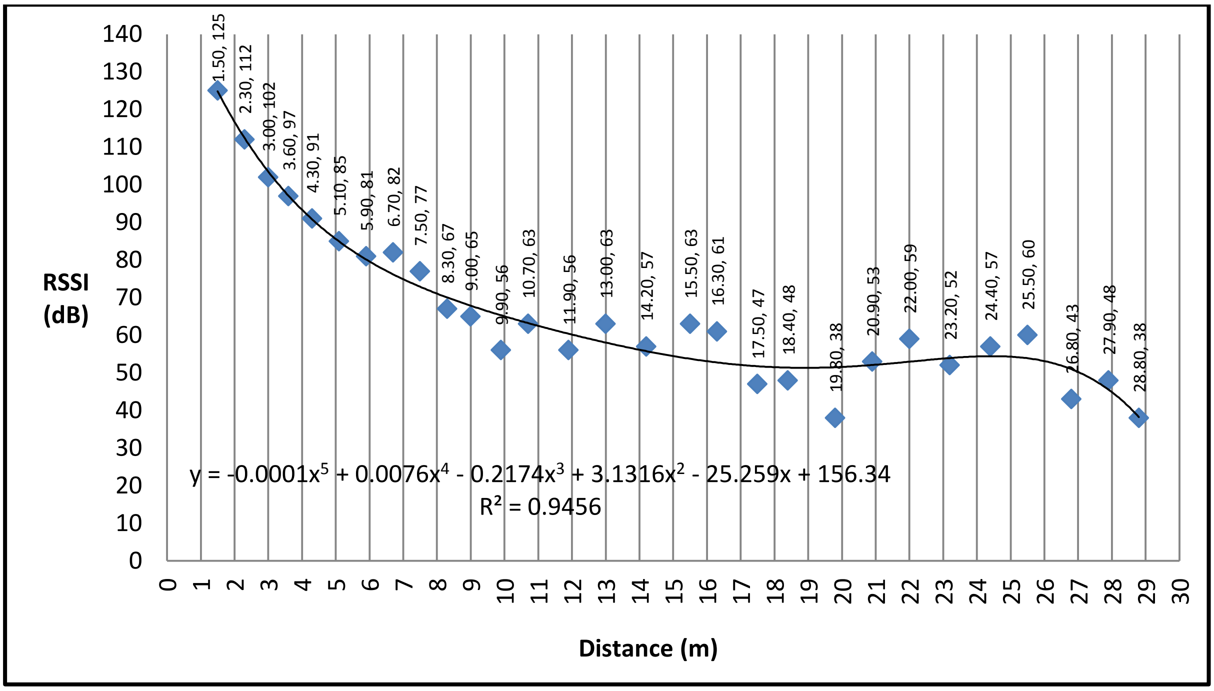


**Figure 54. Test01 Distance-RSSI Diagram**

In this diagram, the linear regression line of data and its formula and R-squared value have been shown.

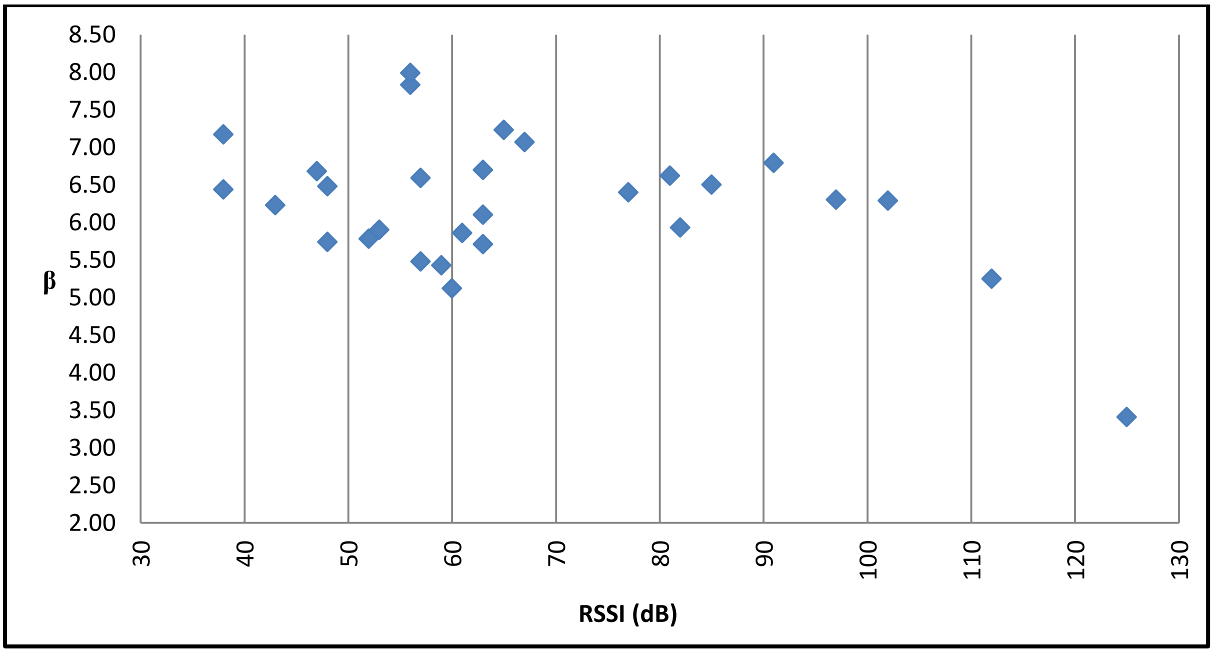


**Figure 55. Test01 RSSI-β Diagram**

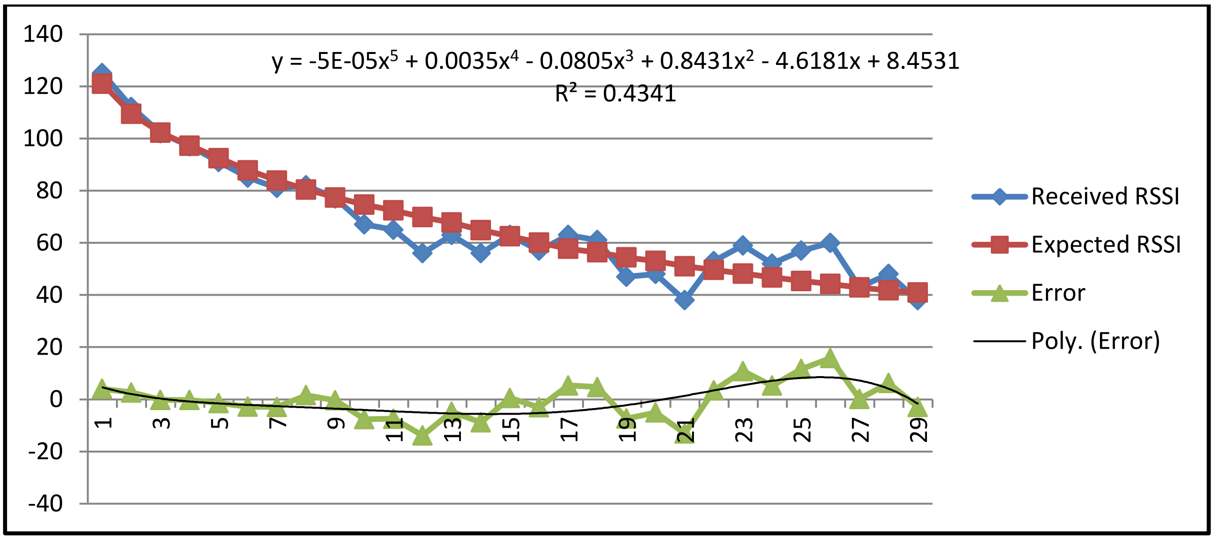


Figure 56. Comparison between Received RSSI and Expected RSSI and their Errors

The result of the another experiment is related to a field in Cantu which its results have been proposed in the below table:

| | | | | |
|---|---|---|---|---|
| Location: | Centro Sportivo Comunale, Via Longoni, Cantu | | | |
| Weather: | Partly Cloudy | | Wind Speed: | 3 Km/h NNE |
| Humidity: | 73% | | Sound Average: | 51 |
| Pressure: | 1023.03 | | Date: | 24/11/2012 |
| Time: | 13:00 | | Average RSSI (1 m): | 134 |
| **Moving Object in Height of 60 cm and Master in Height of 40 cm** | | | | |
| # | Distance(m) | RSSI | β | LOS or NLOS |
| 1 | 5.00 | 101 | 5.15 | LOS |
| 2 | 10.00 | 87 | 5.00 | LOS |
| 3 | 15.00 | 71 | 5.61 | LOS |
| 4 | 22.00 | 64 | 5.44 | LOS |
| 5 | 30.00 | 58 | 5.35 | LOS |
| 6 | 40.00 | 78 | 3.68 | LOS |
| 7 | 45.00 | 70 | 4.05 | LOS |
| 8 | 50.00 | 73 | 3.77 | LOS |
| 9 | 55.00 | 55 | 4.71 | LOS |
| 10 | 60.00 | 49 | 4.95 | LOS |
| 11 | 65.00 | 60 | 4.25 | LOS |
| 12 | 70.00 | 66 | 3.85 | LOS |
| 13 | 75.00 | 56 | 4.32 | LOS |
| 14 | 80.00 | 54 | 4.36 | LOS |
| 15 | 85.00 | 42 | 4.92 | LOS |
| 16 | 95.00 | 50 | 4.40 | LOS |
| 17 | 100.00 | 56 | 4.05 | LOS |
| 18 | 110.00 | 46 | 4.46 | LOS |

Table 11. Results of the test02 outdoor test

The average of β for this experiment is 4.57, and its comparing diagrams between Distance-RSSI and RSSI-β, are in next figures.

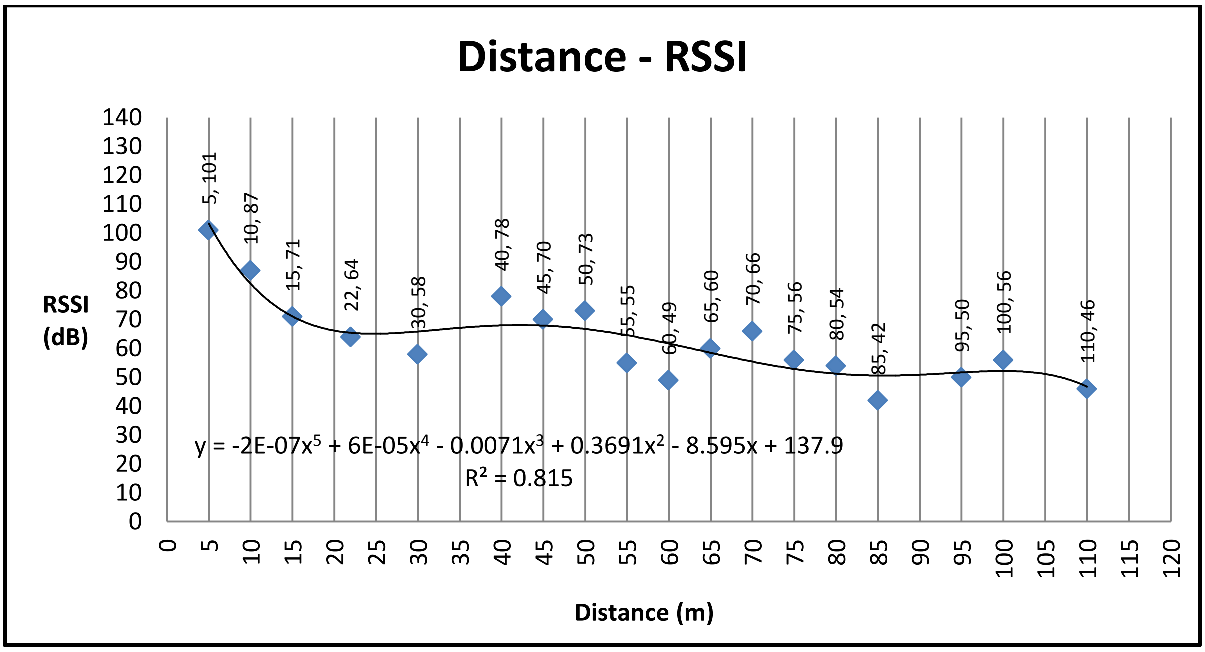


Figure 57. TheDistance-RSSI Diagram

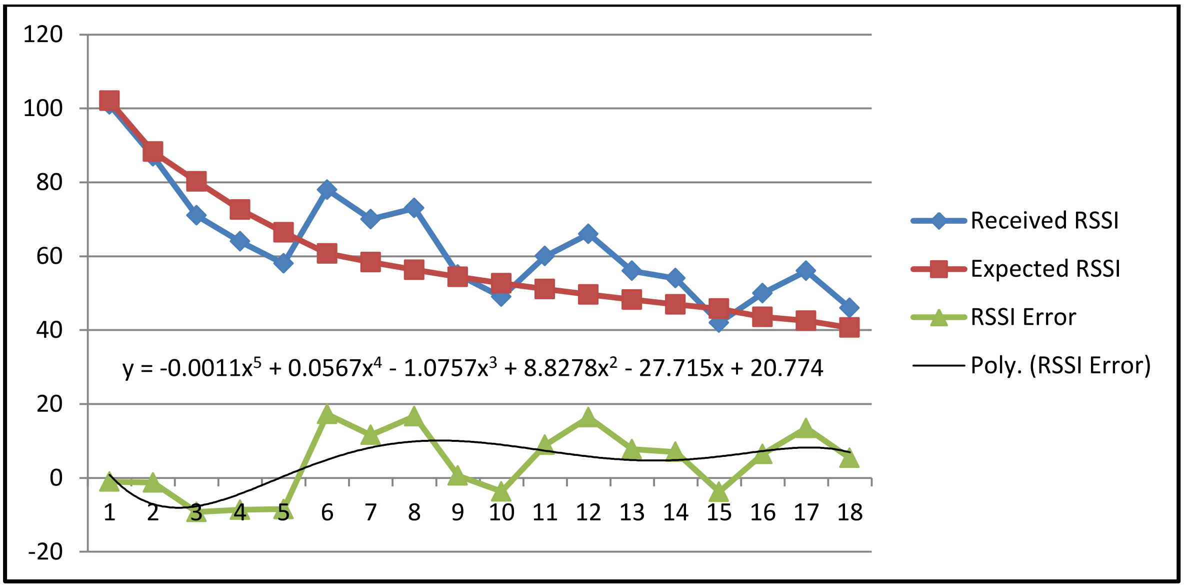


Figure 58. Camparison between Received RSSI and Expected RSSI with their Errors

## 8.2 – Indoor Experiment

In indoor area we choose one of the furnished room in second floor of the university building in Anzani street. In next figure the dimension of the room and the positions of the sensors have been shown. In this filed the effects of reflection, distraction, multi paths and etc. on the RSSI were clear. In this situation we had LOS and NLOS situation together.

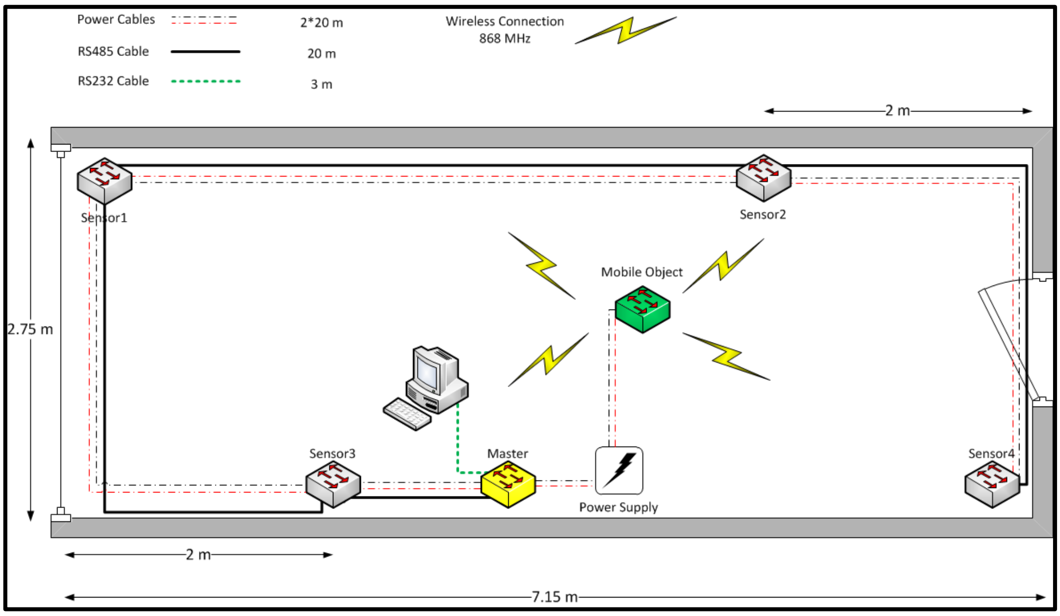


Figure 59. Dimension of indoor area for experiment

The first test in this section is about the localization inside a full furnished office room.

| | | | | |
|---|---|---|---|---|
| Location: | Via Anzani, 22, Second Floor, Room No. 2.15 | | | |
| Weather: | Indoor Area | | Wind Speed: | - |
| Humidity: | inside 35% - outside 75% | | Sound Average: | 49 |
| Pressure: | 1023.03 | | Date: | 1/12/2012 |
| Time: | 15:00 | | Average RSSI (1 m): | 138 |

| # | Distance (meter) | RSSI | β | LOS or NLOS |
|---|---|---|---|---|
| 1 | 1.50 | 123 | 8.52 | LOS |
| 2 | 2.30 | 132 | 1.66 | LOS |
| 3 | 2.40 | 118 | 5.26 | LOS |
| 4 | 2.80 | 133 | 1.12 | LOS |
| 5 | 3.40 | 130 | 1.51 | LOS |
| 6 | 3.50 | 122 | 2.94 | LOS |
| 7 | 3.60 | 133 | 0.90 | LOS |

| | | | | |
|---|---|---|---|---|
| 8 | 4.10 | 103 | 5.71 | LOS |
| 9 | 4.40 | 133 | 0.78 | LOS |
| 10 | 4.80 | 133 | 0.73 | LOS |
| 11 | 6.90 | 125 | 1.55 | LOS |
| Average: | - | - | **2.79** | - |

Table 12. The Results of the First Test in Inside Area

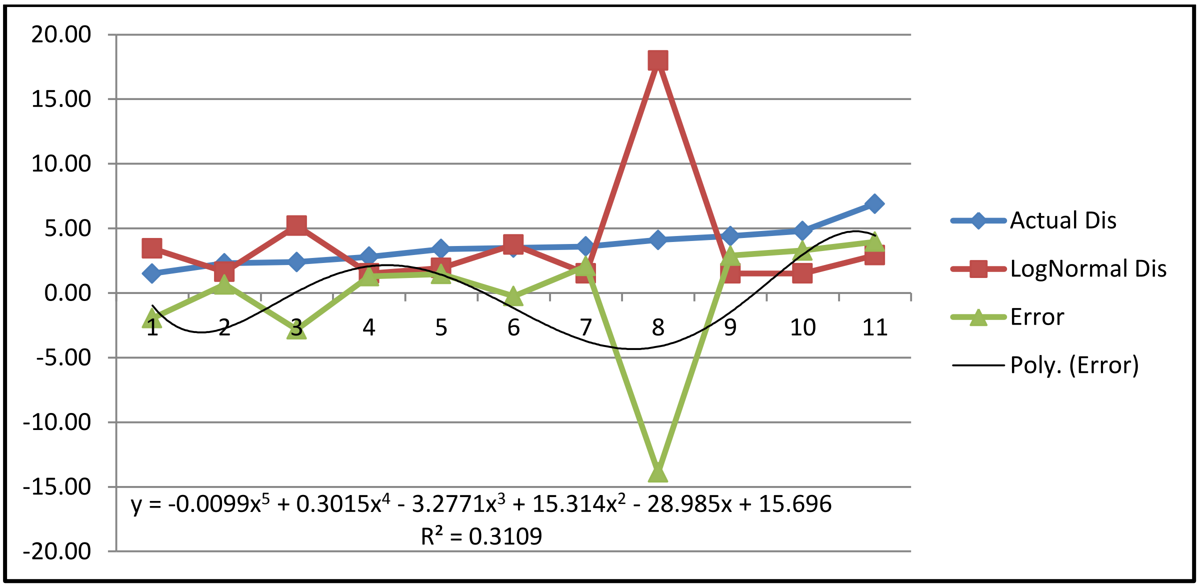


Figure 60. Comparison between Actual Distance and Lognormal Shadow Model

In the next test we measured distance and corresponding RSSI at the corridor inside the office building which is graphically presented in the below figures.

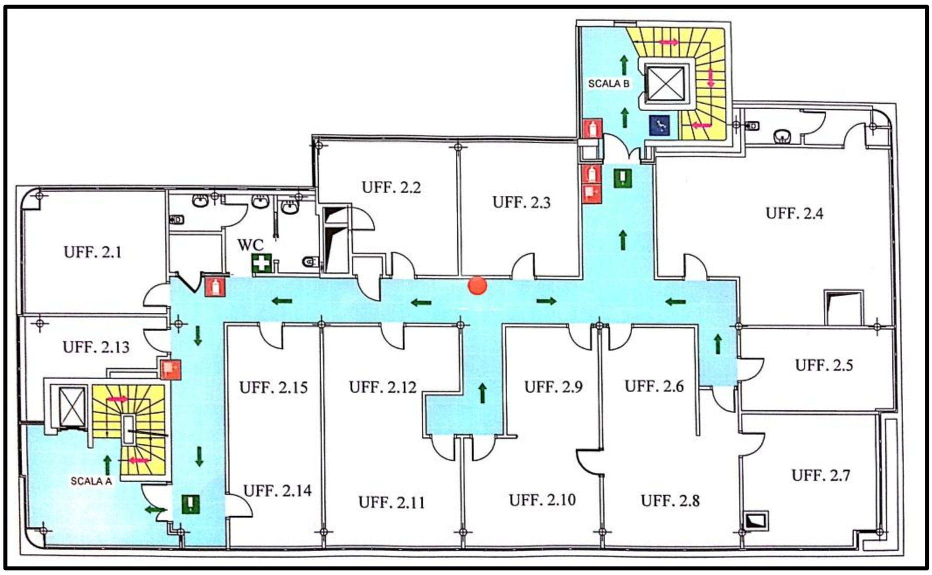


Figure 61. The Schematic of the Floor

| | | | |
|---|---|---|---|
| Location: | Via Anzani, 22, Second Floor | | |
| Weather: | Indoor Area | Wind Speed: | - |
| Humidity: | inside 35% - outside 75% | Sound Average: | 49 |
| Pressure: | 1023.03 | Date: | 1/12/2012 |
| Time: | 16:00 | Average RSSI (1 m): | 138 |

| # | Distance (meter) | RSSI | β | LOS or NLOS |
|---|---|---|---|---|
| 1 | 4.80 | 113 | 3.67 | LOS |
| 2 | 6.00 | 122 | 2.06 | LOS |
| 3 | 7.00 | 118 | 2.36 | LOS |
| 4 | 10.00 | 115 | 2.30 | LOS |
| 5 | 11.30 | 123 | 1.42 | LOS |
| 6 | 12.60 | 110 | 2.54 | LOS |
| 7 | 16.80 | 100 | 3.10 | LOS |
| Average: | - | - | **2.49** | - |

**Table 13.Results of the Inside Corridor Test**

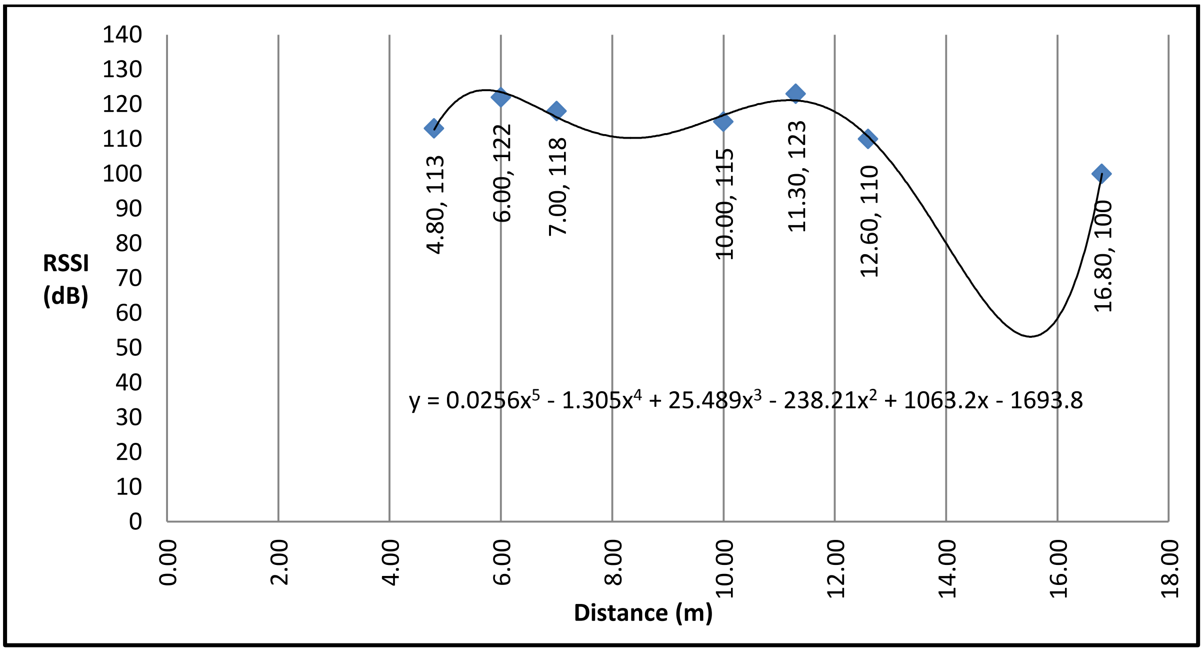


**Figure 62. The Relation between the RSSI and Distance**

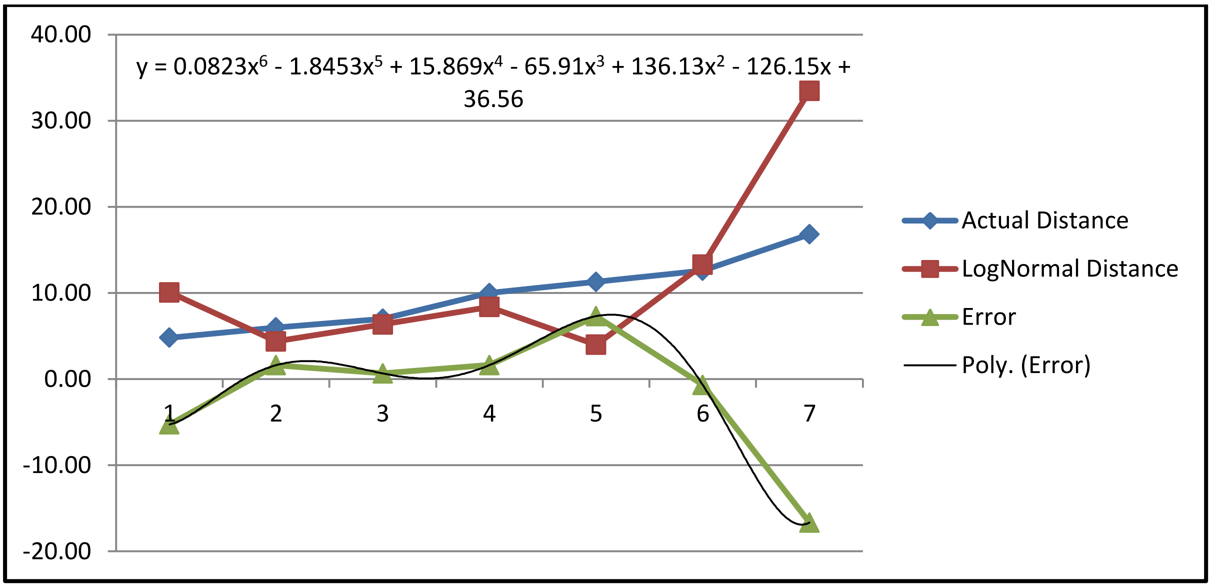


**Figure 63. Comparison between the Actual Distance, Lognormal Shadowing Distance and their Errors**

The last experiment has been done inside the office room considering the effect of the wall/partition, which can be present in the propagation path. The positions of the moving object have been shown in next figure.

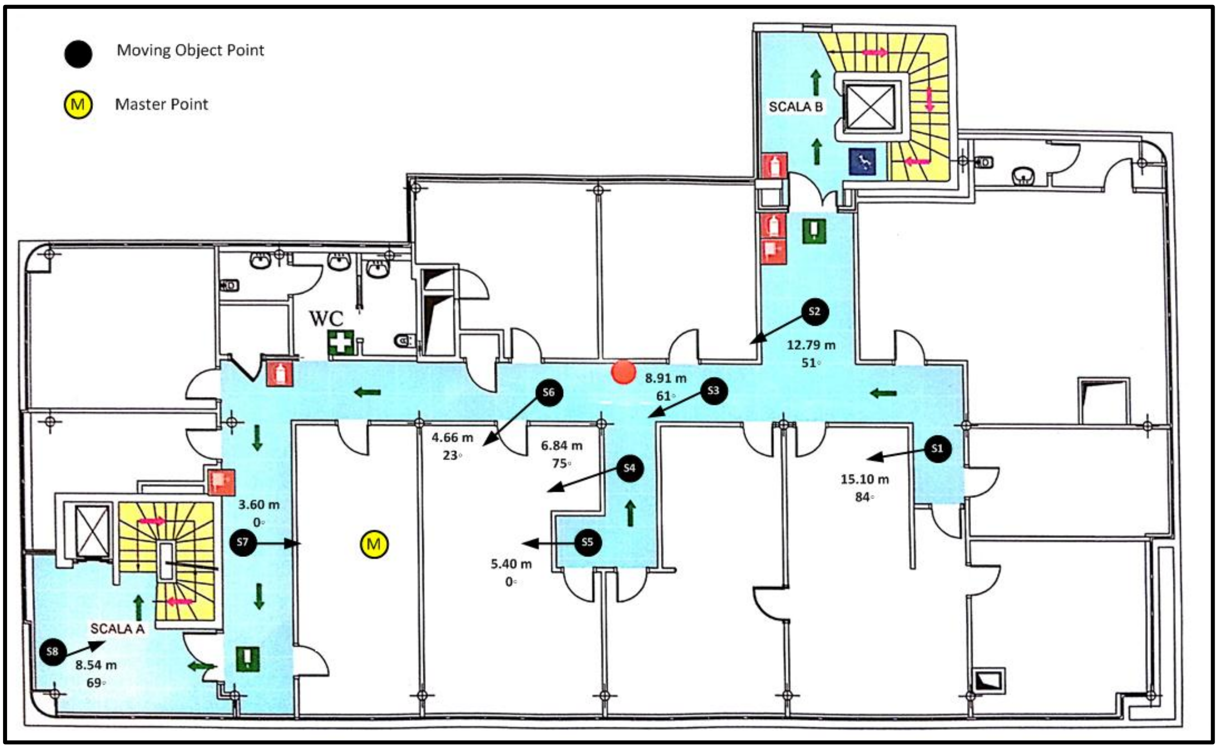


**Figure 64. Positions of the Moving object in Wall Effect Test**

| | | | | |
|---|---|---|---|---|
| Location: | Via Anzani, 22, Second Floor | | | |
| Weather: | Indoor Area | | Wind Speed: | - |
| Humidity: | inside 35% - outside 65% | | Sound Average: | 49 |
| Pressure: | 1023.03 | | Date: | 27/11/2012 |
| Time: | 19:00 | | Average RSSI (1 m): | 138 |
| | | | | |
| # | Distance (meter) | RSSI | β | LOS or NLOS |
| 1 | 15.10 | 99 | 3.31 | LOS |
| 2 | 12.79 | 102 | 3.25 | LOS |
| 3 | 8.91 | 111 | 2.84 | LOS |
| 4 | 6.84 | 104 | 4.07 | LOS |
| 5 | 5.40 | 121 | 2.32 | LOS |
| 6 | 4.66 | 112 | 3.89 | LOS |
| 7 | 3.60 | 120 | 3.23 | LOS |
| 8 | 8.54 | 96 | 4.51 | LOS |

**Table 14. The Results of the Wall Effect Test**

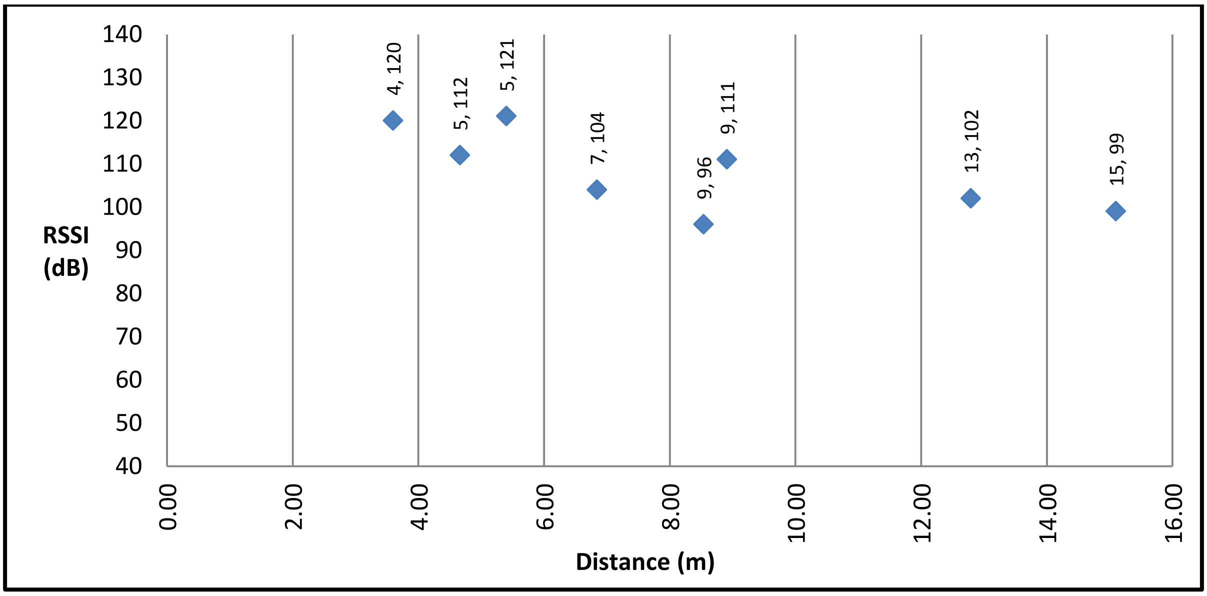


**Figure 65. The Relation between the RSSI and Distance**

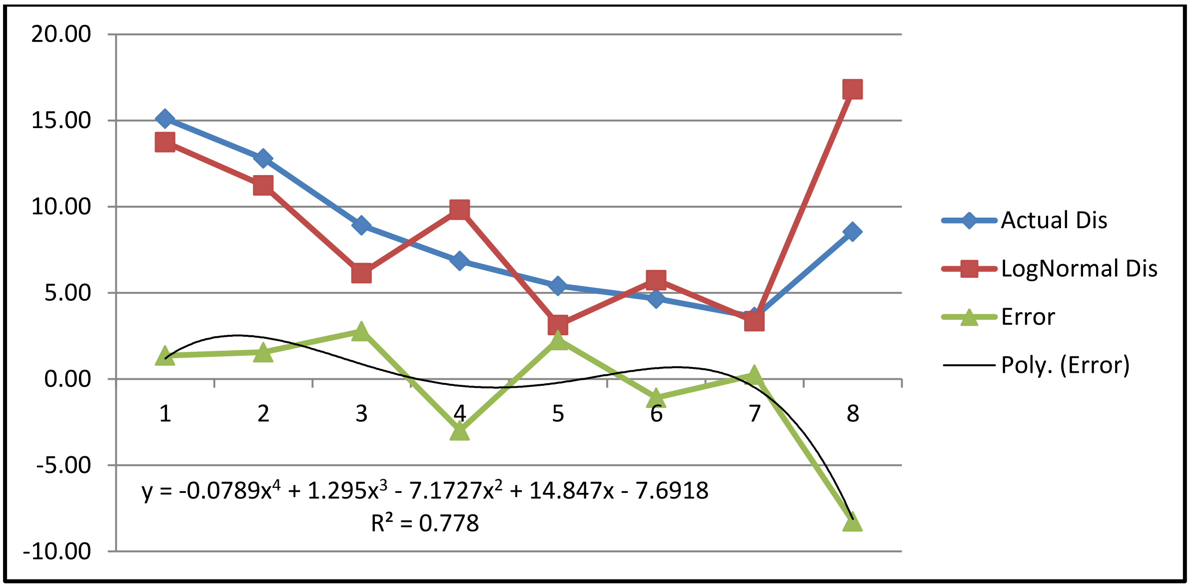


Figure 66. Comparison between Actual Distance and Lognormal Shadowing Distance and their Errors

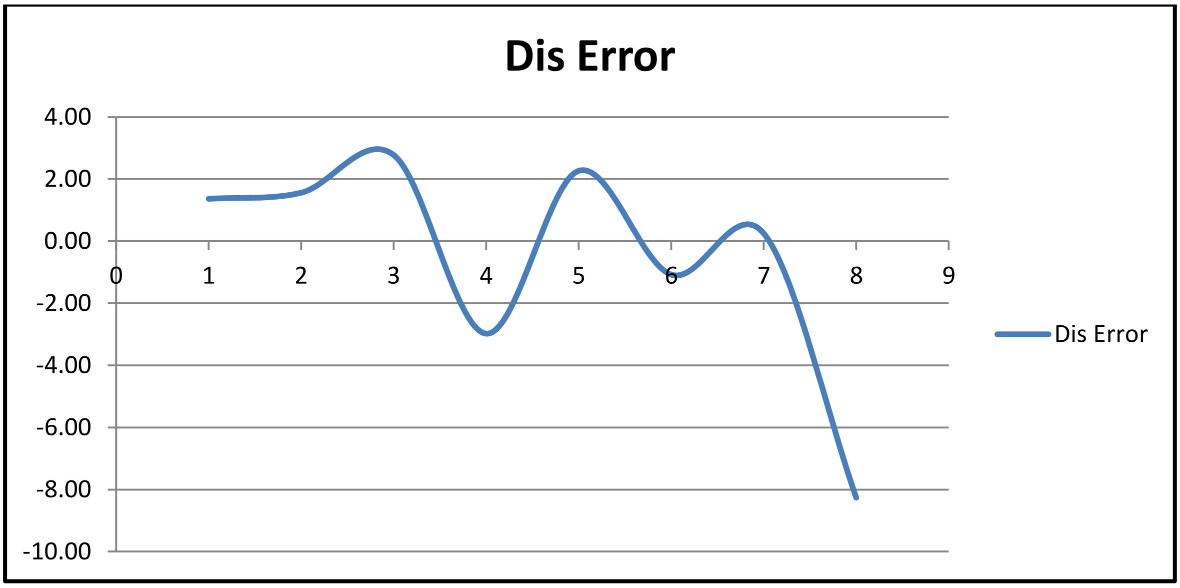


Figure 67. The Measurement Error between Actual Distance and Model Estimated Distance

# ***Chapter 9:*** System Application

## 9.1 – Why Localization?

Each day several people like firefighters and police officers lose their life during their jobs in dangerous situation, or patients and old people in case of vital condition will be in dangerous moment which on time and fast help would be useful for them. It is possible to decrease the numbers of victims in these situation by controlling them momently and help them in the emergency situation. For this reason at first we need to locate their position and after that recognize their health condition or in general monitor their situation. Also in some case may need to trace their routes for other steps.

Or it would be possible to have a system for routing people in big supermarkets to find their needs easily without wasting time for searching it. Or finding the position and tracing the mobile machines in the factories, hospitals and etc.

By equipping these several types of people or machines to the wireless sensor networks and transceivers which should not make trouble for them in terms of protection or searching and it would be possible to have their positions and also health condition for humans at the moment.

## 9.2 - In need of Accuracy

During the late years localization is one of the most critical and important point of several project in different field. In field of telecommunication all wireless sensor networks (WSN) projects which the position of sensors on them are not static, have localization process as one of the main step of their procedure. There are several methods, algorithms, protocols and infrastructures which have been introduced by several companies, researchers and engineers to reach to this goal and all of them try to improve the accuracy and the performance of the systems. The usage of some of these systems is crucial and because of that the localization system should be reliable and confident which in case of emergency can rely on it without any problem or error.

But reach to this level needs to consider all available methods, algorithms and solution to find the most accurate one and it is necessary to optimize these systems repeatedly to make their accuracy stable. There are several possibilities which decrease the accuracy of these systems or change the output to wrong one which in some cases are undefinable, like changing in humidity, covering the sensor networks by human bodies or intense light and etc. that according to methods which have been used in systems are difference and have to consider them.

## 9.3 – Finding and Critical Points

According to previous works and projects in this field, there are three critical functions for these kind of procedures:

- ***Locate the Position of Unknown Beacons (Person or Machine)***

In this function of project the location of mobile object will be find by using the RSSI information which are received by anchors.

- ***Monitor the Health condition of the people***

By receiving the Health Packets from the sensors according the health sensor in server and considering them, the situation of the person at the moment is distinctive.

- ***Tracing the route of the Mobile object (person or machine)***

By storing the position information of the mobile object in each time can show its route in 2D or 3D models.

Finding the accurate, simple and fast method for locating the position of the moving object according to RSSI information is one of the critical points of the systems. Considering the effect of changing the situation of the environment like humidity, light, temperature, or situation of door and windows (open/close) and covering the sensor networks by human body or any other obstacles would be important in this case.

Also Considering the undefined movement in anchors which lead us to the wrong result and considering the Reflection, Collision and interferences, or choosing the best protocol best place for anchors and optimizing it are other critical points.

In addition preparing, managing and controlling the health packets by using a specific middleware for wireless sensor networks and estimating the performance and accuracy of the system are rest of these critical points.

## 9.4 - Where these kind of systems would be useful?

We can mention several situations which a localization system would be useful to improve the performance of the whole systems or decrease the probability of the dangerous situation existence. For instance two important of these situations are as below.

### *9.4.1 – Hospitals and Clinics*

Knowing the positions of the patients especially who need more attention and control like old or disable people at the moment is really critical for the staff of a hospital. Also by adding the health sensor for controlling their medical situation this system will be really valuable for improving the operations of the hospitals or clinics staff.

Also it is possible to equipped the healthcare mobile assets in these areas which theirs positions are not stable and users of them should search to find them, decreasing the wasting time.


**Figure 68. Sample of an localization application in a hospital**

### *9.4.2 – Protection the Firefighters and Police Officers*

If the critical places like prisons, governmental buildings, big supermarkets and etc. have been equipped to a localization and tracing system, in emergency situations like fire or pursuing wanted person there, would be easier and safer. For instance imagine in a big fire in a big building which smoke decrease the eyesight of the firefighters by localizing and tracing the position the master can guide them there.

In other case, it is possible to equip prisoners in a prison to non-removable sensors to track them easily there.

## 9.5 – Actors

In this kind of projects, actors and theirs roles are really critical. We can divide the actors of them according to theirs responsibilities and usages. Some actors have key roles and they do order for running the system and adding new methods on it. Some other are just for preparing new ideas for the system and the last one have running role, in the system. We can add a role in project for inspecting the system and finding the positive and negative points of its operation and measuring the performance and efficiency of that.

- ***Governments***

The important actors of this system are governments who order for installing and implementing these kinds of systems for specific people in critical situation and also in specific places to protect them in emergency situation.

- ***Companies, Universities and Laboratories***

They are several companies, universities and laboratories in the world which work on new systems for finding the location and monitoring the specific persons according to new methods and inventions, generally if field of localization. Universities and laboratories have brain role in this field. The scientist and researchers preparing new methods develop, improve and introducing them to the companies who do the tools roles for making them ready as applicable way by using theirs engineers.

For instance we can mention *ekahau* company which run a real-time location system in one of the India hospital.

- ***People***

All these facilities and systems are used to control, protect, watch and monitor the people and their health condition and situation like patient, old and disable people. These kind of actor use the benefits of these system directly but for some other people like doctors and nurses these system help them to develop and improve their accuracy and decrease their errors and mistakes. These people can do the inspector role and finding the negative and positive points of system ideally and measure the efficiency and performance of the system.

# Conclusion and Future Works

## Conclusion

In this project we tried to find the position a mobile transceiver according to its RSSI signal based on 868 MHz frequency in indoor area. Some sensors which have been situated around an indoor environment, receive the RSSI packets from this transceiver sequentially and send them to the master. Master send these packets to the server and in server side the procedure of the localization is done.

These project has been done before in 2.4 GHz and the error rate was about between 2 to 3 meters according to the environment situation. Changing humidity and temperature, opening and closing the door and windows and also covering the sensor by human body and other obstacle had effects on the results. In our project according to the result the effects of the changing in environment were less which have been proposed in chapter 8.

In this project we had hardware and firmware implementation phase and also software implementation phase in server side. One of the most important part of this project was synchronizing between components of the systems like master, sensors and moving object. Because moving should send its signals through the 868 MHz frequency and sensors receive these signals sequentially and should send them to the master through the RS-485 cable by balancing between two interruption; high for wireless connection and low for wire connection. And finally all packets should send from the master to the server through the RS-232 cable.

In the procedure of the localization, in calibration phase, according to the measured data the system calculate path loss exponent value ($\beta$) and zero-mean Gaussian random variable ($X_\sigma$). The path loss exponent value in Log Normal model helps to have specific model for proposing the empirical data. We can find this model from the comparing the actual distance and estimated distance. By this comparing we can find the shape of error and choose a specific model like polynomial to cover the error. Also for considering the situation of the environment and its changes, the zero-mean Gaussian random variable, solve this problem.

According to existence many obstacles in indoor area and in some cases the unstable situation of environment according to opening or closing the door and windows and also the different situation of the two rooms which are near each other and signals path through both of them, the accuracy of the system would be different and in some case it is impossible to measuring it without adding more sensors or having the stable situation in the area. We took in account some of these situation but for sure there are some other causes that decrease the accuracy of the system.

## Future Works

Without presence of obstacle having sharp edges the lognormal model can estimated the distance inside an indoor environment. Diffraction due flat body and different reflection is estimated by higher degree of polynomial around degree equal to five. But the environment can have sharp edges obstacle and error due to diffraction can be predicted by using higher degree polynomial greater than five. The antenna used in this current project is directional. To optimization the current localization problem with multi directional antenna can have better result which may reduce the computational complexity of the higher degree of the polynomial.

# Appendixes

## Firmware

### *Master*

```
//*****************************************************************************
*************

// Interrupt Handler
//*****************************************************************************
*************

 int IntPos=0;

void Low_Int_Event (void);

void High_Int_Event (void);

#pragma code high_vector=0x08

void high_interrupt (void)

{

_asm GOTO High_Int_Event _endasm

}

#pragma code low_vector=0x18

void low_interrupt (void)

{

_asm GOTO Low_Int_Event _endasm

}

// Low Priority Interrupt Handler
****************************************************************

#pragma interruptlow Low_Int_Event

void Low_Int_Event (void)

{

LED_2=!LED_2;

        if(PIR1bits.RC1IF){

                //mem=1;
```

```
                buff1[f]=RCREG1;
                //if(buff1[f]==STX){TXREG2 = '$';IntPos = f; }
                f++;
                while(!TXSTA2bits.TRMT);
                TXREG2 = '$';
                PIR1bits.RC1IF=0;
                _ser =1;
        }
}
// High Priority Interrupt Handler
*************************************************************
#pragma interrupt High_Int_Event
void High_Int_Event (void){
        if(PIR1bits.CCP1IF){      //Timer comparator Interrupt
                counterSyncAck++;
                PIR1bits.CCP1IF=0;
        }
}

// inside the main we have to enable both interrupt and set master ID=0
ind=0x00000000;                 //      MASTER ID
INTCONbits.GIEH=1;                     // Hi pr. en.
INTCONbits.GIEL=1;                     // Lo pr. en.
//inside the main serial wireless and pairing Is done.
sensorcount=1;
        MACAW=0;
        _ControlFlag!=0;
        while(1){
```

```
                    LED_1=!LED_1;
                    randct++;
                    Delay10KTCYx(0x32);
                    Delay10KTCYx(0x32);
                    Delay10KTCYx(0x32);
Polingfunction ();
                     //LED_2=!LED_2; // *** only for the liveness
                    LED_1=!LED_1;
/*      RSSI   LOC  */
                    if(_ser ==1)
                    {
                              checkPacketSerial();
                              _ser =0;
                              INTCONbits.GIEL=1;
                    }
                    Delay10KTCYx(0x32);
                    SerailDataComAccess();
          }
void  checkPacketSerial()
{
          long rf ;
                    int i =0;
                    //MACAW=1;
                    INTCONbits.GIEL=0;        // %%
                    PIR1bits.RC1IF=0;         //!!!
                    INTCONbits.GIEL=1;
```

```
/*
writeByte1(0x0a);
writeByte1(0x0d);
writeByte1('C');
writeByte1('1');
writeByte1(0x0d);
writeByte1(0x0a);
*/

for(i=0;i<f;i++)
{
if(buff1[i]==STX)
        {
                if(buff1[i+3]==0xD0 && _ControlFlag!=1)
                {
                        //                1.1
ind2=((long)buff1[i+11]<<24)+((long)buff1[i+10]<<16)+((long)buff1[i+9]<<8)+(long)buff1[i+8];
                        ConnectedSensorId =ind2;
                        _ControlFlag=1;
                        f= 0;

                        /*
                        writeByte1(0x0a);
                        writeByte1(0x0d);
                        writeByte1('C');
                        writeByte1('1');
```

```
                    writeByte1('.');
                    writeByte1('1');
                    writeByte1(0x0d);
                    writeByte1(0x0a);
                    */
                    MACAW=3;
                }
                else if(buff1[i+3]==0xF0)
                {
                        //                    1.2
                    //data packet receive successfully
ind2=((long)buff1[i+11]<<24)+((long)buff1[i+10]<<16)+((long)buff1[i+9]<<8)+(long)buff1[i+8];
                    if(ConnectedSensorId==ind2)
                    {
                        // send sensor id and rssi value to PC
rf=((long)buff1[i+15]<<24)+((long)buff1[i+14]<<16)+((long)buff1[i+13]<<8)+(long)buff1[i+12];

    //answerP(STX,0xF0,ind,RF,el2,th1,th2,ctPos);
                        writeByte1(0x0a);
                        writeByte1(0x0d);
                        writeByte1('I');
                        writeByte1('D');
                        writeByte1('=');
                        itoa(ConnectedSensorId,str);
                      sendString1(str);
                        writeByte1('#');
                        writeByte1('R');
                        writeByte1('S');
```

```
                        writeByte1('S');
                        writeByte1('I');
                        writeByte1('=');
                        itoa(rf,str);
                  sendString1(str);
                        writeByte1(0x0d);
                        writeByte1(0x0a);
                        _DataReceiveFlag=1;

                        /*
                         writeByte1(0x0a);
                        writeByte1(0x0d);
                        writeByte1('C');
                        writeByte1('1');
                        writeByte1('.');
                        writeByte1('2');
                        writeByte1(0x0d);
                        writeByte1(0x0a);
                        */
                        f=0;
                         MACAW=4;
      }
            }
            else
            {
                  // useless data received
                  /*
```

```
                                        writeByte1(0x0a);
                                        writeByte1(0x0d);
                                        writeByte1('C');
                                        writeByte1('1');
                                        writeByte1('.');
                                        writeByte1('3');
                                        writeByte1(0x0d);
                                        writeByte1(0x0a);
                                        */
                                        f= 0;
                                        MACAW=5;
                                   }
                              }
               }
}
 void SerailDataComAccess()
{
          switch(MACAW)
          {
                    case 0:
                    /*
                    writeByte1(0x0a);
                    writeByte1(0x0d);
                    writeByte1('C');
                    writeByte1('0');
                    writeByte1(0x0d);
                    writeByte1(0x0a);
```

```
                    */
                    INTCONbits.GIEL=1;
// broadcast mesg to ask every sensor are they have anything to send
   //answerP(STX,0xA0,ind,0,el2,th1,th2,ctPos);
                    // token based broadcast msg
                            if(sensorcount>4)
                            sensorcount=1;
                            if(sensorcount==1)
                            {
                            ConnectedSensorId = sensorID1;
                             }
                      else if(sensorcount==2)
                            {ConnectedSensorId = sensorID2;
                             //MACAW=3;
                             }
                      else if(sensorcount==3)
                            {
                            ConnectedSensorId = sensorID3;
                            //MACAW=3;
                            }
                            else if(sensorcount==4)
                            {
                            ConnectedSensorId = sensorID4;
                            //MACAW=3;
                            }
                                            /*      writeByte1(0x0a);
                                                    writeByte1(0x0d);
```

```
writeByte1('<');
itoa(ConnectedSensorId,str);
sendString1(str);
writeByte1('>');
writeByte1(0x0a);
writeByte1(0x0d);*/
Delay1KTCYx(10);
MACAW=3;
break;

case 2:
Delay1KTCYx(36);          //3ms
_delayCount++;

if(_delayCount>11)
{_delayCount=0; /*sensorcount++;*/ MACAW=0; }
break;
case 3:
/*
writeByte1(0x0a);
writeByte1(0x0d);
writeByte1('C');
writeByte1('3');
writeByte1('I');
writeByte1('D');
writeByte1('=');
itoa(sensorcount,str);
```

```
                                    sendString1(str);
                        writeByte1('*');
                                    writeByte1('I');
                                    writeByte1('D');
                                    writeByte1('=');
                                    itoa(ConnectedSensorId,str);
                                    sendString1(str);
                                    writeByte1(0x0a);
                                    writeByte1(0x0d);
                                                */
                    sensorcount=sensorcount+1;
                // ask to specific sensor  and other sensor remain silence  ...
                    INTCONbits.GIEL=1;
                answerP(STX,0xA1,ConnectedSensorId,0,el2,th1,th2,ctPos);
                        _delayCount=0;
                        MACAW=2;
                break;
                    case 4:
                    /*
writeByte1(0x0a);
                    writeByte1(0x0d);
                    writeByte1('C');
                    writeByte1('4');
                    writeByte1(0x0d);
                    writeByte1(0x0a);
                    */
                // send  ACK
```

```
                    INTCONbits.GIEL=1;
                    answerP(STX,0xA2,ConnectedSensorId,0,el2,th1,th2,ctPos);
                    MACAW=5;
                    break;
                case 5:
                    /*
        writeByte1(0x0a);
                    writeByte1(0x0d);
                    writeByte1('C');
                    writeByte1('5');
                    writeByte1(0x0d);
                    writeByte1(0x0a);
                    */
                    INTCONbits.GIEL=1;
        _ControlFlag=0;
                    ConnectedSensorId=0x00000000;
                    /*sensorcount++;*/
                    MACAW=0;
                break;
        }
    }
```

### *Sensor*

The necessary function regarding sensor is given as below:

```
#pragma interrupt High_Int_Event
void High_Int_Event (void){
    unsigned char len;
```

```
unsigned char tv;
unsigned char *rdpt;
unsigned char w;
//If we got a IRQ1 request
if(INTCON3bits.INT1IF){ //WB Interrupt
        while(!TXSTA2bits.TRMT);
        TXREG2 = 0x0d;
        while(!TXSTA2bits.TRMT);
        TXREG2 = 0x0a;
        while(!TXSTA2bits.TRMT);
        TXREG2 = '+';
        TXREG2 = 'W';
        LED_2=0;
        if(!RXflag){
                //getsxSPI(buff7,WBPL);          //inlined code
                {
                        CSDAT=1;
                        CSCON=1;
                        len=WBPL;
                        rdpt=buff7;
                        while(len)                              // stay in loop until length = 0
                        {
                                CSDAT=0;
                                w=24;
                                while(--w);
                                //*rdptr++ = getcSPI1();   // read a single byte // inlined
code
                                tv = SSP1BUF;                   // Clear BF
```

```
                    PIR1bits.SSP1IF = 0;      // Clear interrupt flag
                    SSP1BUF = 0x00;           // initiate bus cycle
                    while(!PIR1bits.SSP1IF);          // wait until cycle complete
                    *rdpt++=SSP1BUF;                  // return  byte read
                    w=24;
                    while(--w);                                     //delay 2 uS
                    CSDAT=1;
                    len--;
                    w=24;
                    while(--w);                                     //delay 2 uS
                }
                CSDAT=1;
                CSCON=1;
            }
            if(buff7[3]!=0 && (buff7[12]==myLast || buff7[15]==0xFF)){
                mem1=1;
                //memBuffer(buff7,buff8,h); //inlined code
                for(i=0;i<PACKLEN2;i++){
                        buff8[h+i]=buff7[i];
                }
                temp6=0;
                temp7=0;
                h+=64;
            }
        _wireless =1;
        }
        else{
```

```
                    overrun++;
                }
                INTCON3bits.INT1IF=0;
                LED_2=1;
        }
        if(PIR1bits.CCP1IF){      //Timer comparator Interrupt
                temp2++;
                temp6++;
                temp7++;
                counterSyncAck++;
                PIR1bits.CCP1IF=0;
                if(counterSyncAck==T_MAX_5);
        }
}
        /*      RSSI LOC SYSTEM         */
        ind=0x0000000B;
        INTCONbits.GIEH=1;                      // Hi pr. en.
        INTCONbits.GIEL=1;                      // Lo pr. en.
void WirelessSignalReceive()
 {
                                                RF=(int)getRSSI();
                                                /* not implemented yet */
                                                //collision check will be there
                                                //send master for asking permission
[request to send ]
                                                //if receive clear to send then start sending
                                                //meanwhile master send other sensor to
keep storing it data
```

```
                                    //after receiving ack transmission finish .
                                    // if not after certain period of time it go to
step 1.
                                    // master gives every sensor for a amount of
time such as 2ms.(for 32 bit data)
                                    //
                                    ctPos=0x00;
                                    el2=0x00;
                                    th1=0x00;
                                    th2=0x00;
                                    _Transmit=1;

    writeByte1(0x0d);

    writeByte1(0x0a);

    writeByte1('*');

    writeByte1('R');

    writeByte1('S');

    writeByte1('S');

    writeByte1('I')
    itoa(RF,str);
    sendString1(st
    writeByte1('O');
    writeByte1('N');
    writeByte1('E');
    writeByte1(0x0d);
    writeByte1(0x0a);
```

```
                    h=0;
                    RXflag=0;
                    mem1=0;
                    for(v=0;v<PACKLEN2;v++){
                              buff7[v]=0;
                    }
                    INTCONbits.GIEH=0;
                    INTCON3bits.INT1IF=0;  // external interrupt did not occur
                    INTCONbits.GIEH=1;      // Enables all high-priority interrupts

                    INTCONbits.GIEL=1;                              // Lo pr. en.
                    for(s=0;s<PACKLEN2;s++)
                    {
                              buff9[s]=0;
                    }
 }
void  checkPacketSerial()
{
          unsigned char i=0;
                     unsigned char ret=0;
                              writeByte1(0x0a);
                              writeByte1(0x0d);
                              writeByte1('C');
                              writeByte1('1');
                              writeByte1(0x0d);
                              writeByte1(0x0a);
```

```
        for(i=0;i<2;i++)
        {
                                  Delay10KTCYx(0xFA);
        }
        for(i=0;i<f;i++)
        {
    if(buff1[i]==STX)
        {
            if(buff1[i+3]==0xA0) // master is free now
            {
                writeByte1(0x0a);
                writeByte1(0x0d);
                writeByte1('C');
                writeByte1('1');
                writeByte1('.');
                writeByte1('1');
                writeByte1(0x0d);
                writeByte1(0x0a);
            // master is free now
            // master asking for packet to all the sensor . first response
            // first win.....
      _ControlFlag =1;
             f= 0;
             MACAW=0;
            }
            else if(buff1[i+3]==0xA1)
```

```
{
writeByte1(0x0a);
        writeByte1(0x0d);
        writeByte1('C');
        writeByte1('1');
        writeByte1('.');
        writeByte1('2');
        writeByte1(0x0d);
        writeByte1(0x0a);
// check what  id the packet contains
// if yes ....communication active for some times.
// else keep silence
ind2=((long)buff1[i+11]<<24)+((long)buff1[i+10]<<16)+((long)buff1[i+9]<<8)+(long)buff1[i+8];
if(ind==ind2 /*&& _Transmit==1*/ )
{
// now it transmit
        MACAW=2;
}
f= 0;
}
else if(buff1[i+3]==0xA2)
{
writeByte1(0x0a);
        writeByte1(0x0d);
        writeByte1('C');
        writeByte1('1');
        writeByte1('.');
```

```
                    writeByte1('3');

                    writeByte1(0x0d);

                    writeByte1(0x0a);

                //data packet receive successfully

ind2=((long)buff1[i+11]<<24)+((long)buff1[i+10]<<16)+((long)buff1[i+9]<<8)+(long)buff1[i+8];

                if(ind==ind2)

                {

                f= 0;

                    _Transmit=0;

                }

                }

        }

        }

                            writeByte1(0x0a);

                            writeByte1(0x0d);

                            writeByte1('*');

                            writeByte1(0x0d);

                            writeByte1(0x0a);

                    //      MACAW=1; // serial data receive

                            INTCONbits.GIEL=0;      // %%

                            PIR1bits.RC1IF=0;       //!!!

                            INTCONbits.GIEL=1;

}

void SerailDataComAccess()

{

//static unsigned char _Transmit=0;

//static unsigned char _ControlFlag=3;
```

```
//static unsigned char _ResendFlag=0;

//static unsigned char _ResendTries=0;

//RSSI_LOC_SENSORID

switch(MACAW)

{

        //Setting the GIEH bit (INTCON<7>) enables all

        //interrupts that have the priority bit set (high priority).

        //Setting the GIEL bit (INTCON<6>) enables all

        //interrupts that have the priority bit cleared (low priority).

 //INTCONbits.GIEH=1;                // 1 high priority Enable and 0 high priority Disable .

//INTCONbits.GIEL=1;                 // 1 Low priority Enable and 0 Low priority Disable

// Sensor to Master _ControlFlag        0 means not ask permission to master with sensor . 1
means   ask permission to master 3 means unassign

 //Sensor to Master ...              ask for a permission type 0xA0 [0xA0 request to send ]

 // [boradcast to all sensor           [0xA1 clear to send] ask for if any sensor needs active
connection

case 0:

        writeByte1(0x0a);

                        writeByte1(0x0d);

                        writeByte1('C');

                        writeByte1('0');

                        writeByte1(0x0d);

                        writeByte1(0x0a);

        // ask for a permission.

   // if something wants to send !!

        if(_ControlFlag==1 && _Transmit==1)

        {

                        MACAW=3;
```

```
            }
            else
            {
            _ResendFlag++;
      Delay1KTCYx(12);
             if( _ResendFlag > 10)
             {
                            _ResendFlag=0;
                            //_Transmit=1;
                          MACAW=5;
             }
            }
           break;
case 1:
                                Delay1KTCYx(36);      //3ms
                                   _delayCount++;
                                   if(_delayCount>11)
                                   {_delayCount=0;MACAW=0;  }
                                   else if(_delayCount==7)
                                   {
                                   if(_Transmit==1)
                                   {
                                   MACAW=4;
                                   }
                                   }
           break;
case 2:
```

```
                    writeByte1(0x0a);
                    writeByte1(0x0d);
                    writeByte1('C');
                    writeByte1('2');
                    writeByte1(0x0d);
                    writeByte1(0x0a);
                    answerP(STX,0xF0,ind,RF,el2,th1,th2,ctPos);
                    _delayCount=0;
                        MACAW=1;
        break;
case 3:
                    writeByte1(0x0a);
                    writeByte1(0x0d);
                    writeByte1('C');
                    writeByte1('3');
                    writeByte1(0x0d);
                    writeByte1(0x0a);
                    // send request to  master 0xD0
                    answerP(STX,0xD0,ind,128,el2,th1,th2,ctPos);
                        _delayCount=0;
                            MACAW=6;
   break;
case 6:
                            Delay1KTCYx(36);        //3ms
                            _delayCount++;
                            if(_delayCount>11)
                            {_delayCount=0;MACAW=0;   }
```

```
break;
case 4:
                    writeByte1(0x0a);
                    writeByte1(0x0d);
                    writeByte1('C');
                    writeByte1('4');
                    writeByte1(0x0d);
                    writeByte1(0x0a);
                    writeByte1(0x0a);
                    writeByte1(0x0d);
                    writeByte1('t');
                    writeByte1('s');
                    writeByte1('m');
                    writeByte1('t');
                    writeByte1(0x0d);
                    writeByte1(0x0a);
        // Resend data to  master
         answerP(STX,0xF0,ind,RF,el2,th1,th2,ctPos);
         Delay1KTCYx(12);      //1ms
  MACAW=5;
  break;
case 5:
                    writeByte1(0x0a);
                    writeByte1(0x0d);
                    writeByte1('C');
                    writeByte1('5');
                    writeByte1(0x0d);
```

```
                    writeByte1(0x0a);
                    // Clear all and restart process
                INTCONbits.GIEH=1;                    // 1 high priority Enable
                    INTCONbits.GIEL=1;        // 1 Low priority Enable
                    MACAW=0;
                    _Transmit==0;
    break;
default:
        break;
}
}
```

## Server

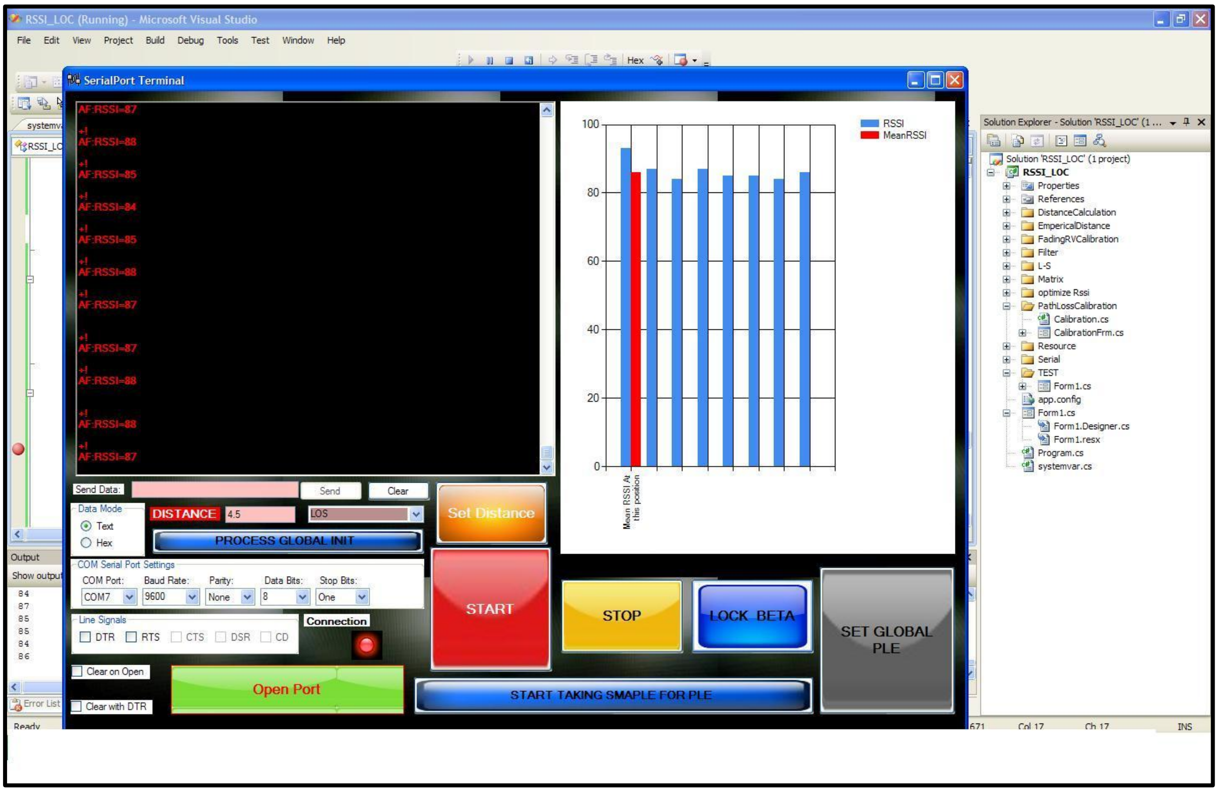


**Figure 69. Localization Server**

Least Square Library:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;
using RSSI_LOC.Matrix;
using RSSI_LOC.PathLossCalibration;


namespace RSSI_LOC.L_S
{

    public class L_SOb
    {


        private List<int> ExperRSSI = new List<int>();
        private List<double> Exprmntdistance = new List<double>();


        ArrayList data = new ArrayList();
        Matrix A1, A2, A3, C;
        Matrix V1 = new Matrix(4, 1);
        Matrix Parameter = new Matrix(2, 1);
        public L_SOb()
        {
             A1= new Matrix(4, 2);
             A2 = new Matrix(2, 4);
             A3 = new Matrix(4, 4);
        }
        public L_SOb(List<int> ExperRSSI, List<double> Exprmntdistance)
        {

            this.ExperRSSI = ExperRSSI;
            this.Exprmntdistance = Exprmntdistance;
        }



        public void setVector(int Rssi, int sensorID)
        {
            V1[sensorID, 0] = Convert.ToDouble( Rssi);

        }
        public void calibinit(double Distance , int sensorID)
        {

            A1[sensorID, 0] = Distance;
            A1[sensorID, 1] = 1;
            A2 = Matrix.Transpose(A1);

            A3 = A2 * A1;
            C = Matrix.Inverse(A3);

            Parameter = C * (A2 * V1);

       }
```

```
public double getPramameter(int i)
{
    // 0 k 1 s
double x = Convert.ToDouble(Parameter[i,0]);
return x;
}



public  List<double> Params()
{
    Calibration objectCal = new Calibration();
    int i = 0;
    double Error= 0.0;
    int vertcln = 0;
    int matrixrow =0, matrixcolm=0;
    // List<double> errorList = new List<double>();
    // vertcln = errorList.Count;
    Matrix Vec1 = new Matrix(this.ExperRSSI.Count, 1);

    Matrix Vec2 = new Matrix(4, 1);
    //Matrix Vec2 = new Matrix(5, 1);

    Matrix AA = new Matrix(this.ExperRSSI.Count, 4);

    //Matrix AA = new Matrix(this.ExperRSSI.Count, 5);


   // calibinit(0.0 , 1);

    foreach (int RSSI in this.ExperRSSI)
    {

                double distance = objectCal.ReturnDistance(RSSI);
                Error = /*Math.Abs*/(distance - Exprmntdistance[i]);

              //Vec1[i, 0] = Convert.ToDouble(/*Math.Log10*/(Error));
                Vec1[i, 0] = Convert.ToDouble(Exprmntdistance[i]);


                AA[i, 0] = Math.Pow(Convert.ToDouble(distance), 3);
                AA[i, 1] = Math.Pow(Convert.ToDouble(distance), 2);
                AA[i, 2] = Math.Pow(Convert.ToDouble(distance), 1);
              AA[i, 3] = 1;

             i++;
    }

    Vec2 = ((Matrix.Inverse((Matrix.Transpose(AA)) *
AA))*(Matrix.Transpose(AA))*Vec1);

   List<double> paramList = new List<double>();
   paramList.Add(Vec2[0,0]);
   paramList.Add(Vec2[1,0]);
   paramList.Add(Vec2[2,0]);
   paramList.Add(Vec2[3,0]);

   return paramList;

}
```

```
    }
}
```

Filter.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections.Generic;
using System.Collections;

namespace RSSI_LOC.Filter
{
    class Filters
    {
        public Filters()
        {
        }

        public ArrayList GaussModel(ArrayList target)
        {
            //ArrayList target;
            double mean=0.0;
            double  std = 0.0;
            double intermval=0.0;
            int i = 0;
            if (target.Count > 0)
            {
                for (i = 0; i < target.Count; i++)
                {
                    intermval = Convert.ToDouble(target[i]);
                   // double.TryParse(, out intermval);
                    mean += intermval;
                }
                mean = mean / target.Count;

              for (  i = 0; i < target.Count; i++)
             {
                 intermval = Convert.ToDouble(Convert.ToDouble(target[i]) - mean);
                  //double.TryParse(Math.Abs(double.TryParse(target[i]) - mean), out
intermval );
                  std += intermval;
              }

                std = (Math.Sqrt(std)) / target.Count;

                 for (  i = 0; i < target.Count; i++)
target[i] = Gaussianfilter(Convert.ToSingle(mean), Convert.ToSingle(std),
Convert.ToSingle(target[i]));
            }
            return target;
     }
        public float Gaussianfilter(float mean, float std, float Rs)
        {
            double y = (double)(1 / (std * Math.Sqrt(2 * Math.PI))) *
Math.Exp((double)(-1 / (2 * std * std)) * Math.Pow((Rs - mean), 2));
            return (Convert.ToSingle(y));
         }
        public void Linearfilter()
        {
```

```
        }

        public int  MeanRSSI(ArrayList target)
        {
            ArrayList ReturnList = new ArrayList();
            double x=0.0;
            int i = 0;
            for ( i = 0; i < target.Count; i++)
            {
                x = Convert.ToDouble(target[i]);
                if(x>0.6 || x<1)
                {
                    ReturnList.Add(x);
                }
            }
            for (  i = 0; i < ReturnList.Count; i++)
            {
                x += Convert.ToDouble(ReturnList[i]);
            }
            int y =Convert.ToInt32 (x / ReturnList.Count);
            return y;
        }
    }
}
```

EmpericalDistance.cs

```
Distance.cs

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace RSSI_LOC.DistanceCalculation
{

    class DistanceCalculation
    {
        double _avgPower = 0.0;
        double _randomGaussinaVar = 0.0;
        int _RSSI   = 0;
        int _coffB = 10;
        int _coffA = 10;
        double Beta = 0.0;
        double _distance = 0.0;
        double _STD = 0.0;

    public void setBeta(double beta)
        {
            Beta = beta;
        }
    public void SetAvgPower(double power)
    {
        _avgPower = power;
    }
    public void SetRSSI(int RSSISet)
    {
        _RSSI = RSSISet;
    }
    public void SetSTD(double std)
```

```
    {
        _STD = std;
    }
    public double  CalculatePDF(double Distance, int coffA, int coffB)
    {

         _coffB =coffB ;
         _coffA = coffA;
         _distance = Distance;
         double x =CalculateWCoff();
        double returnval = CalculateTotalPDF(x);
        return returnval;
    }
    public double CalculateTotalPDF(double W)
    {

    double exp =  (double )(-1)*(Math.Pow((_distance-W), 2)/(2*_STD));
    double PDF =Gaussianfilter( exp );
    return PDF;
    }
    public double CalculateWCoff()
    {
        double x = 0.0;
        double mean = 0.0;
        double std = 0.0;

        double expval = (_avgPower - _randomGaussinaVar - _RSSI) / (_coffB * Beta);
        x=Math.Pow(_coffA, expval);

        return x;

    }
    public float Gaussianfilter(double exp)
    {
        double y = (double)(1 / (_STD * Math.Sqrt(2 * Math.PI))) * Math.Exp(exp);
        return (Convert.ToSingle(y));
    }
    }
}
```

Calibration.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections.Generic;
using System.Collections;
using System.Windows.Forms;

namespace RSSI_LOC.PathLossCalibration
{
    class Calibration
    {

     private double AveragePathLoss; // PLD bar
     private double MasterSensorKnowndistance;

     public Calibration()
     {
         AveragePathLoss =Convert.ToDouble(systemvar.AvgPower);
```

```
        }
        public void setAvgPower(double GivenAvgPowerLoss)
           {
               //AveragePathLoss = GivenAvgPowerLoss;
           }
        public double  PathLosscalculate(double distance ,int RSSI)
        {
            // distance must be in meter
            MasterSensorKnowndistance = distance;
           double x = calculateBeta(RSSI);
           return x;
        }
        private double calculateBeta(int RSSI)
        {
            double ECMBeta = 0.0;

            try
            {
                double logval = Math.Log10(MasterSensorKnowndistance);
                double RSSI_PLD = Convert.ToDouble(AveragePathLoss -
Convert.ToDouble(RSSI));
                 ECMBeta = Convert.ToDouble(RSSI_PLD / (10 * logval));
                systemvar.PleRssi.Add(ECMBeta);

                //MessageBox.Show("calling beta" + ECMBeta.ToString());
            }
            catch(Exception EX)
            {
                MessageBox.Show("Alert : There Is Exception"+EX.Message);
                Console.Beep(1048, 500);
            }
            return ECMBeta;
        }
        public double getPathLossExponent()
        {
            double x = 0.0;
            if (systemvar.PleRssi.Count > 0)
            {

                for (int i = 0; i < systemvar.PleRssi.Count; i++)
                {
                    x += Convert.ToDouble(systemvar.PleRssi[i]);
                }
                if (systemvar.PleRssi.Count > 0)
                {
                    x = (x / systemvar.PleRssi.Count);
                }
                else
                {
                    x = 0.0;
                }

            }
            else
            {

                MessageBox.Show("no data is available ");
            }

            return x;
        }
```

```
     public double ReturnDistance(int RSSI)
     {
         double x = Convert.ToDouble((systemvar.AvgPower - RSSI));
         double y = (10 * (systemvar.BetaLos));
         double z = x / y;
         //double  pw=Convert.ToDouble((39- RSSI)/(10* (Convert.ToInt32(
systemvar.BetaLos))));

         double d = Math.Pow(10.00, z);
         return d;
     }
      public void getDistance(int RSSI)
      {
          double x = Convert.ToDouble((systemvar.AvgPower - RSSI));
          double y = (10 * (systemvar.BetaLos));
          double z = x / y;
       //double  pw=Convert.ToDouble((39- RSSI)/(10* (Convert.ToInt32(
systemvar.BetaLos))));

          double d = Math.Pow(10.00, z);
        MessageBox.Show(d.ToString());
      }
    }
}
```

frmTerminalCal:

```
public frmTerminal()
 {

     settings.Reload();
     InitializeComponent();
     InitializeControlValues();
     EnableControls();
     comport.DataReceived += new SerialDataReceivedEventHandler(port_DataReceived);
        comport.PinChanged += new SerialPinChangedEventHandler(comport_PinChanged);
     //this.chart1.Series.Clear();
     _calibrationLibObj = new Calibration();

     ClearTerminal();

     _calibrationLibObj.setAvgPower(Convert.ToDouble(systemvar.AvgPower));
     _globalStoreMeanRSSI.Clear();
     cmbChannel.SelectedItem = cmbChannel.Items[0].ToString();

 }
private string ByteToASCIIString(byte[] data)
    {
        StringBuilder sb = new StringBuilder(data.Length * 3);
        sb.Append(System.Text.Encoding.ASCII.GetString(data));

        if (ReceiveMsgBuffer.Count < 50 && _smaplstar == true)
        {
          string x = sb.ToString();

          if (x.Contains("RSSI") || x.Contains("ID"))
                ReceiveMsgBuffer.Add(x);
        }

        return sb.ToString().ToUpper();
    }
```

```
private void port_DataReceived(object sender, SerialDataReceivedEventArgs e)
    {

        int bytes = comport.BytesToRead;
        byte[] buffer = new byte[bytes];
        comport.Read(buffer, 0, bytes);
```

function for receive:

```
{
  _smaplstar = false;
            bool _found = false;
            List<string> buffer = new List<string>();
            char[] copyarray = new char[20];
            int i = 0;
            int found = 0;
            int k = 0;
            if (ReceiveMsgBuffer.Count > 0)
            {
                foreach (object y in ReceiveMsgBuffer)
                {
                    string x = y.ToString();
                    if (x.Contains('\n'))
                    {

                        found = x.IndexOf('\n');
                        found += 2;
                        if (!_found) goto foundnewLine;
                        else goto Endline;


                    foundnewLine:
                        //if (found == 2 && _found == false)
                        //   goto PacketstartatBegaining;

                        //  PacketstartatBegaining:
                        if (found < x.Length && _found == false)
                        {
                            bool endoccurance = false;
                            while (found < x.Length)
                            {
                            initagain:
                                if (x[found] == '\n' && endoccurance)
                                {
                                    _found = true;
                                    i = 0;

                                    if (found + 2 < x.Length)
                                    {
                                        found += 2;
                                        endoccurance = false;
                                        goto initagain;
                                    }
                                    else goto jumptonewobj;
                                    //break;
                                }
                                else if (x[found] == '\n' && !endoccurance)
                                {
                                    string st = new string(copyarray);
                                    st = st.Trim('\0');
                                    k = 0; i = 0;
```

```
                                    while (k < copyarray.Length) { copyarray[k] =
'\0'; k++; }

                                    buffer.Add(copyarray.ToString());
                                    endoccurance = true;
                                    if (found + 1 >= x.Length)
                                        goto jumptonewobj;

                                }

                                else
                                    copyarray[i] = x[found];

                                found++;
                                i++;

                            }

                            /* if (_found)
                             goto jumptonewobj;
                        }

                    Endline:
                        if (_found)
                            _found = false;
                        int j = 0;
                        while (j < found - 2)
                        {


                            if (x[j] == '\r')
                            {
                                string st = new string(copyarray);
                                st = st.Trim('\0');
                                k = 0;
                                while (k < copyarray.Length) { copyarray[k] = '\0';
k++; }

                                buffer.Add(st);
                                i = 0;
                                if ((found + 1) == '\n')
                                {
                                    found++;
                                    goto foundnewLine;
                                }
                            }
                            else
                                copyarray[i] = x[j];

                            j++;
                            i++;
                        }

                    }

                jumptonewobj:
                    k = 0;

                }

            }
```

```
                if (buffer.Count > 0)
                {

                    while (k < buffer.Count)
                    {
                        string x = buffer[k];

                        if (x.Contains("ID"))
                        {
                            string ID = x.Split('#')[0];
                            string RSSI = x.Split('#')[1];
                            if (!string.IsNullOrEmpty(ID) && !string.IsNullOrEmpty(ID))
                            {
                                if (Convert.ToInt32(ID.Split('=')[1]) == 10)
                                {
                                   RSSIMsgBuffer.Add(Convert.ToInt32(RSSI.Split('=')[1]));
                                }
                            }
                        }
                        k++;
                   }

                }

                if (RSSIMsgBuffer.Count>0)
                {
                            found = 0;
foreach (int y in RSSIMsgBuffer)
                            {
                                found += y;
                            }

found = (found / RSSIMsgBuffer.Count);
_globalStoreMeanRSSI.Add(found);
                            RSSIMsgBuffer.Clear();
                            RSSIMsgBuffer.Add(found);


                    // for test
                    MeanRssi = found;
               }
            }

        }
              // Emperical Distance Parametere Calibration
                EmpericalDistanceCal objEmp = new EmpericalDistanceCal();
                if (_RSSIParamBuffer.Count > 0 && _DistanceParamBuffer.Count>0)
                    objEmp.parameterCalibration(_RSSIParamBuffer, _DistanceParamBuffer);

                try{

                    if (systemvar.PleRssi.Count > 0)
                {
                    if (ChanelMode == "LOS")
                    {
                        systemvar.BetaLos = _calibrationLibObj.getPathLossExponent();
                        //MessageBox.Show(systemvar.BetaLos.ToString());
                        systemvar.PleRssi.Clear();

                        // Parameter Fx function

                        L_SOb ob = new L_SOb(_RSSIParamBuffer, _DistanceParamBuffer);
```

```
            systemvar.FxparamList = ob.Params();

        }
        else if (ChanelMode == "NLOS" )
        {
            systemvar.BetaNLos=_calibrationLibObj.getPathLossExponent();
            //MessageBox.Show(systemvar.BetaLos.ToString());
            systemvar.PleRssi.Clear();
        }
    }
        // test purposes ...
    if (systemvar.PleRssi.Count ==0)
        {
          ///  systemvar.BetaLos = 4.9;
             _calibrationLibObj.getDistance(Convert.ToInt32(MeanRssi));
            //MessageBox.Show(x.ToString());
        }
    }
     catch(Exception )
        {

       }
```

# Bibliography

[1] C. Wu, S. Chen, Y. Zhang, L. Cheng and H. Wu, "A RSSI-Based Probabilistic Distribution Localization Algorithm for Wireless Sensor Network," *IEEE,* 2011.

[2] Z. Qingxin, D. Qinglong, X. Guangyan and Q. Xiaoyan, "A RSSI Based Localization Algorithm for Multiple Mobile Robots," *IEEE,* 2010.

[3] T. Kazuya and I. Tetsushi, "Estimation Method of Attenuation Constant During Locakization in RSSI," *IEEE,* 2008.

[4] J. Zheng, W. Chengdong, H. Chu and P. Ji, "Localization algorithm based on RSSI and distance geometry constrain for wireless sensor network," *IEEE,* 2010.

[5] P. Barsocchi, S. Lenzi, S. Chessa and G. Giunta, "A Novel Approach to Indoor RSSI Localization by Automatic Calibration of the Wireless Propagation Model," *IEEE,* 2009.

[6] S.-P. Kuo, H.-J. Kuo and Y.-C. Tseng, "The Beacon Movement Detection Problem in Wireless Sensor Networks for Localization Applications," *IEEE,* vol. 8, no. 10, pp. 1326-1338, 2009.

[7] A. Savvides, C.-C. Han and M. B. Strivastava, "Dynamic FineGrained Localization in Ad-Hoc Network of Sensors," *ACM SIGMOBILE,* 2001.

[8] K.-F. Ssu, C.-H. Ou and H. C. Jiau, "Localization With Mobile Anchor Points in Wireless Sensor Networks," *IEEE,* vol. 54, no. 3, pp. 1187-1197, 2005.

[9] Y. Zhang, Z. Fang, R. Li and W. Hu, "The Design and Implementation of a RSSI-Based Localization System," in *Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09. 5th International Conference on*, 2009.

[10] Q. Zhou and A. Lai, "Characterization of Indoor Radio Propagat ion Environments by Cluster Analysis as Preprocessing for Neural Network Model," *IEEE,* 1995.

[11] V. Vivekanandan and V. W. Wong, "Concentric Anchor-Beacons (CAB) Localization for Wireless Sensor Networks," *IEEE,* 2006.

[12] C. Zhong, J. Eliasson, H. Makitaavola and F. Zhang, "A Cluster-based Localization Method using RSSI for Heterogeneous Wireless Sensor Networks," *IEEE,* 2010.

[13] J. Xu, W. Liu, F. Lang, Y. Zhang and C. Wang, "Distance Measurement Model Based on RSSI in WSN," *Wireless Sensor Network,* pp. 606-611, 2010.

[14] A. Redondi, M. Tagliasacchi, M. Cesana, L. Borsani, P. Tarrio and F. Salice, "LAURA - LocAlization and Ubiquitous monitoRing of pAtients for health care support," in *Personal, Indoor and Mobile Radio Communications Workshops (PIMRC Workshops), 2010 IEEE 21st International Symposium*, Istanbul, 2010.

[15] T. Mogi and T. Ohtsuki, "TOA Localization using RSS Weight with Path Loss Exponents Estimation in NLOS Environments," in *Communications, 2008. APCC 2008. 14th Asia-Pacific Conference*, Tokyo, 2008.

[16] G. A. Mizusawa, *Performance of Hyperbolic Position Location Technique for Code Division Multiple Access,* Blacksburge, Virginia: Virginia Polytechnic Institute and State University, 1996.

[17] J.-P. Linmartz, "Propagation," Baltzer Science Publishers, 1996. [Online]. Available: http://people.seas.harvard.edu/~jones/es151/prop_models/propagation.html.

[18] "NATO," [Online]. Available: http://ftp.rta.nato.int/public//PubFullText/RTO/EN%5CRTO-EN-SET-133///EN-ET-133-05.pdf.

[19] P. Biswas, H. Aghajan and Y. Ye, "ntegration of angle of arrival information for multimodal sensor network localization using semidefinite programming," in *39th Asilomar Conference on Signals, Systems and Computers*, 2005.

[20] P. Rong and M. L. Sichitiu, "Angle of Arrival Localization for Wireless Sensor Networks," in *Sensor and Ad Hoc Communications and Networks, 2006. SECON '06. 2006 3rd Annual IEEE Communications Society*, Reston, VA, 2006.

[21] R. Rick, "TDOA Localization Techniques," Harris Corporation, Melbourne, 2004.

[22] K. Sayrafian-Pour and D. Kaspar, "A Robust Model-based Approach to Indoor Positioning Using Signal Strength," *IEEE,* 2008.

[23] A. Oka and L. Lampe, "Distributed Target Tracking Using Signal Strength Measurements by a Wireless Sensor Network," *IEEE,* vol. 28, no. 7, pp. 1006-1015, 2010.

[24] A. Awad, T. Frunzke and F. Dressler, "Adaptive Distance Estimation and Localization in WSN using RSSI Measures," in *Digital System Design Architectures, Methods and Tools, 2007. DSD 2007. 10th Euromicro Conference*, Lubeck, 2007.

[25] T. Lo, J. Litva and H. Leung, "A New Approach for Estimating Indoor Radio Propagation Characteristics," *IEEE,* vol. 42, no. 10, pp. 1369-1376, 1994.

[26] L. Benetazzo, M. Bertocco, A. D. Chiara, G. Gamba, A. Sona and M. Ponzin, "Enhanced use of RSSI-based wireless network nodes for power measurement purposes," in *International Instrumentation and Measurement*, Singapore, 2009.

[27] M. Patzold, N. Avazov and V. D. Nguyen, "Design and Simulation of Measurement-Based

Correlation Models for Shadow Fading," *REV Journal on Electronics and Communications,* vol. 1, no. 2, pp. 80-87, 2011.

[28] S. Y. Park, H.-S. Ahn and W. Yu, "Adaptive Path-loss Model-based Indoor Localization," *IEEE,* 2008.

[29] N. Alsindi, C. Duan, J. Zhang and T. Tsuboi, "NLOS Channel Identification and Mitigation in Ultra Wideband ToA-based Wireless Sensor Networks," in *6th WORKSHOP ON POSITIONING, NAVIGATION AND COMMUNICATION*, 2009.

[30] G. Mao, B. D. Anderson and B. Fidan, "Online Calibration of Path Loss Exponent in Wireless Sensor Networks," *IEEE,* 2006.

[31] J.-H. Choi, J.-K. Choi and S.-J. Yoo, "Iterative Path-Loss Exponent Estimation-based Positioning Scheme in WSNs," *IEEE,* 2012.

[32] "Wikipedia," Wikimedia Foundation, Inc., [Online]. Available: http://en.wikipedia.org/wiki/Main_Page.

[33] "ekahau," [Online]. Available: http://www.ekahau.com/.