© IJEDR 2019 | Volume 7, Issue 1 | ISSN: 2321-9939

Building a Multi-hop Wireless Sensor Network for Water Level Monitoring

[1]Dr. Inacio Henrique Yano, [2] MSc. Jose Ricardo Alves, [3]Dr. Antonio Carlos Demanboro

[1]System Analyst, [2]Electrical Engineer, [3]Professor

[1]Embrapa Informática Agropecuária, Campinas, SP, Brazil,
[2,3]Electrical Engineering Faculty, Pontifical Catholic University, Campinas, SP, Brazil

Abstract—Wireless Sensor Networks (WSN) are very useful for data acquisition in harsh environments or where the maintenance of wired infrastructure would not be viable. Despite of these advantages WSN nodes have a limited range ratio thus to collect data on long distance is necessary to construct a path with many relay nodes to reach the destination. Another limitation of these networks is that they often rely on batteries to operate, which can cause a serious limitation in the network lifetime. In this work it was developed a solution based on a Multi-hop WSN to collect data on long distance, and, also some strategies such as 'sleep schedule', 'data aggregation' and 'hub polling' were implemented to extend the WSN lifetime. This could be done by modifications at RFBee Libraries that reinforces the importance of flexibility and portability of this device.

Index Terms—Hub Polling, Lifetime, Relay Node, RFBee

I. Introduction

The Wireless Sensor Networks (WSNs) are very useful for data acquisition in harsh environments or where the maintenance of wired infrastructure would not be viable. Since the nodes of the WSNs have a restricted range, usually data collection nodes (end devices - ED) need router nodes (RT) to transmit their data to a coordinator or a central base (CB) [1]. Despite of the WSNs utility, the limit of their lifetime constitutes the major problem, considering that they are powered by batteries. The batteries depletion problem may be aggravated when it occurs with RT, because this usually does not happen simultaneously with all network nodes batteries. As a result, if a RT fails, several EDs can not send data to CB. In this way, it becomes evident the importance of extending the RT lifetime in order to avoid orphan nodes formation [2] in multi-hop WSNs.

Among various strategies for extending WSNs lifetime the sleep technique [3] is one of the most effective for extending the nodes lifetime and can be combined with other energy saving techniques [4]. However, its application to RTs presents some difficulty, since it is assumed that they must be awake when EDs transmit their data.

There are some proposals of sleep technique in which there is always at least one RT awake to receive EDs′ data [5]. This is usually efficient for dense networks, which justifies redundancy of RTs for groups of EDs. These solutions, despite of solving the problem of interrupting routes, cause large idle intervals of RTs, since at least one of them will be active all the time.

The solution presented in this paper also uses sleep technique to save energy through the technique of Time Division Multiple Access (TDMA), which allows the combination of activity and sleep periods, providing low power consumption [6]. One of the ways to operate a network with TDMA is to use the polling technique [7], in which in a time slot, the RT queries an ED if it has data to transmit; in positive case, the ED will transmit its data in the next time slot. In the polling technique EDs can only transmit in response to a query from the RT. This artifice extends the WSN lifetime and is suitable for any network size.

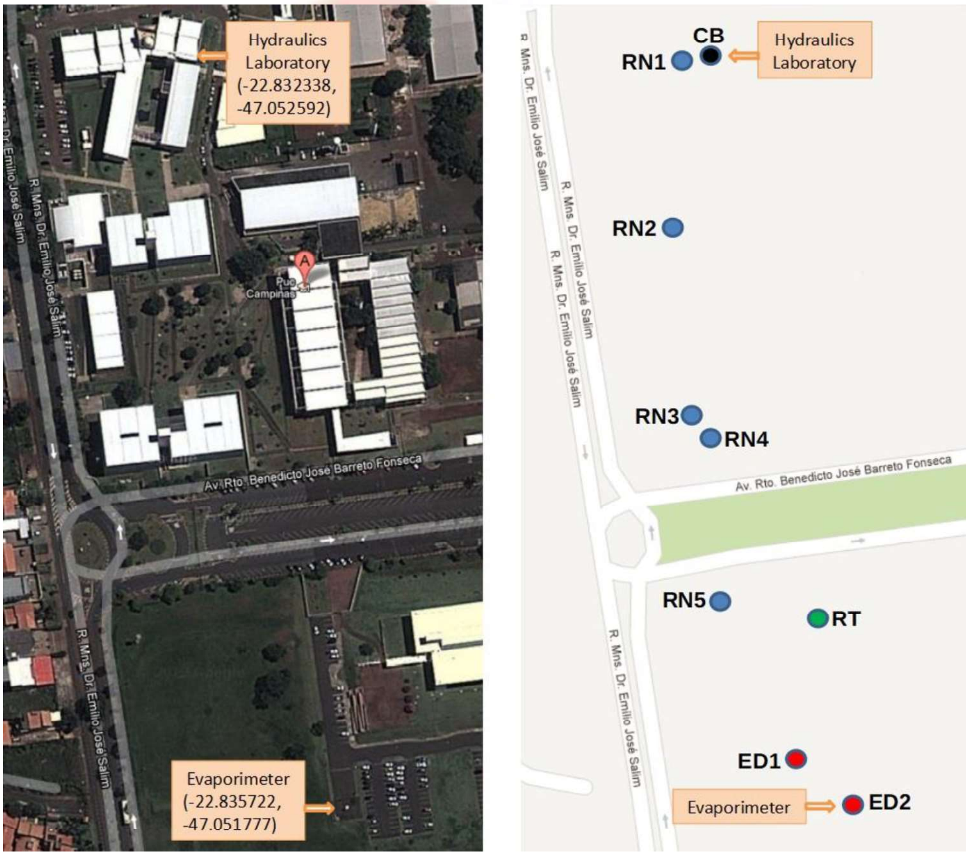
This paper describes the development of libraries for relay data and, also some techniques to save batteries energy for the multi-hop WSN implementation. The purpose of this WSN is the environmental data collection for the Environmentally Sustainable Cities Group (GSAC) of PUC-Campinas and present data of the Santa Candida Experimental Hydrological Basin. Among the data to be collected, the water level of an evaporation tank (evaporimeter) was the most important.

II. The Multi-Hop Network

The purpose of the multi-hop WSN implementation was to allow communication between the evaporation tank and the central base (CB), where data about evaporation will be stored and processed. The evaporation tank (Fig. 1) is located at the weather station next to the parking at the rectory of PUC-Campinas, and the CB is located at the Hydraulics Laboratory of Environmental Engineering.

Figure 1. Evaporation tank located next to the parking at the Rectory of PUC-Campinas

The Fig. 2 shows the hole network. The Hydraulics Laboratory of Environmental Engineering and the evaporimeter are highlighted in Fig. 2(a), the distance between them is approximately 385 m in straight line. Because the range of the radio is around 120 m in open areas [8], some relay nodes (RN) [9] must be placed in this network to retransmit the signal. After the inclusion of relay nodes to retransmit data from evaporimeter to CB, the distance between the end points of the network increased to just over 448 m due to the radio range limitation, and the presence of obstacles.



(a) (b)

Figure 2. Picture and map of the rectory of PUC-Campinas taken from Google Maps (https://www.google.com.br/maps). (a) Picture with the location of the Hydraulics Laboratory of Environmental Engineering and the evaporation tank and (b) map with the location of all WSN nodes.

The Fig. 2(b) shows the schematic of the network nodes spatial distribution. The nodes are represented by a color code that identifies their functions in the network with CB in black, the relay nodes (from RN1 to RN5) in blue, the router node (RT) in green and the data collector nodes (ED1 and ED2) in red.

III. Literature Review

The previous section described the aim and the features of the multi-hop WSN. This section presents the techniques used to save RT energy, which is the node that define the active period of all nodes. Because of this, it is the most sensitive network node and requires specific techniques to be put into sleep [10].

[6] present a study using a polling technique to put RT in sleep mode in the time slot, when there is no activity in the network, with good result in reducing energy consumption. Fig. 3 illustrates the time slots of three EDs being coordinated by one RT. The small time slots with only numbers inside represent requests for data (Polling Time Slot (Tpoll)), the time slots identified by ED represent the frames with data in response to polling requests (Time Slot Transmission (TTX)) and the time slots identified as idles are the sleep time slots (Idle Time Slot (Tidle)), which will occur after a signal send by the RT to all nodes enter in sleep state.

In the first transmission sequence, only EDs 1 and 3 have data to be transmitted. In the second sequence, only the ED3 has data to be transmitted. In the third sequence, no ED has data to be transmitted, and at this time, since there is no return to its data requests, the RT sends a signal to all nodes to enter in sleep mode for a predefined time period. After the period of sleep, the process of requesting data (pollings) is restarted and at this time only the ED2 has data to be transmitted.

This solution presented in [6] is more appropriate for data acquisition in a non-cyclical way to save energy, i.e., EDs may or may not have data to transmit, but must always be consulted because they only transmit when receiving a call from RT. The RT works as a coordinator in classical polling technique, where all EDs send data and enter in sleep mode only after receiving calls from RT.

The polling technique used in this work had some differences from [6], because the EDs have to transmit their data at regular intervals, only the first ED of the network has to receive a call from RT, and all the others EDs of the network send their data in sequence of the first ED, using the data of the predecessor node of the network as a signal to send their data, working similarly to a hub polling network [11]. Each ED automatically enter in sleep mode after sends its data and all EDs were programmed to come out of sleep mode shortly before the next subsequent cycle begins. The RT enters in sleep mode after receives the data of all EDs of the network and come out after a predefined time to restart a new cycle. In this approach, the RT will save much more energy, because it calls only the first ED of the network, considering that the data transmission is the most energy-consuming activity [12].

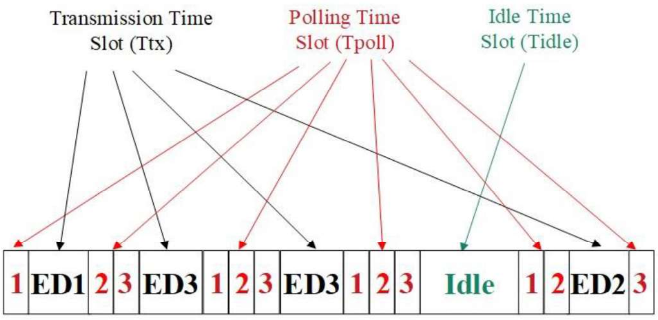


Figure 3. Time Slot of the coordinator node (RT).

Another technique adopted in this work was Data Aggregation (DA), which is a process that aims to reduce network traffic [13], that also provides savings of time slots to retransmission data received from EDs, and consequently, saves energy, mainly because of fewer retransmissions, as well, due to a greater number of time slots in which the RT remains in sleep mode in each cycle. In this work the DA was done without compression, therefore, the data of the EDs only were concatenated before transmitted to CB.

IV. Material and Methods

This section describes the devices and the sensor nodes programming.

Devices

The sensors used in this work were RFBee V1.1 [8] (Fig. 4), powered by two AA-rechargeable 2500 mAh batteries. The RFBee consists of the ATMega 168 microcontroller [15] and the CC1101 transceiver [16], which can be programmed in the same Arduino Integrated Development Environment [17] with the Radiuino open-source libraries [18], which can be downloaded from http://radiuino.cc/wp-content/uploads/2014/01/Bibliotecas.zip.

Figure 4. RFBee V1.1

The device chosen to measure the water level was the Milone eTape Liquid Level Sensor (Fig. 5).

Figure 5. Milone eTape Liquid Level Sensor.

Figure 6 shows the circuit used to calibrate the water level measurement. The sensor calibration was done following the manufacturer's instructions [19]. The Vout was read by the analog input pin 15 and associated to the water level in a bottle using the eTape centimeter ruler. The Rsense electrical resistance decreases in water contact, resulting in a calibration curve with voltage inversely proportional to water level as shown in Fig. 7.

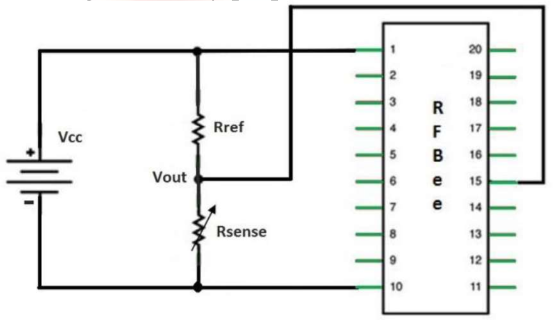


Figure 6. Circuit used to collect Voltage values (V) according to water level in centimeters.

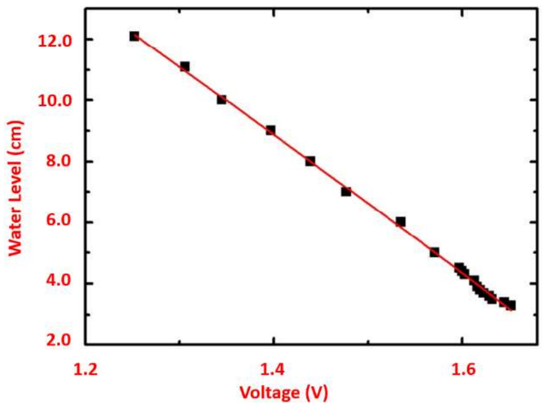


Figure 7. Calibration curve, water level (cm) vs voltage (V).

Sensor Nodes Programming

The choice for RFBee was determined by the easeness of programming the WSN nodes provided by Radiuino libraries. Radiuino has only two libraries functions, each one with a specific role in the WSN. One of the libraries is called BASE and the other is called SENSOR. The RFBee with the BASE library allied to the computer form the first node of the Radiuino platform. In order to simplify the naming the RFBee programmed with the BASE library will be designated as BASE and the RFBee programmed with the SENSOR library will be designated as SENSOR (Fig. 8).

On the computer to which BASE is connected, there is an application named dk101.py [8], written in Python, which is part of the Radiuino platform and whose main functions are:

- Send polling requests to the SENSOR in regular intervals;
- Receive and store the data sent by the SENSOR.

The BASE works as a bridge between the SENSOR and the dk101.py application. The BASE communicates via serial interface with the computer and via radio with the SENSOR. The SENSOR receives the request sent by the BASE, reads the water level and respond to BASE.

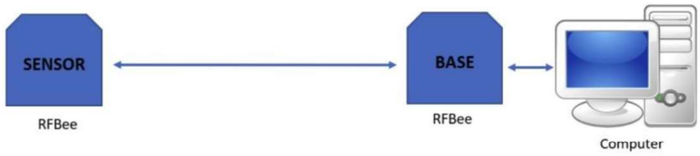


Figure 8. SENSOR and BASE libraries provided by Radiuino Platform

Originally, Radiuino was designed for one hop WSNs (Fig. 8). Because of the long distance, in this proposal, it was necessary to modify the BASE and the SENSOR libraries and create two new libraries due to their specific functions. Figure 9 illustrates a WSN with four types of nodes: Central Base (CB), End Device (ED), Router Node (RT) and Relay Node (RN).

The CB has a modified version of the Radiuino BASE library, in order to receive all frames regardless of whether they are intended for it. In this way, if there is another network available in the same area, it is possible to detect it and change the network channel to avoid collisions [20]. For the End Device (ED), in the Radiuino SENSOR library was included a sleep schedule and a waiting for call functions.

The RT library was created to coordinate the WSN communication. This node incorporated the polling request function from the dk101.py application, for doing the calls with only one hop distance to the EDs. After that, aggregates the EDs data and send them to CB. The polling requests migration to RT reduces network traffic, and, also contributes to the energy savings, because the polling requests do not need to go through several hops to reach the EDs nodes. In order to save RT energy, the hub polling method was chosen instead of roll-call polling, so only ED1 had to be called and after its response, ED2 response in sequence [11], and the polling responses from EDs were aggregated [13] in a single frame, which was sent to the CB.

Since there is a long distance between RT and CB a communication path had to be constructed using five relay nodes (RN). The RN function only was to receive and to retransmit frames. The relay node is only necessary when the distance between two nodes exceeds the range radius limit, this range for RFBee is 50 m for built-in environments or urban areas and 120 m for open-air environments [8]. RT and RN are a modified version of SENSOR library, so they can also be used for data collection. The RFBee libraries and the modified Phyton application can be downloaded from https://github.com/ihyano/rfbee.

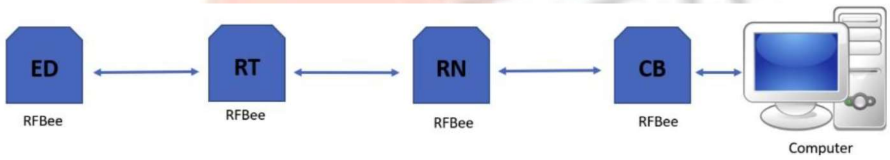


Figure 9. Network example with four types of nodes.

With the CB exception, which is powered by a USB port attached to a microcomputer, all other nodes are powered by batteries and use a sleep schedule function to reduce energy consumption. This reflects the concern in developing a technique in which the cabling or solar panel power supply [14] is impractical, either from a technical or financial point of view.

The [10] was the basis for the Arduino´s sleep schedule function, and the standard time intervals functionality of the watchdog timer (WDT) can be found at [15]. This recorder sets fixed times from 16 ms to 8 s. Therefore, any other sleep time is obtained from the combination of the ten predefined times, which are listed in the Watchdog Timer Prescale Select table on [15]. In this work, a sleep time of 1.5 s was achieved by the sum of 1.0 s (setting WDT bits 1 and 2 to on) and 0.5 s (setting WDT bits 0 and 2 to on).

Figure 10 shows, highlighted in green, small activity times between sleep periods. As described in the previous paragraph, a total sleep period of 1.5 s was the sum of two sleep periods, as a result of twice execution of sleep sub-routine, i.e., an execution for 0.5 s followed by a further of 1.0 s of sleep periods. This is because after the end of the first sleep period of 0.5 s, the node returns to a brief period of activity before entering the second sleep period. This can be monitored with the aid of an oscilloscope (Fig. 10). These two executions of the sleep subroutine are necessary because, as mentioned previously, any sleep period must be a combination of the times standardized by the WDT register.

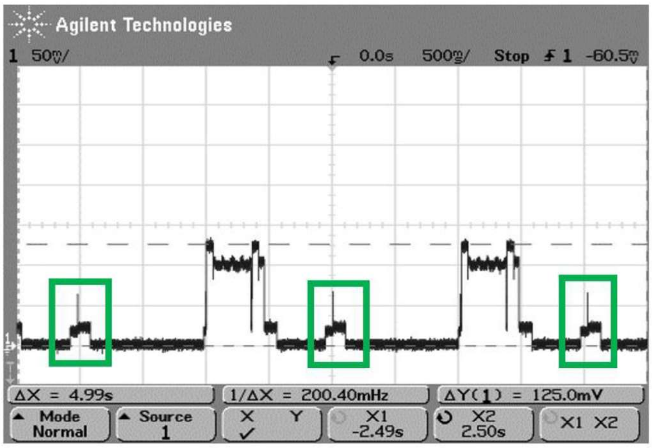


Figure 10. Activity cycles in RT node with short periods of activities during sleep time.

V. Results

The test to verify the communication between the Rectory, where the evaporimeter is located, and the CEATEC, where the data will be stored and processed, was carried out with the provisional placement of the sensors in the locations indicated and identified by letters in Fig. 11 and checking the frames arrivals at each point. The sensors are represented by circles with identification inside (node function and its logical address). The sensors are identified by a color code according to their function in the network (red for ED, green for RT, blue for RN, and black for CB).

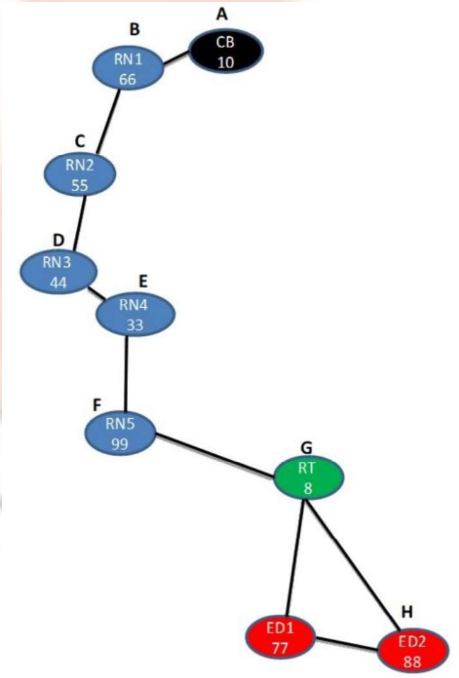


Figure 11. Network topology with nodes identified by a color code.

Figure 12 shows the Received Signal Strength Indicator (RSSI) values measured in each WSN node. The RSSI values were smaller when distances were greater or when there were many obstacles, such as walls. This was the case of the transmission from point B to point A, because point A is inside a room in the mezzanine at the Hydraulics Laboratory of the Environmental Engineering at CEATEC PUC-Campinas.

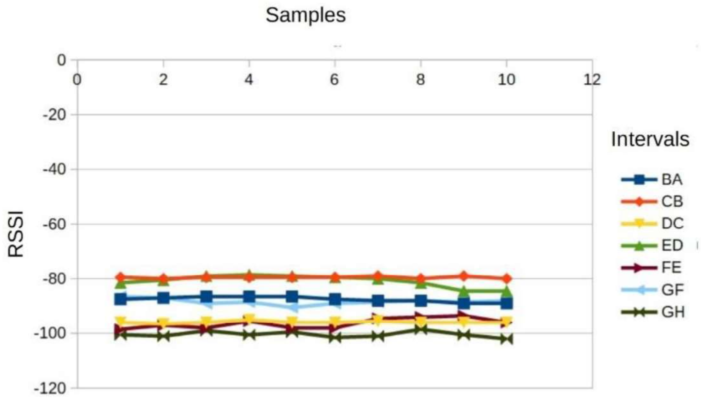


Figure 12. RSSI measurement in each network node.

VI. Conclusion

This paper presented the creation of new libraries for new nodes functions and some strategies chosen to save energy to extend the lifetime of a multi-hop WSN. A sleep schedule function was developed to reduce energy consumption in all nodes powered by batteries. Other techniques applied in this work were the use of hub polling and data aggregation to save RT energy. Thus, the RT will not die long before the others WSN nodes, and when it occurs all the WSN nodes can have their batteries exchanged in one go, since there will not be a large difference between their charges. This has the advantage of avoiding partial network interruptions, that interrupts data collection and make maintenance expensive and frequent.

The WSN implementation in an open environment, in a scenario with six hops, was an important test to verify the robustness and flexibility of RFBees sensors. This implementation will bring several advantages to the Environmental Sustainability Group for the Cities (GSAC) of PUC-Campinas, among them: to collect data several times over the day and mainly to have access to the environment data, where the measurement of the evaporation tank level is the most important.

In the continuity of this work, the technique performance will be evaluated for networks with a larger number of nodes and with different periods of RT activity.

VII. Acknowledgment

The authors thank the Pontifical Catholic University of Campinas for the scholarship support related to this work and to FINEP / CEATEC / HIDRO for the financial support.

References

[1] Piyare, R., & Lee, S. R. Performance analysis of XBee ZB module based wireless sensor networks. International Journal of Scientific & Engineering Research, v. 4, n. 4, p. 1615-1621, 2013.

[2] Gupta, G. P., Misra, M., & Garg, K. An Energy Efficient Distributed Approach-Based Agent Migration Scheme for Data Aggregation in Wireless Sensor Networks. JIPS, v. 11, n. 1, p. 148, 2015.

[3] Faludi, R. “Building Wireless Sensor Networks, Chapter 6 Sleeping – Then Changing the World”, O`Reilly, 2010.

[4] Xing, G., Sha, M., Hackmann, G., Klues, K., Chipara, O., & Lu, C. “Towards unified radio power management for wireless sensor networks”, Wireless Communications and Mobile Computing, 9(3), 313-323, 2009.

[5] XBee, XBee ZNet 2.5 / XBee PRO ZNet 2.5 OEM RF Modules. Available at: https://www.digi.com/resources/documentation/digidocs/pdfs/90000866.pdf, accessed November 30, 2018.

[6] Yang, H., & Sikdar, B. “Performance Analysis of Polling based TDMA MAC Protocols with Sleep and Wakeup Cycles”, Department of ECSE, RPI, Troy NY 12180 USA, ICC 2007 proceedings, 2007.

[7] Kabara, J., & Calle, M. “MAC Protocols Used by Wireless Sensor Networks and a General Method of Performance Evaluation”, Hindawi Publishing Corporation, International Journal of Distributed Sensor Networks, Volume 2012, Article ID 834784, 11 pages, doi:10.1155/2012/834784, 2011

[8] RFBee, RFbee V1.1 - Wireless arduino compatible node. Available at: https://www.seeedstudio.com/RFbee-V1-1Wireless-arduino-compatible-node-p-614.html, Accessed November 20, 2018.

[9] Ain, Q. U., & Jain, A. A NOVEL ENERGY EFFICIENT RELAY NODE PLACEMENT APPROACH FOR WSN, 2016.

[10] Morrissey, D. Sleeping Arduino - Part 5 Wake Up Via The Watchdog Timer. Available at: http://donalmorrissey.blogspot.com.br/2010/04/sleeping-arduino-part-5-wake-up-via.html, Accessed November 14, 2018.

[11] Yano, I. H., de Oliveira, V. C., Becker, M., & Correa, A. S. Applying a hybrid polling approach by software implementation to extend the lifetime of a wireless sensor network. Journal of Computer Science. ISSN: 1549-3636. Vol. 11, Issue 4, p. 699-706. DOI: 10.3844/jcssp.2015.699.706, 2015.

[12] Pantazis, N. A., Vergados, D. J., Vergados, D. D., & Douligeris, C. Energy efficiency in wireless sensor networks using sleep mode TDMA scheduling. Ad Hoc Networks, 7(2), 322-343, 2009.

[13] Maraiya, K., Kant, K., & Gupta, N. Wireless sensor network: a review on data aggregation. International Journal of Scientific & Engineering Research, v. 2, n. 4, p. 1-6, 2011.

[14] Sharma, H., Haque, A., & Jaffery, Z. Modeling and Optimisation of a Solar Energy Harvesting System for Wireless Sensor Network Nodes. Journal of Sensor and Actuator Networks, 7(3), 40, 2018.

[15] ATmega168. ATMega168 – 8-bit AVR Microcontrollers. Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega48_88_168_megaAVR-Data-Sheet-40002074.pdf. Accessed November 30, 2018.

[16] CC1101. CC1101 Low-Power Sub-1GHz RF Transceiver datasheet (Rev. I). Available at: http://www.ti.com/lit/ds/symlink/cc1101.pdf. Accessed November 30, 2018.

[17] Arduino. Arduino – Software. Available at: http://arduino.cc/hu/Main/Software. Accessed November 30, 2018.

[18] Radiuino. Home – Radiuino. Available at: http://radiuino.cc/. Accessed November 30, 2018.

[19] Milone eTape. Milone eTape Liquid Level Sensor. Available at: https://cdnshop.adafruit.com/datasheets/eTapeApp.pdf. Accessed November 20, 2018.

[20] Jong, Y., Zhang, W., & Xu, W. Guard Time Settings in TDMA Family MAC Protocol for Underwater Sensor Networks. In Proceedings of the 2017 The 7th International Conference on Computer Engineering and Networks. 2223 July, 2017 Shanghai, China (CENet2017) Online at href=" https://pos. sissa. it/cgi-bin/reader/conf. cgi? confid= 299"> https://pos. sissa. it/cgi-bin/reader/conf. cgi? confid= 299, id. 34. 2017.