

Intelligent Localization Algorithm for Temperature Monitoring using Wireless Sensor Networks

Mohan Kumar J

Asst. Prof – Selection Grade
School of Information Sciences
Manipal University

Manas Jyoti Sarmah

Post Graduate
School of Information Sciences
Manipal University

P.R. Venketeswaran, PhD
Senior Developer Engineer (C&I)
Welding Research Institute,
BHEL Tiruchirappalli

Gopalakrishna N. Kini, PhD

Professor, Computer Science,
Manipal Institute of Technology
Manipal University

Sundaresan C

Asst. Prof – Selection Grade
School of Information Sciences
Manipal University

Chaitanya C V S

Assistant Professor
School of Information Sciences
Manipal University

ABSTRACT

Wireless Sensor Networks (WSN) are increasingly used in monitoring applications like underground water level, building monitoring, environment monitoring etc. The success of these networks depends largely on the ability of the algorithms to run consistently and over a longer duration with least energy consumption. Location stamping is a very important factor in all wireless networks irrespective of whether it is device based or sensor based. It is important to know the location stamping in WSN where the sensor node has sensed the data. In this paper, a novel and experimental implementation of localization algorithm for temperature monitoring system is successfully attempted. The realization of the algorithm is through a combination of Arduino and Xbee, a Zigbee based wireless communication hardware module. The localization is carried out in an indoor environment of a seven storied building with a built area of 237.5 sq. feet with a centralized coordinator approach. Fuzzy logic is used for categorizing the Received Signal Strength Indicator (RSSI) from the coordinator. The algorithm helps to track and identify the location and the individual nodes and also the parameter value recorded over a period of time consistently.

Keywords - Localization algorithm, RSSI, Coordinator node, embedded logic

1. INTRODUCTION

Wireless Sensor Networks play a prominent role in the field of Ubiquitous computing. This includes the application of embedded processors applied into a wide variety of applications typically characterized by longer geographical and temporal spaces. A very eventful example is the smartphone. To turn this ubiquitous computing into a useful application is the goal of Wireless Sensor Network research. Since the distribution of events and items are in a wider area, aspects deciding the successful outcome will include Deployment, Routing, Power Consumption, Localization and Energy Harvesting identified with the network. These variables have a tradeoff between them and have interdependent and complex behaviour. Localization is the ability of the network to identify the location of the node successfully and since hence this is considered as the variable

of primary importance and other parameters are optimized in relation with the performance of localization algorithm. The importance of the localization algorithm in Wireless sensor networks is reflected in many research works, particularly in [1] – [4]. The research [5] – [7] is categorized into two broad areas of node distribution as Centralized and Distributed. If the algorithm runs on a coordinator node, it is called as Centralized and distributed if it runs on leaf node. It is also possible to classify Localization into range based and range free methods. Typically, range based methods are hardware intensive with higher accuracy. GPS could be used for localization but with limited results indoors. In a Wireless Sensor Network (WSN), thousands of nodes (motes, in WSN parlance) are used and inclusion of GPS will multiply cost severely [8]. As an alternative, range free methods could be tried for optimal solutions albeit with lesser accuracy. Some common applications with range free methods are listed below:

- i. WSN application in healthcare research is widely analyzed in [9]. Monitoring of heart rate during sports training is considered by the author for experimentation using Arduino-nano board is along with nRF24API and Xbee module to implement the WSN. The system can also allow multiple athletes to be monitored simultaneously.
- ii. WSN based on Xbee Zb module performance is analyzed in [10]. The performance metrics evaluated are RSSI, networks throughput, packet delay, mesh routing recovery time and energy consumption in indoor environment. The scenarios of direct transmission from End devices to the coordinator and through the routers as message passing relays is realized in this paper. It shows that compared to the theoretical value of 250 kbps only a 5.4 Kbps speed is achieved. It is concluded that using cyclic sleep mode, power consumption can be reduced.
- iii. An experimental Embedded WSN setup is developed for monitoring Paddy Field Environment in [11]. The pH Sensor, PIR sensor, Humidity Sensor, Temperature Sensor are interfaced with an LPC2148 an ARM based Microcontroller. The paper describes a peer to peer communication of Zigbee network. Then the gateway sends the data to user through the GSM module.

iv. Embedded WSN is also formed for monitoring Temperature in Buildings [12]. The nodes are requested by polling for data. The control is done through web interface using developed using MYSQL and HTML.

In the experimental analysis of these works positioning of nodes may play a vital role. In this paper location based Temperature monitoring system is proposed using Embedded Fuzzy Logic Library. The objective is to achieve better localization with lesser power consumption so that the life of the nodes is extended for a larger period. Embedded Fuzzy Logic Library [13] is being used for the implementation environment.

2. SENSOR MOTE ARCHITECTURE

Sensor mote forms the most important part of the Wireless Sensor Network architecture. The Wireless Sensor Network can be connected to the Server through a gateway where a co-coordinator node is connected. This data can be disseminated to the real time world through internet. The architecture of a Wireless Sensor Architecture is shown in Figure.1.

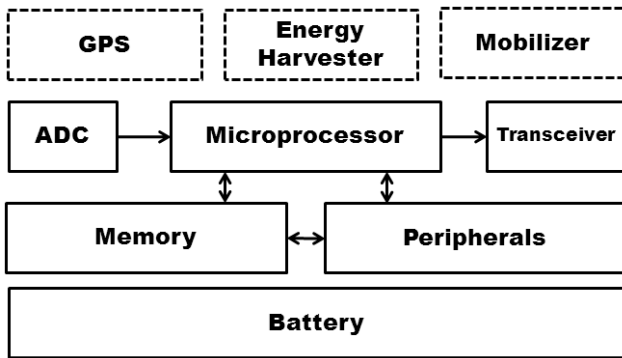


Fig 1: WSN Architecture

A sensor mote comprises of many hardware devices as in Fig. 2. It has a processor with ADC for signal acquisition, processing and generation of signal. Other nodes get to transmit or receive the signals through the ADC/DAC route and processors. The energy to sustain the operations is through a battery backup. Hence, it becomes important to spend the limited energy frugally so that components including GPS, if any, can be operated on the platform for a longer time. Under normal circumstances, GPS and energy harvesting blocks are optional and is based on the requirements of the application. The transceiver radio hardware is based on the WPAN protocol standard. Some of the standards that could be used are Zigbee, Bluetooth and UWB. Zigbee protocol is known for low power consumption and works with 2.4GHz ISM band. It is based on IEEE 802.12.4 where MAC and PHY have been defined. In Fig.2, the network has several motes and can have different functionalities associated with it. The motes in the network can also be called as Full Function Devices (FFD) and Reduced Function Devices (RFD) [14] accordingly. Sensor node comprises of many hardware devices as shown in Figure 1. Generally it consists of a Microprocessor or a Microcontroller with ADC for acquiring the environmental analog signals through sensors. The acquired signal is passed to an ADC to convert into digital value.

A simple wireless sensor network is shown in Figure 2. The Wireless Sensor Network can be connected to the Server through a gateway where a co-ordinator node is connected. This data can be disseminated to the real time world through

internet. This opens up multiple opportunities to connect the network with the external world for communication. This feature is exploited for communication between various devices in the network architecture in this paper.

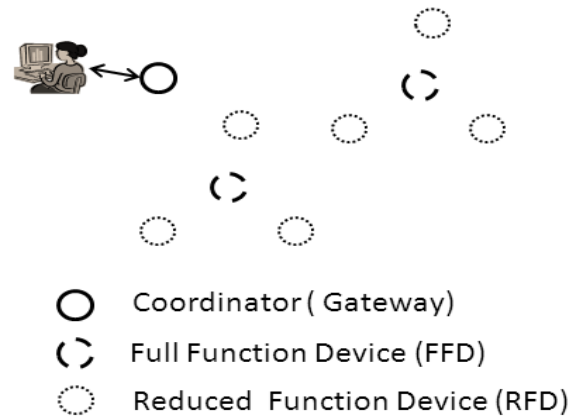


Fig 2: Wireless Sensor Network

3. EXPERIMENTAL SETUP

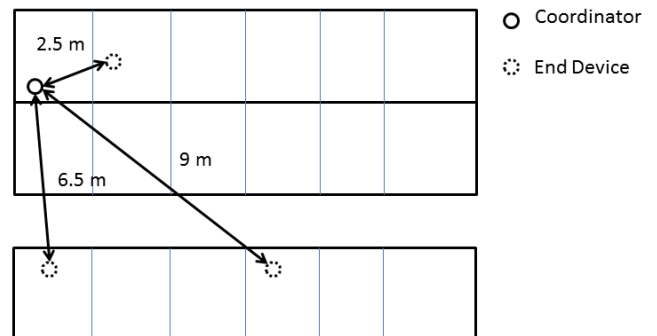


Fig 3: Experimental Setup with Coordinator and end device placed in different rooms

The experiment is conducted in the campus building of School of Information Sciences, Manipal, Karnataka, India. Figure 3 shows the rooms where the sensor nodes are deployed with the coordinator node. The experimental setup consists of nodes distributed in geographical locations identified as end devices. The end devices communicate with the coordinator. For the implementation, we consider the distribution of one coordinator device and three end devices with live feed from the sensor motes. The experimental setup has the following components:

3.1 Coordinator

The coordinator is the initiator for the network. It reads the remotely sensed sensor data and displays in the PC screen. The XBee module in the Coordinator side is configured with ZNET 2.5 COORDINATOR API firmware (an advanced version of IEEE 802.15.4 firmware used in Series-1 XBees), using X-CTU application. The DH and DL fields of the XBee module are set as 0 to put the coordinator in broadcast mode. The AP field is set as 2 to enable API level 2. As the

Coordinator should always be awake, the sleep mode is disabled here. After programming the Arduino Mega ADK with the *Coordinator Side Program*, the module is powered on and its DIN and DOUT pins are connected to the Tx3 and Rx3 pins of the Mega ADK board respectively. The setup of the coordinator is shown in Figure 4.

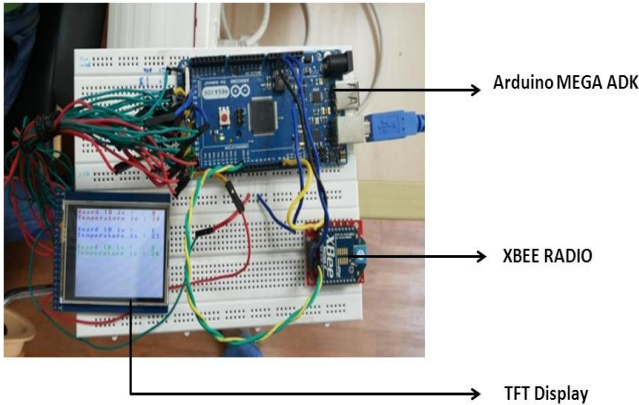


Fig 4: Coordinator Setup

3.2 End Device

Each End Device forms a sensor mote which reads the surrounding temperature of its location and sends the values to the coordinator. The LM35 sensor (used for temperature measurement) is interfaced to an Arduino Duemilanove board which in turn is connected to the XBee module configured with ZNET 2.5 ROUTER/END DEVICE API firmware using X-CTU application. The DL and DH fields of this XBee module is set to contain the DL and DH fields of the Coordinator XBee module respectively and the module is put to cyclic sleep mode, means it waked every prementioned time and goes to sleep otherwise, to save the battery power. The AP field in this case is too set as 2 to enable API level 2. The Duemilanove board is programmed with The *Router Side Program* and its Tx and Rx pins are connected to the DIN and DOUT pins of the XBee module respectively. The end device hardware setup is shown in Figure 5.

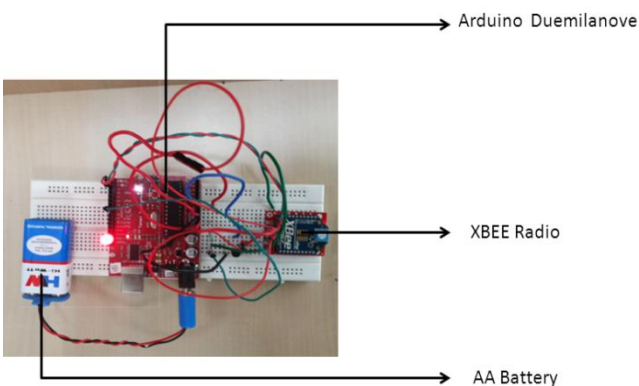


Fig 5: End Device Setup

3.3 Coordinator and End Device Programming

The Arduino Mega ADK board in the Coordinator side is programmed to contain the coordinator side program. The board is programmed using Arduino IDE 1.0.4 and the program includes the <XBee.h> library by Andrew Rapp and the EFL (Embedded Fuzzy Logic Libraries) by Robotic Research Group [13]. This program dictates the XBee to scan the network every 10 milliseconds for any valid data. If the data is available it checks the node ID of the source node which generated the data. Moreover the RSSI pin of the XBee module is connected to one of the digital pins of the Mega ADK board. The program also reads the values from the RSSI pin every 30 milliseconds, stores it in a variable and implements a Fuzzy Logic using the EFL. Triangular Pertinence function [13] is used based on the trial and error measurement with other pertinence this one suits well for the experiment. Three Fuzzy Input Sets, namely sensor1, sensor2 and sensor3 with values (0,15,15,30), (29,39,39,55) and (51,80,80,90) were created as shown in Figure 6 to represent the range of RSSI count values for each End node.

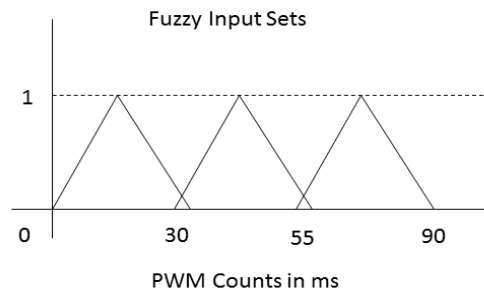


Fig 6: Fuzzy Input Sets

Four corresponding Fuzzy Output Sets namely dist1, dist2 and dist3 with values (0,2,2,4), (4,6,6,8) and (8,9,9,10) respectively are created as shown in Figure 7 to represent the distance assigned to Fuzzy Linguistic Variables.

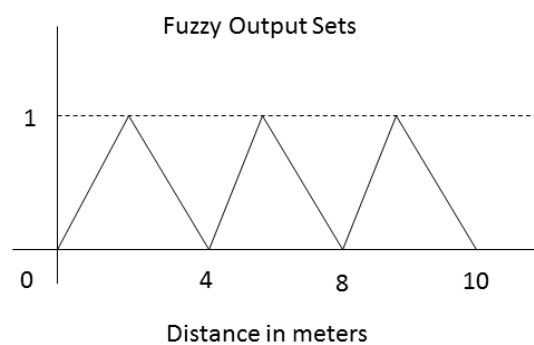


Fig 7: Fuzzy Output Sets

Fuzzy Membership Functions were created to which these input and output assigned Fuzzy Linguistic Variables were added. These assignments are to be used later for Fuzzification and Defuzzification process [21]. A rule base maps the inputs with the output distance values. Once the exact distance is obtained by the program, the room number where the End Devices are placed can be obtained along with

the temperature of the room in degree Celsius. Gobetwino [23] is used to log the sensor data received by the Coordinator node to three different log files. Gobetwino is a “generic proxy” for Arduino that runs on PC with Windows OS only and performs something on behalf of Arduino, in the host PC that Arduino cannot do by itself. It goes in between Arduino and PC and thus the name comes. These text files are later on imported as Excel file to do graphical analysis. The final data gathered by the Coordinator is displayed both on the Serial Console and TFT LCD screen [15]. In each of the sensor mote, an Arduino Duemilanove board is programmed with the End Device Program. This program assigns a unique node ID to each of the End Device and converts the Analog value of the sensor to a temperature value in degree Celsius. Both the node ID and the temperature reading are then transmitted over to the coordinator by the XBee module. The XBee on these nodes are put on Sleep Mode 1 (Pin Hibernate Sleep Mode).

3.4 Embedded Fuzzy Logic Library

Embedded Fuzzy Logic Library is a software library for Fuzzy Logic for embedded systems. Developed by Robotic Research Group (RRG) at the State University of Piauí (UESPI-Teresina) the eFLL (Embedded Fuzzy Logic Library) library is a versatile, lightweight and efficient option to work with Fuzzy Logic in embedded systems[13]. This is used at the co-ordinator programming. All the nodes are polled for the RSSI value. The RSSI value is given as an input to the Fuzzy System, based on which the membership function are created. The output is the distance of the nodes in meters. These are mentioned in through the APIs of the embedded Fuzzy Logic Library (eFLL).

4. HARDWARE COMPONENTS

The hardware parts of the experimentation include XBEE module for wireless communication, Arduino Board for processing, Temperature Sensor for sensing the data, TFT display for displaying node ID and temperature measured at the coordinator side.

4.1 XBee Module

The XBEE module is a Zigbee Protocol based transceiver module which is used for communication in this experimentation. It works with 3.3V battery supply [17]-[18]. The Zigbee Protocol suits well for wireless sensor network due to its low power consumption [20]. Zigbee supports three different types of network topology, but in this work only star topology is considered. As discussed in section 2, Zigbee devices can be configured as FFD or RFD. In practice, this is done using X-CTU software. Generally FFD is supported by a good power and has more capabilities than the RFD. The RFD job is to sense the data and pass it to FFD. The XBee module pin diagram is shown in Figure 8. The pins used for the interfacing with the Arduino boards are RX, TX for the communication with Arduino. To measure the RSSI value the Pin 6 is used. It is an output pin gives RSSI value in terms of PWM count. The XBEE module can work in two modes, AT and API mode. The AT (Advanced Telephony) is likely to be a command mode and API (Application Programming Interface Mode) which is more of packet based/frame based approach. [18]

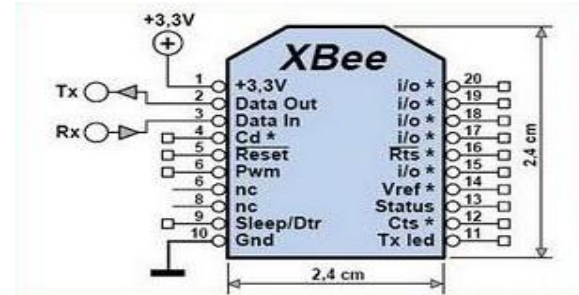


Fig 8: The XBee Series 2 module Pin Diagram

4.2 Arduino

Arduino is an Open Source Hardware Platform which can be used for development of any embedded system applications [22]. Arduino comes with ATMEGA or CORTEX microcontroller as its central element. Arduino IDE is provided for developing applications on this platform. In this project, two Arduino types are used, Arduino Duemilanove and Arduino Mega ADK. Arduino Duemilanove is used for RFD and Arduino Mega ADK is used for building the coordinator. The intention is to expand the project for Android based platform. Also the Embedded Fuzzy library required more memory space. The comparison of features of the Arduino Duemilanove and Arduino Mega ADK is shown in Table 1. The features thus provide justification for the choice of the different boards for coordinator and end device. The XBee module is interfaced with the both the coordinator side and end Arduino devices. The connection is basically a serial communication where RX and TX are connected inversely. The PWM pin of the XBee module is also connected to the Arduino device to get the RSSI value of each node.

Table 1. Comparison of Mega ADK and Deumilanove

	Mega ADK	Duemilanove
Processor	ATMega256	ATMega168
Frequency	16 MHz	16 MHz
Flash Memory	256 Kb	16/32 Kb
EEPROM	4 Kb	0.5/1 Kb
SDRAM	8 Kb	½ Kb
Voltage	5V	5V
IO Pins	54	14
Digital IO with PWM	14	6
Analog Pins	16	6

4.3 TFT Display

The TFT Display is a Liquid Crystal Display which is interfaced with the coordinator to display the Node ID and Temperature in Celsius at each node. The TFT display used at the co-ordinator side is ITDB02-2.4E TFT-LCD module from Ithread Studio. It uses S6D1121, a single chip low power CMOS LCD controller/driver from Samsung. This supports

for 8-bit and 16-bit displays modes. A toggle switch is available on the backside of the display to switch between these two modes. [15]. Certain pre-built library like UTFT and UTouch library have been used for developing the application program. [16]. The display will be fed from the temperature inputs of LM35 temperature sensors from different parts of the spatial distribution. The LM35 temperature sensors [18] are precision IC temperature sensor whose output voltage is linearly proportional to temperature in Celsius. The output of this temperature sensor is fed to one of the Analog pins of Duemilanove board which is programmed to scan this pin for every 15 milliseconds.

5. TESTING AND RESULTS

The experimental setup now consists of a coordinator node monitoring temperature from three end devices. The variable considered is temperature measurement. In the testing stage the experimentation is carried out in three successive phases. Initially, it is tested for receipt of signals and temperature reading is observed for correction. This is considered to be Test Case I and here the Coordinator Board is programmed in such a way that it receives the Board Id and the Temperature from all the nodes after every 30 seconds and displays it. This data gets printed in the Serial Console on PC and the LCD, to which the Coordinator is connected and along with it prints the RSSI PWM value in each case. The tests are repeated for consistency and precision of measurements from three different rooms and RSSI values are verified. After establishing the signal strength from different locations, it is now required to create fuzzy membership functions for the signal values for processing. This is considered as Test Case II where the Fuzzy Logic distribution is constructed using embedded Fuzzy Logic Library for the Coordinator. This fuzzy mapping is based on the PWM values from Test Case I as RSSI values and is used for mapping the input to distance for each node. It is verified that the calculated distance using Fuzzy Logic is almost equal to physical distance of each node. Based on the calculated distance, the Coordinator will detect the room number in which the remote nodes are located and query the temperature. The result will be displayed on the LCD display and the Serial Terminal Console. Thus the locations of the nodes are mapped with respect to the RSSI. To proceed further, it is required to log on the data from three different locations separately without the need to print it every time. Logged data may be found useful for analysis and further tuning the fuzzy algorithm. Hence, another test case, Test Case III is initiated where the Coordinator is programmed in the similar fashion as Test Case II, but the support for Gobetwino is added. The data from each sensor is now logged into three different text files and later used for analyze these results by importing into excel. The data does not get printed in the Serial Terminal Console but is used for programming the LCD. Thus through the three stages of testing, a complete system to receive, display and record the

data is made available and data logged could further be used for improving on the accuracy of the results.

The experimental set up is now positioned for implementation. The locations are identified as Node 0, Node 1 and Node 2. When the nodes are operated with a 9V battery, it is found that the Node 2 and Node 3 are found to survive for approximately one hour duration whereas Node 0 survived for more than 2 hours and 45 minutes. This is recorded in Figure 9. The PWM counts are considered as the indicator of RSSI and it is seen that farther the nodes, lesser is the time duration for survival. The three nodes are placed at three different adjacent rooms at a distance of 2 meters, 6.5 meters and 9 meters respectively. The RSSI PWM count of Node 0 is between 6 and 18 milliseconds, RSSI PWM count of Node 1 is between 20 and 58 milliseconds, and RSSI PWM count of Node 2 is between 73 and 95 milliseconds. This data could also be obtained in Test Case I, but will be available in textual form in Serial Console.

Now, it is important to compare the actual distance of the nodes from fuzzy distances calculated from RSSI inputs. Hence, the distance indicated through fuzzy logic is compared with the actual distance for all three nodes. This is shown in **Figure 9** which details the comparison between the Fuzzy Logic calculated distances in meters v/s the RSSI PWM counts in milliseconds for the three nodes. From the graphs in **Figure 10-12**, observations show that the actual and calculated distance is approximately equal. The distance of the all the three nodes are represented through Fig. to fig. Based on these distance the Coordinator identifies the room number and prints the temperature in each room in the LCD. The data also gets logged to the respective log files. **Figure 13** shows the change in temperature in each over the time period during which the nodes were active. Thus the successful implementation of a Wireless Sensor Network using RSSI Fuzzy logic algorithm was established for an real time indoor temperature monitoring application.

The experimental set up is now positioned for implementation. The locations are identified as Node 0, Node 1 and Node 2. When the nodes are operated with a 9V battery, it is found that the Node 2 and Node 3 are found to survive for approximately one hour duration whereas Node 0 survived for more than 2 hours and 45 minutes. This is recorded in **Figure 9**. The PWM counts are considered as the indicator of RSSI and it is seen that farther the nodes, lesser is the time duration for survival. The three nodes are placed at three different adjacent rooms at a distance of 2 meters, 6.5 meters and 9 meters respectively. The RSSI PWM count of Node 0 is between 6 and 18 milliseconds, RSSI PWM count of Node 1 is between 20 and 58 milliseconds, and RSSI PWM count of Node 2 is between 73 and 95 milliseconds. This data could also be obtained in Test Case I, but will be available in textual form in Serial Console.

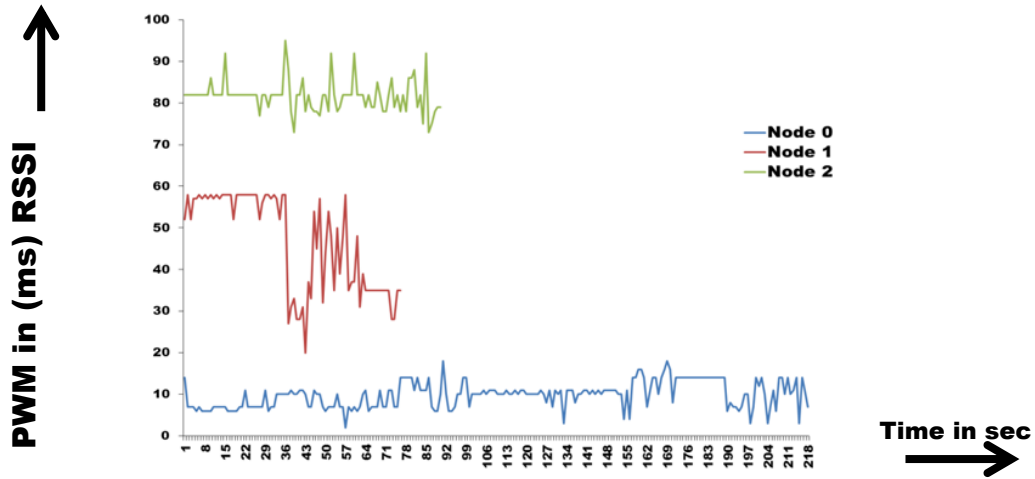


Fig 9: RSSI PWM counts of the Sensor Nodes, received by the Coordinator

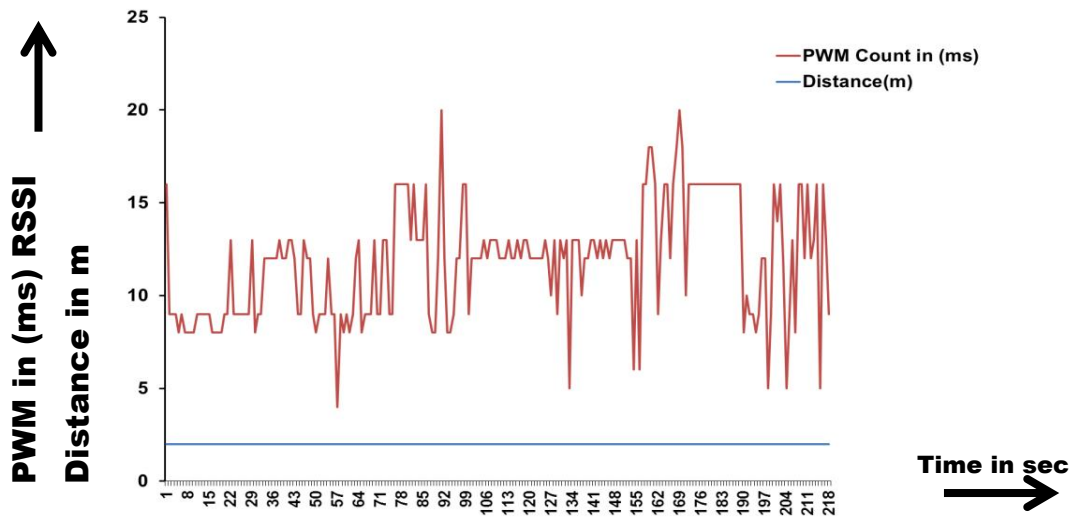


Fig 10: Distance v/s PWM counts for Node 0

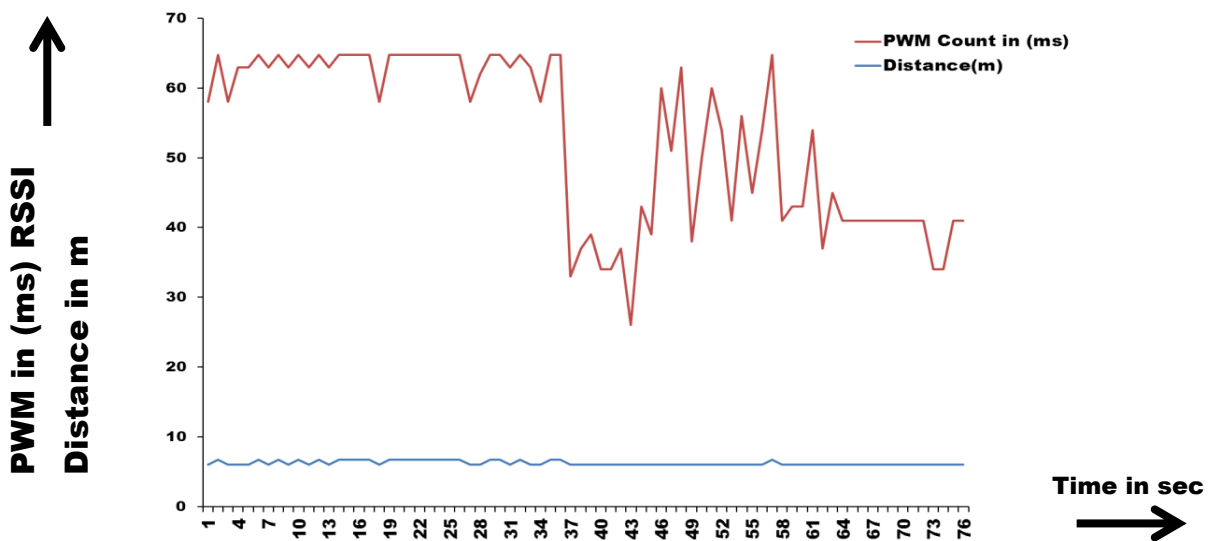


Fig 11: Distance v/s PWM counts for Node 1

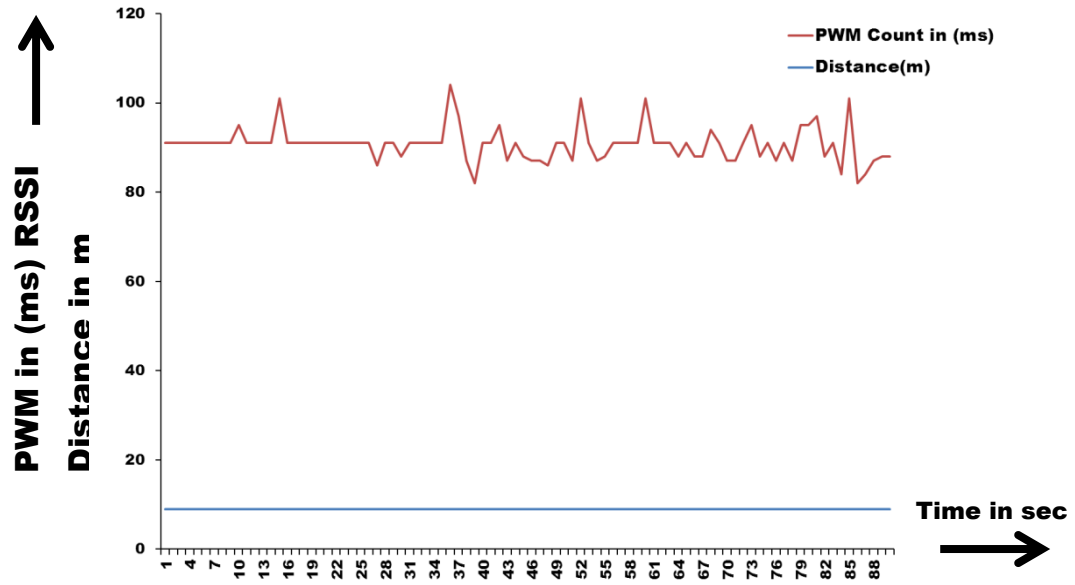


Fig 12: Distance v/s PWM counts for Node 2

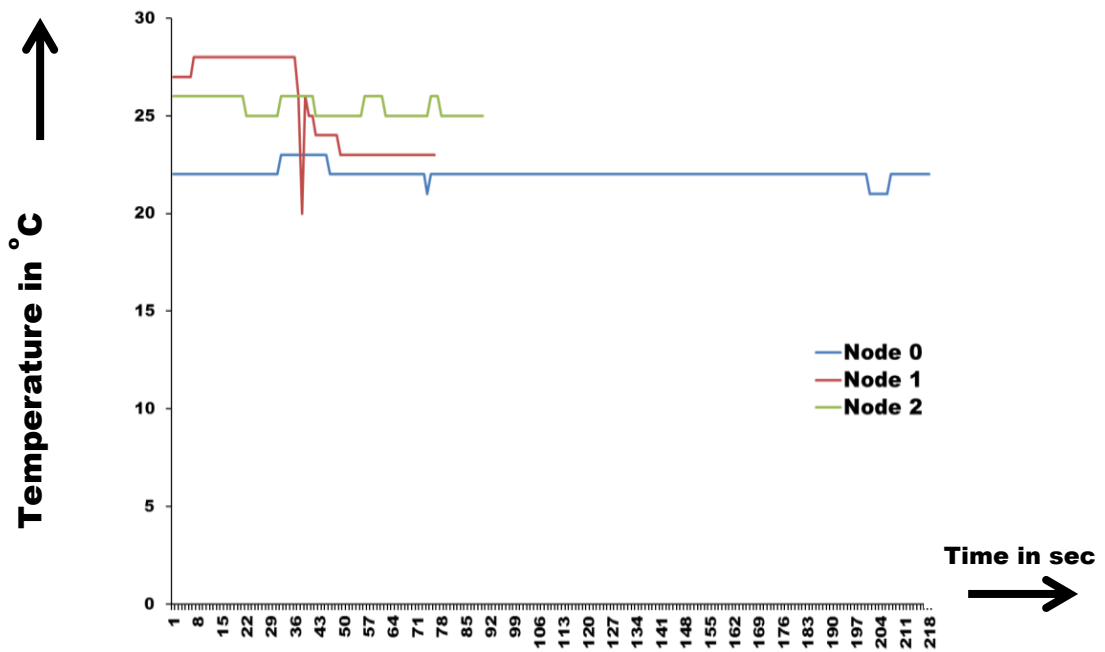


Fig 13: Temperature measured at Nodes 0, 1, 2

7. CONCLUSION

The implementation of a fuzzy based RSSI algorithm for temperature monitoring application using Intelligent Localization algorithm is discussed in this paper. The experimental implementation is performed on remote nodes, all placed inside the School of Information Sciences, Manipal University building. They are placed at *non line of sight* from the Coordinator. The network is constructed based on Star Topology. The network could be linearly scaled without affecting the performance parameters of the system. The system operates well with minimal human interference. However, the disturbances in operating environment like the opening and closing of the doors create changes in the signal strengths. The robustness of the algorithm and the implementation in different topologies along with duplex communication are considered for future scope in implementation.

8. REFERENCES

- [1] D. Niculescu and B. Nath, "DV Based Positioning in Ad Hoc Networks", presented at Telecommunication Systems, 2003, pp.267-280.
- [2] C. Savarese, J. Rabaey, and K. Langendoen, "Robust positioning algorithms for distributed ad-hoc wireless sensor networks", in the Proc. General Track: 2002 USENIX Annual Technical Conference, 2002, pp.317-327.
- [3] Jia Wang and Fu Jingqi, "Research on APIT and Monte Carlo Method of Localization Algorithm for Wireless Sensor Networks". LSMS/ICSEE (2) 2010: 128-137.
- [4] Y. Zeng, S. Zhang, S. Guo, and X. Li, "Secure Hop-Count Based Localization in Wireless Sensor Networks", in Proc. CIS, 2007, pp.907-911.
- [5] Cesare Alippi and Giovanni Vanini, "A RSSI-based and calibrated centralized localization technique for Wireless Sensor Networks", in Proceedings of Fourth IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), Pisa, Italy, March 2006, pp. 301-305.
- [6] Q. Zhang, J. Huang, J. Wang, C. Jin, J. Ye, W. Zhang, 2008, "A new centralized localization algorithm for wireless sensor network," In Proc. of the IEEE ChinaCom 2008, pp. 625--629.
- [7] Mourad F, Snoussi H, Abdallah F and Richard C, "Anchor-Based Localization via Interval Analysis for Mobile Ad-Hoc Sensor Networks," Signal Processing, IEEE Transactions on , vol.57, no.8, pp.3226-3239, Aug. 2009.
- [8] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-Less Low Cost Outdoor Localization for Very Small Devices," IEEE Personal Comm. Magazine, vol. 7, no. 5, pp. 28-34, Oct. 2000.
- [9] Zulkifli, N. S. A., FK Che Harun, and N. S. Azahar. "XBee wireless sensor networks for Heart Rate Monitoring in sport training." In Biomedical Engineering (ICoBE), 2012 International Conference on, pp. 441-444. IEEE, 2012.
- [10] Rajeev Piyare, Soeng-ro Lee, "Performance Analysis of Xbee ZB Module Based Wireless Sensor Networks", International Journal of Scientific & Engineering Research, pp. 1615-1621, Volume 4, Issue 4, April 2013.
- [11] A Santhosh , S Sujana A, "Xbee Based WSN with GSM Technology to Monitor Paddy Field Environment" , International Journal of Computer Applications,pp.14, December 2013 Volume 83, No5.
- [12] Vongsagaon Boonsawat, Jurarat Ekchamanonta, Kulwadee Bumrunghket and Somsak Kittipiyakul,"Xbee Wireless Sensor Networks for Temperature Monitoring", In the Second Conference on Application Research and Development (ECTI-CARD 2010), Chon Buri, Thailand. 2010.
- [13] Embedded Fuzzy Logic Library Documentation <http://zerokol.com/product/51e93616e84c5571b7000018/4/en> , March 2014.
- [14] Drew Gislason, "Zigbee Wireless Networking", Newnes Publications, 2008
- [15] ITDB02-2.4E TFT-LCD Module: <http://imall.iteadstudio.com/im120419004.html> (March 2014)
- [16] UTFT and UTouch Library for Arduino: <http://henningkarlsen.com/electronics/library.php?id=51> (March 2014)
- [17] Product Manual v1.x.4x - ZigBee Protocol - by Digi International: <https://www.sparkfun.com/datasheets/Wireless/Zigbee/XBee-2.5-Manual.pdf>
- [18] LM35 Data sheet: <http://www.ti.com/lit/ds/symlink/lm35.pdf> (March 2014)
- [19] Jianwu, Zhang, and Zhang Lu. "Research on distance measurement based on RSSI of ZigBee." Computing, Communication, Control, and Management, 2009. CCCM 2009. ISECS International Colloquium on. Vol. 3. IEEE, 2009.
- [20] Mayalarp, N. Limpaswadpaisarn, T. Poombansao, and S. Kittipiyakul, "Wireless mesh networking with XBee," in 2nd ECTI-Conference on Application Research and Development (ECTI-CARD 2010), Pattaya, Chonburi, Thailand, 10-12 May 2010.
- [21] Mendel, Jerry M. "Fuzzy logic systems for engineering: a tutorial." Proceedings of the IEEE 83.3, pp. 345-377, 1995.
- [22] Arduino API References www.arduino.cc, March 2014.
- [23] Gobetwino User Manual,0.5 version, <http://mikmo.dk/gobetwino.html> (March 2014)