
DEVELOPMENT OF A LOW POWER CONSUMPTION SMART EMBEDDED WIRELESS SENSOR NETWORK FOR THE UBIQUITOUS ENVIRONMENTAL MONITORING USING ZIGBEE MODULE

By

Ajao A. Lukman, Agajo J., Kolo J. Gana, Inalegwu C. Ogbale, and Edem E. Ataimo

Department of Computer Engineering,
Federal University of Technology, Minna, Nigeria

Email: ajao.wale@futminna.edu.ng

ABSTRACT

An emerging ubiquitous technology has called for the development of a Smart Embedded Wireless Sensor Networks (SEWSN), which has gained a tremendous attention over the years but has a shortcoming including power consumption, mobility, and end-to-end communication. The ability of environmental conditions monitoring is fundamental to research about climate variability in the greenhouse, gardens, zoology, pharmaceutical process and others. Being able to document a baseline and changes in environmental parameters monitoring and weather condition in a real-time at a remote location is increasingly essential which has not been addressed totally. In this paper, we proposed the development of an experimental Smart Sensing Platform (SSP) for a real-time monitoring of environmental parameters using ZigBee module (IEEE 802.15.4). It also introduces an approach to achieved low power consumption in a wireless sensor system. The embedded system consists of a digital humidity and temperature sensor (DHT11) for acquiring the environmental parameters, the XBee module for RF transmitter and receiver, ATmega328 for control unit and so on. The practical results show that the system achieved a real-time data acquisition, efficient energy management and ends-to-ends communication, which capable of storing data and dynamically plot the graphical information for statistical analysis. The transmitting current of the module is determined to be 25.8 (mA), the sleeping mode current is at 1 (μ A) and the current used in listening mode was 31.5 (mA).

Keywords: ATmega328, Greenhouse, Wireless Sensor Network, Ubiquitous technology, ZigBee module

INTRODUCTION

The implementation of an intelligent and automation system has improved the human livings in this era compared to the last decade by transforming several activities that are labour intensive, time wastes and others into simple touch control and remote monitoring using embedded system wireless technology [Aboaba et al., 2015]. The development of an embedded system based wireless sensor networks (WSNs) using different physical and media access control layer technology (such as ZigBee, Z-Wave, Bluetooth and Wi-Fi) in the internet of

thing architecture has play significant roles of applications in the environment monitoring and surveillance, domestic home appliances control, medical systems, pharmaceutical process and robotic exploration [Liu L. et al., 2012].

The Internet of Things (IoT) is a universal network technology that allows things to be connected and communicate to each other over the 6LowPAN platform. It enables things, devices to be monitoring and control of the physical environment by collecting data, processing

data, and analyzing the data generated by sensors or smart objects.

In the various areas of application of these systems, power consumption and others has been a challenge and treat to the sensor node deploy into the field for monitoring and others functionality. Although, the use of battery for sensor nodes powered is greatly eases deployment of the nodes in the constrained environment or hazardous area where the applications are required through wireless network, but the battery replacement shows to be unpractical. This is one of the focus and concerns of the researcher to discover different techniques for low energy consumption to improve the life-span and for the longer lifetime of the sensor node [Xiao F. et al., 2012 & Jia J. et al., 2013].

The duty cycling techniques is implemented at media access control (MAC) layer of internet of things based network stacks, and is widely adopted to reduce energy consumption in wireless sensor network architecture. In this method, sensor nodes are configured to periodically alternate between the active states (Listening) and sleep states (Idle). When the sensor nodes are in active states for data transmission or receiver, the radio will put in sleep mode or turn off to save more energy. These concepts lead to the motivation and development of different philosophy on a succession of MAC protocols including Sensor MAC (SMAC) [YeW. et al., 2004] and Timeout MAC (TMAC) [Van, D. T., & Langendoen K., 2003], Wireless sensor MAC (WiseMAC) [El-Hoiydi, A. & Decotignie J. 2004], Compressed Handshake MAC (CH-MAC) [Li, Z. et al., 2015] and many others.

ZigBee is a low power, low rate wireless communication standard for Personal Area Network based on IEEE 802.15.4 standard. It is widely applicable in the wireless sensor network based internet of thing (WSN-IoT) architecture for the control, monitoring, and end-end communication in the different platform of smart technology such as in the home automation, Pharmaceutical process, industries, healthcare

monitoring, agriculture, environmental climatic changes and energy saving in building environments. ZigBee wireless technology is a low-cost device, low data rate of about 250Kbps, low energy consumption and low bandwidth (10 -100m), which operates on two frequency bands. The 868/915 band support 20-40 kb/s data rate and 2.4GHz frequency band that support 250kb/s data rate. The media access control protocol based on layer two of IEEE 802.15.4 standards can be configures in two different ways, which are beacon enable mode or beaconless mode [Eduardo, C. et al., 2010]. In the beacon mode configuration, the coordinator node will broadcast a frame periodically to identifying the matching node and allows the synchronization of packets to the neighboring sensor nodes. The interval between the two beacons in a network are varied from 15 milli-seconds to 252 seconds. This interval of synchronization is governed and slotted by carrier sense multiple access/collision avoidance (CSMA/CA) in a contention access period (CAP). It allows communication between the coordinator and router node. In the contention free period (CFP), the time slot is directly set aside for the assigned neighboring nodes to guarantee quality of services during searching of application time. So, beacon mode is mostly suitable and recommended for the scenario of the wireless sensor network based layer two ZigBee that including coordinator and the intermediate routers in a multi-hop cluster-tree of IEEE 802.15.4/ZigBee that are powered by the batteries [Eduardo, C. et al., 2010].

The beaconless mode for layer two configuration is the most suitable applications for typical wireless sensors network, where the coordinator (ZigBee transmitter module) is powered by the main source (battery) in the deployed field environments. This will help the Coordinator to maintain and control its radio receiver every time is ready to communicate with any sensor node. Due to the permanent activity of the Coordinator allows the clients to be inactive (in the sleep mode) for unlimited periods of time in order to save more energy. Therefore,

the sensor motes can wake up at their own configuration interval of time, either on a periodical or event-driven basis for the essence of sensed data to be transmitted to the Coordinator.

Conclusively, most of the commercial IEEE 802.15.4/ZigBee motes does not support beacon mode implementation and configuration, in this laboratory experimental research work of low power consumption for smart embedded wireless sensor network for the environmental monitoring using ZigBee module will adopt beaconless networks techniques.

In [Krishna, Y. B. & Nagendram, S. 2012], a low powered RF ZigBee based voice control system for home automation was developed. In this research, the ZigBee receives the voice command as input and then sends the data to an ARM9 controller which converts the input data to the required format. After the conversion ARM9 controller sends the data to the ZigBee module for the transmission to the base station. The proposed voice command system is simply converted into digital form and then transmit serially as a packet through the wireless mechanism. At the receiver end, the digital information is reconstructed into the voice and send to the computer through a sound card. The Visual Basic programming on a computer uses Microsoft Speech API library for the voice recognition. After voice recognition, the system generates the control characters which is used to switch OFF/ON home appliances.

A system which uses biological sensors to monitor the health condition of patients remotely was discussed in [Anusha, K. & Balakrishna K. 2013]. The system consists of the patient module and manager module. The pressure, heart beat and temperature parameters that was acquired by the biological sensors were sent to a remote personal computer (PC) continuously through a wireless technology device (ZigBee) interfacing with microcontroller chip. The Manager can be informed about the health measurements through GSM module with available internet facilities.

METHODOLOGY

The methods adopted for this experimental development of low power consumption for smart embedded wireless sensor network for the ubiquitous environmental monitoring system using ZigBee module are comprises of several facilities integrated together to make IoT architecture functioning (which includes embedded technology, network technology and information technology). Although, they are popularly categorized as hardware system and the software system.

The firmware code was designed in the Arduino software IDE and the result is view in the Arduino serial monitor. The Arduino Atmega328 consists of 15 parallel pins which are arranged as (TX1-D12) at one end, and (VIN-D13) at another end. The TX and RX pin of XBee ZigBee transmitter and receiver module are connected accordingly to the pin 1 & 2 (TX/DI & RX/DO) of Arduino Nano respectively for appropriate communication. Also, the DATA pin of DHT11 is connected to the pin8/D8 of ATmega328 transmitter module, the GND and VDD pin are grounded and connected to power respectively. The details connections of LCD module with others components involved are as depicted in the figure 1 of the Proteus (VSM) 8.0 circuit diagram.

The mathematical expression of several facilities and the services integrated in the ubiquitous architecture are given in the equation 1.

$$\sum_{k=1}^n f(U_{system}) = \sum_{k=1}^n \sum_{i=1}^k (E_{st}) + \sum_{k=1}^n \sum_{i=1}^k (N_{st}) + \sum_{k=1}^n \sum_{i=1}^k (I_{st}) \quad (1)$$

where,

$\sum_{k=1}^n f(U_{system})$ is the summation and integration of

several ubiquitous facilities functioning in the system architecture (embedded system, network facilities and

programming languages), $\sum_{k=1}^n \sum_{i=1}^k (E_{st})$ is the summation and integration of different embedded system technology in the ubiquitous architecture (such as Atmega328, sensor nodes), $\sum_{k=1}^n \sum_{i=1}^k (N_{st})$ is the summation and integration of different network facilities used in the ubiquitous architecture (includes ZigBee (Tx & Rx), Bluetooth, Wi-Fi) and $\sum_{k=1}^n \sum_{i=1}^k (I_{st})$ is the summation and integration of different information system technology in the ubiquitous architecture (e.g JNL, micro C, MySQL and protocols like CoAP, HTTP etc).

The ZigBee Module Configuration

In the design and configuration of XBee module for the IoT platform, it requires two modes of settings which are "Coordinator" and "Router". The transmitter (Tx) module is configure as Coordinator and receiver (Rx) shield is identified and configure as Router before communication can take place between the two shields. The wireless ZigBee UART serial communication port and Future Technology Devices International (FTDI) USB-to-serial chip is used to transmit and receive data between the shield and Arduino ATmega328. This network architecture consists of one Coordinator (master) and many Routers (node). The Router's PAN ID was set to the same PAN ID of the Coordinator for the easy network connection. Hereafter, The TTL USB Serial cables was used to provide connectivity between Arduino USB and serial UART interfaces of XBee module during configuration and PL-2303HX USB-TTL driver was installed. Also, the ZigBee module configure V5I software was installed to allow the communication between the remote station and controller (ZigBee modules 'Rx and Tx').

The steps taken for both XBee ZigBee transmitter (Tx) called coordinator and receiver (Rx) called routers are outlined as follows:

1. The Tx of the ZigBee module was connected to Rx of USB-TTL converter, and the Rx of ZigBee module is connected to Tx of USB-TTL converter. The GND of ZigBee module was connected to GND of USB-TTL, PWR pin of ZigBee module is connected to 3.3V of USB-TTL converter as shown in the figure 1 "(a)" and "(b)".
2. The USB-TTL converter was connected to the computer and the ZigBee Module Configure V5I software was launched to begin configuration.
3. The Coordinator setting was selected, and then restart the XBee ZigBee module.
4. The PAN ID coordinator is change and set to the users' choice values (such as 0x1234) (range: 0x0001-0xFF00). This is to prevent conflicts with the default values of XBee module (default is 0x199B), then put the module OFF.
5. Repeat the procedure 1 and 2 for the other router (XBee ZigBee Rx).
6. The PAN ID of the router is change to the same value as in step 4, and search for the network automatically, then put the module OFF (note that by default all ZigBee module is configured as a router).
7. Finally, ON the Coordinator mechanism (Tx) first, followed by the Router module (Rx) immediately to automatically join the network. For the router to accept the new PAN ID, it must be connected to the network. The figure 2 depict the block diagram of logical configuration of ZigBee module procedures and figure 3 shows the simulation of the system circuit diagram.



Figure 1: A:Pin configuration of USB-TTL Converter

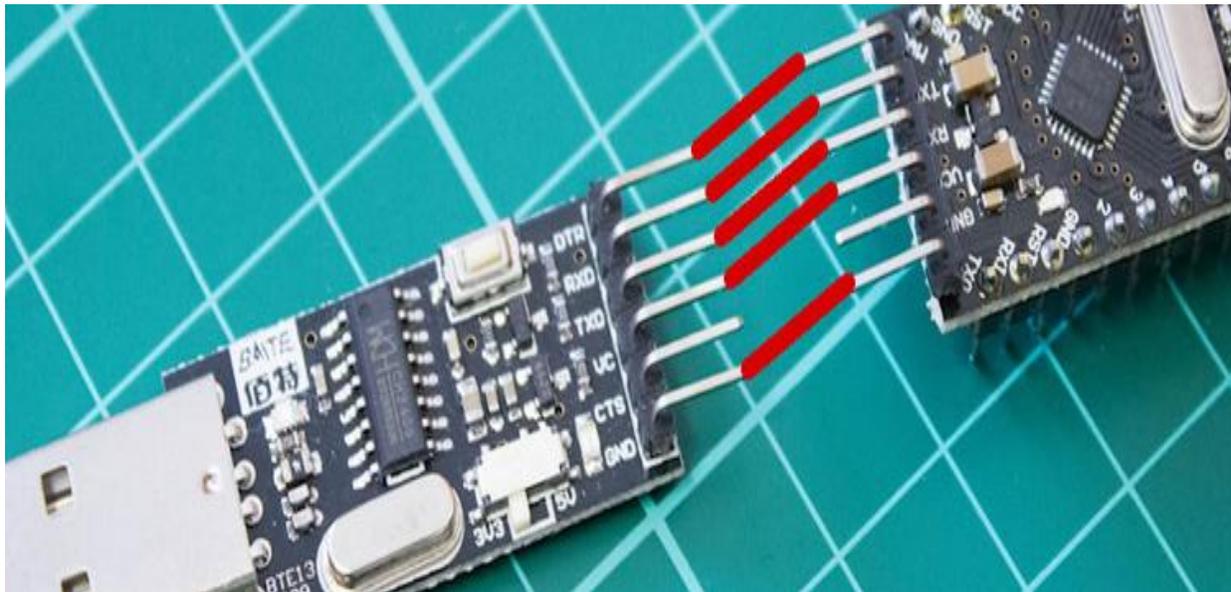


Figure 1B: Snapshot of USB-TTL Connection

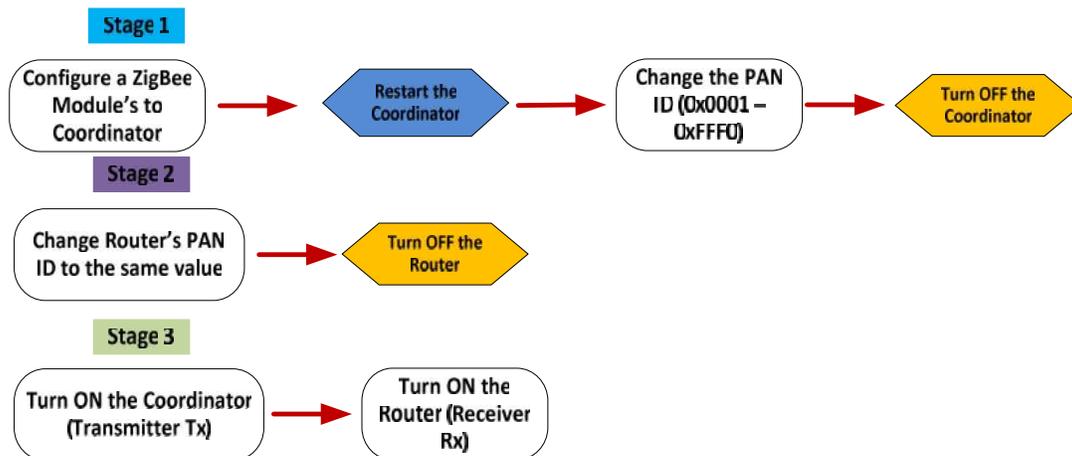


Figure 2: Logical Configuration of ZigBee Module

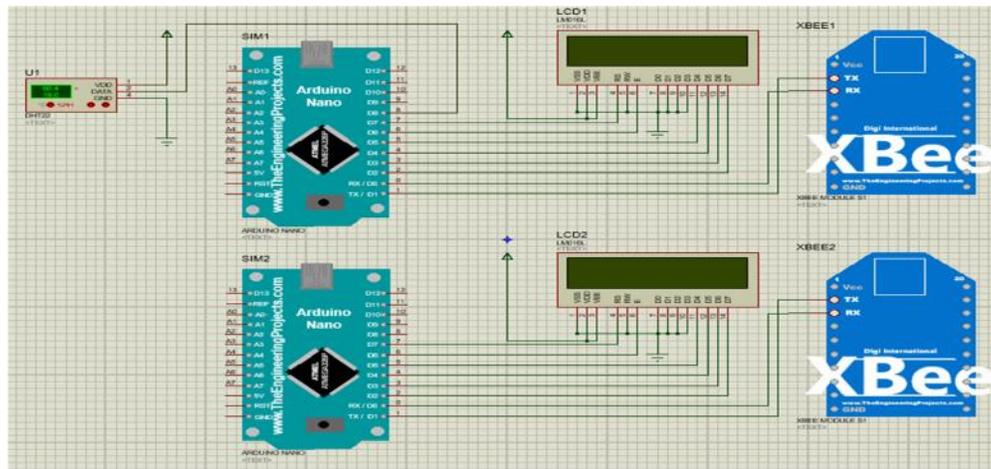


Figure 3: Proteus virtual simulation modelling of XBee transmitter and receiver circuit diagram.

Hardware System Design

The hardware module comprises of several components (such as Atmega328 based Arduino development board, the Xbee ZigBee RF transmitter and receiver (DRF1605H), DHT11 temperature and humidity sensor, 2*16 Liquid Crystal Display (LCD), PDA or Laptop computer and others supportive active and passive devices) which all are interfacing and control by Arduino ATmega328 microcontroller based development board.

The ZigBee transmitter module consist of a digital humidity and temperature sensor DHT11 for sensing the

temperature and humidity of its immediate environment, a liquid crystal display LCD used for displaying a real-time information, as well as the ZigBee module for receiving the data being transmitted over the network. The hardware architecture of both transmitter and receiver modules are depicted in figure (4a & 4b), and figure 5 shows the flowchart of the hardware system implementation and functions.

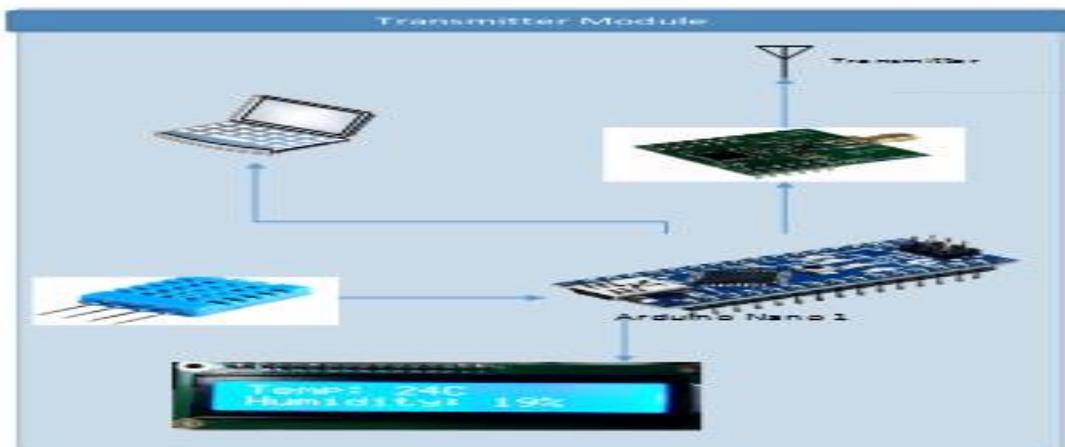


Figure 4a: Transmitter module system

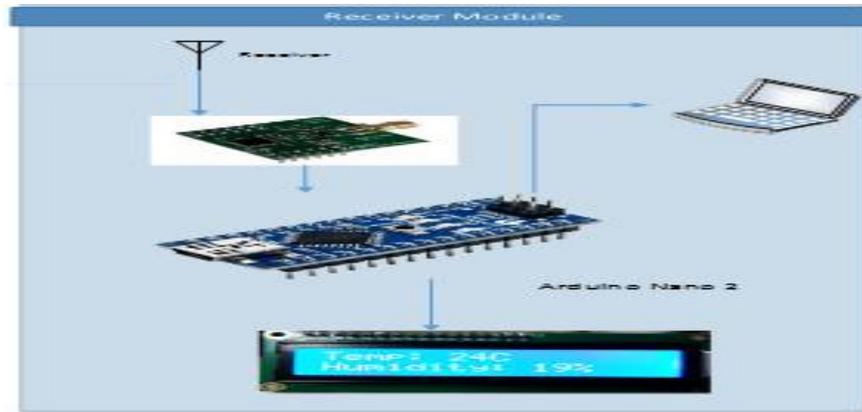


Figure 4b: Receiver module system

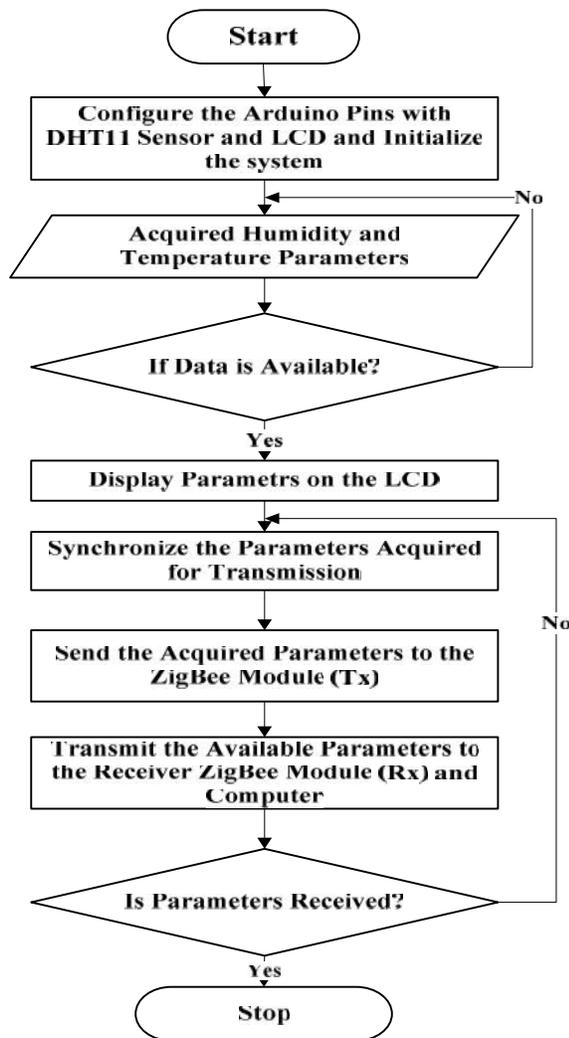


Figure 5. Hardware system configuration flowchart.

2.2.1 Arduino Nano (ATmega328)

The Arduino Nano is a new technology development board embedded with ATmega328 microcontroller chip, very portable with many functions includes 32KB flash memory, 2KB SRAM, 16MHz clock speed, 8 analog input/output pins, 22 digital input/output pins, 6 PWM and power consumption is 19mA. This device can be powered with operating voltage (3.3 – 5V), it has an internal pull-up resistor of 20-5kΩ and can allow input voltage range (7-12V) with operating DC current of 40mA. The serial ports pin 0 (Rx) and 1 (Tx) are used to receive and transmit TTL serial data by using FTDI USB-to-TTL serial chip. It consists of external interrupts pins 2 and 3 which can be configure to change an interrupt from rising edge to falling edge and vice-versa. The PWM pins are 3, 5, 6, 9, 10 and 11 used to provides 8-bit output Analog Write () functions. Also, the serial peripheral interface (SPI) communication pins are 10 (SS), 11 (MOSI), 12 (MISO), and 13 (SCK), while other pins support I2C (TWI) communication such as A4 (SDA) and A5 (SCL). The Arduino Nano architecture is shown in the figure 6.

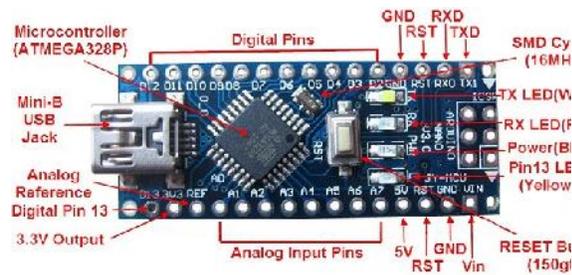


Figure 6. Arduino Nano architecture [Source: Arduino Nano, 2016]

Wireless ZigBee Module (802.15.4)

ZigBee is a wireless mesh network technology that is used for sensors communication based on the IEEE 802.15.4 protocol, this technology was designed for a remote control smart application in the IoT-based model such as smart home, smart agriculture etc. It operates at a frequency of 2.4GHz ISM, low data-rates (250kbps) over a constrained application of 10-100m range. It is robust and offers low-power operation of about 1µA, the transmit, Idle/receiving current of 40mA at 3.3V, high scalability using stochastic address assignment (SAA) with high node counts, high security using symmetric-key exchange (SKE) and is well positioned to take advantage of wireless control and sensor networks between machine-to-machine in the IoT applications. The XBee ZigBee RF Module is in receive mode when it is not transmitting data, which make the device to be shifted into either transmit, sleep or command mode used when the SPI port is not available.

DHT11 Sensor

The DHT11 sensor is a calibrated digital signal output device used for measuring of environmental temperature and humidity. This consist of three pins (such as V_{dd} for power, DO for data line connected to MCU and GND for ground), to ensure high reliability and excellent long-term stability as shown in the figure 2.18. The soil temperature level of hotness or coldness has great impact on soil precipitation, watering system and soil physical properties. Therefore, Temperature is measured in

degree Celsius (°C), degree Fahrenheit (°F) and Kelvin (K). Humidity is measured in percentage (%), where both of the physical parameters play an importance roles in seed germination and plant growth as optimum temperature is required for their growth and development.

DHT11 sensor use a single data-line format for one communication process that exist between the device and microcontroller unit (MCU) during synchronization. This sensor usually transmits a higher data of 40bit maximum which consists of integral and decimal value as expressed in equation (2).

$$DHT_{11} = 8 \int_n^{40} RH + 8RH_d + 8 \int_n^{40} T + 8T_d + 8CS \quad (2)$$

where, DHT₁₁ is the data format for the humidity and temperature, $8 \int_n^{40} RH = 8$ bit integral relative humidity

(RH) data, $8RH_d = 8$ bit decimal relative humidity (RH)

data, $8 \int_n^{40} T = 8$ bit integral temperature (T) data, $8T_d =$

8bit decimal temperature (T) data, $8CS = 8$ bit check sum.

Analog-to-Digital Converter (ADC) Module

An Analog-to-Digital Converter (ADC) can be describe as a feature used to convert an analog voltage level on a microcontroller pin to its equivalent digital value or number. The analog pins connection available on the Atmega328 microcontroller can be described as (AD - A7). Although, ADCs can vary greatly between various categories of microcontroller, but the ADC on the Arduino is a 8-bit analog-to-digital conversion (ADC). That is, it has ability to detect ($2^8 = 256$ discrete analog levels) or 10-bit ADCs ($2^{10} = 102$ discrete levels) or 16-bit ADCs ($2^{16} = 65,536$ discrete levels).

The simple technique that ADC adopt to achieve the operation is that, the analog voltage supply charges up an internal capacitor it measures the time it takes to discharge across an internal resistor. Then, the

microcontroller is used to monitors the number of clock cycles that pass before the capacitor is discharged. This number of cycles is coded as a returned digit during completion of ADC process. The ratio-metric value of ADC can be expressed as in equation (3).

$$\frac{ADC\ resolution}{System\ Voltage} = \frac{ADC\ reading}{Analog\ voltage\ measured} \quad (3)$$

Since, analog to digital conversions are system voltage dependent, then we predominantly use the 8-bit ADC of the ATmega328 Arduino board that operate on 3.3-volt system. Therefore, it can be simplified as given in "(4)";

$$\frac{255}{3.3} = \frac{ADC\ reading}{Analog\ voltage\ measured} \quad (4)$$

For instance, if the analog voltage reading is 0.25V, then ADC can be calculated as expressed in "(5)";

$$\frac{255}{3.3V} = \frac{y}{1.25V} \quad (5)$$

Software System Implementation

The Software design section includes the main functions of the system designed in the Arduino IDE for firmware (ATmega328) coding and the application programming Interfaces (APIs) for the data storage and remote users analysis. The remote user application programming interfaces (APIs) was designed to create a database for the environmental parameters monitoring in a real-time basis and plot the graph of parameters acquired against real-time dynamically for the users analysis using java native interface (JNI) like NetBeans 8.0 IDE which allows java framework. The Java Native Interface (JNI) is a program design outline that supports running of Java code and allows java native applications and libraries written in another programming language (such as mnemonics code, C and C++) to be called severally for their active roles on a Java Virtual Machine (JVM).

The application programming interface (API), was designed in JavaFX using JRE 8.0 version for the system design. It also requires Rx/Tx serial API together with DLL file in other to recognize the hardware attachments such

as Arduino Nano (ATmega328) and both Xbee ZigBee RF module (Tx & Rx).

The API coding in NetBeans 8.0 IDE was perform using Java programming language for the graphical interface, (Arduino.serialList) for serial port communication. This allow the users to access data acquisition remotely with the help of wireless technology (Xbee ZigBee module) and plot dynamic graph for the DHT11 parameters against real-time for details analysis. The data acquisition flowchart is depicted in the figure 7.

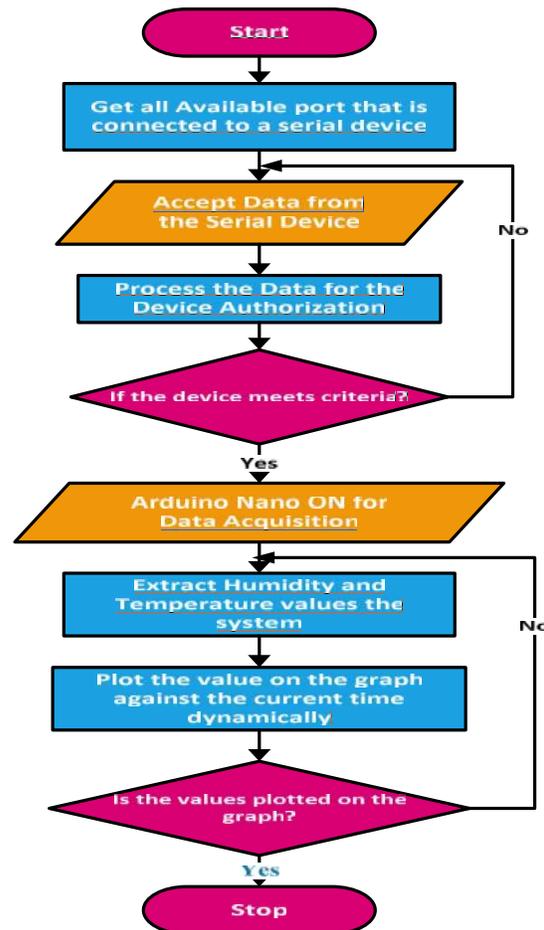


Figure 7. Flowchart for the APIs and dynamic graph plotted

Power Consumption Configuration

The Xbee ZigBee module, both transmitter (Tx) "Coordinator", the receiver (Rx) "Router" and end devices (DHT11 sensor) are configured at the physical and media access control layer (MAC) to reduced power

consumption during communications of the system in a wireless personal area networks (WPAN). The general framework of IEEE 802.15.4 has capacity of transmitting data over a minimum distance of 10m to 100m at the rate of 250Kbps or reduced to 20Kbps at the frequency of 868/915MHz to achieved extremely low power consumption in the CC2530 ZigBee module.

When the system is powered, it will initialize by put the end device (sensor and microcontroller) to the listening mode, the ZigBee transmitter Tx (Coordinator) is in a deep sleep mode for a period of time and receiver Rx module (Router) is in off mode. After the elapsed listening time of end devices, it will synchronize the messages and activate the coordinator to be in a listening mode for the packet transmission (P_{tx}) as well as the Router will be initialized to receive the packet sent. After this process done, the Coordinator will be put into the sleep mode and the Router still in listening status to complete its cycle (frame relay) until all packet received and displayed before moving to the off status as shown in the figure 8. The purpose of this method or techniques is to maximize and prolong lifetime of the sensor motes battery and wireless personal network area architecture.

The figure 9 show the power supply connection with digital multi-meter, and the battery life expectancy can be calculated as expressed in "(16)".

From this process, it is discovered that more energy is expended during the period of packet receiving than in sleep mode and packet transmission period as expressed in equation "(6)", "(7)" & "(8)" [Luca, B. 2007]

$$P_{rx} \gg T_{sleep} \gg P_{tx} \tag{6}$$

$$P_{tx} \ll T_{sleep} \ll P_{rx} \tag{7}$$

$$P_{rx} = \frac{P_{tx}}{T_{pkt}} \tag{8}$$

Then, T_{pkt} = packet transmission time (s) which is assumed to be 100ms as worst setting.

The energy cost for receiving packet (P_{rx}), energy cost for sleep mode (E_{sleep}) and energy cost for transmitting packet (P_{tx}) can be mathematically expressed as in equation "(9)", "(10)", & "(11)"

$$E_1 = P_{rx} dT_f \tag{9}$$

$$E_2 = T_{sleep} [T_f (1 - d) - NT_{pkt}] \tag{10}$$

$$E_3 = NP_{tx} \tag{11}$$

Also, the total cost of efficient energy management configuration can be expressed as in "(12)" & "(13)".

$$E_{cost} = E_1 + E_2 + E_3 \tag{12}$$

$$Ec = P_{rx} dT_f + T_{sleep} [T_f (1 - d) - NT_{pkt}] + NP_{tx} \text{ (mAh)} \tag{13}$$

where,

Ec = Energy cost, P_{rx} = receiving cost (mA), d = duty cycle, T_f = Message frame interval (s),

N = Number of sensor node neighbours, T_{sleep} = Sleeping time (mA),

P_{tx} = Single packet transmission costs (mAh), T_{pkt} = Synchronization packet time length (s)

Therefore, the changes between the states are synchronous which is expressed as in "(14)", and the duty-cycle is given as in "(15)".

$$T_f = T_i + T_s \tag{14}$$

$$D_c = \frac{T_i}{T_i + T_s} \tag{15}$$

where

T_f = Message frame period, T_i = Listening period or idle mode

T_s = Sleeping period or power saving mode, D_c = Duty cycle

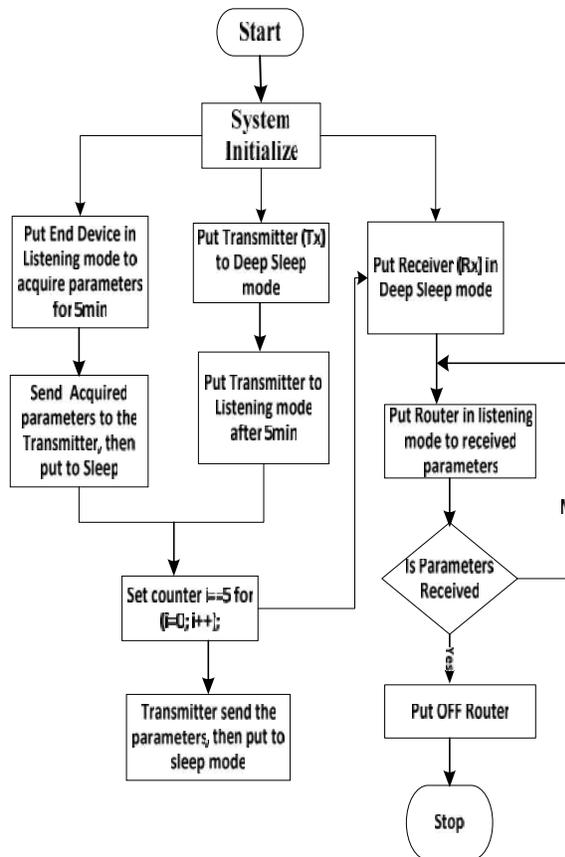


Figure 8. Flowchart diagram for energy management configuration

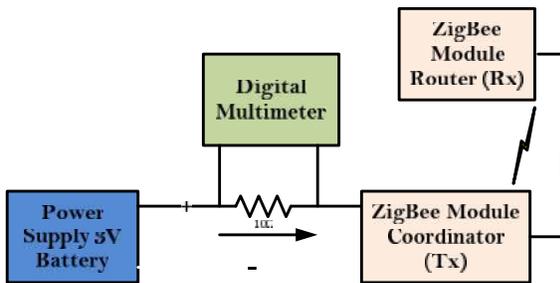


Figure 9. Energy consumption experimentation

$$Battery_{lifetime} = \frac{Battery_{capacity} (mA/h)}{Load Current (mA)} * 0.70$$

(16)

RESULTS AND DISCUSSIONS

The main goal of this research experimentation is to developed and implement a wireless sensor network based ubiquitous architecture for a real-time data acquisition of the environmental monitoring using ZigBee modules. Also, to configured the system at media access control layer for low power consumption and to improve end-to-end communication.

System Implementation

In this research work, several components (both hardware and software) has been put together to achieved the environmental monitoring system in a real-time based on wireless sensor network and technology using ZigBee module. The system was implemented and tested during the experiment which is depicted in the figure 10 “(a)” an “(b)”, and the results of environmental parameters (temperature and humidity) are presented in the table 1. Also, a dynamic plotted graph based on the environmental data acquisition in a real-time basis are presented in the figure (11-13). Although, the environmental reading was taken within 60 minutes interval for 6 hours to ensure the accuracy of the system performances, and the average values of both temperature and humidity was calculated as shown in the table 1.

The IEEE 802.15.4/ZigBee layer two protocol render a wide range functions and opportunities in the wireless sensor network based internet of things, which initiates the configuration of the Medium Access Control (MAC) sublayer to achieved low power consumption during transmission (Tx), sleeping mode and receiving (Rx) mode, and also assist greatly in end-to-end communication of devices, interoperability’s, and scalability. In this sense, the 802.15.4 standard analyze two classes of motes in the ubiquitous platform. The Full-Function Devices (FFD) and the Reduced-Function Devices (RFD). The FFDs is recognize as Coordinators in the

network (Transmitter) which responsible for the communications of a set or “cluster” of nodes or neighbor nodes using star topology. While the role of RFD (Receiver) in the network is reserved for simple devices with some limited resources in order to permits the end-to-end communication nodes with just one FFD acting as its Coordinator.

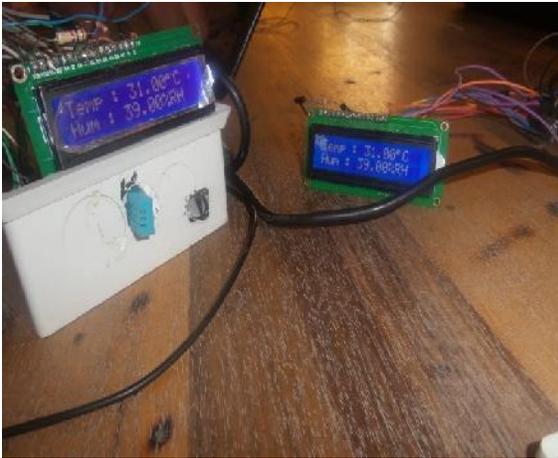


Figure 10a. ZigBee (Tx and Rx) module system soldering



Figure 10b. ZigBee (Tx and Rx) module system packaging

Table 1: Results of environmental parameters (temperature and humidity)

Real-Time Measurement (PM)/Hour	Temperature Values (°C)	Humidity Values (%)	Average Temperature Value (°C)	Average Humidity Value (%)
12:00	30, 31, 32,	60, 57, 55	31.00	57.33
1:00	34, 33, 30	72, 70, 69	32.33	70.33
2:00	33, 33, 32	69, 68, 69	32.67	68.67
3:00	31, 30, 31	57, 60, 61	30.67	59.33
4:00	29, 30, 28	56, 57, 50	29.00	54.33
5:00	29, 28, 29	52, 49, 48	28.67	49.67
6:00	28, 29, 27	58, 50, 47	28.00	51.67



Figure 11. Dynamic plotted graph for current temperature (°C)



Figure 12. Dynamic plotted graph for current humidity (%)

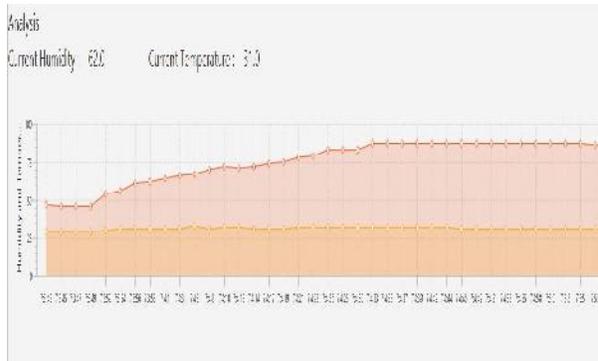


Figure 13. Composite dynamic plotted graph for current temperature (°C)

Battery Life Expectancy for the ZigBee-based WSN

In the laboratory experimentation of the ZigBee based wireless sensor network for the efficient energy management, the battery life time was carried out and analyzed. The power consumption of the ZigBee module and the data transmitted was tested using digital multi-meter with 1KΩ resistor connected in series to the 3VDC battery for the accurate estimate of voltage drop and to estimate the ZigBee consumption current since both of the transmitter and the receiver are adequately configured to sleep mode when the end device are listening. The table 2 shows comparison power consumption measurement with others existing works in the datasheets.

Table 2. The comparison of experiment performed for the current consumption measurements with others different ZigBee transceivers

Characteristics	Measurement (CC2530)	CC2480+MSP430 [Eduardo C. 2010]	TI CC2420 [Texas, 2010]
Operating Voltage (V)	3.6	3.6	21-3.6
Power Output (dBm)	0	0	-24 to 0
Sleep mode Current (µA)	1	0.75	1
Transmission Current (mA)	25.8	30.5	17.4
Listening mode Current	31.5	32.5	18.8

Characteristics	TI CC2480 [Texas 2010]	TI CC2520 [Texas 2010]	EM260 [Ember, 2010]	JN5121 [Jennic, 2010]
Operating Voltage (V)	2.0-3.6	1.8-3.8	2.7-3.6	2.1-3.6
Power Output (dBm)	0	Up to 5	0	-32-4.5
Sleep mode Current (µA)	0.5	0.12 (max)	3.5	1
Transmission Current (mA)	27	25.8	49	28-30
Listening mode Current	27	22.3	44	28-34

CONCLUSION

In this paper, we proposed the development of a low power consumption smart embedded wireless sensor network for the ubiquitous environmental monitoring and real-time data acquisition using ZigBee module. The system consists of two stations, the "Coordinator" that consists of "end devices", which collects the humidity and temperature parameters in a real-time basis, encode it and stored in its memory before transmit to the base station (ZigBee transmitter module). The "Router" received the same parameters send from the coordinator and compared to be accurate with the transmitter (Tx) parameters, which is then plotted the graph dynamically in the NetBeans IDE coded with Java Native Languages (JNL). This work shows that the coding of JNL with layer two wireless technology can be helpful to monitor and achieved real time data acquisition remotely with low energy consumption. The media access control (MAC) configuration for low power consumption and end-to-end communication has shown to be effective with a practical and simple formula that permits to forecast the battery lifetime of a sensor mote as a function of the duty cycle and the size of the data to be transmitted. Also, this research shows the analysis of the current consumption in Milliamp (mA) of others ZigBee devices with the CC2530 transceiver and Atmega328 microcontroller.

The future work will employ more field installation of sensor nodes connecting to the base station, allowing other parameters to be acquired and monitored in real time with energy consumption rate. Also, the battery life expectancy will be fully justified based on the current consumption rate (mA) per packet transmission in a second, so as to determine the durability periods (Days) of the battery when several packets transmitted per seconds in a day.

REFERENCES

[1] Aboaba A. A. Amoo L. A. & Ajao L. A., (2015). "Design of an Automatic Temperature Controlled Switch for Air-

Conditioning Systems", ATBU, Journal of Science, Technology & Education (JOSTE); Vol. 3 (2), pp. 157-168.

[2] Liu, L., Wang, R., & Xiao, F. (2012), Topology control algorithm for underwater wireless sensor networks using GPS-free mobile sensor nodes, Journal of Network and Computer Applications, vol. 35, no. 6, pp. 1953-1963.

[3] Xiao, F., Wu, M., Huang, H., Wang, R., & Wang, S. (2012), Novel node localization algorithm based on nonlinear weighting least square for wireless sensor networks, International Journal of Distributed Sensor Networks, vol. 2012, Article ID 803840, 6 pages, 2012.

[4] Jia, J., Zhang, G., Wu, X., Chen, J., Wang, X., & Yan, X. (2013), On the problem of energy balanced relay sensor placement in wireless sensor networks, International Journal of Distributed Sensor Networks, vol. 2013, Article ID 342904, 9 pages, 2013.

[5] Ye, W., Heidemann, J., & Estrin, D. (2004), Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactions on Networking, vol. 12, no. 3, pp. 493-506.

[6] Van, D. T., & Langendoen, K., (2003), An adaptive energy-efficient MAC protocol for wireless sensor networks, Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys'03), pp. 171-180, ACM, Los Angeles, Calif, USA, November 2003.

[7] El-Hoiydi, A., & Decotignie J., (2004), Wise MAC: An Ultra-Low power MAC protocol for multi-hop wireless sensor networks, ALGOSENSORS, LNCS 3121, pp. 18-31,

[8] Li, Z., Chen, Q., Zhu, G., Choi, Y., & Sekiya, H., (2015), A low latency, energy efficient MAC protocol for wireless sensor networks, International Journal of Distributed Sensor Networks, pp. 1-9.

[9] Eduardo, C., Jose M. C., & Gonzalo C., (2010), Modeling of current consumption in 802.15.4/ZigBee Sensor Motes, MDPI Sensor, pg.5443-5468, doi:10.3390/s100605443.

[10] Krishna, Y. B., & Nagendram, S., (2012), Zigbee based voice control system for smart home, international

journal computer technology & applications, vol. 3, no. 1, pp.163-168.

[11] Anusha, K. & Balakrishna K., (2013), An adaptive embedded system for monitoring patients, International Journal of Application or Innovation in Engineering & Management (IJAIEM), Volume 2, Issue 7.

[12] Arduino Nano, (2016), Revision history of Arduino Nano ePro Labs, Retrieved April, 2016, from <https://wiki.eprolabs.com/index.php>.

[13] Luca, B., Davide, D. P., Giovanni C., & Gianfranco M., (2007), Wireless sensor networks for on-field agricultural management process, Nestens S. r. l., Italy, pg. 1-18.

[14] Texas Instruments, CC2420 Single-Chip 2.4 GHz IEEE 802.15.4 Compliant and ZigBee Ready RF Transceiver. Retrieved January 28, 2010, from <http://focus.ti.com/docs/prod/folders/print/cc2420>.

[15] Texas Instruments, CC2480A1 Status (Z-Accel 2.4 GHz ZigBee Processor) datasheet. Retrieved January 28, 2010, from <http://focus.ti.com/docs/prod/folders/print/cc2480a1>

[16] Texas Instruments, *CC2520 (Second generation 2.4 GHz ZigBee/IEEE 802.15.4 RF transceiver)*. Retrieved January 28, 2010, from <http://focus.ti.com/docs/prod/folders/print/cc2520>.

[17] Ember Corporation, EM260 802.15.4 and ZigBee Compliant Network Co-processor datasheet. Retrieved January 28, 2010, from <http://www.ember.com/pdf/ember-EM260.pdf>.

[18] Jennic, Application Note: JN-AN-1001, Calculating JN5121 Power Consumption. Retrieved January 28, 2010, http://www.jennic.com/files/support_files/JN-AN-1001-Power-Estimation-lv3.pdf.