

CC2431 Location EngineBy K. Aamodt

1 KEYWORDS

- *CC2430*
- *CC2431*
- *ZigBee*
- *Location Engine*

2 INTRODUCTION

This document describes the location engine implemented in the CC2431. CC2431 is a ZigBee system on chip, so it will be natural to use the location engine in a ZigBee network. This manual is written to be as general as possible and will not describe any protocol specific considerations.

The main purposes of this document it to present some basic aspects of the location technology, and provide some hints and tips for easy developing of systems using the CC2431 location engine. This document should be read as an extension to the CC2431 and CC2430 data sheets.

Table of Contents

1	KEYWORDS	1
2	INTRODUCTION.....	1
3	LOCATION ENGINE.....	3
3.1	NODE TYPES.....	4
3.1.1	<i>Reference node</i>	<i>4</i>
3.1.2	<i>Blind Node</i>	<i>4</i>
3.2	THE LOCATION HARDWARE	4
3.2.1	<i>Input.....</i>	<i>5</i>
3.2.2	<i>Output.....</i>	<i>5</i>
4	RECEIVED SIGNAL STRENGTH INDICATOR (RSSI)	6
4.1	OFFSET	6
4.2	LINEARITY.....	6
4.3	THEORETICAL SIGNAL PROPAGATION.....	7
4.4	RSSI – PRACTICAL CONSIDERATIONS	7
4.4.1	<i>Simple ways to filter the RSSI values.....</i>	<i>7</i>
4.4.2	<i>Calculated RSSI vs. measured RSSI</i>	<i>8</i>
5	DIFFERENT PARAMETERS – INFLUENCE	9
5.1	A – RSSI VALUE MEASURED ONE METER FROM THE SENDER	10
5.1.1	<i>Measuring A</i>	<i>10</i>
5.1.2	<i>A versus calculated position</i>	<i>11</i>
5.2	N – SIGNAL PROPAGATION COEFFICIENT	12
5.2.1	<i>Measuring n.....</i>	<i>13</i>
5.3	NUMBER OF REFERENCE NODES	14
6	SOFTWARE ALGORITHMS	15
6.1	SELECTION OF “BEST” REFERENCE NODES.....	15
6.2	EXTENSION OF THE COVERED AREA.....	15
6.3	LEVEL/ FLOOR INDICATION	16
7	CONTROL SYSTEM/ CENTRAL	18
8	GENERAL INFORMATION	19
8.1	DOCUMENT HISTORY.....	19
9	IMPORTANT NOTICE	20

3 LOCATION ENGINE

The location algorithm used in the CC2431 Location Engine is based on Received Signal Strength Indicator (RSSI) values. The RSSI value will decrease when the distance increases.

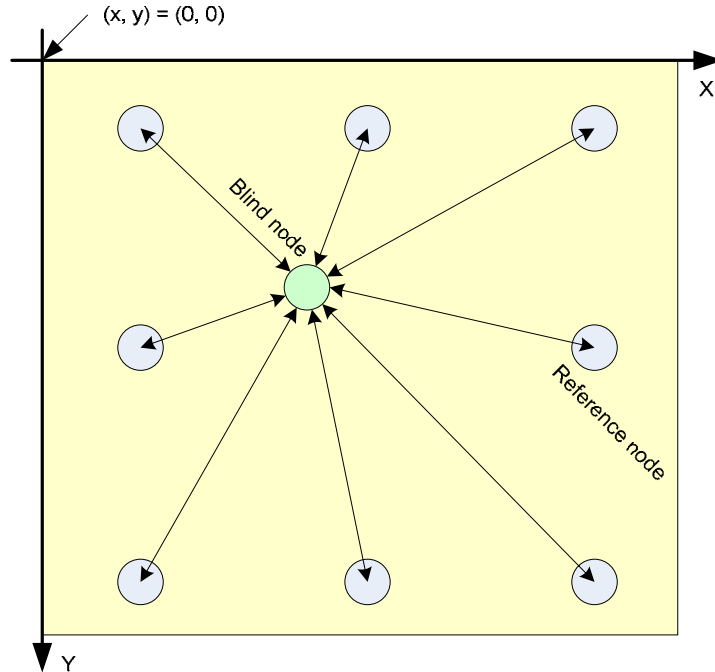


Figure 1: Location Estimation

Figure 1 shows a simplified system for location detection. “Reference node” is a static node placed at a known position. For simplicity this node knows its own position and can tell other nodes where it is on request. A reference node does not need to implement the hardware needed for location detection, it will not perform any calculation at all. A “Blind node” is a node built with CC2431. This node will collect signals from all reference nodes responding to a request, read out the respective RSSI values, feed the collected values into the hardware engine, and afterwards it reads out the calculated position and sends the position information to a control application.

The minimum data contained in a packet sent from a reference node to a blind node shall be the reference nodes’ X and Y parameters. The RSSI value is calculated by the receiver, i.e. the blind node.

The main feature of the location engine is that the location calculation can be performed at each blind node, hence the algorithm is decentralised. This property reduces the amount of data transferred in the network, since only the calculated position is transferred, not the data used to perform the calculation.

To map each location to a distinct place in the natural environment, a two dimensional grid is used. The directions will, in the following, be denoted X and Y. In all the figures X is defined to be the horizontal direction and Y the vertical. The CC2431 Location Engine can only handle two dimensions, but it’s possible to handle a third dimension in software (i.e. to represent floors in a building). The point named $(X, Y) = (0, 0)$ is located in the upper left corner of the grid.

3.1 Node types

3.1.1 Reference node

A node which has a static location is called a reference node. This node must be configured with X and Y value that correspond to the physical location.

The main task for a reference node is to provide a “reference” packet that contains X and Y coordinates to the blind node, also referred to as an anchor node.

Since this node is not using the hardware location engine at all, it is not necessary to use a CC2431 for the purpose. This means that a reference node can be run on either a CC2430 or a CC2431. Since CC2430/31 is based on the same transceiver as CC2420, even a CC2420 together with a suitable microcontroller can be used as reference node.

3.1.2 Blind Node

A blind node will communicate with the closest reference nodes, collecting X, Y and RSSI for each of these nodes, and calculate its position based on the parameter input using the location engine hardware. Afterwards the calculated position should be sent to a control station. This control station could be a PC or another node in the system.

A blind node must be using CC2431.

3.2 The location hardware

The location engine utilizes an extremely simple interface seen from the software layer; write parameters in, wait for the calculation to performed, and read out the calculated position out. This chapter will discuss the different parameters and how the shall be interpreted.

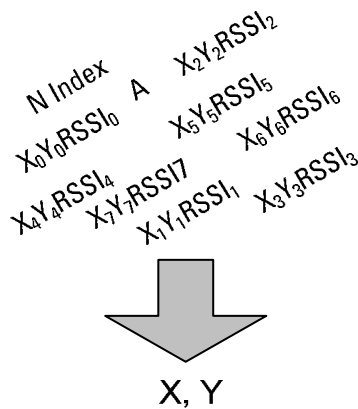


Figure 2: Location Engine, input and output

3.2.1 Input

Table 1 shows all necessary input to the location hardware. All the values will be described in details later in this document. The following is a brief introduction.

Name	Min. value	Max. value	Description
A	30	50	The absolute RSSI value in dBm one meter apart for a transmitter.
n_index	0	31	This value represent the signal propagation exponent, this value depends on the environment.
RSSI	40	95	Received Signal Strength Indicator this value is measured in dBm. The location engine using the absolute value as input.
X, Y	0	63.75	These values represent the X and Y coordinates relative to a fixed point. The values are in meters and the accuracy is 0.25 meters.

Table 1: Hardware inputs parameters

3.2.2 Output

Name	Min. value	Max. value	Description
X, Y	0	63.5	These values represent the calculated X and Y coordinates relatively to a fixed point. The values are in meters.

Table 2: Location Engine Output

4 RECEIVED SIGNAL STRENGTH INDICATOR (RSSI)

When CC2430/31 receives a packet it will automatically add an RSSI value to the received packet. The RSSI value is always averaged over the 8 first symbol periods (128 μ s). This RSSI value is represented as a one byte value, as a signed 2's complement value. When a packet is read from the FIFO on the CC2431 the second last byte will contain the RSSI value that was measured after receiving 8 symbols of the actual packet. Even if the RSSI value is captured at the same time as the data packet is received, the RSSI value will reflect the intensity of received signal strength at that time, not necessarily the signal power belonging to the received data. This gives the opportunity for the RSSI value to be erroneous when a large number of nodes are talking on the same channel at the same time as the RSSI value is captured.

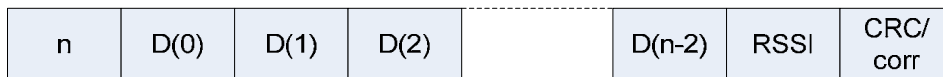


Figure 3: Received data packet

CC2430/31 contains a register termed RSSI. This register holds the same values as described above, but it is not locked when a packet is received, hence the register value should not be used for further calculations. Only the locked RSSI value attached to the received data can be interpreted as the RSSI value measured exactly when the data is received.

4.1 Offset

The RSSI value described above is represented as signed 2's complement. The value can not be read and interpreted as the received signal strength as it is. To convert the actual read out value to the received signal strength an offset must be added. This offset, which is given by the data sheet is approximately -45, furthermore this offset will depend on the actual antenna configuration.

4.2 Linearity

Measurements performed in TI's laboratory shows that the RSSI values measured by the chips fit nicely with the signal input power. The linearity curve can be found in the CC2430 data sheet plotted as input power versus RSSI value.

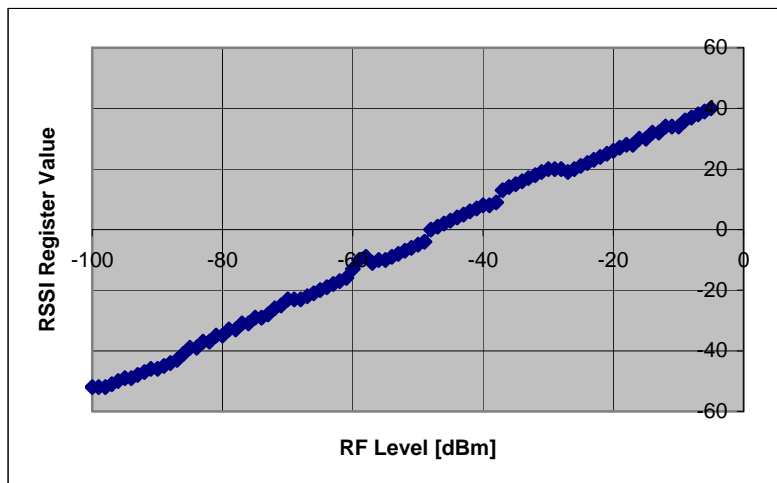


Figure 4: Typical RSSI value vs. input power

4.3 Theoretical signal propagation

The received signal strength is a function of the transmitted power and the distance between the sender and the receiver.

The received signal strength will decrease with increased distance as the equation below shows.

$$RSSI = -(10n \log_{10} d + A)$$

- n: signal propagation constant, also named propagation exponent.
- d: distance from sender.
- A: received signal strength at a distance of one meter.

A wider discussion of A and n can be found in chapter 5.

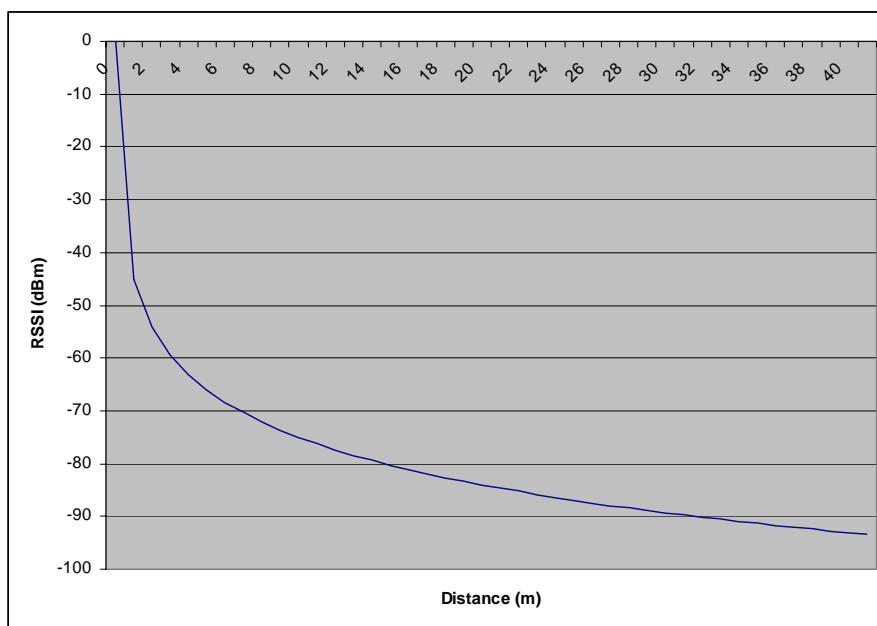


Figure 5: RSSI versus distance for A = 40, and n = 3

4.4 RSSI – Practical considerations

Section 4.3 described the theoretical RSSI value as a function of the distance. This section will discuss how the RSSI value can be expected to be measured in the real world. When using the ideal formula for signal strength it's pretty straightforward to do the calculation, but when using real values uncertainty must be taken into account. Most of this uncertainty is handled by the hardware, but some software handling should be added to increase the accuracy. The methods presented in this section have one main goal: obtain an RSSI value that correlate to the distance in the best possible way.

4.4.1 Simple ways to filter the RSSI values

Various filters can be used to smooth the RSSI value. Two common filters are simple averaging and feedback filters. Averaging is the most basic filter type, but it requires more data packets to be sent. Feedback filters uses only a small part of the most recent RSSI value for each calculation. This requires less data, but increases the latency when calculating a new position.

The average RSSI value is simply calculated by requiring a few packets from each reference node each time the RSSI value are measured and calculated according to the equation below.

$$\overline{RSSI} = \frac{1}{n} \sum_{i=0}^{i=n} RSSI_i$$

If a filter approximation shall be used, this can be done as shown below. In this equation the variable a is typically 0.75 or above. This approach ensures that a large difference in RSSI values will be smoothed. Therefore it is not advisable if the assets that should be tracked can move long distance between each calculation.

$$RSSI_n = a \cdot RSSI_n + (1 - a) \cdot RSSI_{n-1}$$

4.4.2 Calculated RSSI vs. measured RSSI



Figure 6: Theoretically vs. measured RSSI, distance given in logarithmic scale

The figure shows, from left to right, the theoretical RSSI value, next when a slowly varying components, and finally when adding fast varying components, for example under influence of multipath components. The rightmost figure shows the signal that is closest to reality. Notice that the figures are not showing any real measurement, it is only drawn to indicate some of the problems with using RSSI values to calculate position.

5 DIFFERENT PARAMETERS – INFLUENCE

The location engine implemented in CC2431 is using two different parameters in the position calculation process. They are named “A” and “n”, and they are both discussed in the following.

The examples in this section are using values from a real experiment. The experiment used eight nodes, located as shown below. The blind node was placed in centre of the reference node grid, and the RSSI values were typically measured at the locations shown in the table.

	X	Y	RSSI
Node 0	20.00	20.00	-71
Node 1	20.00	24.00	-59
Node 2	20.00	28.00	-55
Node 3	20.00	32.00	-69
Node 4	24.00	20.00	-72
Node 5	24.00	24.00	-81
Node 6	24.00	28.00	-50
Node 7	24.00	32.00	-60

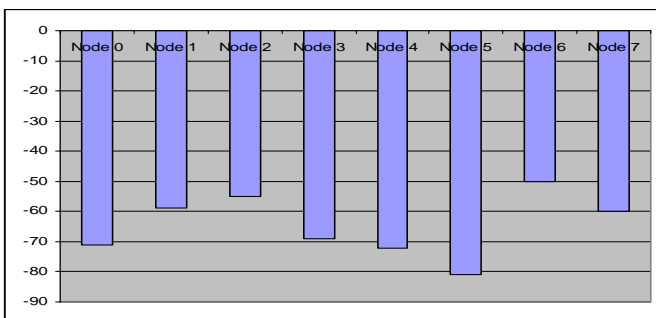


Figure 3: Reference nodes used in the example

In the example Node0, Node3, Node4 and Node7 were placed at equal distances from the blind node. Node1, Node2, Node5 and Node6 were also placed at equal distance from the blind node, but some meters closer. As the values show, this is not directly represented in the measured values.

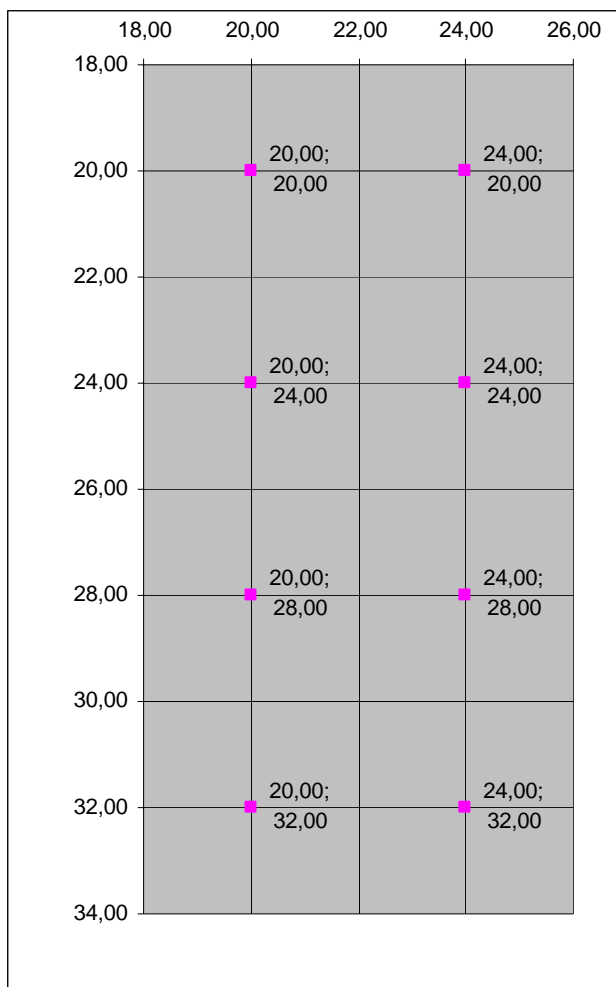


Figure 7: Reference node placement in grid

5.1 A – RSSI value measured one meter from the sender

A, which is an empirical parameter can be determined by measuring the RSSI value one meter from the transmitting unit.

5.1.1 Measuring A

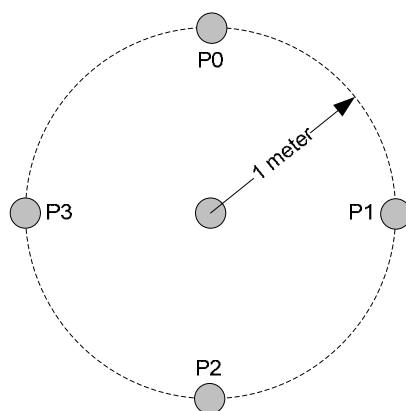


Figure 8: Measuring A

The value A should theoretically be equal in all directions. It is highly unlikely that the antennas on both the transmitter and the receiver are completely isotropic, thus an averaged value should be used.

Figure 9 shows typical RSSI values measured one meter from sender. The figure shows values measured at position P0, P1, P2 and P3 from Figure 8. The conclusion from this figure is that the antenna is not isotropic, so an averaged value should be used as the parameter A. For the devices used in this test, the averaged value is approximately -46. These measurements were done with RSSI_OFFSET equal to -45.

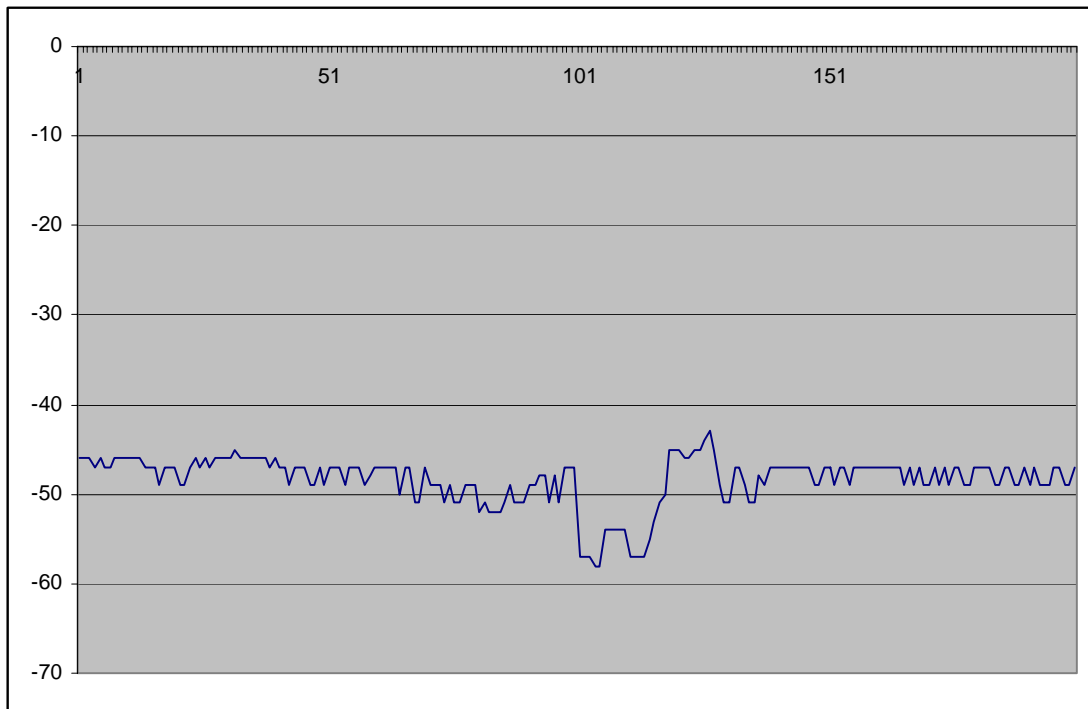


Figure 9: Measured RSSI at a distance of one meter

5.1.2 A versus calculated position

The figure below shows how an erroneous value for the parameter A will affect the calculation of a blind node's position. The figure is not showing which A value that is most correct to use. The blue dot shows that the blind node calculates its own position differently by using different values for A, relatively big differences in X and Y can be obtained. Generally an A value between 45 and 49 gives most accurate answer for an indoor office environment. In this example the calculated positions with A between 45 and 49 were approximately (22, 28), while the real position was (22, 26).

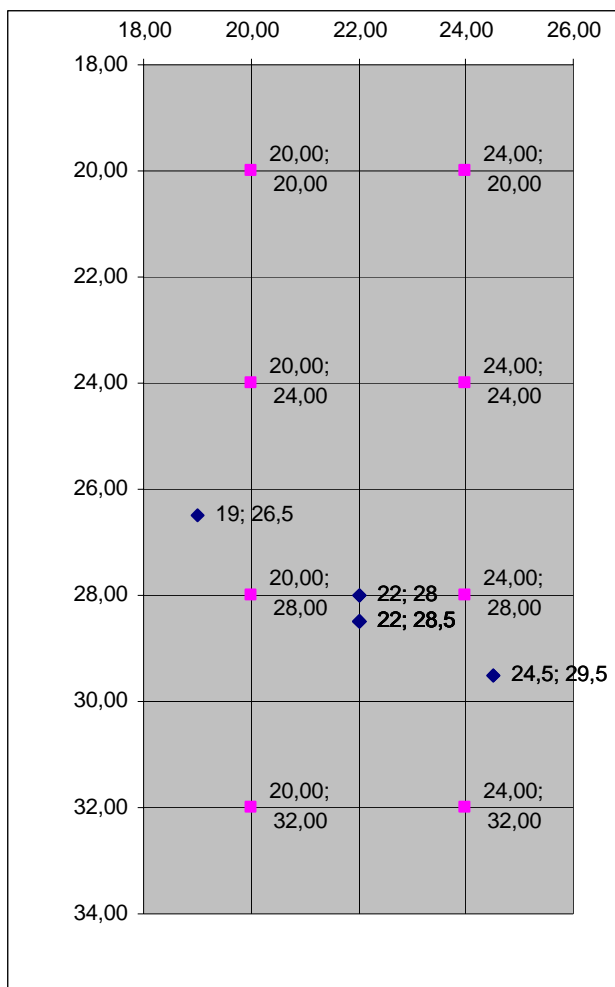


Figure 10: A versus calculated position

5.2 N – Signal propagation coefficient

The parameter named N is a parameter that describe how the signal strength decreases when the distance from the transmitter increases. N is highly dependent of the environment. E.g. the thickness of walls will influent a lot. This value can only be determined empirically.

The location engine implemented in CC2431 is not using N directly, instead it is using a value termed n_index. The relation between N and n_index can be seen in the table below. This conversion table is used to reduce the complexity for the actual hardware implementation.

n_index	N	n_index	N
0	1.000	16	3.375
1	1.250	17	3.500
2	1.500	18	3.625
3	1.750	19	3.750
4	1.875	20	3.875
5	2.000	21	4.000
6	2.125	22	4.125
7	2.250	23	4.250
8	2.375	24	4.375
9	2.500	25	4.500
10	2.625	26	4.625
11	2.750	27	5.000
12	2.875	28	5.500
13	3.000	29	6.000
14	3.125	30	7.000
15	3.250	31	8.000

Table 4: N and n_index

From Table 4 N can be chosen from 1.0 to 8.0.

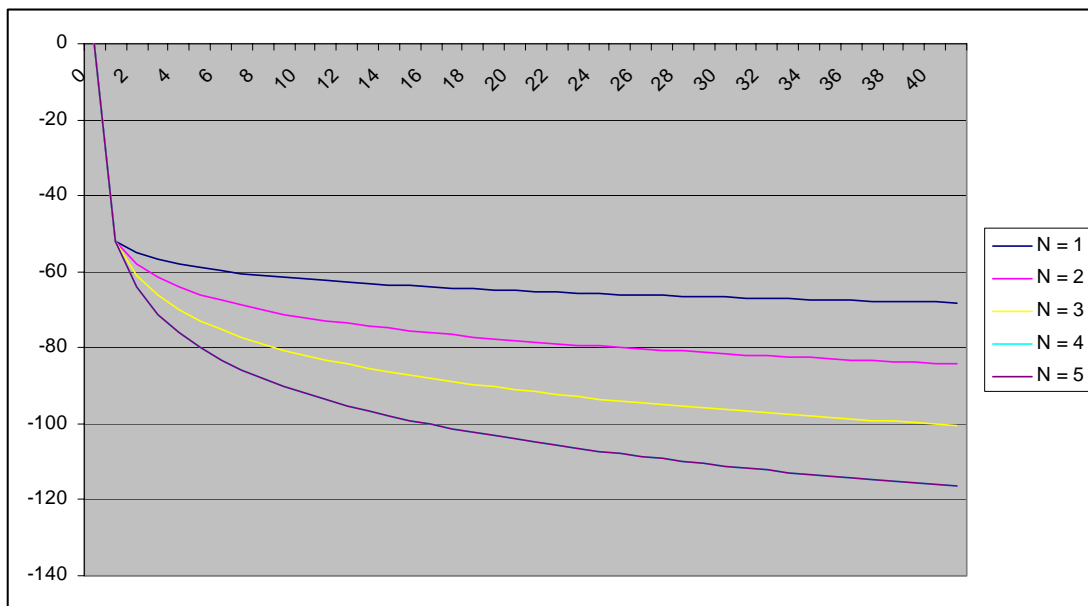


Figure 11: N versus distance

Figure 11 shows how the choice of N affects the theoretical RSSI value at a given distance, where the distance is given in meters. N should be fitted to the actual environment as accurate as possible.

5.2.1 Measuring n

It is challenging, if not impossible, to find a default value that is optimum n_index for all environments. The easiest way will probably be to install all reference nodes in the environment where the system should be used and thereafter test with different values of n_index to find the value that fit the best with the actual environment.

Experiments have shown that an `n_index` value between 15 and 25 gives good results.

5.3 Number of reference nodes

As a rule of thumb; as many reference nodes as possible should be used. At least three is necessary to give a reliable result. If too few nodes are used, then the influence from each of the nodes is higher, and one erroneous RSSI value can change the calculated position in a significant way. Erroneous RSSI values in this context means an RSSI value that does not fit well with the theoretical value, e.g. due to multipath or if the signal is blocked by walls or similar.

If the blind node is located outside the grid of reference nodes, it is highly probable that the result will differ too much from the real position to be usable. It's not advisable to track objects that are located outside the grid.

6 SOFTWARE ALGORITHMS

Some general algorithms can be implemented in software, they are described on a high level in the following. Please notice that this application note not describes any thing specific to the network topology in use.

6.1 Selection of “best” reference nodes

The nearest nodes should be used for calculation. This means that the eight reference nodes with highest RSSI value should be used, all other nodes should be discarded. If eight nodes not can be reached, as many nodes as possible should be used.

6.2 Extension of the covered area

The hardware location engine can handle X and Y values up to 64 meters, or more precisely up to 63,75 meters in both directions. This is too small an area for many practical applications. Therefore it will be necessary to extend the area. This can easily be archived by a simple software pre-processor algorithm.

Each node represents X and Y with 2 bytes. With an accuracy of 0.25 meters, this gives a maximum range of 16 384 m.

	Max	Accuracy
X	$2^{14} = 16\,384\text{m}$	0.25 meter
Y	$2^{14} = 16\,384\text{m}$	0.25 meter

Table 5: Dimensions

Figure 12 shows an example of how the algorithm works. The figure shows reference nodes placed at each 30th meters in both the X and Y directions. On the figure the light green node is the blind node, all other nodes are reference nodes.

- Step one is to locate the one node with the highest RSSI value and calculate an offset that “move” this node to the centre of a 64x64 meters square. Because of the known RSSI value for the signal from this node, the distance to that node can easily be estimated. In Figure 12 the position must be in the white circle.
- The next step is to locate the other reference nodes to be used in addition to the “strongest” one. In the figure the nodes coloured in dark blue are used. All nodes are offset with the value found in step 1.
- All values found are fed into the location engine hardware and the resulting position is read out.
- The last step is to add the same offset to the position calculated. After these calculations are done, the location of the blind node is known in the global grid.

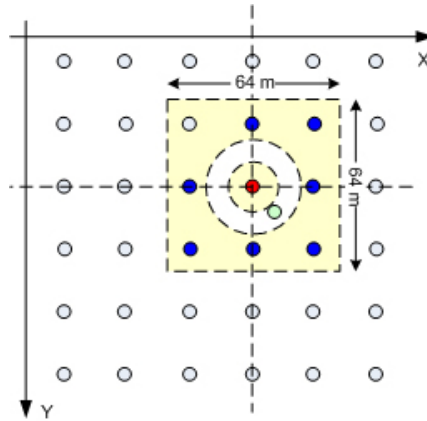


Figure 12: 64x64 meters mapping

6.3 Level/ floor indication

The hardware location engine is designed for calculating of positions in two dimensions. This chapter gives some hint and ideas on how this can be extended in software to handle a third direction, e.g. floors in building.

To indicate levels one coordinate must be added in addition to X and Y. In the following this coordinate is denoted Z. For most purposes it will be enough to represent Z with a one byte value, this gives 256 different levels.

It is assumed that the received signal strength from a node placed on another floor than the blind node is lower than the signal strength from a node in the same floor. This implies that the density of reference nodes will be pretty high. It is not assumed that all reference nodes on the same floor as the blind node are stronger than all reference nodes at all other floors.

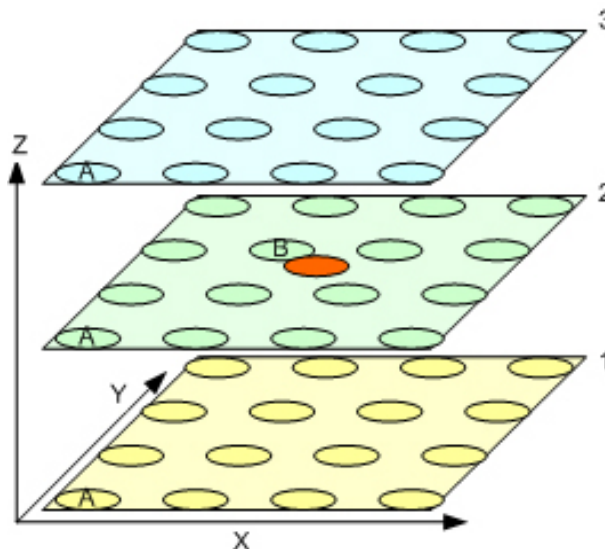


Figure 13: Location in a three dimensional network

Figure 13 shows a simplified network used for position tracking between levels. The three nodes marked with an 'A' all have the same X and Y values, but different Z. All nodes colored with the same color have the same Z, but different X and Y coordinates.

An algorithm that can be used is to first locate the nearest reference node and read out the level indication reported from this node. This level is assumed to be the level the blind node is located. The blind node then needs to make sure that only nodes in the same level are fed into the location engine. Anyway, an algorithm must be used to select which reference nodes that should be used, level indication will only add one criteria to this algorithm. The main key in this algorithm is to decide which level the blind node is on. This can be achieved in different ways. If the mentioned assumption is not valid, some other algorithm must be implemented.

In this example it is assumed that the closest reference node will provide the blind node with the strongest signal. Figure 13 shows a network distributed over three floors. The red node on floor 2 is assumed to be a blind node. The algorithm described above would in this case do the following:

- (1) Find the closest node, this node is labeled B in this case
- (2) Node B will tell the blind node that it is located on floor 2
- (3) The blind node will then assume (know) that it also is located at floor 2
- (4) All other reference nodes used for the calculation must also be located at floor 2 (colored in green in the figure), this must be sorted out by the blind node
- (5) The blind node calculates its position in the normal way, only by using X and Y directions
- (6) The calculated X and Y coordinates, together with the decided Z-coordinate, is reported as the three dimensional position

7 CONTROL SYSTEM/ CENTRAL

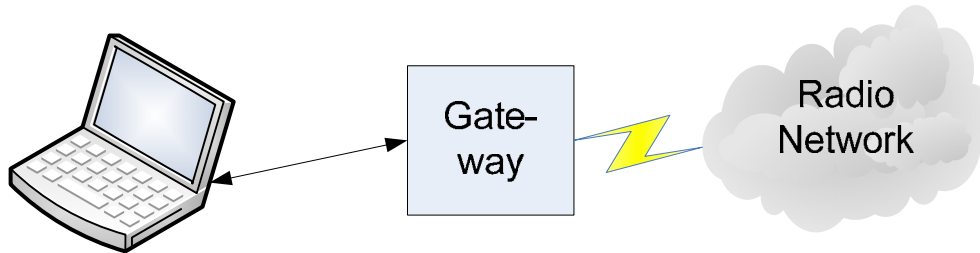


Figure 14: Interfacing the radio system from a PC

To collect calculated data and interact with the network of radio nodes, some kind of control system is necessary. A typical control unit will be a computer. Since a PC not typically has an embedded radio transceiver, a transceiver needs to be connected externally.

It is two main approaches to build a gateway; (1) the gateway can be a passive device that only listens on traffic in the network, or (2) it can be an active part of the network. The passive approach will mainly be a packet sniffer, the gateway will only capture RF packets and provide this information to the user. The active approach gives the user more flexibility, but it's harder to implement.

What functionality that resides in this transceiver, called gateway, is up to the application. Some typical minimum functionality can be:

- Configuring of nodes.
- Check the status of network, e.g. how many nodes are connected.
- Require the blind nodes to perform calculations if they not are set up to do it themselves on regular intervals.

The device which performs as a gateway can be a part of the calculation network itself, and act as reference node or as a blind node. This means that gateway can be both a gateway and a reference node or blind node.

Since all position calculations are performed in the blind nodes, the control device will not perform any position calculations at all. Its only purpose is to give the user the ability to interact with the network.

8 GENERAL INFORMATION

8.1 Document History

Revision	Date	Description/Changes
1.0	2006-07-10	Initial release.