

Indoor Position System based on BitCloud Stack for Ambient Living and Smart Buildings

A. Verónica Medina, Isabel M^a Gómez, Manuel Romera, José A. Gómez and Enrique Dorrozero

Escuela Técnica Superior de Ingeniería Informática
Dpto. de Tecnología Electrónica
Avda. Reina Mercedes s/n, 41014 Sevilla, Spain
{vmedina, igomez}@us.es, manromera@gmail.com, jgomezdte@us.es,
enriquedz@dte.us.es

Abstract. Nowadays there is a research on ambient living and smart building applications to make people life more comfortable. This kind of applications, in general, requires the position where a person is in a building to act according it and to any physical parameters that are measured from the person, to determine, for instance, if the air condition has to be switched on or off, the room temperature has to be put up or down or the music has to be changed from classical to pop, or vice versa. This paper presents an indoor position solution based on BitCloud Stack, a full-featured, second generation embedded software stack from Atmel.

Keywords: IEEE 802.15.4, RSSI, Centroid, Indoor position, ZigBee, WSN

1 Introduction

WSNs (Wireless sensor Network) are present in many applications. They are used in Ambient Living [1-4] or Smart building [5-9] researching fields for solving data acquisition process, depending on its applications, it can be used ambient or user sensors and actuators for making decisions. The knowledge of the subject's position it is very useful in these kind of systems because depending on it the decisions to be made are different, this paper present an indoor position solution for doing so.

In Section 2 an overview of different alternatives to calculate the indoor position is presented and paper goal is explained. System overview is shown in Section 3. Implementation and data collection issue is explained in section 4 and the system test is exposed in section 5. Conclusions are established in section 6.

2 Positioning Determination

As stated in [10,11], a number of indoor location tracking systems have been proposed in the literature, based on RF signals, ultrasound, infrared, or some combination of modalities. Given a model of radio signal propagation in a building or

other environment, received signal strength can be used to estimate the distance from a transmitter to a receiver, and thereby triangulate the position of a mobile node. However, this approach requires detailed models of RF propagation and does not account for variations in receiver sensitivity and orientation.

An alternative approach is to use empirical measurements of received radio signals, known as RSSI, Receiver Signal Strength Indicator), to estimate location. By recording a database of radio “signatures” along with their known locations, a mobile node can estimate its position by acquiring a signature and comparing it to the known signatures in the database. A weighting scheme can be used to estimate location when multiple signatures are close to the acquired signature.

All of these systems require that the signature database are collected manually prior to system installation, and rely on a central server (or the user’s mobile node) to perform the location calculation. Several systems have demonstrated the viability of this approach, one of those is MoteTrack [10,11].

MoteTrack’s basic location estimation uses a signature based approach that is largely similar to RADAR [12] that obtains a 75th percentile location error of just under 5 m, but decreased the location error by 1/3 in MoteTrack.

MoteTrack is implemented on the Mica2 mote platform using the TinyOS operating system [13]. They chose this platform because it is designed for low-power operation, is relatively small, and can be deployed unobtrusively in an indoor environment.

Our goal is not to improve upon the accuracy of MoteTrack signature-based localization scheme, but rather to use a similar scheme but using other motes, Meshnetics’ one [14], that uses different RCB (MCU and transceiver) and, also, different software, i. e., the BitCloud Stack, a ZigBee PRO certified platform.

3 System Overview

In this section we give an overview of the system, shown in Figure 1, that is based on low-power, embedded wireless devices, Meshnetics’ sensor “motes”. The advantages of this platform over other motes is that they are equipped with extra sensors and others could be easily connected to them if the application requires it, so for prototyping these motes work quite well. Other advantage of this mote is that the supplier has developed the ZigBee RFC4 stack architecture [15] in a software pack called BitCloud Stack.

In our system, a building or other area is populated with a number of Meshnetics motes acting as fixed nodes, one of them acting as coordinator, C. Fixed nodes send to C periodic beacon messages, beacon 2, which consist of an n-tuple of the format {MobileID, RSSI}, where n is the number of mobile nodes. MobileID is a unique identifier of a mobile node, and RSSI the value received from the last beacon message, beacon 1, received from that mobile node in an specific fixed node.

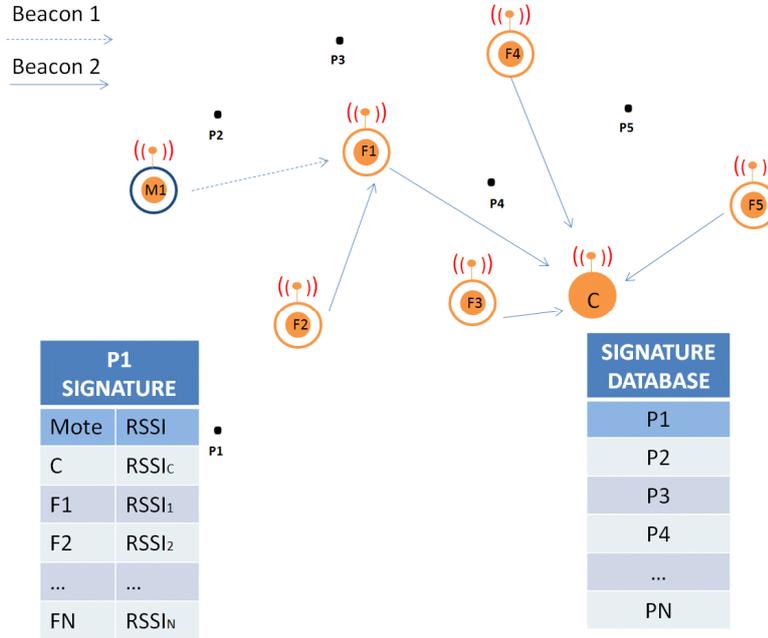


Fig. 1. System Overview. M1 is a mobile node, F1-F5 are fixed nodes, and C is the coordinator, also a fixed node. M1 periodically sends a beacon message, beacon 1, to inform the others node that is present, all fixed node that receives it save the RSSI of that message in a table. Fixed node periodically sends a message to C, beacon 2, to inform about the RSSI that they receive from mobiles node, M in this case.

The location estimation problem consists of a two-phase process: an offline collection of reference signatures followed by online location estimation. As in other signature-based systems, the reference signature database is acquired manually by a user with a mobile node and a P_c connected to C. Each reference signature, shown as black dots in Figure 1, consists of a set of signature tuples of the form $\{\text{sourceID}, \text{meanRSSI}\}$, where sourceID is the fixed node ID and meanRSSI is the mean RSSI of a set of beacon messages received over some time interval. Each signature is mapped to a known location by the user acquiring the signature database (P1-P5 in Figure 1).

3.1 Location estimation

Given a mobile node's received signature s , received from fixed nodes, and the reference signature set \mathbf{R} , the mobile node's location can be estimated as follows. The first step is to compute the signature distances, from s to each reference signature $r_i \in \mathbf{R}$. We employ the Manhattan distance metric,

$$M(r, s) = \sum_{t \in T} |RSSI(t)r - RSSI(t)s| \quad (1)$$

where T is the set of signature tuples presented in both signature, $RSSI(i)r$ is the RSSI value in the signature appearing in signature r , and $RSSI(i)s$ is the RSSI value in the signature appearing in signature s .

Given the set of signature distances, the location of a mobile node can be calculated in several ways. We consider the centroid of the set of signatures within some ratio of the nearest reference signature. Given a signature s , a set of reference signatures R , and the nearest signature $r^* = \operatorname{argmin}_{r \in R} M(r, s)$, we select all reference signatures $r \in R$ that satisfy

$$\frac{M(r, s)}{M(r^*, s)} < c \quad (2)$$

for some constant c , empirically-determined. The geographic centroid of the locations of this subset of reference signatures is then taken as the mobile node's position. Small values of c work well, generally between 1.1 to 1.2.

4 Implementation and Data Collection

Our system is implemented in Meshnetics' motes, Meshbean development board. We have used this mote because they have leds, buttons, additional sensors and can easily be connected other sensors that can be used for the purpose applications of this indoor position system, ambient living and smart buildings, so for prototyping works quite well. They also have a USART accessible by a USB connector, so a Pc can be connected via a USB port, emulating it a COM port, for both programming and receiving information, in our case beacons and sensor values.

A Meshnetics' mote is shown in Figure 2, in this case, it has an integrated PCB antenna, but we has used others that aren't, this affects only the range of coverage. This mote have a MCU wireless, called ZigBit, a compact 802.15.4/ZigBee modules featuring record-breaking range performance and exceptional ease of integration. It integrates both the ATmega1281 microcontroller [16] and AT86RF212 transceiver [17] of ATMEL so the AVR tools [18] are necessary for programming purposes.

The BitCloud Stack has been used for software, a full-featured, next generation embedded software stack from Atmel. The stack provides a firmware development platform for reliable, scalable, and secure wireless applications running on Atmel hardware kits. BitCloud is designed to support a broad ecosystem of user designed applications addressing diverse requirements and enabling a full spectrum of software customization. Primary application domains include home automation, commercial building automation, automated meter reading, asset tracking, and industrial automation.

BitCloud is fully compliant with ZigBee® PRO and ZigBee standards for wireless sensing and control. It provides an augmented set of APIs which, while maintaining

compliance with the standard, offer extended functionality designed with developer's convenience and ease-of-use in mind. As seasoned ZigBee technology experts, Atmel created BitCloud to dramatically lower the developer learning curve, factor out the unnecessary complexity and expose as much power of the underlying hardware platform as possible.

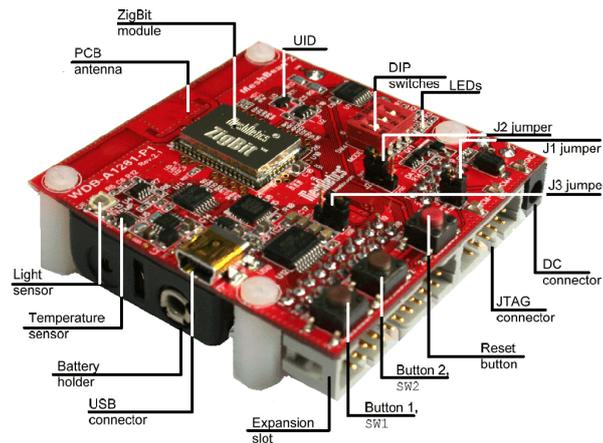


Fig. 2. Meshbean development board.

In ZigBee there are three kinds of devices, each one having its own purpose:

- Coordinator (C): A full function device (FFD) that it is in charge of creating the PAN (Personal Area Network) and typically is the point of the WSN to acquire all sensors information from all the other motes to be shown in a computer. The icon uses to represent this device is a filled circle, Figure 1 shown one.

- Router (R): A FFD that it is in charge of routing when the range of coverage requires this capability, so it is possible to have dynamic topologies. The icon uses to represent this device is a small filled circle inside a circle, Figure 1 shown six.

- End device (ED): A reduced function device (RFD) that is always slept (to reduce consumption) and only wakes up to do a specific task, for instance, to send sensor information to the WSN, typically directed toward C. The icon use is a not filled circle.

So a ZigBee WSN is composed of one C, many EDs and many Rs. Each kind of devices can receive what the other transmit if they are in the same range of coverage, because the transmission media is share by all one, but not all the information receive is processed (the explanation of why this is that way is out of the scope of this paper).

Atmel also provides a demo, called WSN demo, to help programmer task. The demo is prepared to create a dynamic ZigBee topology, controlled by the ZigBee's Network layer. There is no positioning function in this demo so we have to modify the WSN Demo source code to do so.

4.1 WSN demo modifications

As explained in section 3, to determine position, we require two kinds of beacons, beacon1 and beacon2. Beacon 1 is used to inform other devices that a mobile mote is present and beacon2 is used to inform C the RSSI value that a fixed mote receives from a mobile one for location estimation. To send both beacons in WSN we have to use the information saved in a table at the network layer called neighbor table. This table registers all the FFD, this is motes that are C or R, that are in the range of coverage of a determine mote and for each one it registers the RSSI value of the received signal from that mote. Periodically, a FFD device sends a MAC layer message to inform other that is in the WSN, so that message is used by neighbor motes to measure the RSSI value of the received signal and to save it in their own neighbor table. So beacon 1 is sent automatically by the protocol stack. As only FFD sends this kind of message the mobile motes have to be R, as shown in Figure 1.

The WSN demo code has been changed to send periodically beacon 2 messages, for doing so, a search has to be done in the neighbor table to find out if the mobile mote is in its range of coverage, if so, the beacon 2 is sent to C with the information required as explained in section 3. As neighbor table is only in FFD, fixed motes have also to be R.

4.2 Deployment

We have deployed our system over half floor of our Department Area, measuring roughly 225 m². The first step was to test how the different kind of materials affect the RSSI value, taking into account that results, and that a mote could cover an area of 4-5 meters, we determine that with a number of 7 fixed motes was enough (Figure 3).

As shown in Figure 3, there are 11 rooms (room number G1.25 is the telecommunications room and doesn't belong to our Department), being six of them double size than the other ones and counting the halls as room, numbered from 1 to 3 (left to right). We determined that with two signature points for small rooms and four signature points for big rooms were enough. Halls were more troubles because signals propagated easily so we empirically determined to have a point every 3 meters. In total, we have 46 signature points each one places as shown in Figure 4, red dots represent fixed mote, being C the mote called "mota 0" and black dots represent signature points.

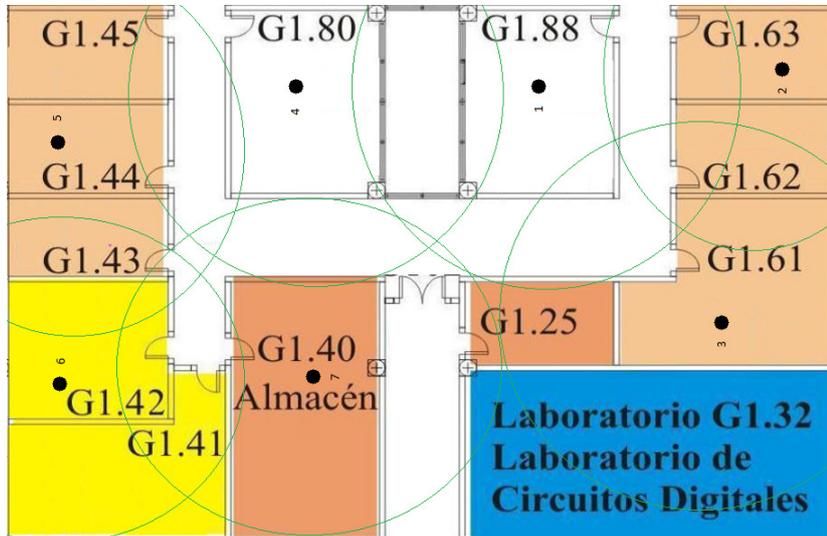


Fig. 3. Position of fixed notes in the Department Area. Black dots represent fixed notes and they are the center of the circles that represents the range of coverage of each one.

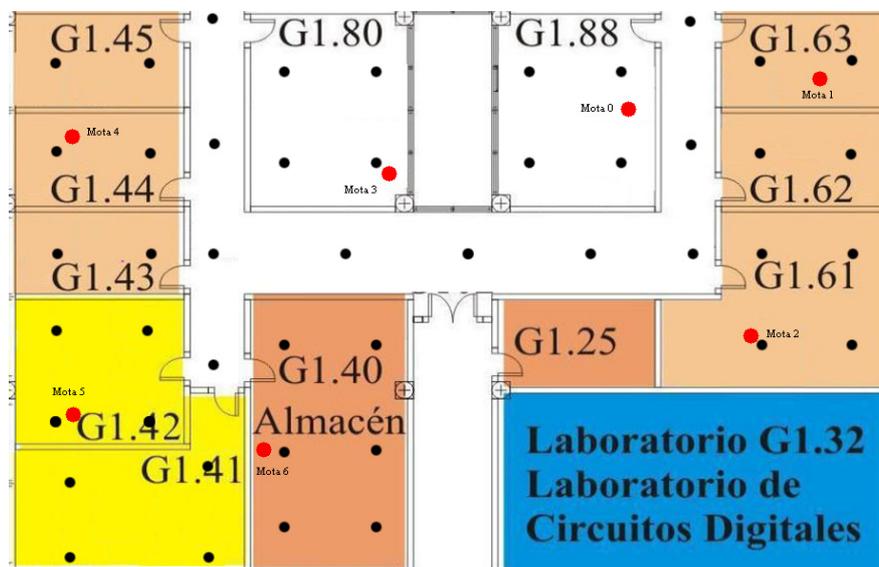


Fig. 4. Signature points.

Once all the signature points were determined, we made the signature point database, it was an off-line process where we run a Windows console application that took signature information for a period of time and saved it in a file, so using this file

other application could calculate the mean value to be saved in the signature point database.

Once the signature point database was created, the system was ready to determine the online position of a mobile mote. Figure 5 shows the computer software prototype used in the online phase, as it is shown, we not only determine the position of more than one mobile mote, number 1 to 4, but also the value of two of the sensors that that motes have, light, called "luz" in Figure 5, and temperature.

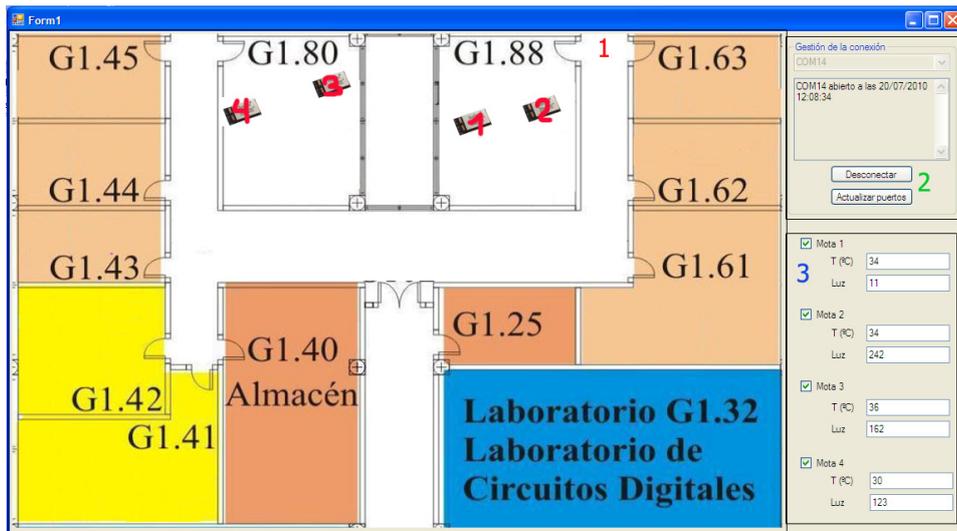


Fig. 5. Positioning determination. Phase on-line.

5 System Test

We have tested the system in order to know if it is able to determine if a mobile mote is placed in a room, i.e., it doesn't matter exactly where it is inside de room, so the precision required is not very high. This has been this way, because the kind of applications for whom our indoor position solution is going to be used don't require more precision.

We decided to test 30 points in where we detected signal problems (RSSI values that changed frequently) to check how the positioning algorithm works. It is determined that the algorithm presents the right position in 23 points but in 7 points, it makes a bad position determination, Figure 6 shows that points, making numbers our precision is about 77%. Further work is going to be done to optimize the algorithm in order to have more precision.

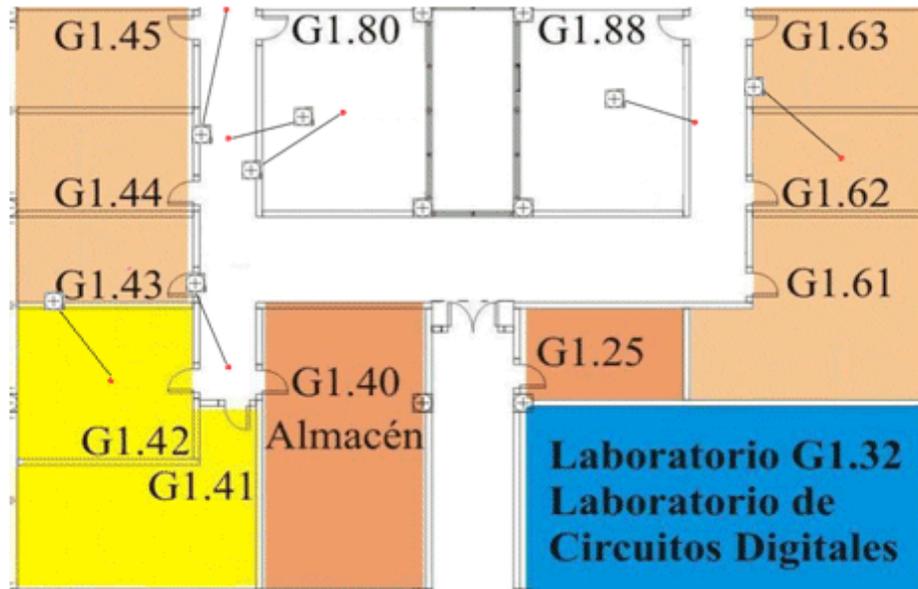


Fig. 6. Points with troubles. Dots represents the actual position, \boxplus represents the determined position.

6 Conclusions

In this paper we have presented an indoor position system based on BitCloud Stack over a WSN. Although the results obtained are good we have to improve the algorithm in order to be more precise. Also, some research has to be done in order to make mobile mote ED (RFD), because now it is R (FFD), so the power consumption is very high and it should be a problem because mobile node is battery power. There are some parameters of BitCloud Stack that we aren't able to modify, at least we haven't been able to, so a deep study of this stack has to be done to check if we can do it and, if not, keep on researching how we could work out. A first solution is to work over the Mac layer using OpenMac software instead of over the application layer of the BitCloud Stack (ZigBee Stack). This software is also provided by Atmel [19].

As presented, the system could be easily applied to any applications of Ambient Living or Smart Building adding the specific sensors to the mote.

Acknowledgements

This work has been carried out within the framework of two research programs: (P08-TIC-3631) – Multimodal Wireless interface (IMI) funded by the Regional Government of Andalusia and Efficient and Health Intelligent Technologies Oriented

to Health and comfort in Interior Environments (TECNO-CAI) approved project at the fifth call of CENIT program by the Innovation Science Ministry of Spain (CDTI and Ingenio 2010 Program).

References

1. Hristova, A.; Bernardos, A.M.; Casar, J.R, Context-aware services for ambient assisted living: A case-study. Applied Sciences on Biomedical and Communication Technologies, 2008. ISABEL '08. First International Symposium on Digital Object Identifier: 10.1109/ISABEL.2008.4712593 Publication Year: 2008 , Page(s): 1 - 5.
2. Figueiredo, C.P.; Gama, O.S.; Pereira, C.M.; Mendes, P.M.; Silva, S.; Domingues, L.; Hoffmann, K.-P, Autonomy Suitability of Wireless Modules for Ambient Assisted Living Applications: WiFi, ZigBee, and Proprietary Devices. Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on Digital Object Identifier: 10.1109/SENSORCOMM.2010.34. Publication Year: 2010 , Page(s): 169 - 172.
3. Hong Sun; De Florio, V.; Ning Gui; Blondia, C., Towards Building Virtual Community for Ambient Assisted Living. Parallel, Distributed and Network-Based Processing, 2008. PDP 2008. 16th Euromicro Conference on Digital Object Identifier: 10.1109/PDP.2008.28 Publication Year: 2008 , Page(s): 556 - 561.
4. Sun, H.; De Florio, V.; Gui, N.; Blondia, C., PRomises and Challenges of Ambient Assisted Living Systems. Information Technology: New Generations, 2009. ITNG '09. Sixth International Conference on Digital Object Identifier: 10.1109/ITNG, 2009.169 Publication Year: 2009 , Page(s): 1201 - 1207.
5. Martin, H.; Bernardos, A.M.; Bergesio, L.; Tarrío, P., Analysis of key aspects to manage wireless sensor networks in ambient assisted living environments. Applied Sciences in Biomedical and Communication Technologies, 2009. ISABEL 2009. 2nd International Symposium on Digital Object Identifier: 10.1109/ISABEL.2009.5373643 Publication Year: 2009 , Page(s): 1 - 8.
6. Dietrich, D.; Bruckner, D.; Zucker, G.; Palensky, P., Communication and Computation in Buildings: A Short Introduction and Overview. Volume: 57 , Issue: 11 Digital Object Identifier: 10.1109/TIE.2010.2046570 Publication Year: 2010 .
7. Chen, Po-Wei; Ou, Kuang-Shun; Chen, Kuo-Shen, IR indoor localization and wireless transmission for motion control in smart building applications based on Wiimote technology. SICE Annual Conference 2010, Proceedings of Publication Year: 2010 , Page(s): 1781 - 1785.
8. Han Chen; Chou, P.; Duri, S.; Hui Lei; Reason, J., The Design and Implementation of a Smart Building Control System Digital Object Identifier: 10.1109/ICEBE.2009.42 Publication Year: 2009 , Page(s): 255 - 262.
9. Snoonian, D , Smart buildings. Spectrum, IEEE Volume: 40 , Issue: 8 Digital Object Identifier: 10.1109/MSPEC.2003.1222043 Publication Year: 2003 , Page(s): 18 - 23.
10. Konrad Lorincz and Matt Welsh, MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking. To Appear in Springer Personal and Ubiquitous Computing, Special Issue on Location and Context-Awareness, October 2006. ISSN: 1617-4909 (Print) 1617-4917 (Online), DOI: 10.1007/s00779-006-0095-2.
11. Konrad Lorincz and Matt Welsh, MoteTrack: A Robust, Decentralized Approach to RF-Based Location Tracking. In Proceedings of the International Workshop on Location and Context-Awareness (LoCA 2005) at Pervasive 2005, May 2005. [PDF]
12. Bahl P, Padmanabhan VN (2000) RADAR: an in-building RF-based user location and tracking system. In: INFOCOM, pp 775–784.

13. Hill, J., Szewczyk, R., Woo, A., Hollar, S., Culler, D.E., Pister, K.S.J.: System Architecture Directions for Networked Sensors. In: ASPLOS 2000. (2000) 93–104.
14. <http://www.meshnetics.com/>
15. ZigBee RF4CE Specification. Version 1.00. ZigBee Document 094945r00ZB, March 17th, 2009
16. http://www.atmel.com/dyn/resources/prod_documents/doc2549.PDF.
17. http://www.atmel.com/dyn/resources/prod_documents/doc8168.pdf.
18. http://www.atmel.com/dyn/products/tools_card.asp?tool_id=2725.
19. http://www.atmel.com/dyn/products/tools_card.asp?tool_id=4675.