Design and Implementation of Wireless Sensor Network for Wildlife Monitoring (May 2009)

N.V. Pillay [1] and T. Muddenahalli N [2]

***Abstract* – Wireless Sensor Networks (WSN) is a popular and interesting field of research in the study of wildlife. In this paper, the design of a WSN that will be used to remotely monitor wildlife is proposed. The sensor nodes designed were small, low powered units to allow them to be mounted onto animals. Each sensor node was designed to monitor the body temperature of the animal it was mounted to and the light intensity of the animal's external environment. This sensed data was then relayed to a base station node for further processing. A user application program running on the base station computer was developed to receive the data from the base station node and store it in a structured text file format for further analysis. Both the hardware and software designs of the WSN and the data sample collected from the sensor nodes relating to temperature and light are presented.**

***Index Terms* — Graphical User Interface (GUI), Identification No (ID), Medium Access Control (MAC), Direct current (DC), Transistor-Transistor logic (TTL), Industrial Scientific & Medical (ISM), Serial Peripheral Interface (SPI), Open Systems Interconnection (OSI), Cyclic Redundancy Code(CRC), Universal Synchronous Asynchronous Receiver Transmitter (USART), Complementary Metal Oxide Semiconductor (CMOS).**

I. INTRODUCTION

WIRELESS SENSOR NETWORKS is an emerging field of research which combines sensing, communication and processing on a single device, called a sensor node [1, 2]. These nodes are often small in size, and their individual capabilities are limited (most likely in processing capability and power availability). As a result, nodes work together amongst themselves to allow certain phenomena that the WSN is monitoring to be recorded or acted upon.

II. A WIRELESS SENSOR NETWORK FOR WILDLIFE MONITORING

A. Application background

WSNs are becoming increasingly popular in the field of wildlife monitoring [3]. The goal of this design project is to develop a WSN to monitor the environmental conditions of an animal's habitat such as light intensity, as well monitor the body temperature of the animal itself, and then relay this data back to a base station for further processing. In this application of the WSN, nodes were mobile as they had to be attached to animals, namely cattle. Cattle were chosen for the prototype as they were less likely to damage the sensor units and could be interacted with more easily than carnivorous animals. Light conditions and the body temperature of the animal were chosen as the parameters to be monitored to research the effect of these parameters on the behaviour pattern of animal being studied. Other parameters such as location and movement are not discussed in this paper as the project was allocated a low budget and such parameters required higher cost equipment to be monitored.

B. System Functionality

Each sensor node will be mounted onto an animal. These nodes will sense light intensity conditions of the animal's environment and the body temperature of the animal it is mounted to. Nodes will be in standby mode to reduce energy consumption, only becoming active at certain times to perform measurements and forwarding of data across the network to the base station [4]. The received data will be analyzed to produce results on a specially designed, easy to use GUI at the base station.

C. System Specifications

1) Functional Requirements

- Light intensity and body temperature readings should be taken periodically and sent to the base station for further processing.
- Each node should be able to forward/relay data packets from other nodes out of range of the base station.
- Nodes should be self-configuring and have a unique ID to distinguish between each other.
- Each node should alert the base station when its battery is low.

2) Non-Functional Requirements:

- Nodes should be robust as they will be used in the environment/on wildlife.
- User-friendly and easy to understand interface at base station to process received data.
- Collars should be easy to fasten onto animal, and comfortable for animal to wear.
- Nodes should conserve energy as far as possible.

D. Physical Design of Node

Since the node was designed to be used on animals, it required an attachment mechanism. Each sensor node is thus encased in a hard plastic box (figure II-1), and attached onto a collar which is used to mount the node onto the animal. The light sensor is protected by a transparent covering and

[1] N.V. Pillay, the Department of Information Management, Eskom Enterprise Division, Sunninghill, RSA (email: nolan.pillay@eskom.co.za, nolanvictorpillay@gmail.com; phone: +27(0) 73 550 0894, +27(0) 11 800 2925; fax: 086 572 6946; postal address: P.O. Box 472, Verulam, 4340, South Africa).

[2] T. Muddenahalli N is currently an academic staff member at the School of Electrical, Electronic & Computer Engineering, University of Kwazulu Natal, Durban, RSA. (email: muddenahalli@ukzn.ac.za; phone: +27(0) 31 260 7008).

the temperature sensor is fitted on the inside of the collar so that it came into direct contact with the animal's body. Silicone is used to seal applicable parts of the sensor node to prevent them from being damaged in rainy weather.

Figure II-1: Sensor node prototype hardware

III. Sensor Node Design

A. Module Break-down

The architecture of each sensor node is depicted in Figure III-1[1].

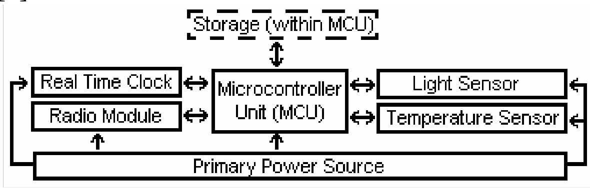

Figure III-1: Sensor node architecture

B. Architecture description

In this section, the architecture module of the design is briefly explained.

1) Communication Module

The *NRF905-C433* radio transceiver [5, 6, 12 and 13] which operates in the 433MHz ISM band was used on the prototype sensor node. Interfacing to the NRF905 was performed by the use of SPI between the device and the host microcontroller. The configurable power output level of the device was set to 6dBm to achieve a balance between a good range and moderately low power consumption [6].

2) Sensors

A low voltage light sensor and temperature sensor are used on each node. The purpose of the light sensor is to monitor the light intensity conditions of the animal's environment. This sensor is used to investigate the effect of varying light conditions on the behavioural patterns of the animals being studied. A Light Dependant Resistor (LDR) [8], configured in a potential divider circuit, was chosen for this sensing purpose as it was inexpensive, robust, simple to use, and responded to visible light conditions (that which can be seen by the human eye) with good sensitivity. The Analogue-to-Digital Converter (ADC), of the host microcontroller on each node, used to convert the raw analogue sensor reading into an 8-bit binary value which is then converted to a light intensity value at the base station using a lookup table. The light sensor was calibrated using the Lutron LX-105 Light Meter.

The DS18B20 temperature sensor was selected for use in each node. This sensor can operate on a low voltage (~3V) and offers an output of up to 12-bit resolution [9, 10], making it a very accurate sensor. In addition, due to its 1-wire® interface, it only requires a single port pin on the host microcontroller for communication. The raw output from the sensor was stored in two bytes and forwarded to the base station for further collaborative processing within the User-Interface program.

3) Processing Module

The processing module in each sensor node is the "brain" of the node. It comprises of a single 8-bit low-powered ATmega16L microcontroller [11] which was selected because of its low intrinsic power, various peripheral features and its easy-to-use instruction set. The ATmega16L has 512Bytes of onboard EEPROM, thus eliminating the need for any external storage device. During the design phase, a node's unique ID is stored here (section IV-B-1).

4) Real Time Clock

The DS1302 low-powered real time clock is used to log the time and date of each measurement on each sensor node [6]. The device is also used in the timing synchronization needed for data transfer between nodes (section IV-B-2).

5) Power Source

Each sensor node is equipped with a 9V alkaline battery which is regulated using a LM2905 3V regulator. The 3V output is used to power all components within the node and thus serves as the node's energy source. Each node has functionality to monitor its battery status using the onboard comparator of the microcontroller unit (MCU) to compare the node's battery voltage to the internal band gap reference voltage of the MCU. Battery status that is sent to the base station is one of two values, i.e. a "0" signalling that the battery is healthy, or a "1" signalling that the battery is low.

IV. Communication Protocol Specification

The Communication protocol is divided into four layers from the OSI model. These four layers are described below:

A. Physical Layer

The physical layer is responsible for inter-node communication. The radio transceiver used is responsible for how information is transmitted across the medium. Figure IV-1 shows the structure of each physical packet that is transmitted.

10-bit	32-bit	128-bit	16-bit
Preamble (NRF905)	Receiver Address (NRF905)	Payload	CRC (NRF905)

Figure IV-1: Structure of Physical Packet

The preamble is used by the radio transceivers to synchronize the sender and receiver's frequencies so that they may communicate. The receiver address is used by a receiving node to determine whether the data is for itself or not. The payload contains the data which is to be sent along with some other essential parameters as described in the MAC layer. The 16-bit CRC is automatically added by the radio circuitry to perform error checking [12, 13].

B. MAC Layer

The MAC layer is responsible for coordinating when nodes are to transmit and receive data and when they are to become active or go back to sleep.

1) Node Identification

In the proposed design, each node has a unique identification within the network. This is stored within the node's memory in the form of an 8-bit number. The number is assigned at the design phase and the node will keep this ID for the duration of its life. Since IDs are 8 bits in length, the network can accommodate a maximum of 256 nodes.

2) Node discovery & Synchronization

All nodes within the network will be mobile and thus the network topology is not fixed. Nodes will always have to perform node discovery to determine the next hop for data transmission. When a node switches to an active state, it will attempt to discover its neighbours. A form of time-slotting is used to realise this functionality [7].

In order to realise data transfers to the base station from each node, we assume the following:

- Nodes are fully mobile, and may move at any time
- Speed at which nodes move is relatively slow compared to the rate at which data transfer occurs.
- All nodes are willing to participate in data exchange.
- Each node possesses its own unique ID that is seen by all active nodes.

The real time clock on each node will help determine when a node is to take readings from its sensors and when it transmits this data to other nodes or the base station. Every half hour, on the real time clock, two cycles of 250ms periods occur. Each period is broken into ten 25ms timeslots. During each timeslot, only 2 nodes will be communicating and transferring data at a time. This will help reduce the congestion of the network and the number of nodes that are trying to communicate in each time slot.

Assume that a half hour has just occurred on the node's real-time clock. The first 25ms timeslot of the first 250ms period now begins. The node will then generate a random number to determine whether it goes into transmit or receive mode for the rest of that time slot. Nodes that are in transmitting mode will send out a broadcast packet to alert other nodes within their neighbourhood of their existence. After sending out this broadcast packet, these nodes will then wait for an acknowledgement (ACK) packet from a neighbouring node which is in receive mode.

Nodes in receive mode will listen for a broadcast packet from any transmitting nodes within their neighbourhood, and reply to this broadcast packet with an ACK packet. After sending this ACK packet reply, the receiving nodes will then wait for an ACK packet from the transmitting node. Once the transmitting node receives the ACK packet from the receiving node, it will then include in the ACK packet that it sends to the receiving node, a value representing the new frequency to which the two nodes will then communicate in. This will complete the three-way handshake that has to occur for nodes to begin any data exchange. Other transmitting nodes that have timed out waiting for the ACK packet will contend for the next time slot searching for reachable nodes to transfer data to.
After communication is complete between the paired nodes, both nodes step back to the original broadcast frequency and do not take part in slot contention during the rest of the two period cycles. These nodes will however, still listen for the base station broadcast packets so that they may send their sensed data to the base station in remaining time of the 2 period cycles.

During the next 9 timeslots within the period, other nodes follow the same procedure like the nodes that communicated in the first timeslot. The reduction in the number of nodes that contend for each time slot helps to reduce the possibility of collisions in the medium. If for some reason, a node wasn't able to communicate with another node during the first period cycle, it will try again in the second period cycle using the same rules as the first cycle. All nodes go back to sleep [14] after the second period cycle, to awaken on the event of the next half hour.
Each node will perform sensing measurements every half hour period. Thus a maximum of 48 readings will be taken per sensor node day. The real-time clock will help maintain timing synchronization across the nodes in the network. (Section C – 1).

The above protocol uses a mixture of routing concepts. Like the SPIN [15] protocol, a node sends data to its neighbours, but only the closest neighbour node (determined by quickest response to a broadcast). A receiving node will not send back data to a sending node on the next frame since it will remember the sending node's unique ID, and will not accept a handshake from a node with the same ID in the next frame. The receiving node sends the data to the base station when a connection is acquired.

3) Node discovery by base station

The base station will send out a broadcast every minute. Nodes within range will discover the broadcast packet and attempt to communicate with the base station. Nodes will use the listen-before transmit mechanism to communicate. This is made possible by the carrier detect pin on the NRF905 [12] which is set high by the radio whenever there is a carrier in the medium. A 3 way handshake as described in the node discovery phase (section IV – B – 2) must take place for communication to occur. Data transfer between a node and the base station is also only at a preset frequency determined at design time. Nodes are assumed to be a maximum of 2 hops away from the base station, due to the grazing range of the animals.

C. Network Layer

1) Broadcast Packets:

Broadcast packets will be sent out from nodes time to time to alert other nodes of their presence. Broadcasts will also be sent out by the base station at certain time intervals to alert neighbouring nodes that they are within range of the base station. The time and date information will be forwarded to allow nodes to synchronize their real time clocks with that of the base station. The transmitter ID will also be sent so that receiving nodes will know from which node the broadcast is from. Note that one should not confuse the idea with receiver address and node ID. In order to facilitate node discovery and broadcasting, all nodes have the same address. It is the node ID (that is stored in the node's memory) that is unique to each node. Hence, for a node to send a broadcast it simply has to load the payload appropriately as described by the above packet structure and sends it. No receiver node ID is required as all nodes will be intended to receive the data. All broadcasting is done at a frequency of 433.2MHz. The frequency of each slot (after handshaking) is given by the equation below:

$$communicationfreq_n = 433.2MHz + (n \times 500Hz) \;\ldots (1)$$

For example, the frequency of communication occurring within the third timeslot (n=3 in the above equation, where n is the slot number) in the second period would be 433.2MHz (which is the base frequency) plus 3 times 500 kHz giving a frequency of 434.7MHz.

2) Acknowledgement packet:

Acknowledgements are used to allow nodes to determine whether they are in communication lock with another node or if their last sent message was received correctly.

3) *Data packet*:

Data packets will include the sensed data as well as the time-stamp, battery status and transmitter node ID, to ensure that the data arriving at the base station is put into the correct flat-file.

D. *Application layer*

This layer is responsible for determining the sensed data payload in the data packet. Temperature is output in the form of an MSByte and an LSByte. Time and date from each node are also sent in the data packet giving a time-stamp to the sensed data. A single byte is used to store each parameter, i.e. the day is stored in one byte, with the month in another and so on. This allows the base station microcontroller to easily manipulate the incoming data. The application layer at the base station is also responsible for passing received data from sensor nodes to the PC via the serial interface. The base station microcontroller thus had to remove only the required data and push it through the serial port in the format described in Figure IV-2: Format of data sent from base station microcontroller to PC below.

	Byte	Data
	1	NodeID
	2	Battery Status
	3	Day
	4	Month
	5	Year
	6	Minute
	7	Hour
	8	Second
	9	Temp_MSByte
	10	Temp_LSByte
	11	Light Intensity
for future use	12	
	13	
	14	
	15	
	16	

Figure IV-2: Format of data sent from base station microcontroller to PC

Description of the format of data received at the base station User-interface application program (user-app):

- *Node ID*

This unique number found in the memory of every sensor node is used to differentiate between nodes in the network. The user-app also uses this node ID to name the text file in which data for a specific node is stored.

- *Battery Status*

The battery status is either a "0", which indicates a healthy battery, or a "1", which implied that the battery is low.

- *Day, Month and Year*

These three bytes represent the date at which the sensed data in the packet was obtained. These values can be used by the user-app to filter viewing of data to certain dates only.

- *Hour, Minute and Second*

These three bytes indicate the time at which the sensed data in the packet was obtained (i.e. they represent the timestamp of the sensed data).

- *Temp_ MSByte and Temp_ LSByte*

These two values represent the raw temperature data from the node. The user-app manipulates this data to output the actual body temperature of the animal.

- *Light Intensity*

This value is the raw light intensity reading. The user-app will convert this reading to a unit in Lux using a look-up table and interpolation routine.

- *Bytes 12 to 16*

These bytes are reserved for future use. Perhaps the number of nodes in the network will increase beyond 255, whereby more than just one byte will be required to represent the node's unique ID. These extra bytes may be configured for such a purpose at a later stage.

V. BASE STATION

A. *Architecture of Base Station*

All messages sent from sensor node to base station are relayed via a sink node called the base station node, which is connected serially to the base station.

B. *Base Station Node Components*

The USART on the base station node's microcontroller was connected to the serial port of the base station (PC) via a MAX232 level shift converter using a male-to-female DB9 serial cable. This level-shift converter was required to prevent the TTL level components on the microcontroller from being damaged by the CMOS-level voltages in the base station PC. Since the base station will have to be constantly powered as not to miss any data reception, the primary 3V power source at the base station will be obtained by regulating a standard 5V DC power supply using an LM2905 3V regulator.

C. *User Interface*

The User-interface application program (user-app) (shown in Figure V-1) is implemented in Microsoft Visual Basic 6 ®. The user-app allows the user to view the sensed data collectively and to perform comparative analysis.

1) *Design Approach*

The design approach undertaken for the implementation of the user-application at the base station is a top-down approach to break down the system into subsystems to perform data acquisition, data storage and data analysis.

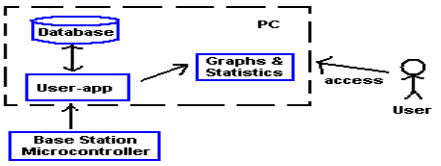

Figure V-1: Software System Functionality

2) *Implementation*

The two most essential project components that are used in the user-app were the MS Chart® and MS Comm® activeX controls (mscomm32.ocx and mschrt20.ocx respectively). These allowed data to be transferred through the serial port and provided a means to generate various plots which could be used for data analysis.

a. *Data Acquisition*

Data is transferred from the base station node's MCU to the PC via a serial connection. The user-app, being created in an event driven programming language, reacts to certain events to provide its functionality. The user-app was designed to respond to incoming data on the serial and was performed using the ***MSComm_OnComm*** sub procedure. The incoming data is read from the serial port in the form of a string of ASCII characters, which is then converted into integer values to allow ***RXData*** to store the incoming data in the form of separate numerical values. In addition, a text file format was chosen to store the sensed data from each node, named according to its node ID. Nodes having existing files had the sensed data appended to the file, whereas new nodes had a new file created and the sensed data written to it.

```
1,OK, 1,January, 2009, 2: 12: 40, 26.0625,      180
1,OK, 1,January, 2009, 2: 12: 41, 26.0625,      161
1,OK, 1,January, 2009, 2: 12: 43, 26.0625,      210
1,OK, 1,January, 2009, 2: 12: 44, 26.0625,      222
```

Figure V-2: Database structure – for single node (node 1)

The automatic generation of flat files eliminated the error that could have occurred from users entering wrong file names. Accessing specific information from the database is done using the node ID and date specified by the user. The file matching the node ID is opened and all records having the date specified by the user have their temperature and light parameters stored in a 48-element array (48 readings per day), which is then used to plot graphs as required by the user.

b. *Plotting of graphs, Statistical Calculations and Choice of node data*

The user application was designed to provide functionality to display graphs showing changes in the animal's body temperature and changes in its environment's light intensity. Both parameters may be viewed per day or per month. An additional feature provided allows the user to select a specific node whose data he/she wants to view by means of entering the node's ID in a text box. The date of sensed data could be selected in a similar fashion. In the application a button named ***View Statistics*** on the graph window was created to enable the user to view statistics such as average temperature, average light intensity as well as minima and maxima values per day.

VI. LITERATURE REVIEW

In [17], the authors presented contention based and hybrid versions of CSMA and TDMA in that they maintain the synchronized time slots. S-MAC [17] was used the UCB Rene Mote [18], Atmel AT90LS8535 microcontroller [19] and TR1000 from RF Monolithic [20] for MAC testing purposes. S-MAC [17] applies message passing to reduce contention latency for sensor-network applications that require store-and-forward processing as data moves through the network. In [21], authors developed a lightweight MAC called B-MAC which is used as the default MAC for Mica and outperforms S-MAC for broadcast traffic. Both B-MAC and S-MAC were implemented in TinyOS. In B-MAC, to measure latency, each node is connected to an oscilloscope. Upon submission of a packet to the MAC protocol, a hardware pin was toggled. When a packet is received, a different pin is toggled. Using an oscilloscope, authors captured the time between each pin toggling to yield the total latency. Routing in WSN is a key technology and very challenging problem. Many routing algorithms have been proposed to satisfy the requirements of sensor networks [22]. A source routing protocol in which the sender knows the complete hop-by-hop route to the destination is used in DSR where as AODV adopts using traditional routing tables, one entry per destination [23] to maintain routing information.

VII. TESTING & RESULTS

In order to test the design, three sensor nodes as discussed in section II were developed. A breadboard was used to connect and test all the components (shown in figure VII-4) before actual Vera-board and PCB designs were done. The results which we obtained were satisfactory as per the design specifications and all components were working as planned. Figure VIII-1 shows the output of the microcontroller at the base station, which was verified by viewing incoming characters on terminals. The wireless transceivers were tested using the STK500 development board from ATMEL [16]. The sensed data readings were taken every second and then output onto the screen. This procedure involved deliberately causing the temperature and light conditions to increase and decrease so that corresponding results could be seen on the user application at the base station.

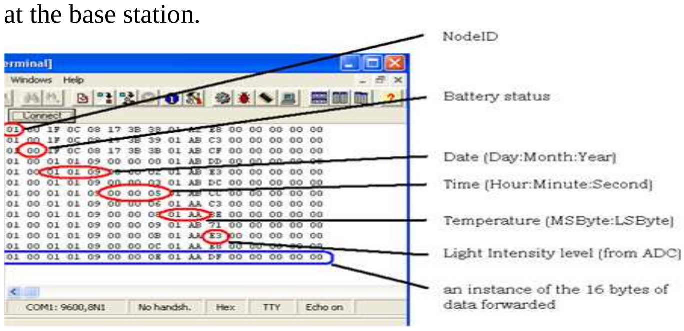

Figure VII-1: Testing of data reception from serial port on a terminal in the Codevision C Compiler

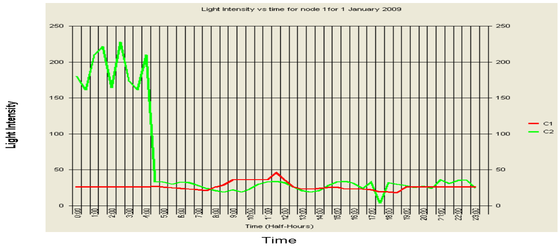

Figure VII-2: Results for prototype sensor node testing

Figure VII-2 shows body temperature and light intensity vs. time for node 1 under a test procedure. Figure VII-3 shows the data written to the database as viewed in a textbox in the user-application.

Node ID	Battery Status	Date	Time	Temperature	Light
1	OK	1,January, 2009	0: 15: 24	25.250	30
1	OK	1,January, 2009	0: 15: 25	25.250	21
1	OK	1,January, 2009	0: 15: 27	25.250	22
1	OK	1,January, 2009	0: 15: 28	25.0625	29
1	OK	1,January, 2009	0: 15: 30	25.250	21
1	OK	1,January, 2009	0: 15: 31	25.250	21
1	OK	1,January, 2009	0: 15: 33	25.250	29
1	OK	1,January, 2009	0: 15: 34	25.0625	24
1	OK	1,January, 2009	0: 15: 36	25.0625	19
1	OK	1,January, 2009	0: 15: 37	25.250	27
1	OK	1,January, 2009	0: 15: 39	25.0625	33
1	OK	1,January, 2009	0: 15: 40	25.0625	275
1	OK	1,January, 2009	0: 15: 42	25.250	1006
1	OK	1,January, 2009	0: 15: 43	25.0625	1074
1	OK	1,January, 2009	0: 15: 45	25.250	46
1	OK	1,January, 2009	0: 15: 46	25.250	12
1	OK	1,January, 2009	0: 15: 48	25.250	13
1	OK	1,January, 2009	0: 15: 49	26.0625	12
1	OK	1,January, 2009	0: 15: 51	27.125	11
1	OK	1,January, 2009	0: 15: 52	28.0625	11
1	OK	1,January, 2009	0: 15: 54	28.0625	12
1	OK	1,January, 2009	0: 15: 55	29.500	12
1	OK	1,January, 2009	0: 15: 57	29.0625	28
1	OK	1,January, 2009	0: 15: 58	29.0625	36
1	OK	1,January, 2009	0: 16: 0	29.500	33

Figure VII-3: Results for prototype sensor node testing as appearing on User-app

Figure VII-4: Hardware testing

VIII. CONCLUSION & FUTURE WORK

In this paper, both the hardware and software designs for a WSN designed for wildlife monitoring have been presented. The results presented would highlight the outcome of the system. The sensor nodes that were designed are small and robust, able to be used as required in the specifications. The system design could be expanded in the future by introducing more nodes within the network and upgrading the system to a multi-network capable one. In addition, with an increased budget, different sensors could also be added to each node as the current design allows for expansion of more data to be transferred from node to node/base station.

IX. REFERENCES

[1] F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks", IEEE Communications Magazine, pp 40(8):102-114, August 2002.

[2] R. Rayudu, "Wireless Sensor Node Design Considerations," Workshop on Ubiquitous computing (UbiComp2008), C-DAC, Bangalore, India, 2008.

[3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in the proceedings of 1st ACM international workshop on Wireless sensor networks and applications, USA, 2002.

[4] P.K. Saraswat, A. Kumar, S. Polana, A. Singh, "WildCENSE – Sensor Network for Wildlife Monitoring," DA-IICT, Guiarat, India, 2006.

[5] Polygon Technologies, "NRF905-C433 datasheet v1.1," www.pteq.net, 2007.

[6] NORDIC Semiconductor, "Application note: NRF Radio Protocol Guidelines," nan400-07, www.nordicsemi.no, 2006

[7] J. Hill, "System Architecture for Wireless Sensor Networks," PHD thesis, University of California, Berkeley, 2003.

[8] RS Components, "Light Dependant Resistors datasheet," RS Components, Northants, March 1997.

[9] MAXIM, "DS1302 datasheet," Maxim Integrated Products, USA 2005, www.maxim-ic.com

[10] G.M Paretas, "Using DS18B20 digital temperature sensor on AVR microcontrollers," September 2007.

[11] ATMEL, "ATmega16L datasheet," www.atmel.com/dyn/resources/prod_documents/doc2566.pdf, 2008.

[12] NORDIC Semiconductor, "NRF905 datasheet," www.nordicsemi.no, 2008.

[13] NORDIC Semiconductor, "Application note: NRF Radio Protocol Guidelines," nan400-07, www.nordicsemi.no, 2006.

[14] M. Halpern, K. Saleem, "Battery Power Issues for WSNs," Sensor Networks Program, Melbourne NICTA Lab, Australia.

[15] J. N. Al-Karaki, and A. E Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey", Iowa State University

[16] ATMEL, "STK500 user guide," www.atmel.com, 2000.

[17] W. Ye, J. Heinemann, and D. Estrin, "An energy efficient MAC protocol for Wireless sensor networks, in Proceedings of the IEE INFOCOM, New York, June 2002.

[18] http://www.cs.berkeley.edu/˜awoo/smartdust/.

[19] http://datasheet.digchip.com/054/054-05481-0-AT90LS4434-4AC.pdf.

[20] http://www.rfmonolithics.com/products/data/tr1000.pdf.

[21] J. Polastre, J. Hill, and D. Culler, **"**Versatile Low Power Media Access for Wireless Sensor Networks", In Proceedings of the Second ACM Conference on Embedded Networked Sensor Systems (SenSys), November 3-5, 2004.

[22] Z. Jin, Y. Jian-Ping, Z. Si-wang, L. Ya-Ping and L. Gauang, "A Survey on Position Based Routing Algorithms in Wireless Sensor Networks", Algorithms, Vol,2,Pages.158-182, February,2009.

[23] Md. Anisur Rahman, Md. Shohidul Islam, Alex Talevski, "Performance Measurement of Various Routing Protocols in Ad-hoc Network", Proceedings of the International Multi Conference of Engineers and Computer Scientists 2009 Vol 1,IMECS 2009, March 18 - 20, 2009, Hong Kong.

N.V. Pillay was born in Durban, South Africa on 12/09/1987. He graduated with a BSc. Engineering (Computer Engineering) degree at UKZN, RSA in 2009.
He is currently an ENGINEER IN TRAINING at Eskom Enterprises, Sunninghill, South Africa.