2011 International Conference on Communication Engineering and Networks
IPCSIT vol.19 (2011) © (2011) IACSIT Press, Singapore

Image Transfer in Wireless Sensor Networks

A.A.Shahidan, N.Fisal, A.H.A.H.Fikri, Nor-Syahidatul N.Ismail and Farizah Yunus

UTM-MIMOS Center of Excellent, Faculty of Electrical Engineering
Universiti Teknologi Malaysia, 81310 UTM Johor
Johor Bahru, Johor, Malaysia
ashahidan@utm.my, sheila@fke.utm.my, hadifikri@gmail.com, syanadia@fkegraduate.utm.my, farizah@fkegraduate.utm.my

Abstract— Image transfer in Wireless sensor Network (WSN) is very useful for information gathering since information obtained from analysis of the images is significant in certain situation. In general, WSN utilize an eight bit microcontroller and IEEE802.15.4 compliant radio for processing and transferring the data to remote location. These specifications posed a serious restriction in transferring multimedia data since it require a huge bandwidth and processing capabilities. The purpose of this project is to develop the image transferring mechanism for realizing JPEG motion data transfer in WSN where the video is basically generated from the sequence of compressed image. Hence, image sequence transfer with data buffering mechanism is implemented. The scope covers developing sensor node circuit which is equipped with external memory and designing software for the sensor node which is capable to functioning as intended. Image sequence is produced at the control station after processing the received data.

Keywords-Image Sequence; Wireless Multimedia Sensor Networks; Wireless Sensor Networks

1. Introduction

Since a few years ago, wireless sensor network technology has been developed. Research communities give their attention on developing wireless sensor network for many purposes. The introduction of the wireless sensor network has become a new paradigm in information-gathering method. This is because wireless sensor network consists of many self-organized sensing nodes that cooperate with each other to gather information [1]. Each node is equipped with devices which are used to monitor and collect the data, process the collected data and then transmit the data to the adjacent nodes. There are many applications of wireless sensor network such as multimedia surveillance sensor networks; traffic avoidance, enforcement and control systems; advanced health care delivery; automated assistance for the elderly and family monitors and many more [2].

Such various applications have lead toward designing many type of sensor node. One of them is the sensor node with built in camera. This kind of node is developed after the availability of small size and low price new CMOS technology camera in the market. The camera has the ability to capture still image and also video stream from the environment. The development of this kind of sensor node has enhanced the wireless sensor network capability in its application [3]. However, the development of sensor node with image sensor is not complete if the sensor node has limited capacity of temporary storage. This is because handling the data in form of image usually required large buffer to store large size of data. Therefore, it is favorable to have a sensor node with image sensor and a large buffer size. Users now can obtain the visual data instead of the scalar data which is in form of numbers and graph. The example of the developed sensor nodes related to data buffer and image sensor are eCAM, Cyclops, TelG and RISE [4, 5, 6, 7].

IEEE 802.15.4 compliant radio is widely used for WSN implementation because of its low cost, and low power attributes [8]. The standard specified that the maximum data rate that may be achieved is up to 250kB/s only. Therefore, the data that need to be transmitted is usually small. For multimedia data, the data rate is considered to be too slow. Hence, the data that need to be transmitted must be compressed or coded in such a way that the transmission speed can be compromised [9]. The transmission of multimedia application at low rate and low power using IEEE802.15.4 has already been proven in [10]. An 8-bit microcontroller, as the central processing unit for multimedia processing is not relevant at all. Only by using an interface which processes the multimedia data before passing it to the microcontroller can allow the use of such platform. Transmission delay is one of the main concern in conventional WSN, however for wireless multimedia sensor network, the processing delay prove to be more significant.

This happens since there is a limited storage in microcontroller and its processing speed [9]. The memory space in the microcontroller is used for storing previous image data while the data processing is in progress. The storage must be cleared first before the next image data can be stored and processed by the microcontroller. Therefore, a few second is needed until the process is finished and the memory can be flushed. For application in the real world, such condition must be avoided. This is important in order to ensure that the sensor nodes are reliable and efficient for image capture. Suppose the users are able to capture more images without have to wait for several amount of time. For that reason, an external memory can be used as the temporary storage and solve the limited memory space problem. All images data collected from camera can be temporarily stored in this memory before being processed by microcontroller. To summarize, bandwidth requirement, processing speed, power consumption efficiency, and latency must be carefully analyzed and tackled.

2. Hardware Development

Sensor nodes are the crucial part of wireless sensor network since they are distributed in the sensor field while having communication with each other so that the data can be transferred. The sensor nodes consist of several components which have certain functions so that the nodes are able to work properly. The main components used are microcontroller, transceiver, sensors, external memory, analog to digital converter, and power source. Fig. 1 shows the sensor node in block diagram.

Microcontroller Unit (MCU) is the core component in every single sensor node. The functions of this component are to control other devices including interfaces, collect the data from devices and send the data to the devices, and also process the data. Atmega644PV, the MCU of the sensor node, consists of two USART interfaces which are needed to interface with camera and transceiver. Besides, it operates in low power consumption with low operating voltage, 3.3V. The requirement of larger memory spaces to store temporary data from MCU is solved using AT45DB321D data flash operated with 3.3V operating voltage via Serial Peripheral Interface (SPI) to MCU. For communication between the sensor nodes, the Xbee OEM transceiver is used. The transceiver is able to operate in 4 modes which is transmit mode, receive mode, idle mode and sleep mode which lead to power saving, plus the operating voltage is 3.3V.

In order to retrieve graphical information in the sensor field, C328-7640 VGA resolution camera module is used. It operates through USART with 3.3V operating voltage and able to generate several size of JPEG compression images up to 640x480 pixels.

3. Software Design

All sensor node activities such as routing, data input/output, data processing and many more are controlled by MCU. Such activities need to be attempted almost at the same time. Here MCU need to have multitasking capability since all activities must be managed and coordinated appropriately. To do so, MCU need to be programmed with powerful and robust software to conduct multitasking job and the best way here is to design an operating system (OS) for MCU [11].

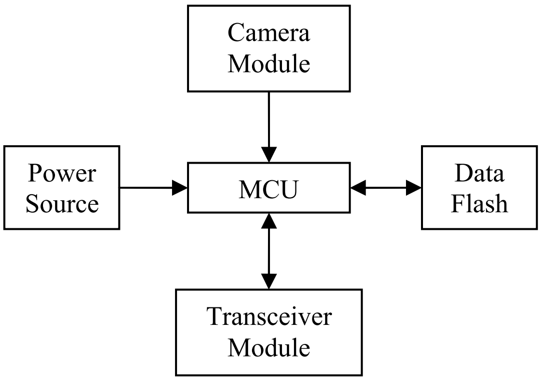


Figure 1. Block diagram of the sensor node.

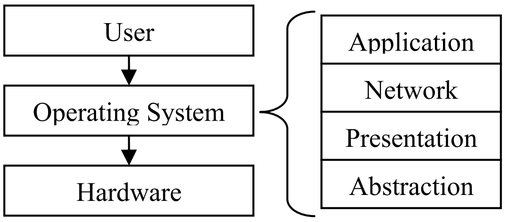


Figure 2. Layering structure of the operating system

Basically, OS in sensor node function as an interface between applications software and hardware which offer several services related to the applications of the sensor node. After receiving command from user, an interface called application programming interfaces (API) which in turn will be called by applications software to request certain services from the OS. Then, certain parameters will be passed and the result from related operation will be obtained. The interfacing between user and hardware is made through four layers of the OS which are application layer, network layer, presentation layer and abstraction layer. Fig. 2 illustrates the layering structure of the OS.

Application layer basically is the nearest layer that connects the end user to the system. Generally, there will be certain applications software that enable user to communicate, give commands to the sensor node and provide result to the user. In this project, the data is received from presentation layer. The application layer function is to provide the data to the end application software on the computer known as Graphic User Interface (GUI). GUI will process the received data, generate image from the data in JPEG format which is stored in computer hard drive and finally display the image at computer screen.

Network Layer in this project is designed for node-to-node and end-to-end packet delivery. Here, flooding routing protocol is used. Based on the conditions as stated in literature review, every sensor node will rebroadcast the received data until the data is delivered at the end node.

As stated before, information received by application layer came from presentation layer. The information is actually being formatted by presentation layer to a format recognized by application layer. In short, presentation layer could be analogous as translator who will translate the information received to something understood by application layer. Generally, this layer is where the encryption data (if any) is being read and reformatted for upper layer process.

The abstraction layer plays the role as a functions adapter between hardware and presentation layer and it is hardware dependence. Here, the presentation layer functions will remain unchanged although the hardware use is changed. This is because the abstraction layer will generalize the hardware functions to a specific command used by presentation layer. The uses of the abstraction layer will simplify the presentation layer functions and at the same time make the presentation layer to be hardware independence.

4. Software Architecture

The operating system can be divided into three parts including sensor part, network part and application part as illustrated in Fig. 3 below. Each part consists of several modules with certain function which will be discussed later. In general, sensor part is the part where the data is gathered, and network part is responsible for data delivery through the network while all data processing is done in application part.

4.1 Sensor

Sensor part is responsible for retrieving image data from the hardware (C328R camera module). The functions include issuing command, receiving data and passing data from sensor part to application part.

Basically the OS will communicate with C328R using 6 bytes command set which is started with sync command. The sync command will be sent repeatedly until acknowledgement is received from C328R. After that, initial command consists of several parameters which are Color Type, RAW Resolution and JPEG Resolution, will be sent to C328R. Then, Set Package Size command will be issued to C328R to set the size of packet that will be generated by the camera before it is sent to MCU. After that, Get Picture command will be sent to the hardware. Once Get Picture command is received, C328R will start capturing image. The image data is passed to MCU byte by byte through serial communication which is USART. During processing, the data will first be queued in buffer before it is collected by camera module. Here, the data must be buffered due to multitasking processes where the MCU will only collect the data once the task that instruct it to do so is reached. Otherwise, the data must be buffered first. The same process will be repeated for every single byte of data received from C328R.

4.2 Network

Network part is responsible for data transmission path setup which is referring to communication between nodes to the end node which is done via hardware wireless transceiver module (Xbee OEM RF Module). Here the OS will play the role in order to establish the flooding protocol used in the network. There are two data processing mechanisms in network part which are transmitting data mechanism and receiving data mechanism.

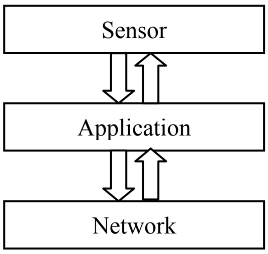


Figure 3. Parts involve in operating system

Transmitting data start with collecting data from application part and send the data to the wireless transceiver. Once a packet of data reaches at the Network part, it will be processed by appending the data packet with network protocol data unit (network PDU). Receiving data mechanism is started when the transceiver of a node receives the data. The data it will be sent to MCU through USART. The MAC destination address of the packet will be examined in order to verify whether the address is the same as the address of the node. If the MAC destination address differs, then the packet will be dropped. However, if it is the same, then the packet will be examined again by duplicate checker mechanism to ensure that the packet never reaches at the node yet. If it does, then the packet will be dropped. After duplicate checker, the network destination address of the packet will be verified. If it is correct then the packet will be passed to application part. However, if the wrong network destination address is received, then the packet will be rebroadcasted. Fig. 4 shows the flow chart of the receiving data mechanism.

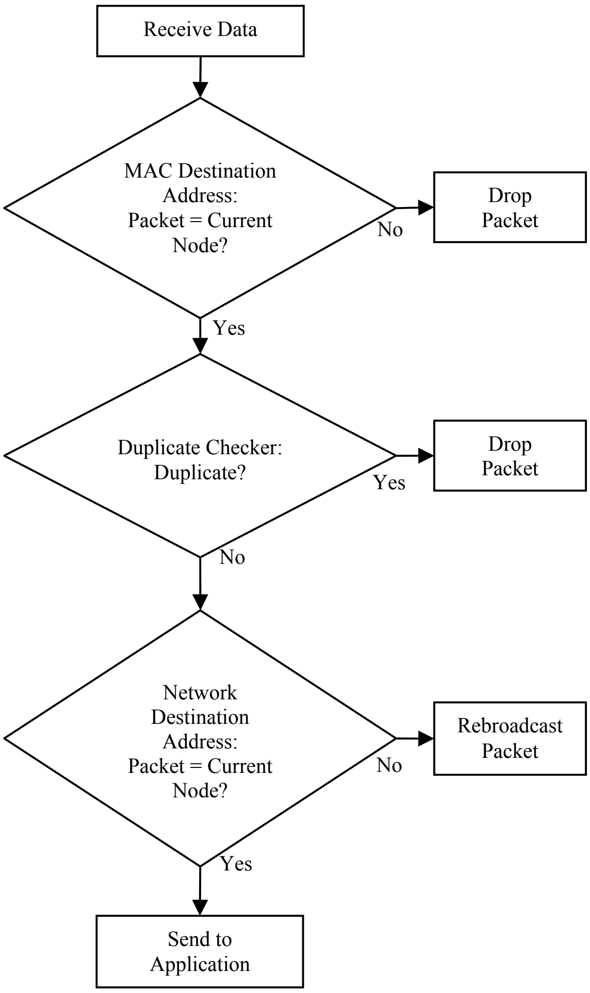

Receive Data
MAC Destination Address: Packet = Current Node?
No
Drop Packet
Yes
Duplicate Checker: Duplicate?
Yes
Drop Packet
No
Network Destination Address: Packet = Current Node?
No
Rebroadcast Packet
Yes
Send to Application

Figure 4. Flow chart of the receiving data mechanism

4.3 Application

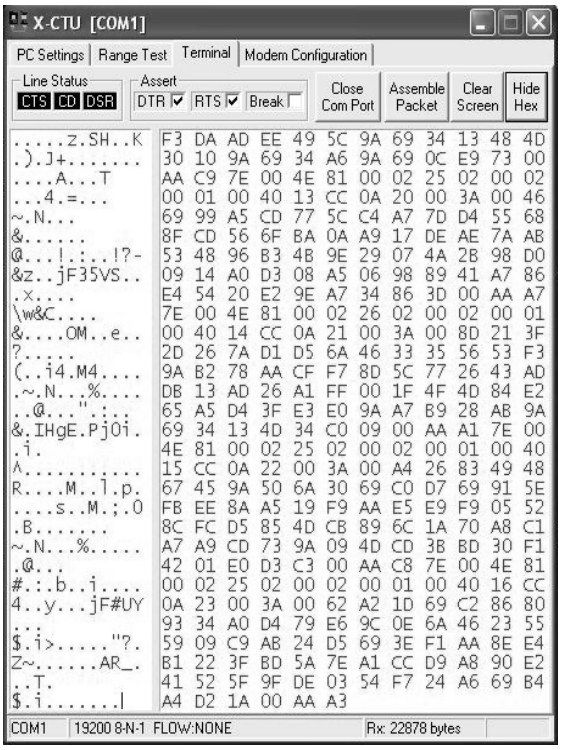
Application part is responsible for data processing such as data input/output control, data destination and data buffer. It is also responsible for data processing such as setting packet destination, determining packet size and buffering data. It can be said as controller of the overall system. Basically, many of the system commands are initiated in application part. For example, it initiates command for sensor part to obtain image sequence data. Besides, the part also handle data buffer process where it is responsible for determining target address for storing and retrieving data, and also sending appropriate command to Data Flash through Serial Peripheral Interface (SPI) link between software and hardware.

The following describe the process of data buffer. As a data packet is received from sensor part, application part, the address or location where the data will be stored will be determined. Four bytes of Addressing Sequence consist of one command opcode byte and three bytes of starting address will be to the Data Flash through SPI module. After sending the addressing sequence, data that want to be stored in the Data Flash will be transferred. The process of retrieving the data is similar with storing where the addressing sequence consist of command opcode and address will be sent by application part, and then the data will be received from the Data Flash.

5. Result

5.1 Hardware

The sensor node PCB is designed through several times of improvement and troubleshooting. The sensor node circuit was prepared with certain cautions such as avoiding excess solder paste used and avoiding long contact between solder iron and surface mount ATmega644PV during soldering. This is to endure that the process would not damage the MCU and other components. The sensor node size is miniaturized by using small components such as surface mount resistor and LEDs.



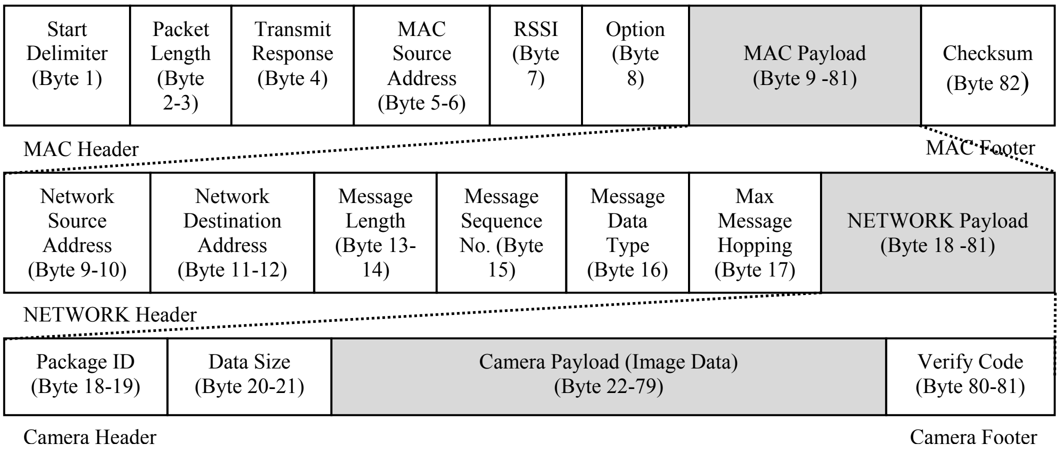
```
.....z.SH..K    F3 DA AD EE 49 5C 9A 69 34 13 48 4D
.).J+.......    30 10 9A 69 34 A6 9A 69 0C E9 73 00
....A...T       AA C9 7E 00 4E 81 00 02 25 02 00 02
...4.=...       00 01 00 40 13 CC 0A 20 00 3A 00 46
~.N...          69 99 A5 CD 77 5C C4 A7 7D D4 55 68
&......         8F CD 56 6F BA 0A A9 17 DE AE 7A AB
@...!.:..!?-    53 48 96 B3 4B 9E 29 07 4A 2B 98 D0
&z..jF35VS..    09 14 A0 D3 08 A5 06 98 89 41 A7 86
.x....          E4 54 20 E2 9E A7 34 86 3D 00 AA A7
\w&C....        7E 00 4E 81 00 02 26 02 00 02 00 01
&....OM..e..    00 40 14 CC 0A 21 00 3A 00 8D 21 3F
?.....          2D 26 7A D1 D5 6A 46 33 35 56 53 F3
(..i4.M4....    9A B2 78 AA CF F7 8D 5C 77 26 43 AD
.~.N...%....    DB 13 AD 26 A1 FF 00 1F 4F 4D 84 E2
..@...".:..     65 A5 D4 3F E3 E0 9A A7 B9 28 AB 9A
&.IHgE.PjOi.    69 34 13 4D 34 C0 09 00 AA A1 7E 00
.i.             4E 81 00 02 25 02 00 02 00 01 00 40
^...........    15 CC 0A 22 00 3A 00 A4 26 83 49 48
R....M..l.p.    67 45 9A 50 6A 30 69 C0 D7 69 91 5E
....s..M.;.O    FB EE 8A A5 19 F9 AA E5 E9 F9 05 52
.B.......       8C FC D5 85 4D CB 89 6C 1A 70 A8 C1
~.N...%.....    A7 A9 CD 73 9A 09 4D CD 3B BD 30 F1
.@...           42 01 E0 D3 C3 00 AA C8 7E 00 4E 81
#.:.b..i....    00 02 25 02 00 02 00 01 00 40 16 CC
4..y...jF#UY    0A 23 00 3A 00 62 A2 1D 69 C2 86 80
...             93 34 A0 D4 79 E6 9C 0E 6A 46 23 55
$.i>....."?.    59 09 C9 AB 24 D5 69 3E F1 AA 8E E4
Z~......AR_.    B1 22 3F BD 5A 7E A1 CC D9 A8 90 E2
..T.            41 52 5F 9F DE 03 54 F7 24 A6 69 B4
$.i.......|     A4 D2 1A 00 AA A3

COM1   19200 8-N-1  FLOW:NONE          Rx: 22878 bytes
```

Figure 5. Output data displayed by X-CTU Software

5.2 Software

The software designed shows positive result where the image sequence data is successfully received at the end node. The output data is displayed by X-CTU software as shown in Fig. 5. The received data is examined in order to ensure that the structure of the data packet is correct according to the packet format. The received packet should start with eight bytes of MAC PDU; follow by nine bytes of Network PDU and four bytes of camera PDU. Then, there will be 58 bytes of image data and lastly two bytes of camera verify code and a checksum byte. Fig. 6 illustrates the receive packet format.



Start Delimiter (Byte 1)	Packet Length (Byte 2-3)	Transmit Response (Byte 4)	MAC Source Address (Byte 5-6)	RSSI (Byte 7)	Option (Byte 8)	MAC Payload (Byte 9 -81)	Checksum (Byte 82)

MAC Header

MAC Footer

Network Source Address (Byte 9-10)	Network Destination Address (Byte 11-12)	Message Length (Byte 13-14)	Message Sequence No. (Byte 15)	Message Data Type (Byte 16)	Max Message Hopping (Byte 17)	NETWORK Payload (Byte 18 -81)

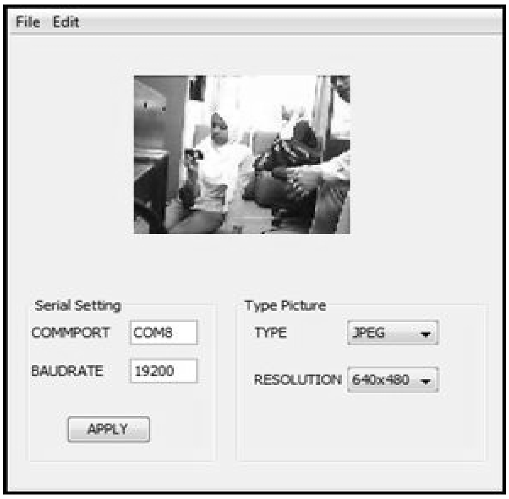
NETWORK Header

Package ID (Byte 18-19)	Data Size (Byte 20-21)	Camera Payload (Image Data) (Byte 22-79)	Verify Code (Byte 80-81)

Camera Header

Camera Footer

Figure 6. Format of Received Packet



COMMPORT COM8

BAUDRATE 19200

TYPE JPEG

RESOLUTION 640x480

Figure 7. Output of the GUI Software

The end node is connected to the computer and there is software to display the image using Graphical User Interface (GUI) which functions as platform to receive the data received by end node. The software is able to process the received data and generate images from the processed data. Basically there is no error detection or error control implemented in this system. Therefore the GUI software will only process whatever data received from the end node. If there is error or packet lost occur let say during transmission data, it can only be seen in the image generated by the GUI software. The reason behind this logic is to reduced retransmission and latency. From the results, the frame rate achievable average is about 1 frame in every two seconds. This is considered to be tolerable when taking into account the transmission speed and the processing capability. Fig. 7 shows the output of the GUI software.

6. Conclusion

Images transfer in wireless sensor network need self-organized sensing nodes that cooperate with each other to ensure the system reliability in gathering image data. In this project, the sequence of image data can be transferred through the sensor networking to the end node. The data received at the end node has been shown in X-CTU software. The programming for the MCU is working properly where the data can be buffered into data flash. This has led toward the realization of image sequence data transfer in wireless sensor network. In this work, the multi-hop data transfer has also been implemented and it is working as intended. The data is transferred from the sensor node to the end node through one intermediate node which is located at the middle of both sensor node and end node.

7. Acknowledgments

The Author would like to thank to the Ministry of Higher Education (MOHE) of Malaysia, Universiti Teknologi Malaysia (UTM), MIMOS Berhad and Research Management Centre (RMC) for the sponsorship and Telematics Research Group (TRG) for the full support and good advice.

8. References

[1] Wang-Yan, “Study on Model and Architecture of Self-Organization Wireless Sensor Network," in Wireless Communications, Networking and Mobile Computing, 2008.

[2] Ian F. A., Tommaso M., Kaushik R. C., “A Survey on Wireless Multimedia Sensor Network,” Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, 2006.

[3] Tian H., “A Scalable Sensor Network Architecture for Robustness & Time-Energy Awareness,” Charlottesville: University of Virginia, 2003.

[4] Chulsung P., Pai M. C., “eCAM: ultra compact, high data-rate wireless sensor node with a miniature camera,” New York: ACM, 2006.

[5] Mohammad R., Rick B., Obimdinachi I. I., Juan C. G., Jay W., Deborah E., Mani S., “Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks,” California: ACM, 2005.

[6] Abdul Hadi Fikri Bin Abdul Hamid, Rozeha A. Rashid, Norsheila Fisal, S. K. S. Yusof, S. H. S. Ariffin, Liza Latiff, "Development of IEEE802.15.4 based Wireless Sensor Network Platform for Image Transmission," in International Journals of Engineering & Technology IJET, vol. 9, 2009

[7] D. Zeinalipour-Yazti, S. Neema, V. Kalogeraki, D. Gunopulos, W. Najjar, “Data Acquisition in Sensor Networks with Large Memories,” Washington: IEEE Computer Society, 2005.

[8] Carlos F. García-Hernández, Pablo H. Ibargüengoytia-González, Joaquín García-Hernández, and Jesús A. Pérez-Díaz, “Wireless Sensor Networks and Applications: a Survey,” International Journal of Computer Science and Network Security, vol. 7 no.3, March 2007.

[9] Satyajayant Misra, Martin Reisslein, and Guoliang Xue, "A Survey of Multimedia Streaming in Wireless Sensor Networks," in IEEE Communications Surveys & Tutorials, Vol. 10, No. 4, Fourth Quarter 2008

[10] Rozeha A. Rashid, Norsheila Fisal, Abdul Hadi Fikri Bin Abdul Hamid, "Wireless Multimedia Sensor Network Platform for Low Rate Image/Video Streaming," Jurnal Teknologi Special Edition (Instrumentation and Sensor Technology), Vol. 54, 1-12, 2011.

[11] Larry O' Cull, Sarah Cox, “Embedded C programming and the Atmel AVR,” Clifton Park, New York: Thomson/Delmar Learning, 2003.