

Design and Implementation of Wireless Sensor Network Based on MATLAB Interfaced with Arduino

Dr. Mahmood F. Mosleh^{1,a}, Dr. Rashid A. Fayadh^{2,b}, Shahad A. Hamid^{3,c}
^{1,2,3}College of Electrical Engineering Techniques, Middle Technical University
Aldora, Baghdad, Iraq
^adrmahfa@yahoo.com, ^br_rashid47@yahoo.com, ^cshahad.ahmed1993@yahoo.com

Abstract. Wireless Sensor Network (WSN) have a great influence in various applications in recent years. Monitoring the surrounding environments is one of the most important uses in many aspects of modern life under the great technological improvement which saves effort and time. In this research, a facility of interfacing of new version of Matlab with the Arduino microprocessor has exploiting to implement a flexible network. The data of each sensor is collected in wireless manor exploiting the ZigBee protocol with its XBee hardware platform through a router which connected wirelessly through coordinator to Personal Computer (PC) as a base station. The gathering data is interpreted using Matlab to store and/or displayed as a schedule or graphical. The proposed network consists of many routers each one is associated with a group of sensors directly with one hope or indirectly with multi-hope given the network high flexibility in expansion to cover variable area on demand. In this research, one router with four sensors are used as a part of the proposed network for experimental implementation. The results show that the Matlab with Arduino support the performance of such network to process the collected data for storing and displaying. Also, it can be designed a program for any function serving user requirements.

Keywords: Matlab, Arduino, WSN, monitoring

I. INTRODUCTION

WSN has important applications such as remote environmental monitoring and target tracking, especially in last years with the help of sensors that are smaller, cheaper, and intelligent. The technology of wireless sensor networks evolves very fast gratitude to the advancement of digital electronic technology and wireless communication technology. WSNs are collected of a large number of sensor nodes that are deployed spatially through indoor or outdoor environment. These sensor nodes can be used to detect the surround conditions such as gas, humidity, light, temperature, sound and so on. It means that there are various application areas for WSNs. The challenges of WSNs are the limitation of energy because of small battery used, the transmission bandwidth and small size of memory. The node in WSN can't transmit a large number of data so that it used a system with low power consumption such as Wi-Fi, Bluetooth and ZigBee [1]. A simulation model is better to evaluate the performance of WSN. Among of many tools that used recently for such simulation is the MATLAB package. In WSNs, this tool can be used as interfaced technique between coordinator that receive data from all sensors and sends it to the base station which is used in this research as a Personal Computer (PC). Also, can be designed a special program included more than one function and control all processing to execute a full task of WSN. In addition, a monitoring of environment can be done along a limited period using such technique with alarm can be designed easily by some of commands added to the special program.

Many research has been implemented a WSN with Matlab tools for the purpose of performing some tasks such as, [2] who suggested an algorithm for WSN in Matlab environment. This model can reflect the reality networks effectively to verify the performance of the localization algorithms including location node numbers, run time and location error. In [3] the authors mainly focuses on the correlated measures made to reduce human interaction with machines by using industrial observation robot. This robot collects the parameters from the machines and transmits to control room using Zigbee protocol. The control room interface with the robot is designed using MATLAB GUI model. MATLAB GUI interface shows the parameter readings from the robot and plotted in graph. The robot mainly observes smoke, temperature, gases and light intensity that defect in manufacturing machines and security purposes using cameras.

This paper attempt to design and implemented a WSN based on interface with Matlab to collect the information from various nodes using the ZigBee protocol based on IEEE 802.15.4 report as an efficient communication link. Such network has an advantage for extension in any size of area to cover and monitor the environments including recorded data for any parameters such as lighting, gas, motion and distance.

II. WSN Overview

A WSN can be defined as a network of devices, denoted as nodes, which can sense the environment and communicate the information collected from the monitored field through wireless links. The data is forwarded, possibly via multiple hops, to a sink (sometimes denoted as controller or monitor) that can use it locally or is connected to other networks (e.g., the Internet) through a gateway. The nodes can be stationary or moving, aware of their location or not and homogeneous or not [4]. During the design of Wireless Sensor Networks one has to consider the effects of energy consumption, system topology, fault tolerance, communication, and synchronization, scheduling strategies, quality of service and coordination protocols [5]. WSNs are the combination of embedded system and wireless communication which allows data transmission among the sensor nodes over ad-hoc wireless networks as shown in Fig. 1. The ad hoc networks requires no existing infrastructures unlike those in WLAN or cellular networks. The brain of each WSN node is the microcontroller which processes readings from its own sensors and, in some cases, readings from adjacent nodes as well since the sensors in different nodes located near each other may be highly correlated and hence the amount of sensor data needed to be transmitted from the two sensor nodes can be cooperatively reduced [6].

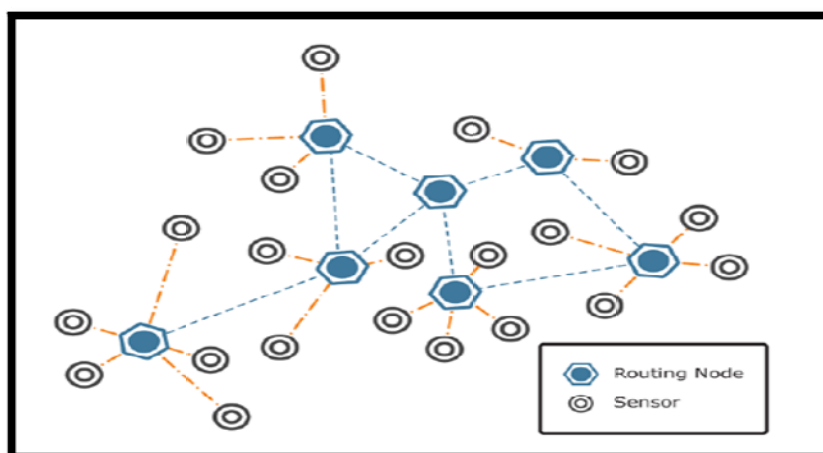


Fig. 1. Ad Hoc Wireless Sensor Networks

III. MATLAB Interfacing with Hardware

Many approaches are used for simulation and modeling WSN. They operate in different levels of simulating: hardware emulation, operating system and application level. The environment for building a simulation model was MATLAB. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Simulink is an environment for multi domain simulation and embedded systems. It provides an interactive graphical environment and a customizable set of block libraries that facilitates the designing, simulation, implementation, and testing a variety of time-varying systems, including communications, controls, signal processing, video processing, and image processing [7].

MATLAB with Arduino

The MATLAB Support Packages for Arduino Hardware to communicate with Arduino board through USB cable. In this case the Arduino becomes as data acquisition or to interface the other parts of the network to the MATLAB directly without need to additional component. The support packages are available for 32 and 64-bit Windows, 64-bit Mac OS and Linux. To apply the interfacing between MATLAB and Arduino through USB port, the steps listed in the next subsection can be done to install the Arduino as external hardware.

Installing of Packages

In this research it has been exploit the facility of interfacing the Matlab with Arduino for purpose of reading and writing the data of sensors collecting from the network which can be immediately seen in the MATLAB. The following steps must be applied to install the Arduino to be interfaced with the Matlab:

- 1- First, start MATLAB and click the "Add-Ons" which lead to drop down menu. In the drop down menu click "Get Hardware Support Packages". It will start the package installer window as shown in Fig. 2.
- 2- In this step it may be need to connect to the internet then select "Install from Internet" and then next as shown in Fig. 3.

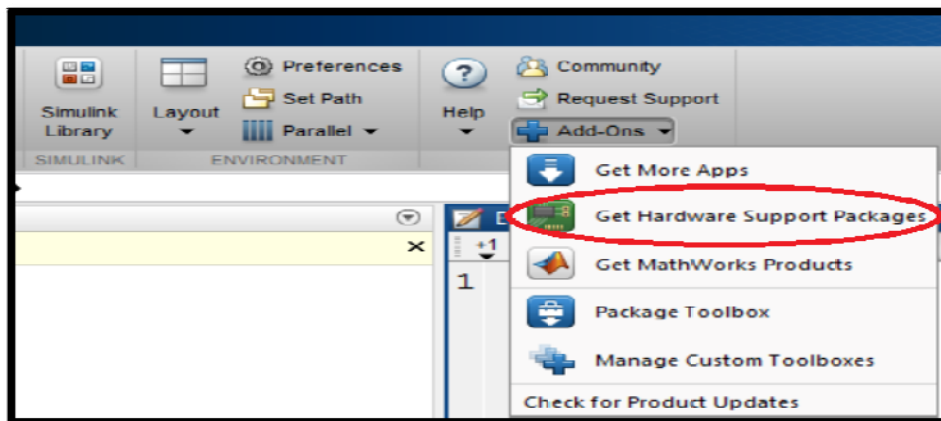


Fig. 2. Get hardware support package

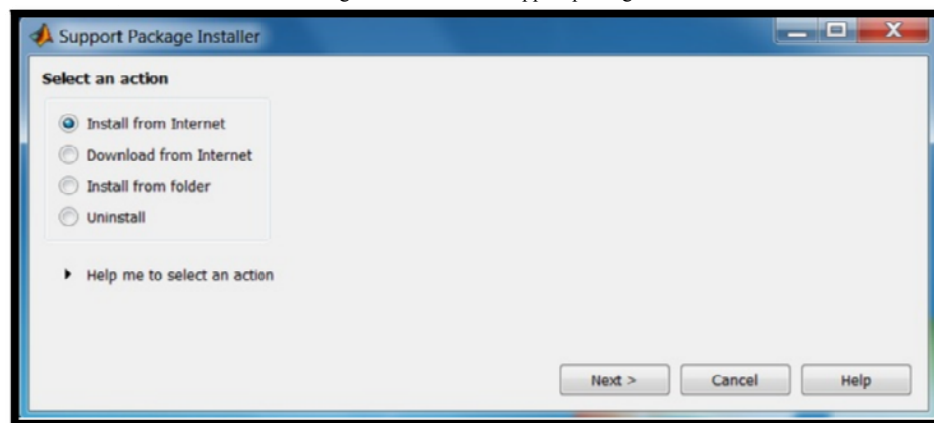


Fig. 3. Install packages

3-Select the Arduino package and then check and select all packages displayed and click next to continue installation as shown in Fig. 4.

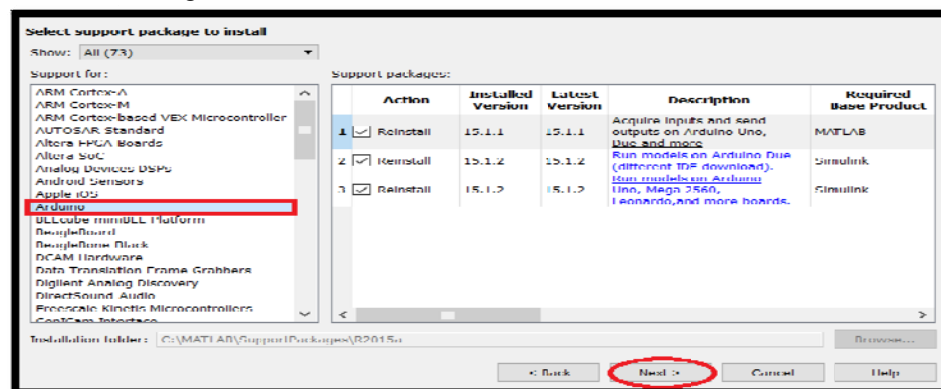


Fig. 4. Select Packages of Arduino

4- Now the installer ask to log in MathWorks account or create account during installation. Accept the license agreement on the next screen and continue to download the packages. Then MATLAB downloads and installs all the required packages.

Testing the package instilled

Once the packages are installed, connect Arduino board to PC and type the following command in MATLAB command window:

```
>> a = arduino()
```

If there are more than one Arduino connected to the PC, specify the board type that will be communicating with:

```
>> a = arduino('com3', 'uno')
```

MATLAB will then attempt to communicate with Arduino board. If successful, it will display the properties of the Arduino board connected to the PC. This information displays the port on which Arduino board is connected, the model of the Arduino board, and available pins and libraries available for the board as shown in Fig. 5:

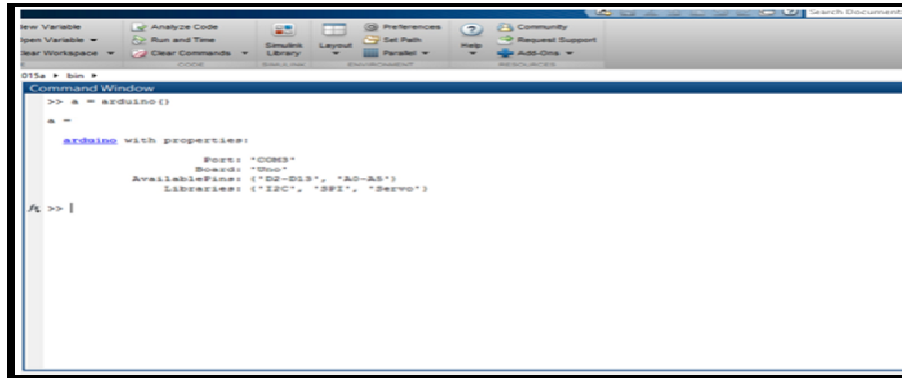


Fig. 5. Arduino board connected to PC

IV. The Proposed system

The proposed system use in this research is shown in Fig. 6.

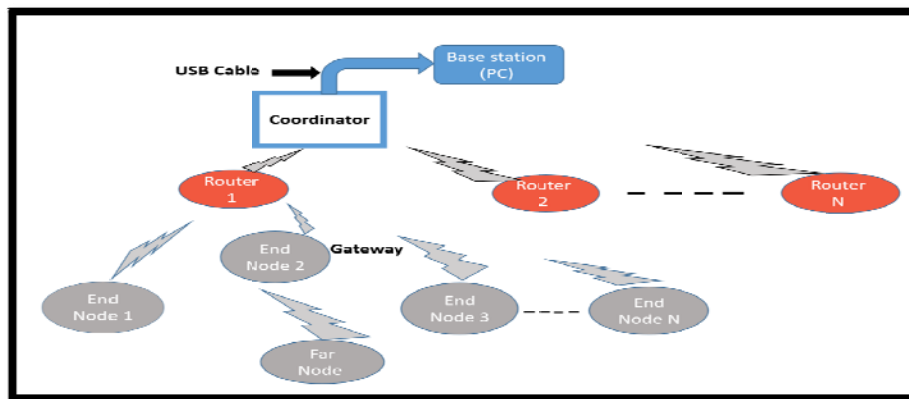


Fig. 6. Proposed system

End Node

The components of end node are sensor, Arduino, XBee and Xbee shield. The sensor which is transduce the surrounding environment into electrical signal either in digital or continuo signal. The second part of node component is Arduino for collecting and processing the data in addition to produce a suitable commands. The last component is the Zigbee based IEEE 802.15.4 which is used as communication part. In the following subsections each part will be explained in detail as shown in Fig. 7.

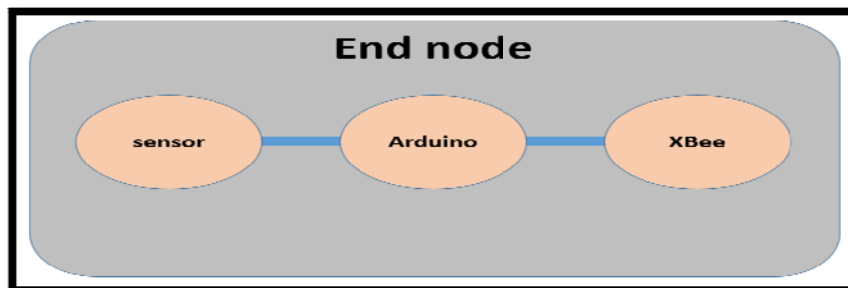


Fig. 7. End Node

Sensor

Many types of sensors can be used in the proposed network, but for experimental, it has been used only four types in this research.

i- Gas Sensor

This sensor is a MQ-7 semiconductor of carbon monoxide as shown in Fig. 8. Its features are high sensitivity to carbon monoxide, stable, long life, low cost and suitable for different applications. The sensor could be used to detect different amount of gases contains CO, the specification of MQ-7 is listed in table 1 [8]:

TABLE I Gas Sensor Specifications

Parameter	Value
Circuit voltage	5v
Using temperature	-20C to +50C
Storage temperature	-20C to +50C
Relative humidity	Less than 95% RH

This sensor has 4 pins like GND (Ground), +5V (VCC), AOUT (Analog Pin) and DOUT (Digital Pin).

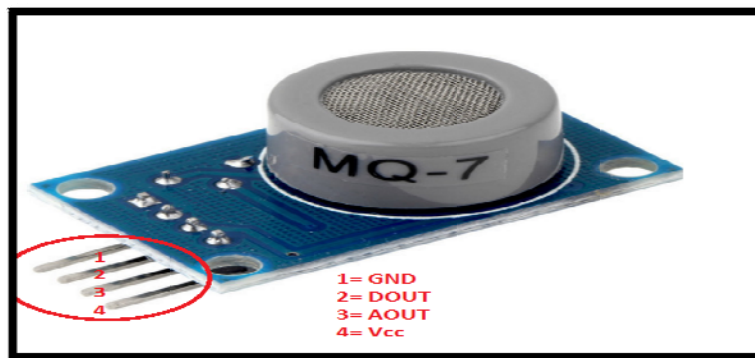


Fig. 8. MQ-7 Sensor

ii- Light Sensor

It is a photocell which used to sense the light and send a signal to the Arduino which in turn active the Matlab to record and display the activation time. The feature of this sensor is cheap, small, easy to use, low-power and don't wear out. It is often referred to photoresistors, a Cadmium-Sulfide (CdS) cells, Light-Dependent Resistors (LDR) and photo-resistors as shown in Fig. 9 [9].

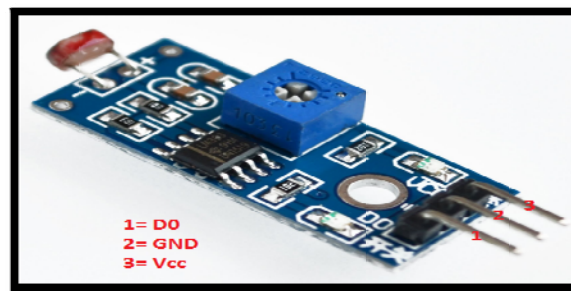


Fig. 9. Light Sensor

iii- Distance Sensor

It is an ultrasonic ranging module HC - SR04 provides (2- 400) cm, and having a ranging accuracy 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The electric parameter of such sensor are listed in table 2 [10]:

TABLE II The Parameters of Distance Sensor

Voltage Source	DC 5V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm

There are four pins in this sensor which are VCC (Power), Trig (Trigger), Echo (Receive), and GND (Ground) as shown in Fig. 10.

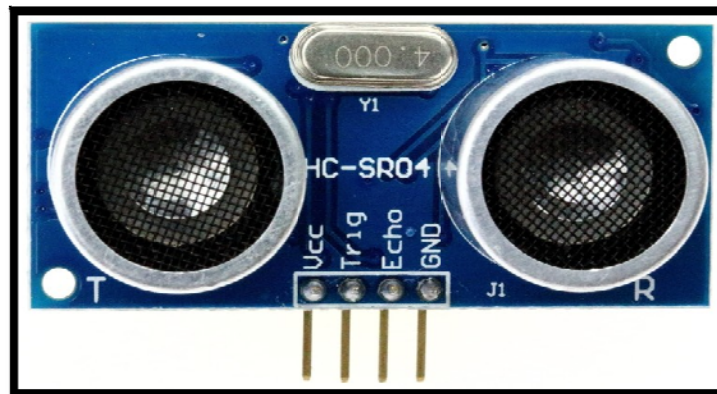


Fig. 10. Distance Sensor

iv- Motion Sensor

This sensor can be used to detect the motion of a human who moves in its range. The features of this sensor are cheap, small, low-power, easy to use and don't wear out. For those reasons they are commonly found in appliances and gadgets use in homes or businesses. It is often referred to as Passive Infrared (PIR), Pyroelectric, or IR motion sensors as shown in Fig.10 [11].



Fig. 11. Motion Sensor

Arduino

Arduino is an open source physical computing platform based on a simple input/output (I/O) board and a development environment that implements the processing language. Arduino can be used to develop standalone interactive objects or can be connected to software on the computer [12]. It is a small microcontroller board with a USB plug to connect to the computer and can either be powered through the USB connection from the computer or from a 9V battery. It can be controlled from the computer or programmed and then disconnected to work standalone. On the other hand there are a number of connection sockets that can be wired up to external electronics, such as motors, relays, light sensors, laser diodes, loudspeakers, microphones...etc. These connectors are arranged like this so that called "shield" boards can be plugged on to the main board [13].

The type of Arduino is depend on many factors like microcontroller, size of memory, number of pins...etc. In this research, the used Arduino Mega 2560 is a microcontroller board shown in Fig. 12 based on the ATmega2560. It has 54 digital input/output pins (of which 14 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), 16 MHz crystal oscillator, USB connection, power jack, ICSP header, and a reset button [14].

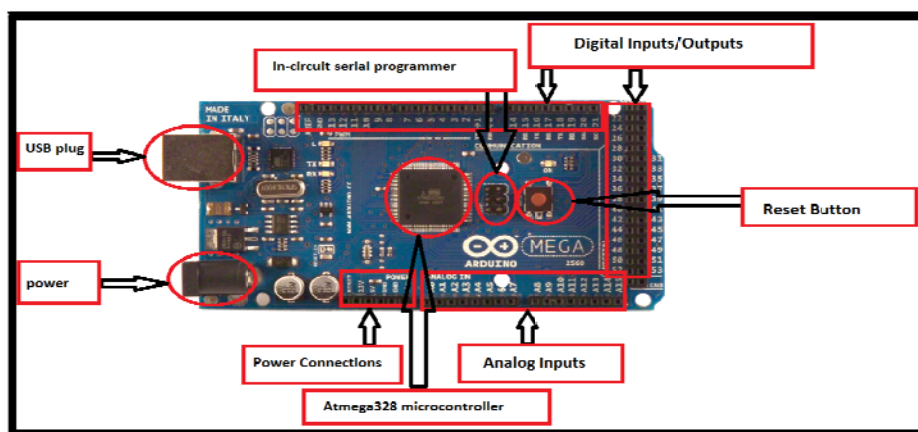


Fig. 12. Arduino MEGA [14]

All characteristics of Arduino mega use in this subsection are listed in table 3 [14]:

TABLE III Characteristics of Arduino Mega

Microcontroller	ATmega2560
Operating Voltage	5V
Input voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins (of which 14 provide PWM output)	54 pins
Analog Input Pins	15 Pins
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

In order to execute all functions for this research, the program shown in the algorithm of Fig. 13 is designed in general form for all tasks.

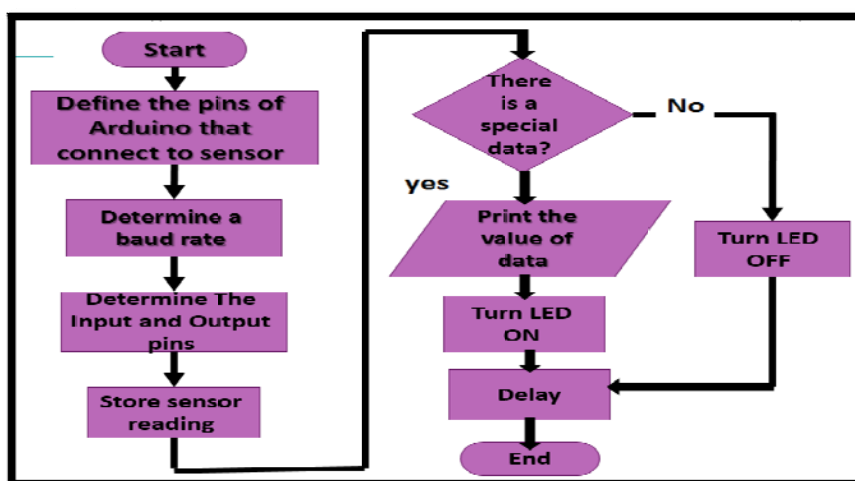


Fig. 13. End node Algorithm

Zigbee Standard IEEE 802.14.5

Zigbee is one of the most widely utilized WSN standards with low power, low data rate, low cost and short time delay characteristics, simple to develop and deploy to provide a robust security and high data reliability. The IEEE 802.15.4 standard employs 64-bit and 16-bit short addresses to support theoretically more than 65,000 nodes per network. ZigBee network can have up to 653356 devices, the distance between ZigBee

devices can be up to 50 meters, and each node can relay data to other nodes. Which creates a possibility of making a very large network which covering significant distances [15]. The Zigbee is a general protocol that includes many techniques called (XBee). In this research it has been use the following types of XBee products which are shown in Fig. 14.

i- XBee Series 2 (S2)

Xbee S2 module is used for embedded solutions providing wireless end-point connectivity to devices. This module incorporates the ZigBee PRO feature with set mesh networking protocol. Series 2 modules allows to create complex mesh networks. However, the infrastructure of a ZigBee network is more complex and requires more configuration to fully implement. This module can give range of 40 meters indoor or 120 meters outdoor. This XBee wireless device can be directly connected to the serial port (at 3.3V level) of a microcontroller. This module supports data rates of up to 250kbps. It is designed for high-throughput (35kbps) applications requiring low latency and predictable communication timing [16].

ii- XBee S2C

XBee is a 2mW wire antenna S2C which is the latest of its kind in Series 2 and offers point to multipoint device connectivity with ease providing cost-effective wireless solutions for electronic devices. This version (S2C) of XBee ZigBee Series 2 module utilizes EM357 transceiver. The EM357 transceiver is faster, has lower current draw, and has significantly more RAM and Flash. These advanced features allow EM357-based modules to support more network traffic, have more memory for code updates, and operate more efficiently. This XBee module, similar to its predecessors have 2 mm pin spacing. It can be used one of its adapter boards and easy interfacing within any project. Series 1 and Series 2 XBee modules have the same pin-out. However, Series 1 modules cannot communicate with Series 2 modules [17].

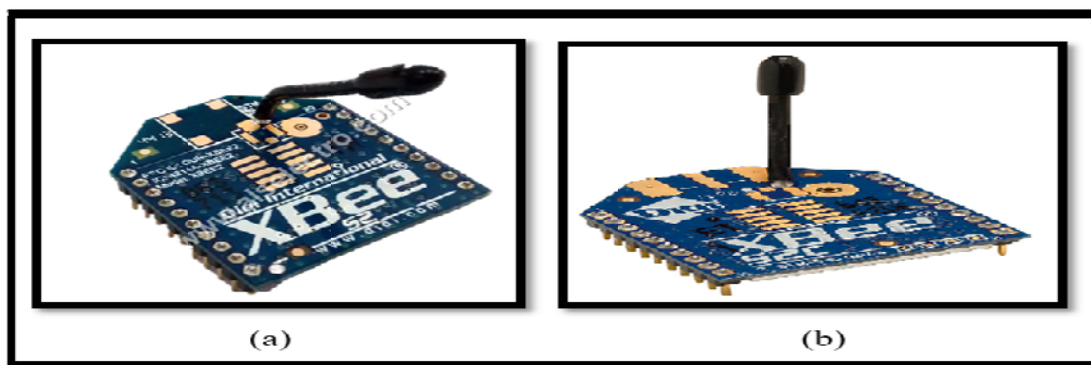


Fig. 14. (a) Xbee S2 (b) Xbee S2C

The following comparison between the two products shown in table 4 [17]:-

TABLE IV The Comparison between Two Products of XBee.

Specification	XBee S2	XBee S2C
Indoor/Urban range	up to 133 ft. (40 m)	up to 200 ft. (60 m)
Outdoor RF line-of-sight range	up to 400 ft. (120 m)	up to 4000 ft. (1200 m)
Transmit Power Output	2 mW (+3dbm)	6.3mW (+8dBm) Boost mode 3.1mW (+1dBm) Normal mode
RF Data Rate	250 Kbps	250 Kbps
Receiver Sensitivity	-98dBm (1% PER)	-102dBm (1% PER) Boost mode -100dBm (1% PER) Normal Mode
Supply Voltage	2.8 - 3.6 V	2.1 - 3.6V
Transmit Current (typical)	40 mA (3.3 V)	45 mA (+8dBm) Boost Mode 33 mA (+5dBm) Normal Mode
Idle/Receive Current (typical)	40 mA (3.3 V)	31 mA (+8dBm) Boost Mode 28 mA (+5dBm) Normal Mode
Power-down Current	1 μ A	1 μ A
Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Network Topologies	Point to point, Star, Mesh	Point to point, Star, Mesh

Both types of XBee are installed to the Arduino through XBee shield shown in Fig. 15 to ensure the conductivity without any losses

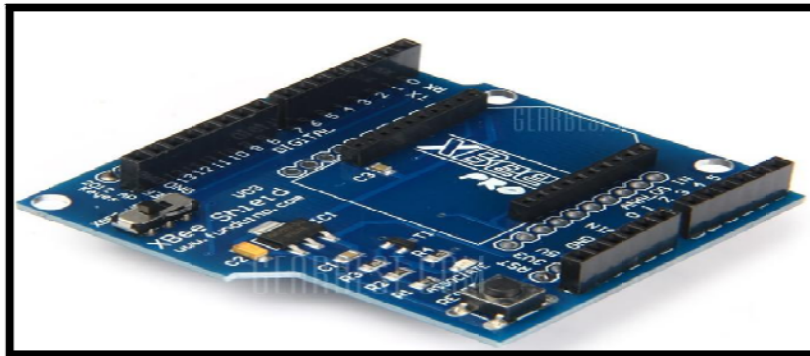


Fig. 15. XBee Shield

Router

It is possible to have more than one router in this research to give the proposed network for feature through which to increase the area under control by increasing the number of routers. Each router is linked to a set of end nodes, in each one a sensor can perform a specific task and sends the data either according to the program that was designed for the Arduino or responding to the program that has been designed in Matlab to perform several sequence commands. In addition, the user can intervene at any moment in order to monitor or take any decision. On the other hand any end node may be performs as a gateway to joint any far node which is out of router range which leads to more extending the coverage area of the network. Also, it is possible to have more than one router with its group of nodes as in router1 to duplicate the network area. For the purpose of programming the router to perform all above the tasks, a program has been designed shown in the algorithm of Fig. 16. The structure of the router include of Arduino, Xbee shield and Xbee as shown in Fig. 17. As well as running an application function, a router is act as an intermediate router between coordinator and end node to gateway to pass the data [18].

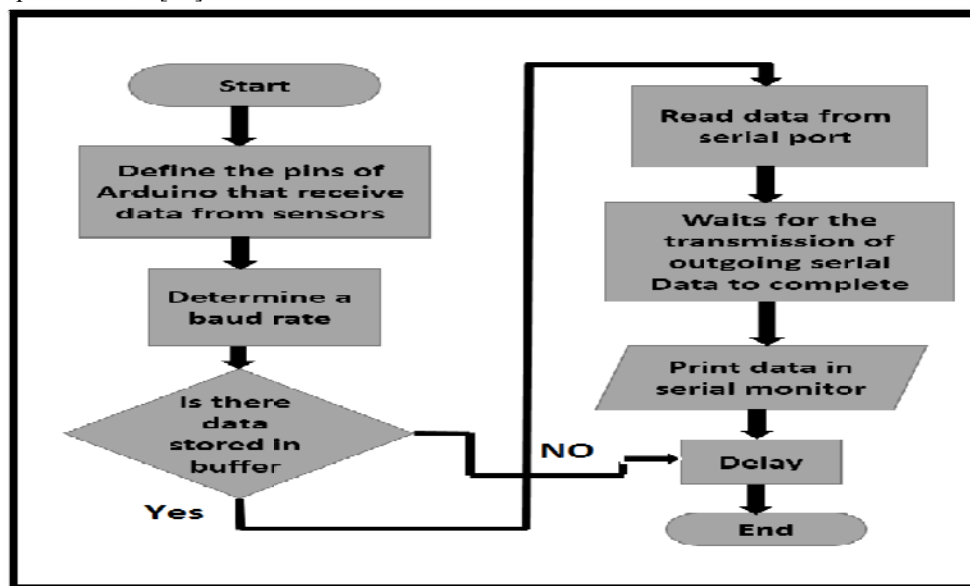


Fig. 16. Router Algorithm

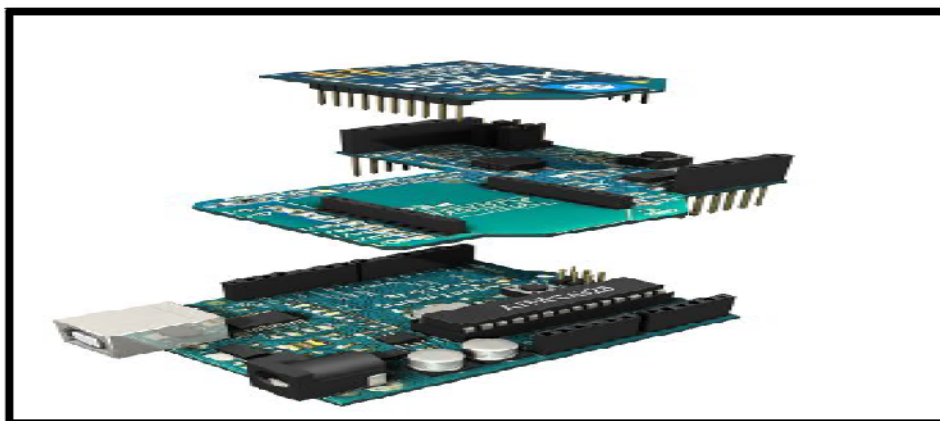


Fig. 17. The Router Construction

One of the Arduino family is the Arduino Uno which is the most extended board as shown in Fig. 18. The board carries an 8-bit microcontroller, and it comes with 14 digital input/output pins and 6 analog inputs. Six of the digital pins can be programmed to send pulse width modulation (PWM). The Uno board also comes with internal peripherals able of running the UART, SPI, and I2C communication protocols. Programs using the Arduino Uno board (Fig. 18) can be as big as 30 Kbytes and run at 16 MHz [19]. The reason for using Arduino Uno in router node is not needing to have a large numbers of pins.

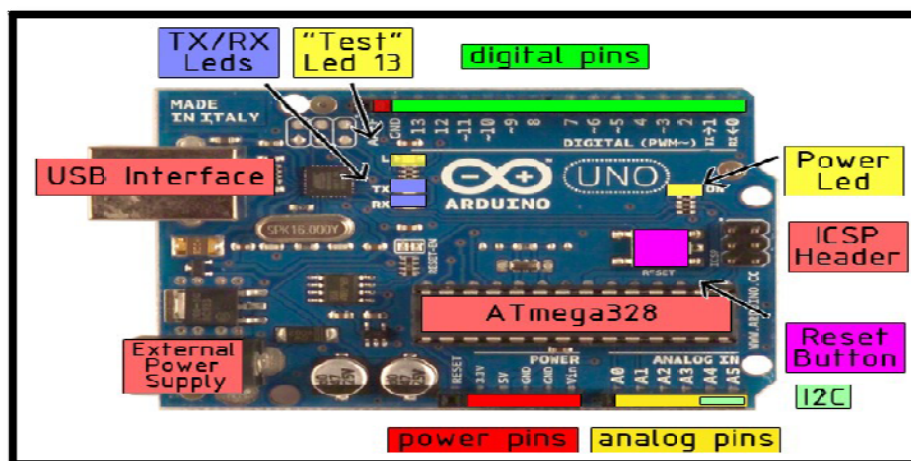


Fig. 18. Arduino UNO Board

All characteristics of Arduino Uno use in this subsection are listed in table 4 [20]:

TABLE V Characteristics of Arduino Uno.

Microcontroller	ATmega328
Operating Voltage	5V
Input voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins (of which 14 provide PWM output)	16 pins
Analog Input Pins	6 Pins
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 Hz

Coordinator

It is a gateway device that received the data from each router and passed to the base station. It is constructed of XBee (S2) installed on XBee adapter connected to the USB port of PC as shown in Fig. 19. To install this XBee it must defined its address to each router in X-CTU software to enable such coordinator to receive the data from each router that define the address of such coordinator in its X-CTU software.

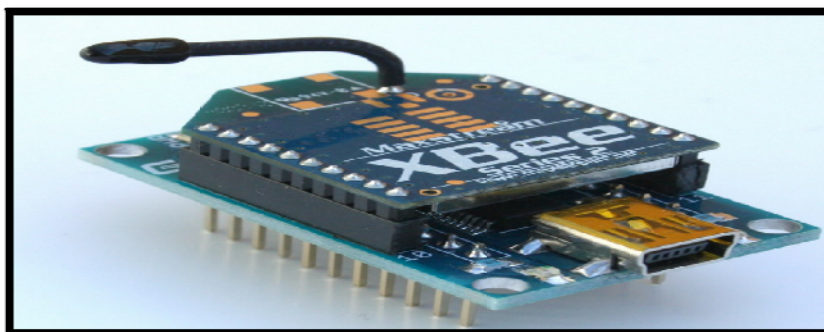


Fig. 19. XBee Coordinator

Base Station

Matlab has many features including storing, sending commands, designing monitoring software, storing and sending data for specific times. Therefore, a special Matlab program has been designed in this research to receive data from sensors as shown in the algorithm of Fig. 20.

There are two types of reading according to the function of the sensor. The first type includes the motion and lighting sensor which records the reading when there a change is happened for example, turn on or off lighting and a body moved in the range of PIR sensor in this cases the Matlab will display suddenly those new reading as logical value. The second type includes distance and gas sensor, they read the distance and gas amount at regular period and plot the results if need so the Matlab program has been designed for data analysis of sensors as shown in the designed algorithm of Fig. 21.

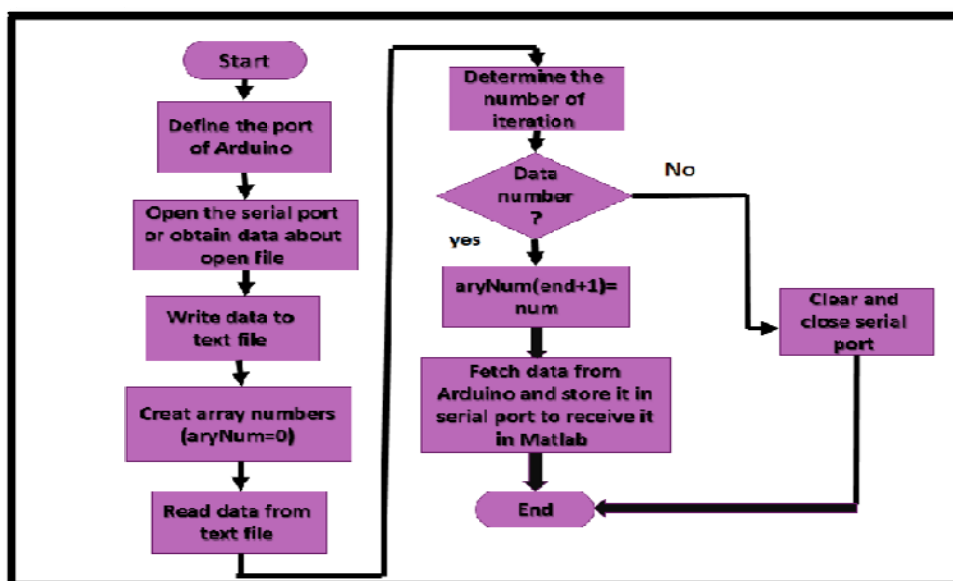


Fig. 20. Algorithm for Receive data from Sensor

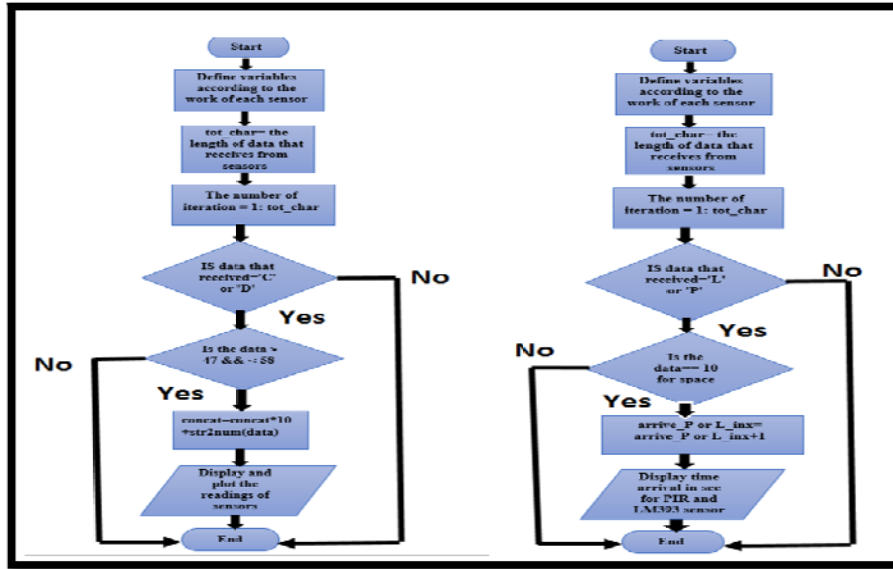


Fig. 21. Algorithm flow chart for data analysis of sensors

VI. Results and discussions

As mentioned previously, it has been used four sensors (Distance, Motion, Light, and Gas) connected to router1 for experimentally application. First, the result recorded in this section for each sensor individually to clear out the capability of each sensor reading. Fig. 22 shows the reading of each sensor that displayed on serial monitor of Arduino.

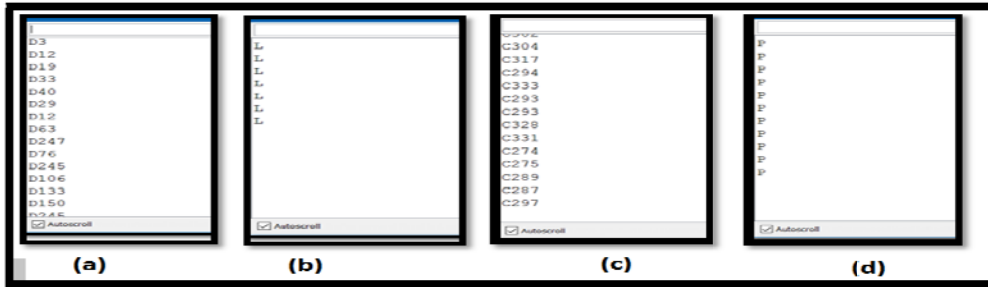
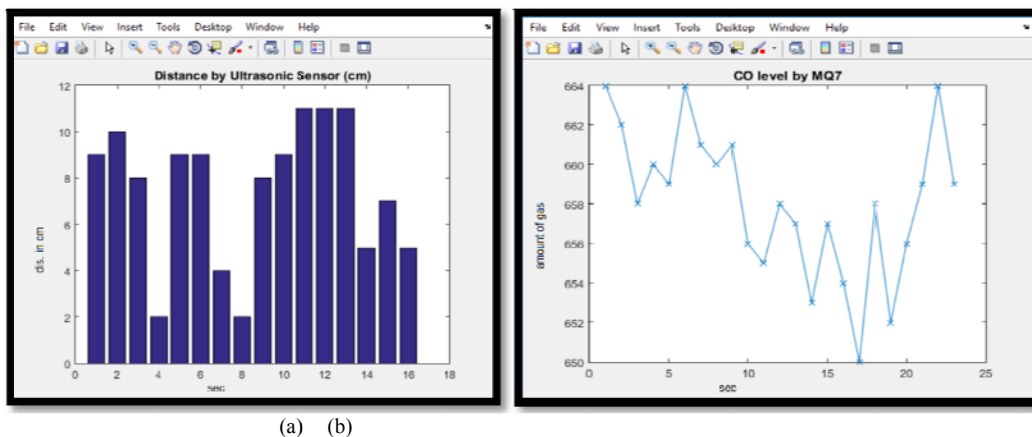


Fig. 22. The reading of sensors in serial monitor of Arduino Distance sensor. (b) Light sensor. (c) Gas sensor. (d) PIR sensor.

The results shown in the Fig.22 indicate sensor readings. Each sensor is indicated by a letter with and without numbers, for example, the distance sensor uses the letter (D) and number beside it to indicate the amount of distance in centimeter, same as for gas sensor. The light and motion sensors use (L) and (P) respectively without any numbers to indicate the appearance of light or person.

For the purpose of displaying the previous reading with time. The Matlab can show each values with its time reading. Also, it can be cleared out the reading by plotting the result vs. time. Fig. 23 shows the plotting of distance and gas sensors reading for a period of second.



(a) (b)
 Fig. 23. Plotting of sensors Distance sensor. (b) Gas Sensor.

In the above figure, (a) which represents the distance between the sensor and any other object and these readings is changing every second, (b) represents the amount of gas in the room and the readings change with time, every second a new reading is recorded. Fig. 24 displayed the time arrival in second for PIR and LM393 sensors.

```
Time arrival in sec for PIR and LMS sensors are:  
  
ans =  
  
    0     6  
    8    12  
   11    22  
   18    31  
   19     0  
   20     0  
   28     0  
   29     0
```

Fig. 24. Time arrival for PIR and LM393 sensors

Fig. 25 shows the overall network connecting and running with its results displayed in the laptop screen. This network can be used for various application like designing a program to leave the network for operating stand alone and recording the results depending on the user need. In addition, the user can be install a threshold value and put a command that be enabled after any value approach this threshold to do something like alarm or activate other device for self-protection.

In advanced, this network can be connected to any other communication network like Wi-Fi, Internet and GSM to monitor and remote control of the proposed network to meet user requirements.

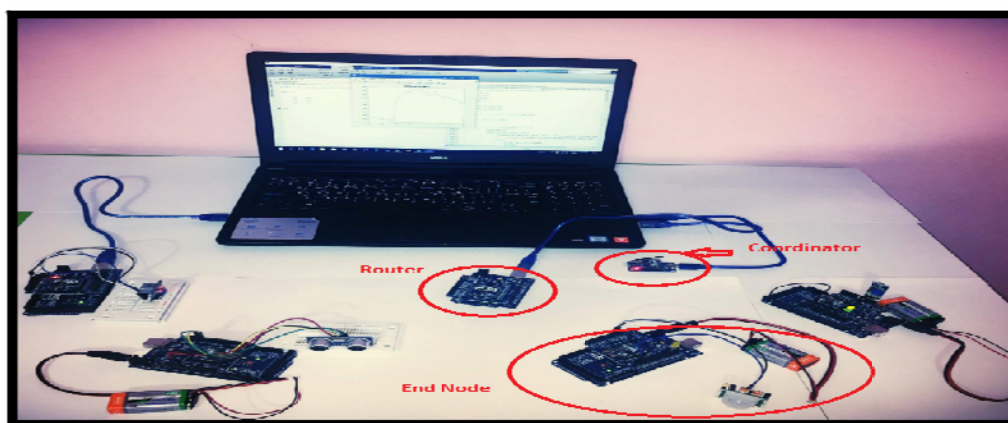


Fig. 25. Hardware design of a network

VII. Conclusions:

In this research, it has been designed a WSN for monitoring any area. It is a flex network which can extended according to the area, exploiting the facility of Matlab version R2016a interfaced with Arduino to receive the data from all sensors through its specific router that pass the information to the Matlab. Analyzing and storing can be done in the Matlab and display it as the user need. Also, a program was designed to apply any function like monitoring, shooting alarm according to threshold value of any sensor reading and display results in a limited or every times. The hardware implementation in this research included four sensors as a part of very large network consists of huge sensors divided hierarchically. The results show the implemented part of such network can response to the demand of user to display the reading in the manner of user's need. In this research the results are displayed in a serial monitor, reading with time and graphically.

REFERENCES

- [1] Zhejun Feng, "Design and Implementation of an Experimental Platform for Performance Analysis in Wireless Sensor Networks", M.Sc. Thesis, KTH Royal Institute Technology, 2011.
- [2] Fangwei Liu, Zhilong Shan, "Node Location Model and Simulations in Matlab for Wireless Sensor Networks", *Advanced Materials Research*, Vol. (694-697.1060), (2013).
- [3] Vijaya Kk, Surender S, "Industry Monitoring Robot Using Arduino Uno with Matlab Interface", Department of Electronics and Instrumentation, St. Joseph's College of Engineering, *Advances in Robotics & Automation*, 2016.
- [4] Chiara Buratti, Andrea Conti, Davide Dardari, Roberto Verdone, "An Overview on Wireless Sensor Networks Technology and Evolution", *Sensors*, 2009.
- [5] Leelavathi G, Shaila K, Venugopal K R and L M Patnaik, "Design Issues on Software Aspects and Simulation Tools for Wireless Sensor Networks", *International Journal of Network Security & Its Applications (IJNSA)*, Vol.5, No.2, March 2013.
- [6] Vongsagon Boonsawat, Jurarat Ekchamanonta, Kulwadee Bumrungkhet, and Somsak Kittipiyakul, "XBee Wireless Sensor Networks for Temperature Monitoring", Sirindhorn International Institute of Technology, Thammasat University, 2013.
- [7] Qutaiba Ibrahim Ali, Akram Abdulmaowjod and Hussein Mahmood Mohammed, "Simulation & Performance Study of Wireless Sensor Network (WSN) Using MATLAB", *International Conference on Energy, Power and Control (EPC-IQ)*, College of Engineering, University of Basrah, Basrah, Iraq, November 30 - December 2, 2010.
- [8] Datasheet of Carbon Monoxide Sensor ISO 9001-2008 Certified Company Available in the Site <http://forum.researchdesignlab.com/datasheet/carbon%20monoxide%20sensor%20mq7.pdf>.
- [9] Photocell Tutorial by Adafruit in Electronics Available in the Site <http://www.instructables.com/id/Photocell-tutorial/>.
- [10] Datasheet of Ultrasonic Ranging Module HC - SR04 by Elec Freaks Available in the Site <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>.
- [11] Datasheet of Motion Sensor by Lady Ada Available in the Site <https://learn.adafruit.com/pir-passive-infrared-proximity-motion-sensor/overview>.
- [12] Massimo Banzi, "Getting Started with Arduino the Open Source Electronics Prototyping Platform", 2nd edition.
- [13] Simon Monk, "30 Arduino Projects for the Evil Genius", 2010.
- [14] Datasheet of Arduino Mega, "Arduino MEGA 2560" by Allide Electronics Available in the Site <https://www.arduino.cc/en/Main/arduinoBoardMega2560>.
- [15] Shanmugaraj Madasamy, R. Prabakaran, "Study on ZigBee technology", Conference Paper, April 2011.
- [16] "XBee (S2) 2mw Wireless Communication Module" by Nex Robotics Available in the Site <http://www.nex-robotics.com/products/wireless-devices/xbee-s2-2mw-wireless-communication-module-wire-antenna.html>.
- [17] "XBee S2C Low-Power Module, with Wire Antenna for ZigBee Mesh networks" by Hacktronics India Available in the Site <https://hacktronics.co.in/2-4-ghz-wireless-sensors/xbee-s2c-low-power-module-with-wire-antenna-for-zigbee-mesh-networks>.
- [18] Anshul Agarwal, Mukesh Agarwal, Manju Vyas, Richa Sharma, "A Study of Zigbee Technology", *International Journal on Recent and Innovation Trends in Computing and Communication*, ISSN 2321 – 8169, Volume: 1 Issue: 4, IJRITCC | APR 2013.
- [19] Andreas Goranson and David Cuartielles Ruiz, "Professional Android Open Accessory Programming with Arduino", Published by John Wiley & Sons, Inc. , Canada, 2013.
- [20] Datasheet of Arduino Uno, "Arduino UNO" Available in the Site <https://www.arduino.cc/en/main/arduinoBoardUno>, 2017.