Kalman Tracking Prototype based on Zigbee

Juan M. Camús[1], P. Corral[1], A.C.D.C. Lima[2]
[1]Grupo Universitario de Investigación en Redes Inalámbricas, Universidad Miguel Hernández de Elche (Spain)
[2]Universidade Federal de Bahia, Dpto. Enginharia Elétrica, Salvador (Brazil)
pcorral@umh.es

Abstract.- This paper presents a prototype development that uses wireless communication protocols in order to perform distance estimation, objects location and, finally, objects and persons tracking. This system is based on Zigbee, a technology that allow us to have minimum consumption and low cost, that can track a mobile indoor within an error round twenty-five centimeters in 90% of the cases using Kalman Filter.

I. Introduction

An indoor tracking system allows you to know the trajectory that follows a Mobile at indoor conditions. Apart from adding functionalities to existing networks or, as the search of devices or the presence detection, the objective of an indoor tracking system is to try to mitigate the logic limitations of GPS location in interiors.

For that reason, there is a lot of bibliography on the range estimation and location in this type of surroundings. Using ultrasounds technology we can obtain better results like [1], which achieve precisions of 15 cm. The problem of such systems is that are dedicated surroundings: their nodes are only useful for the position estimation. In the present systems, nevertheless, one tends to integrate all the possible functionalities in a single system.

With this philosophy, systems of range estimation or location in interiors have been developed on already existing networks, like Bluetooth in [2] or WLAN with standard 802.11 in [3]. In this type of networks, the achieved precisions usually are within a meter.

Our election has been to integrate a tracking system on ZigBee, a commercial specification based on the standard IEEE 802.15.4. This standard fulfills our requirements exactly. In first place, we can use this network as a home automation system. In that way, it is common to have several nodes in same environment.

In addition, its philosophy is based on minimum consumption and low cost, and the system that we have implemented adds a minimum hardware that does not increase in price the final node. So, our system will try to reach precisions similar to which obtains other non-dedicated networks, but being added to the advantage of the minimum cost and low power.

The article is divided as it follows: in section II, we make a small summary of the standard 802.15.4. In section III we explain the hardware that we use to achieve our objective. In IV and V we show how we estimate the distance, and the location, respectively. In VI the Kalman Filter is showed. In VII we describe the implemented system and, finally, in VIII the conclusions are said.

II. Wireless Personal Area Network: Zigbee

The personal area networks (PAN) are networks for interconnection of devices near a person that habitually have smaller reaches of 10 meters. Between the radio networks of this type, we have the well-known Bluetooth, with a medium transmission rate, and UltraWideBand, whose standard still is unfinished, and that it allows high rates of transmission. For our objective, we choose to use the standard IEEE 802.15.4 [4], that describes layers physical and MAC of a radio network of low rate of transmission (LR-WPAN) and therefore of low cost. Between its basic characteristics:

- Maximum rate of data transfer of 250 Kbps.
- 16 channels in the frequency band of 2,4 GHz.
- Access to channel CSMA/CA.
- QPSK Modulation, DSSS.
- AES Encryption.

Three types of nodes in a network 802.15.4 are distinguished:

- ZigBee coordinator (ZC) or FFD (Full Function Device): The most capable device, the coordinator forms the root of the network tree and might bridge to other networks. There is exactly one ZigBee coordinator in each network since it is the device that started the network originally. It is capable of storing information about the network.

- ZigBee Router (ZR): As well as running an application function a router can act as an intermediate router, passing data from other devices.

- ZigBee End Device (ZED) or RFD (Reduced Function Device): Contains just enough functionality to talk to its parent node (either the coordinator or a router); it cannot relay data from other devices. This relationship allows the node to be asleep a significant amount of the time thereby giving you the much quoted long battery life.

On standard 802.15.4, a group of 25 well-known companies, like Philips and Motorola, created the ZigBee specification [5] that contributes the superior levels (network and application) of the WPAN, with functions like the routing or the extreme security to end. Between the solutions available in the market, our project leaves from contributed by Microchip with its kit of evaluation of ZigBee PICDEM Z.

III. Hardware Explanation

In order to implement the decided system, a tracking system integrated in a network 802.15.4, the basic components are a microcontroller, a transceiver and an antenna; in addition to the devices that we will use to measure the time to apply it to our algorithm and calculate the position of the mobile terminal.

This way, part of the evaluation kit PICDEM Z, that provides two ZigBee nodes, but we just used one of the them, the one who works as FFD, because the other one did not have the devices to measure the time and we decided to built our RFD device, the one who was going to work as mobile terminal. In addition, we will built the RFD nodes used to position de mobile terminal. The main difference between the RFD device and the RFD nodes is that the first one has the counter system. In Figure 2, the electrical scheme of counter system is shown and in Figure 3, the layout of the whole RFD device appears.All the devices have the microcontroller PIC18F4620 and transceiver CC2420 from Texas Instruments.

The configuration of the transceiver and its handling are carried out from the microcontroller. The communication microcontroller-transceiver makes use of the communication protocol SPI (Serial Communication Interface). It is a serial synchronous communication, in which the microcontroller acts like master, sending the clock signal and indicating when the communication by means of the pin is made Select Chip.

In addition to these two lines, other two are only required, for outgoing data and incoming data.

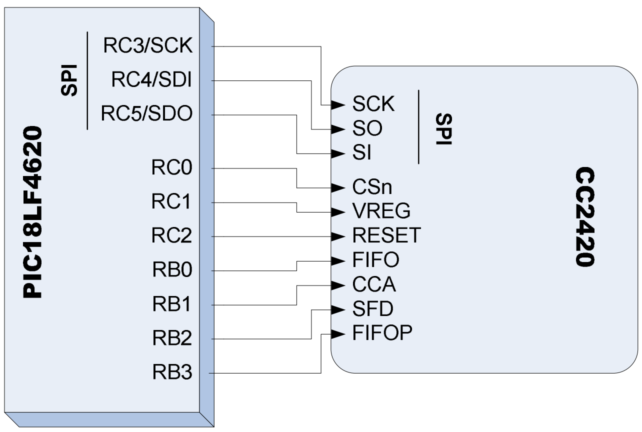

Fig. 1. Microcontroller-Transceiver Connections.

In each node, the secondary card (daughter card) where the transceiver is, we also have a printed antenna of similar characteristics to the recommended one by Chipcon and that is omnidirectional in the horizontal plane.

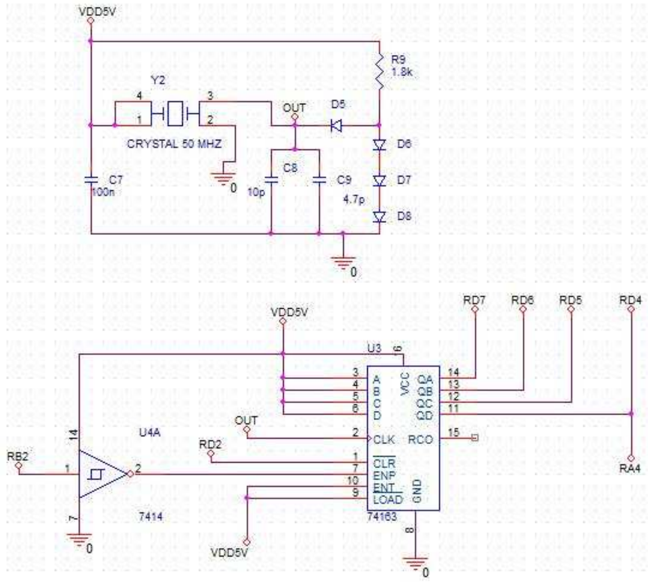

Figure 2: Electrical Design of auxiliary Hardware

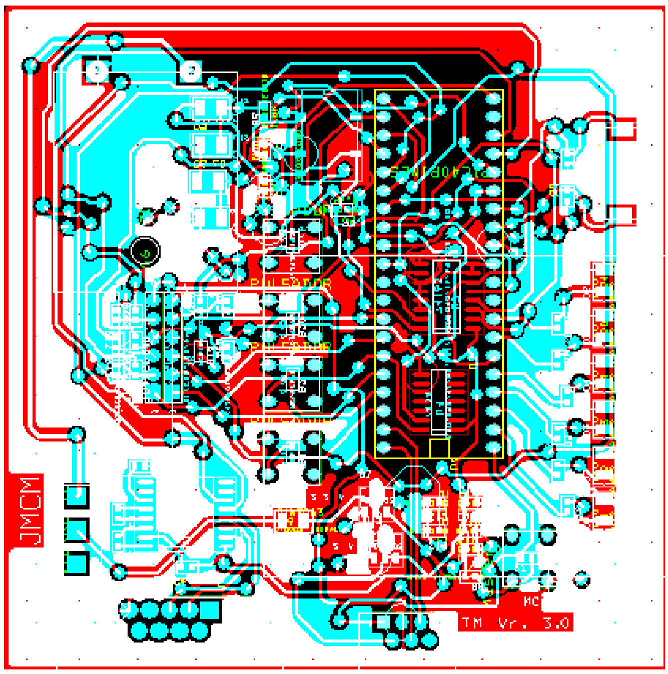

Figure 3: Mobile Terminal's Layout

IV. Distance Estimation

In our system, the base to calculate firstly localization and finally tracking was the measure of the distance between the RFD device and the FFD device. From the diverses techniques used to calculate the distance of a mobile (TOA, AOA and RSSI [6]), we decided to use TOA (Time of Arrival). This technique is based on the measurement of the time of arrival of a signal transmitted by a mobile terminal to different base stations. In order to carry out the calculation, a possibility is to measure the time of roundtrip of the signal (RTT, Round Trip

Time). This way, the whole range travelled by the signal is calculated as the product of the time used in arriving at the base stations and the speed of light.

To calculate the RTT, we had to create a separate hardware that were able to count the time, and it had to be low cost and low consume. We decided to use a count with an auxiliary oscillator just for it, and when we sent the frame, the microcontroller activate the counter through the SFD signal of the transceiver.

Once known the hardware, the basic procedure consists of sending a frame used by the standard and waiting for an answer. If we know the time that have taken in arriving the answer and the speed to which the signal travels and we avoided the time of processing of the frame, we can obtain the distance that separates both nodes:

$$d = \frac{c \cdot t}{2} \qquad (1)$$

where d is the distance that separate both nodes, c the speed of the light and t the time that takes the signal in crossing the space that separates the nodes in roundtrip (for that reason it is divided by 2).

The problem is there is a time that past while the FFD device is creating the frame to answer. Let us observe the following diagram:

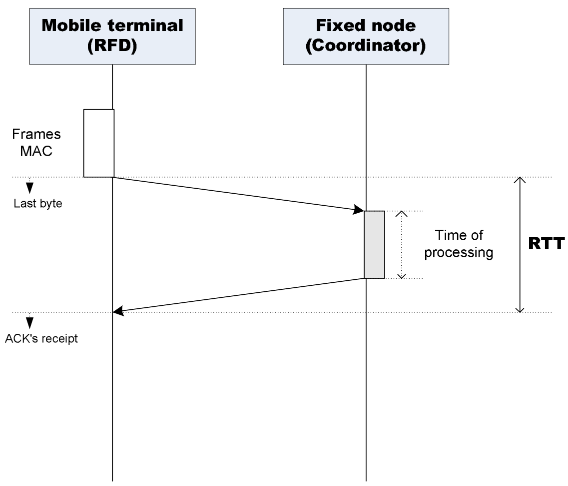

Fig. 2. Round Trip Time.

As we see, the RTT is the time passed from the shipment of the last byte of the MAC frame, to the reception of the answer. Modifying the previous equation to adapt it to our case, theoretically we could obtain the distance between two nodes of the following form:

$$d = \frac{c \cdot \left(RTT - t_{procesamiento} \right)}{2} \qquad (2)$$

In order to avoid having another hardware of count in the fixed node, due to processing times of non-constant frames, we used AUTOACK function available with transceiver CC2420. When activating it, according to [7], the transceiver gives back an ACK with the direction of the node that has sent it the original frame, exactly 12 symbols (1 symbol = 4 bits) after receiving it.

V. Localization

When we were able to measure the distance, we moved on and we thought about the idea of doing localization. We copied the method that GPS use to calculate the position of a mobile. It is Multilatheration. But in our case, we just used trilateration (Figure 4) because, as we have said before, we wanted a system that was low cost and we studied that the minimum number of nodes that you need to find the position of your mobile is three.

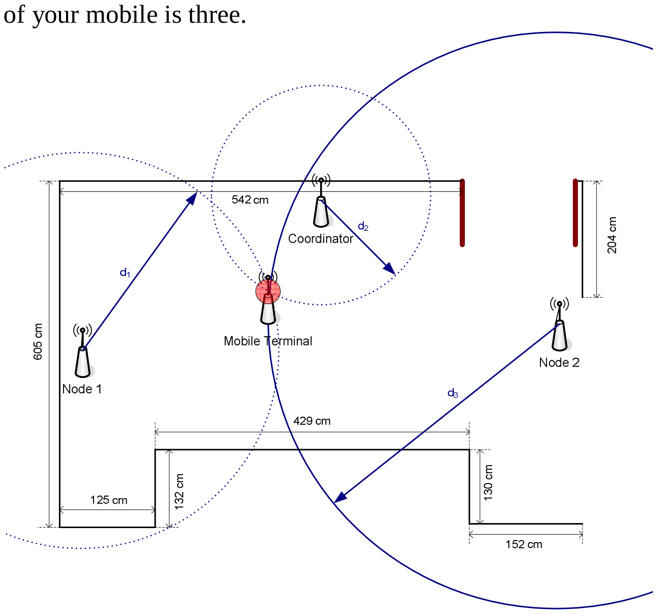

Figure 4: Trilateration

The trilateration basis is that if you find the intersection point of the circles with radio, the distance between your mobile and the other nodes, you will have the position of the mobile. Mathematically, you have to solve the second grade equation as follows.

$$(x - x_i)^2 + (y - y_i)^2 = r_i^2 \qquad (i = 1,2,3)$$

(x, y) :Mobile Terminal Position

(x_i, y_i) :ith Node Known Position

r_i :ith Distance between Node and Mobile Terminal

Computationally, solving this equation's system would be inefficient, for that we will use the following mathematical procedure, we programmed in C++. We created a lineal system that we could represent is by matrix way.

$$\overline{A}\vec{x} = \vec{b}, \qquad (3)$$

with,

$$\overline{A} = \begin{pmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \\ \vdots & \vdots \\ x_n - x_1 & y_n - y_1 \end{pmatrix}, \quad \vec{x} = \begin{pmatrix} x - x_1 \\ y - y_1 \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} b_{21} \\ b_{31} \\ \vdots \\ b_{n1} \end{pmatrix} \qquad (4)$$

However, if the case is that the distances are not precise, the results are unacceptable. So, we use the Newton's method to solve the equation that we had seen before.

The Newton method, also called No-Lineal Squared Minimum, show in different studies [8] that it is the trilateration method more precise. Newton's method is an iterative method, where you start form an initial point of the curve that we want to get the zero. That initial point should be reasonably close to zero. Following, we calculate the point where it cuts the abscises axis with the tangent line, using the function's derivate. The new point will be, generally, a better approximation to the root of the function. We can apply this proceed as much times as we can. In Figure 5 is shown the Newton's method in unidimensional case.

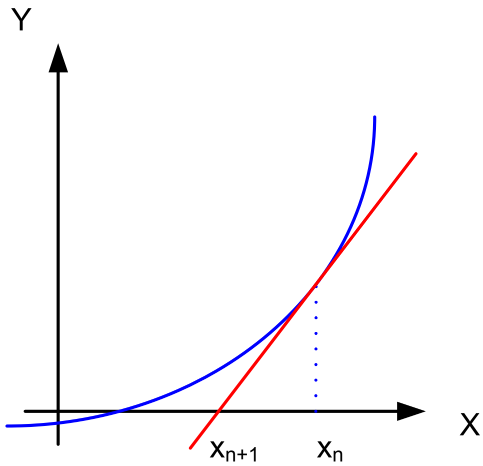

Figure 5. Newton's Method

For the creation of the code for our application, we started from stack of Microchip, modifying and adding the necessary code; and with precaution of not interfering in the on-speed operation of a network ZigBee, since we looked for a system non-dedicated exclusively to the location. Basically, it consists of a set of archives .c, archives of head .h and other files, in which the inferior levels of the standard the 802.15.4 and basic ones of ZigBee are implemented. From this way, the end user obtains a support to create its ZigBee applications without having to worry about basic functions of ZigBee such as to program the form in that the coordinator looks for a free channel to form a network, or the series of necessary messages so that a RFD can adhere to the same one.

For estimation distance's code, we created our archive .c, where firstly we joined the net created by the FFD device and after that we sent a MAC frame, created by ourselves, with direction the FFD device. Meanwhile, in the RFD we activated the count system with the SFD bit from the tranceptor and with an extern oscillator we will be able to know how many clock's ticks has ocurred during the RTT time, and using the mathematics explained before you are able to know the distance.

Starting location has three ways:

- By pressing the corresponding button on the Coordinator
- By giving the command via PC
- By request of the User Terminal

Location process in FFD node is going to pass through different states, since receives the start order until he goes back the normal state. We have developed a .c program, which follows the next organigram (Figure 6), where we can see that the different states are in capital letters. If you don't receive any answer of at least three nodes, we have to abort the location understanding that there is not enough nodes to do multilateration, so then, we send an error message to the RFD device and let the other RFD nodes that probably have received the message to start the location go back to SLEEP state. Likewise, if the RFD device do not receive the directions that we have stored, we understand that it is isolated and we will abort the location. In our program, the maximum number of tries is three, de time out time for the nodes is 3 seconds and for the mobile terminal is 15 seconds.

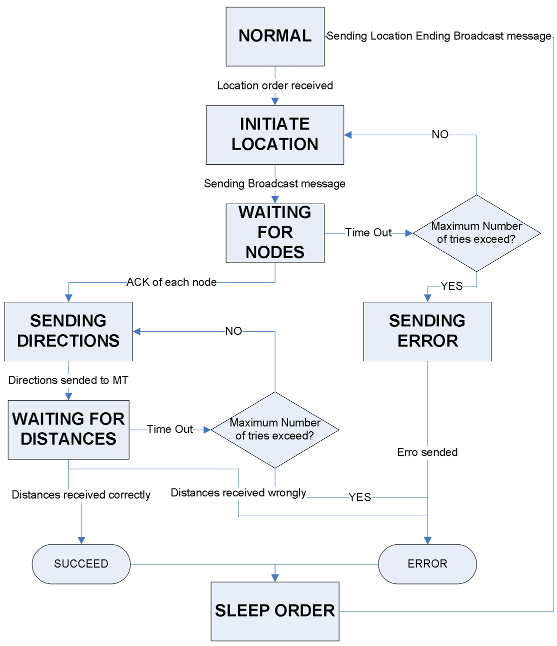

Figure 6: Location FFD Node Organigram

In the case of the RFD device or mobile terminal, we will follow a different organigram (Figure 7). The states are in capital letters too and are based on [9]. The measure estimation is the same, but now the number of transmissions and the directions from the nodes come from the FFD device or coordinator. We have two limit times in the case that it was impossible the communication with the coordinator: the first one when we send the request for initiate localization, and the other to wait the confirmation that the distances have been received properly. Both are of 4 seconds long, and the number of tries is 3.

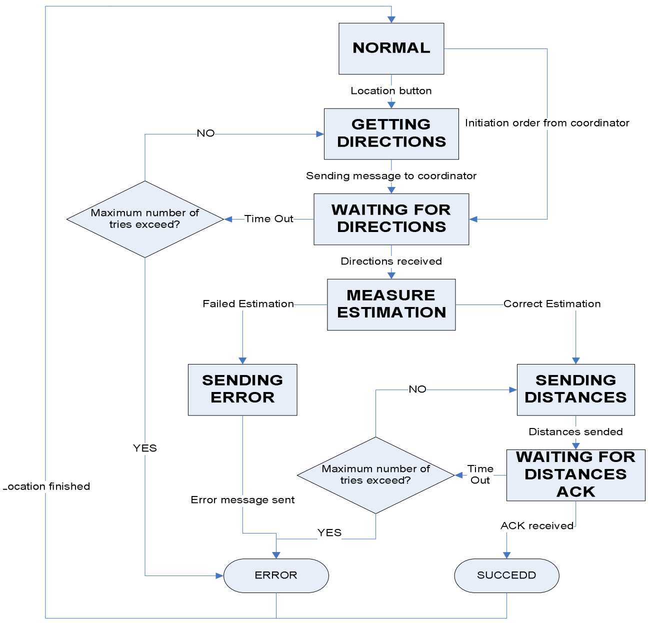

Figure 7: Location RFD Device Organigram

The rest of RFD nodes (Figure 8) share the same source code. Its working is the same that a normal RFD node; but, to make sure that the transceiver is connected, during the measure estimation have to stand awake. To go back to its normal state, it has to wait an indication that the locations has finished.

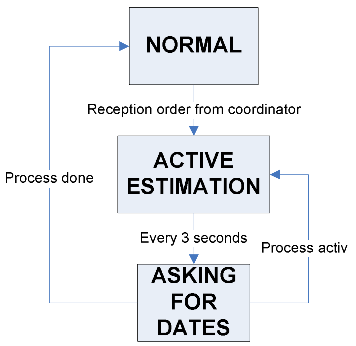

Figure 8:Location RFD Node Organigram

However, a RFD node just receives information when the coordinator requires it; and this just happens when it goes out from SLEEP mode. For that, we have programmed a state that it is activated each certain time during the localization process, when this node sound out the coordinator to check out if it is done.

VI. Kalman Filter

Before talking about our system, it is necessary to introduce the bases on the tracking system, which is tracking with Kalman Filter. As the system is known to be not really precise, as it has a resolution of 25 centimeters and easily subject to errors, due to the weakness to multipath and the lack of correction algorithms, one of the potentially useful additions to the system is the Kalman Filter [10]. It allows calculating a new trajectory on the base of the previous known positions, so that a possibly wrong calculated displacement of the User Terminal can be corrected with this "intelligent" predictive algorithm.

The Kalman Filter uses a form of feedback control to estimate the state of the system. The process through which we obtain the *k*-th state is divided in two main parts, the *time update* and the *measurement update*. The *time update* – or *predictor* – is responsible for letting the current state and error covariance estimates advance in time, to calculate the *a priori* estimates for the next time step. The *measurement update* – or *corrector* – is the feedback, thus introduces a new measurement into the *a priori* estimate to obtain an improved *a posteriori* estimate.

Each part has its own set of equations.

- Time Update Formulas

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{5}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{6}$$

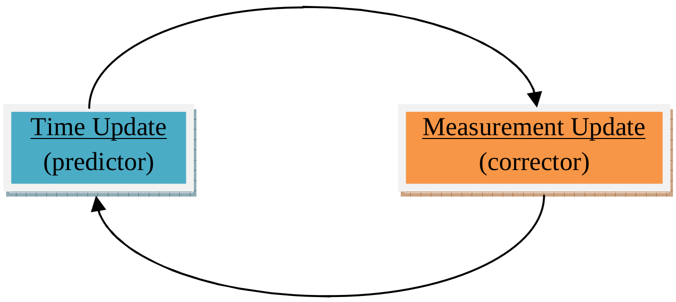

Figure 9: Simplified Kalman Filter Cycle

Here we can clearly see how the state and covariance *a* priori estimates are modified on the base of the *a posteriori* values of the previous step, as $\hat{x}_k^-$ depends from $\hat{x}_{k-1}$ and P_k^- from P_{k-1}.

- Measurement Update Formulas

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{7}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(\vec{z}_k - H\hat{x}_k^-) \tag{8}$$

$$P_k = (I - K_k H)P_k^- \tag{9}$$

The steps of the measurement update are:

- Calculate the Kalman Gain K_k by formula (7)
- Measure the process to obtain z_k, then generate the *a posteriori* state estimate as in (8)
- Finally, compute P_k, the *a posteriori* error covariance estimate via (9).

In each cycle, composed by a time update and a measurement update, the *a priori* estimates are predicted with the use of the *a posteriori* estimates obtained in the previous cycle.

What is good about Kalman Filter is that it is a recursive algorithm that works over the current state estimate by applying a modification that depends from all the past measurements, while there are other filters that operate on all of the data for each estimate [11].

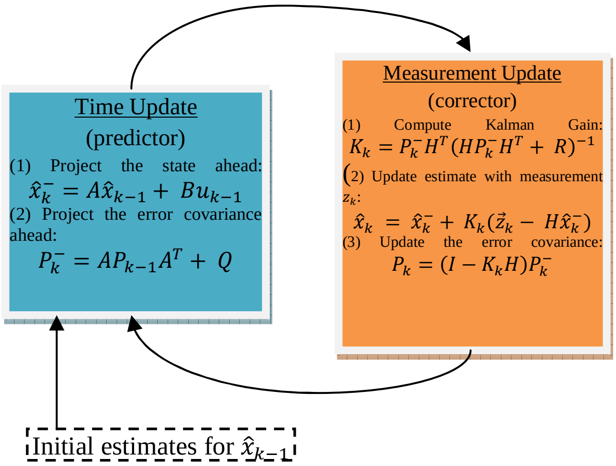

Figure 10: Detailed Kalman Filter Cycle

VII. Implemented System

When we started with the tracking system, firstly, we did a research about what method would be better to do our goal [12], we checked if it was better doing tracking with multilatheration or with Kalman Filter. In Figure 11, we can see the result, the black line is the one done with multilatheration, the red one is the real trajectory and the green one is the one done with Kalman Filter. As we can see here, the green line follows better the real trajectory. In this situation, Kalman is using all the nodes that are around and Trilateration is just using the three closed nodes. And even in this situation, Kalman has less error.

In order to obtain the same error as trilateration with the same velocity (pedestrian velocity 2.2 m/s) we have to do the iteration cycle at least two times. It could be a problem if we did the Kalman Filtering inside the microprocessor, but it is not our problem, because we are going to do it in a PC that do not have problems with processor time. We are not going to have problems with the velocity, because we are working in an indoor situation, and the studies says the common velocity is the pedestrian, round 2.2 m/s.

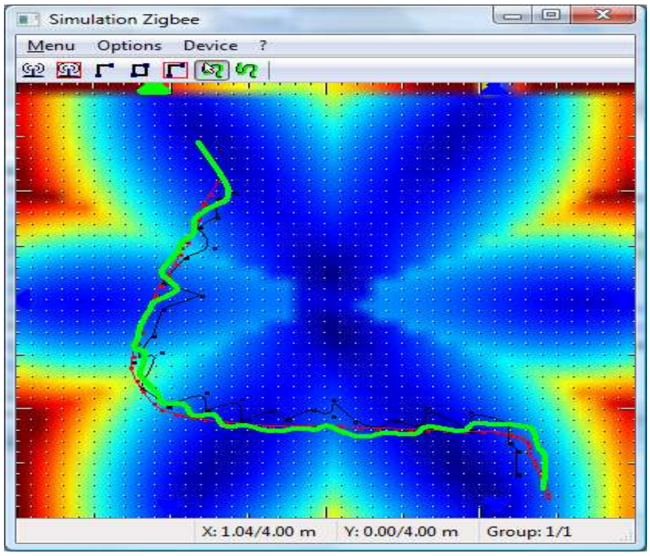

Figure 11:Tracking Simulation

Kalman Filter has different problems, the main one is that it depends on how fast the mobile is moving, because the points that Kalman calculates are so separated that in the next iteration's cycle the error that it induces to the algorithm will be wrong and the next calculated point will be useless. Another problem, is that you are not able to calculate a local position, and when you have to give Kalman the first position of the mobile, you are not able to calculate it itself. In our project, we use trilatheration to calculate the first position and this will be the first point of Kalman. Finally, when we solved all the problems about Kalman, we programmed the tracking code for each node.

The organigram for coordinator or FFD device is shown in Figure 12. There, the states that the program is going are in capital letters and marked. The main difference with the organigram in localization is that we have to difference between start and stop. The coordinator is going to continue receiving distances from the Mobile Terminal until the stop signal enter in action.

That is the moment when the coordinator send the sleep signal to the other nodes and answer to the computer seeing that it is ready again to start a new tracking. About the waiting time, here is practically the same as localization, the waiting time for the nodes is 3 seconds, and for mobile terminal to receive the distances is 15 seconds each iteration. The maximum number of tries to initiate tracking is 3, and if it cannot happen, the coordinator sends a signal to the computer saying that the tracking is impossible at the moment.

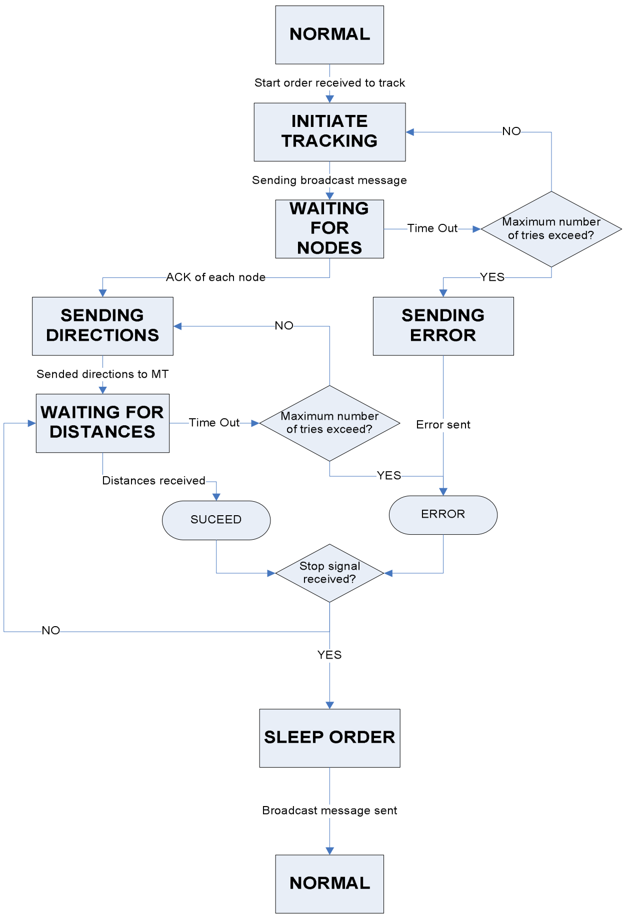

Figure 12: Coordinator Organigram

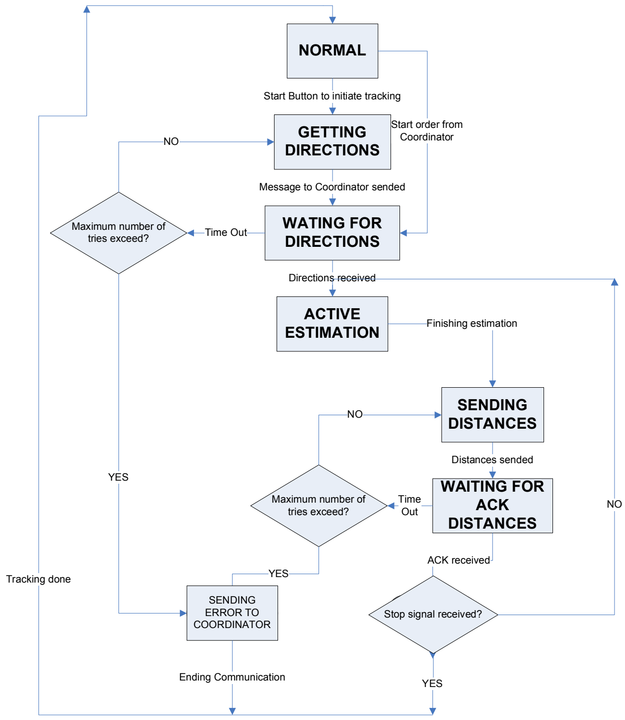

Figure 13: MT Organigram

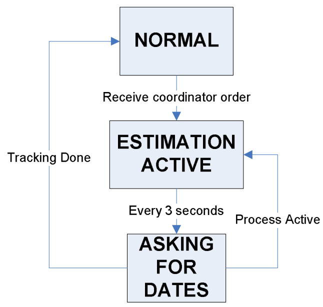

Figure 14: AP Organigram

In the RFD device (Figure 13) is where there are the most significant changes respect of localization. The problem was that the RFD device should be calculating regularly the distance estimation and sending it to the coordinator as fast as possible. We had to try to eliminate all the time that the RFD device was idle and check that had not receive the stop signal, and if it happen, stop calculating and go to sleep mode. The part of distance estimation is the same and the number of tries is the same 3. Here we have changed the time that the Mobile Terminal is waiting for the ACK of the Coordinator saying that has received, it is just 0.5 second, and it is recursive, if you do not receive it in this time, the Mobile Terminal start again the algorithm, and if it has not receive 6 ACK, stops the tracking and send an error message. We have also changed that if the Measure Estimation is wrong do not stops the tracking, it just starts again, because Kalman Filter is going to correct it.

Respect of the RFD node (Figure 14), the behaviour is the same as in the localization case, with the particularity that when tracking is active they do not go to sleep mode because the RFD device needs to be calculating as fast as possible, and it will reverberate in a higher energy consume.

If you want to start tracking, you have 3 options:

- Pressing start button in Mobile Terminal.
- Pressing start button in Coordinator
- Sending a message by RS232 to the Coordinator from the PC

To stop tracking is the same, but pressing the stop button and sending the stop message from the computer. In the figure 15, we can see the whole interaction between the nodes explained in a real case.

The last step of our investigation was creating a device as much smaller as possible. The final device is shown in Image 16. It is a device made in SMD Technology and as we can compare in the photographs is really small. The box near to the SMD device is less than a tobacco box.

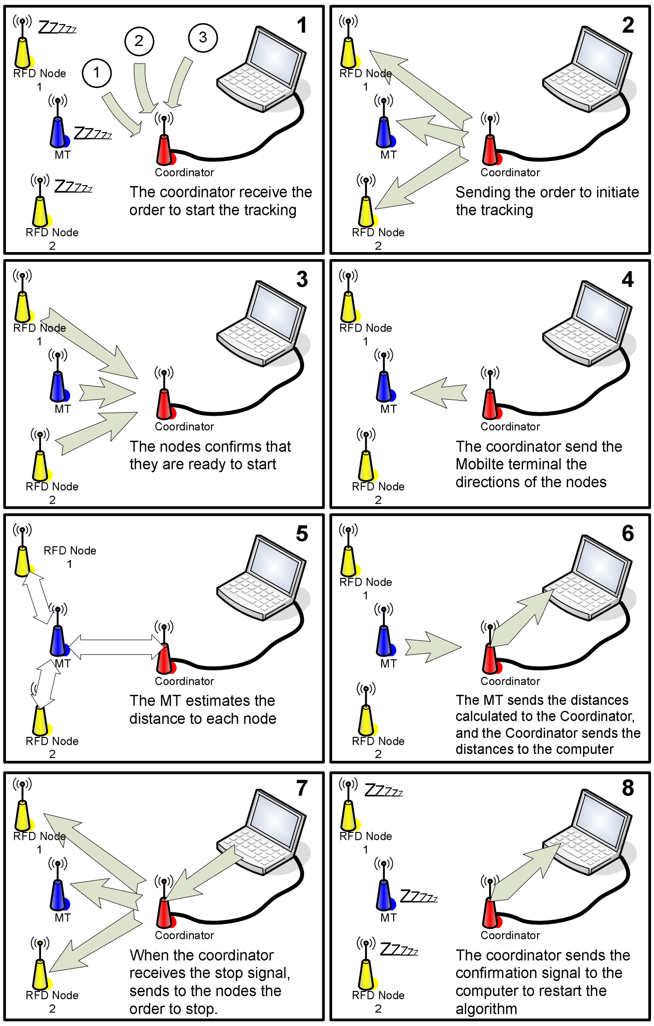

Figure 15: Complete Algorithm

Image 16: MT (Mobile Terminal) Zigbee Tracking.

VIII. Conclusions

In this study, a tracking system implemented on a ZigBee network has been performed. The results prove that applying the measure of RTT to this kind of network can achieve accurate ranging capabilities: in 90% of the cases, a maximum error of +/- 0.25 meter is obtained.

On the other hand, in this work, we developed a prototype using Kalman Filter in order to track the object in real time. We found that, regarding Kalman Filter: it is able to obtain good results at low speeds (pedestrian velocity), that is typical in indoor environments.

Acknowledgments

The current project has been financed and supported by Miguel Hernández University.

Bibliography:

[1] Sherratt, R.S. Makino, S. *"Numerical precision requirements on the multiband ultra-wideband system for practical consumer electronic devices"* IEEE Transactions on Consumer Electronics, Vol. 51, Issue 2, pp: 386- 392, May 2005.

[2] Yasuhiro Fukuju et al., "*DOLPHIN: An Autonomous Indoor Positioning System in Ubiquitous Computing Environment*", IEEE Workshop on Software Technologies For Future Embedded Systems, 2003.

[3] Miguel Rodríguez et al., "*Blueps: Sistema de localización en interiores utilizando Bluetooth*", URSI 2005.

[4] M. Ciurana et al., "*A ranging system with IEEE 802.11 data frames*", Radio and Wireless Symposium, IEEE, january 2007.

[5] IEEE 802.15.4, "*Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*", February 2003.

[6] L.F. Da Cruz, et al. "An evaluation of RSSI based indoor localization systems in wireless sensor networks", SBAI, September 2009.

[7] *CC2420DBK User's Manual: Demonstration Board Kit*, rev. 1.3, Chipcon Products from Texas Instruments, 2004

[8] F.Izquierdo, et al. "*Performance Evalution of a TOA-based trilateration method to locate terminals in WLAN*", 1st International Symposium on Wireless Pervasive Computing, IEEE, 2006, pp.217-222.

[9] P.Corral, et al. "*Distance Estimation System based on Zigbee*" IEEE Computational Science and Engineering 2008.

[10] J. Hightower, G. Borriello, "Location Systems for Ubiquitous Computing", IEEE Computer, 34(8):57-66, 2001.

[11] I. Atzeni, "Simulation of an Indoor Tracking System Based on 802.15.4", Master's Thesis, Università degli Studi di Cagliari, 2009

[12] Juan M. Camús, P. Corral, Alberto Congiu, Italo Atzeni, "*Simulación de Segumiento Zigbee en Tiempo Real*", XXV Symposium Nacional de la Unión Científica Internacional de Radio, Bilbao (Spain), September 2010.