CLOCK SYNCHRONIZATION: ESTIMATION OF NON-DETERMINISTIC DELAYS IN WIRELESS MESSAGE DELIVERY

Shiu Kumar[1] and Seong Ro Lee[2]

[1]Department of Electronics Engineering, Fiji national University, Suva, Fiji
[2]Department of Electronics Engineering, Mokpo National University, South Korea

ABSTRACT

Clock synchronization is significantly essential as they require universal time on WSN nodes for time measurement, event ordering and coordinated actions, and power management. This paper gives an insight of solving the problem of the non-deterministic delays that exist in the wireless message delivery. Sensor nodes consisting of Arduino Mega and 2.4 GHz nRF24L01+ radio modules are used, and based on the estimation of non-deterministic delays a clock synchronization protocol for WSN is proposed. The results obtained are quiet promising compared to the existing synchronization protocols for WSNs.

KEYWORDS

Clock Synchronization, Non-deterministic Delays, Timestamp, Wireless Communication, WSNs (Wireless Sensor Networks).

1.INTRODUCTION

Recent innovative changes in electronics and communication technologies have led to an increased attention to the field of WSNs (wireless sensor networks), making it a popular research topic. Having a broad range of applications and systems with enormously varying requirements and characteristics, WSNs can be applied to a huge diversity of fields such as military environment, habitat monitoring, medical and health care[1,2], industrial fields[3], home networks[4], disaster management, surveillance[5] and scientific environment for example detecting chemical[6], biological, radiological, nuclear, and explosive materials. Wireless sensor nodes have constraints such as limited energy as they are battery powered, have limited bandwidth and are small in size with limited storage and computational power.

Clock synchronization is one of the major research topics [7] in this domain, which requires giving all the network nodes a universal time scale to operate. This universal time scale is generally achieved by occasionally synchronizing the clocks at each node to a reference time source. An emblematical synchronization system relies on the transmission of numerous time-stamped radio messages from one node to the others. However, these radio message exchanges that take place during synchronization are subject to numerous non-deterministic delays such as send time, access time, propagation time, transmission time, reception time, receive time, and interrupt handling time as highlighted in[8]. These delays are a fundamental problem in clock synchronization and needs to be minimized or eliminated in order to increase the performance of

DOI : 10.5121/ijcnc.2015.7109

the synchronization technique. Very little research has been done in this regard as estimating the time spent on each of the non-deterministic components can be difficult.

Time synchronization of the nodes has been an open research issue in WSNs[7], which is also very important for energy conservation[9]. Lack of time synchronization can significantly reduce not only the network's lifetime but also its functions[7]. Due to the limited power supply, low power consumption is also an issue, thus in addition to using low-powered hardware; devices are often periodically put into sleep mode to preserve power. Hence, wireless sensors must be time-synchronized in order to schedule some actions (e.g. wake-up at the same time), to communicate during assigned time slots to localize and track objects or events. Furthermore, the communication range of wireless sensor devices is limited, they tend to be small in size and are limited in computational abilities, as well.

Time synchronization is very important in a distributed wireless network for the following reasons: data fusion, time interval calculation, contest-free access to radio channel and coordination[10]. To facilitate time synchronization, two different approaches exist[7]: (1) Distributed clocks can be synchronized to an actual time standard like Coordinated Universal Time (UTC), referred to as physical clock. (2) Distributed clocks can be relatively synchronized to one another, referred to as logical clocks. Synchronization is typically based on some sort of message exchange among sensor nodes. However, these message exchanges that take place during synchronization are subject to numerous delays. Such latency depends on the amount of time taken from the moment when the sender timer is read to the instant when the receiver timer is updated. The total latency in terms of end-to-end single-hop transmission includes the following terms:

- Send time: it is the nondeterministic time spent by a transmitter to construct the message and access the network interface.
- Access time: considers the time taken to access the wireless communication channel and depends on the network traffic and features of the communication protocol.
- Propagation time: time taken for the message to be transmitted from source to destination.
- Receive time: time taken by the receiver to receive and process the message.

These components are addressed by clock synchronization protocols. These delays are called packet delays, which is a key problem in time synchronization and being able to predict time spent on each can be difficult. In order to attain higher accuracy; eliminating or minimizing any of these packet delays is essential, thus greatly increasing the performance of the synchronization technique.

This paper focuses on the estimation of the non-deterministic delays in the wireless message delivery for the purpose of clock synchronization in order to improve the performance of the synchronization techniques. The remaining of the paper is organized as follows. In Section 2, the existing time synchronization protocols for WSNs are briefly discussed. The proposed method of estimating the non-deterministic delays in order to achieve higher synchronization accuracies is presented in detail in Section 3 while the results obtained are presented and discussed in Section 4. Finally, the conclusion of this research is presented in the last Section.

2. RELATED WORKS

Researchers have recently reported various time synchronization techniques for WSNs to tackle this issue. Time synchronization algorithms can be divided into two main categories depending on their method of synchronization; receiver-receiver and sender-receiver synchronization. In a sender-receiver synchronization method, the sender usually known as the server transmits the timestamp information, which is utilized by the receivers (clients) to synchronize their local clocks. Whereas in a receiver-receiver synchronization method, the receivers exchange their timestamp information's amongst them to synchronize their local clocks. However a reference beacon containing no timing information is used. The following paragraphs highlight some of the time synchronization protocols that have been reported.

The RBS protocol is based on the receiver-receiver synchronization method that synchronizes a cluster of wireless sensors within transmission range of the reference sensor node[11]. A third party broadcasts the reference beacon to all the receivers in range. The receivers timestamp the time when the reference beacon is received and hence all the receivers exchange this time information with their neighboring nodes, which is sufficient information to be able to calculate the offset and synchronize their local clocks. This method eliminates the send time and access times associated with packet delay, however the propagation time and receive time uncertainties still exists. A synchronization accuracy of 11µs has been reported for the RBS method implemented on Berkeley motes.

The FTSP method proposed in[12] is a sender-receiver synchronization method aimed for resource limited wireless platforms requiring stringent accuracy, using low communication bandwidth. MAC-layer time stamping has been utilized together with error compensation including clock skew estimation. The FTSP method records multiple time stamps on both the sender and receiver, which are made at each byte interval after the SYNC bytes as they are transmitted or received. However, it employs a single radio message with only the final error corrected timestamp embedded in it. The method was tested on the Berkeley Mica2 platform and with only 6 timestamps, the average and maximum per-hop synchronization errors were 1.4 µs and 4.2 µs respectively.

TPSN [13] is a timing synchronization algorithm for sensor networks based on the sender-receiver synchronization comprising of two phases. The first phase, known as the level discovery phase, involves creating a hierarchical topology of the network where every node is assigned a level. Only one node is assigned level 0, which is called the root node, while the other nodes are assigned level *i*. In the second phase called the synchronization phase, each node from level *i* is synchronized with another node from level *i*-1 to achieve a network wide time synchronization where all nodes are synchronized to the root node. The average single hop accuracy of the TPSN method reported is 16.9 µs.

To mention, a few other time synchronization methods reported are Pairwise Broadcast Synchronization[14], On-Demand Synchronization (ODS) [15], Clapping and Broadcasting Synchronization (CBS)[16], Delay Measurement Time Synchronization (DMTS) [17], Pairwise Lightweight Protocol[18] and Energy-efficient and rapid time synchronization (EETS) [19].

3. NON-DETERMINISTIC DELAY ESTIMATION

The sensor nodes used in this research consist of Arduino Mega 2560 and the Nordic single chip 2.4 GHz nRF24L01+ transceiver operating at an air data rate of 256 kbps. The NRF24 library was used and the Arduino communicates with the transceiver using the SPI (Serial Peripheral Interface) pins. The hardware wiring diagram is shown in Figure 1.

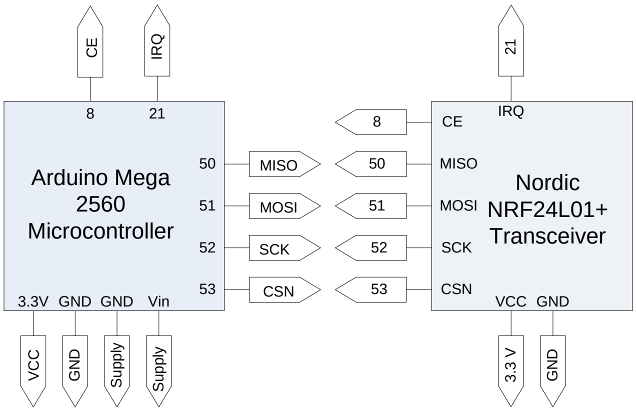
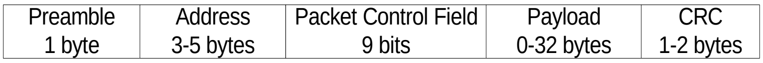


Figure 1. Hardware wiring of the sensor node.

3.1. The Packet Layout of the Wireless Message

Generally, wireless messages are sent in packets containing all the required information. This is the reason why the non-deterministic delays are also referred to as uncertainties in packet delays. The nRF24L01+ transceiver's embedded baseband protocol engine called Enhanced Shock Burst is used for this research and the packet layout used is shown in Figure 2.



Preamble 1 byte	Address 3-5 bytes	Packet Control Field 9 bits	Payload 0-32 bytes	CRC 1-2 bytes

Figure 2. Packet layout of the wireless message [20].

The length of this packet has to be known in order to estimate the upload, transmission and reception time. For this research, the packet contained 1 preamble byte, 5 address bytes, 9 Packet Control Field bits, 1 byte CRC (Cyclic Redundancy Check) and a payload of 20 bytes consisting of the time information. Time stamping at the MAC (Medium Access Control) layer is usually used by researchers to minimize the errors. The overall steps involved in sending a wireless message from one sensor node to another is shown in Figure 3.

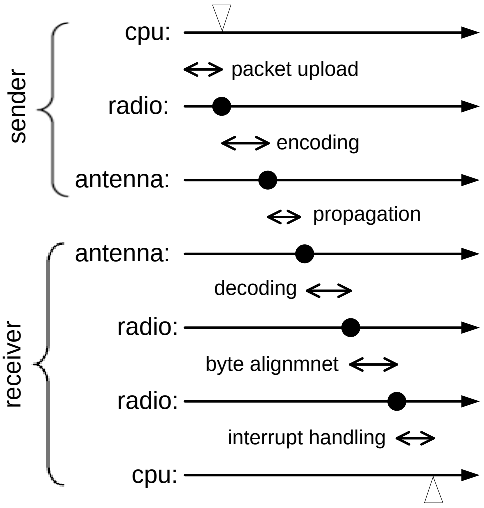


Figure 3. The timing of the transmission between the sender and the receiver.

3.2. Estimating the Non-deterministic Delays in the Sender

The timing sequence of the nRF24L01+ transceiver shown in Figure 4 was thoroughly investigated in order to determine the components of the non-deterministic delays in the sender and receiver.

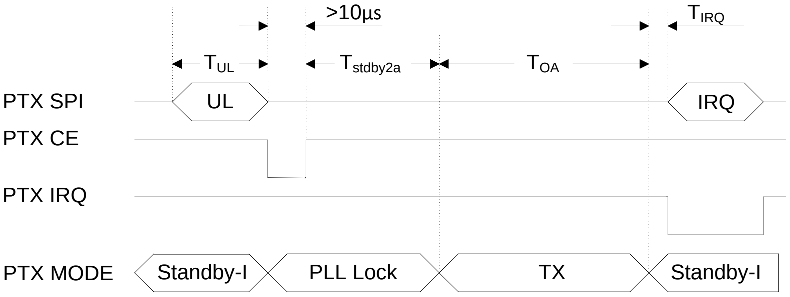


Figure 4. Timing sequence of the nRF24L01+ radio module[20].

The terms PTX refer to the Primary Transmitter, PTX CE is the Chip Enable pin, PTX mode indicates the mode in which the transmitter is at various stages, PTX IRQ is the interrupt pin, T_{UL} is the time taken to upload the payload to the PTX FIFO and T_{OA} is the time taken to transmit the radio message over the air. $\mathrm{T}_{\mathrm{stdby2a}}$ is the time taken for the radio to get into transmit mode while the $\mathrm{T}_{\mathrm{IRQ}}$ is the time taken for the interrupt to be generated after the data in the PTX FIFO has been transmitted over the air.

In order to estimate the sender uncertainties, all the parameters mentioned needs to be known or calculated. The value of T_{stdby2a} and T_{IRQ} are given to be 130 µs and 17 µs respectively [20]. Firstly, experiments were carried out in order to determine the relation between the wireless message payload size and the time taken to upload it to the nRF24L01+ transceiver using the spiBurstWrite() function of the NRF24 library. Low level programming was used to output high and low at one of the output pins of the Arduino Mega at start and finish of the upload process. An oscilloscope was used to record these and the relation given by Equation (1) was obtained, where PL is the payload size. Secondly, the time for which the PTX CE pin is low (denoted as T_{CE})

was also measured for all the possible payload sizes and it was discovered that this value was constant, having a value of 21.12 µs as shown in Figure 5.

$$T_{UL}\ (\mu s) = 2.06(PL) + 20.48 \qquad (1)$$

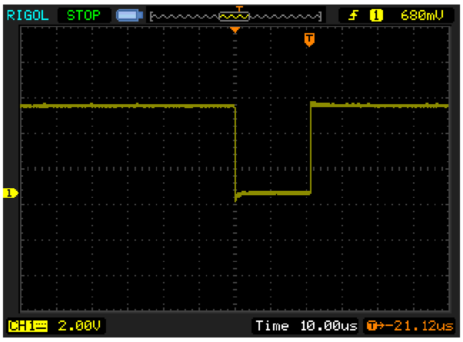


Figure 5. Result from oscilloscope showing the value of T_{CE} .

Further experiments were carried out in order to obtain the estimate of the relationship between the payload size and the time taken to transmit the wireless message packet over the air. The time taken after the PTX CE pin goes back to high after going low to the point when the PTX IRQ pin goes high was measured using an oscilloscope. This was done for the entire possible payload size at an air data rate of 256 kbps. The relation obtained from the results can be represented as given by Equation (2).

$$T_{OA}\ (\mu s) = 32(PL) + 260.15 \qquad (2)$$

3.3. Estimating the Non-deterministic Delays in the Receiver

On the receiver node, the PRX (primary receiver) IRQ pin goes high as soon as the wireless message packet is made available in the receiver's FIFO. Therefore, $T_{IRQ\text{-}D}$, the time between the PTX IRQ and the PRX IRQ pins were measured using the oscilloscope in order to estimate the propagation time and the message validation time that includes the decoding and byte alignment time. The value was found to be 6.185 ± 0.575 µs, which is independent of the wireless message payload size and was obtained by analysing one hundred samples of the collected results. However, this value varies slightly for each time the wireless message is transmitted due to the presence of interference from other sources such as Wi-Fi and other RF modules that were operating during the time of testing. It is also due to the reason that the nodes were performing other tasks during experimentation.

Finally, the PRX IRQ pin was used to generate an interrupt on the Arduino, which allows it to read the wireless message that is received. As mentioned, the sensor nodes were also performing other tasks during the experimentation in order to reflect the real time environment. Therefore, the time taken for the interrupt to be generated (T_{INT}) after the PRX IRQ pin goes high was also measured using an oscilloscope. Hundred samples of these measurements at varying payload sizes were recorded and the average value of 3.91 ± 0.39 µs was obtained.

3.4. Synchronizing the Sensor Nodes

All the components investigated in the previous section have their values or relationship with respect to the payload determined. In order to test the effectiveness of being able to estimate the non-deterministic delays in the wireless message delivery, the results obtained were used to synchronize the sensor nodes clocks. The general stages shown in Figure 6 were used for synchronizing the clocks of the sensor nodes.

A timestamp is made by the initial sender node, to which the uncertainties in the sender consisting of the payload upload time, T_{IRQ-D} and 1* (consists of T_{UL}, T_{CE}, T_{stdby2a}, T_{OA}, and T_{IRQ}) are calculated and added. This takes place during the packet formation process and it accounts for the estimate of the time taken for the message to be received by the receiver until the interrupt on PRX IRQ pin is generated. A timestamp is also made by the receiver node after the interrupt is generated. After the message is read, another time stamp is made in order to determine the time taken to read the received wireless message. This time taken to read the message together with T_{INT} value are than added to the timestamp received and the receivers node is than adjusted to this new time.

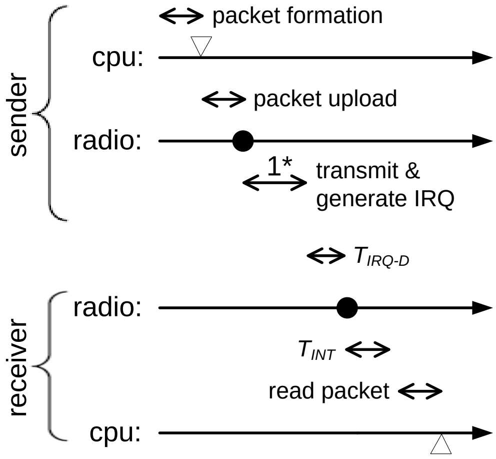


Figure 6. General stages of the wireless message delivery.

4. RESULTS AND DISCUSSION

The estimation of the non-deterministic delays in the wireless message delivery is presented in the previous section. These were used to synchronize the sensor nodes clocks to a reference nodes time using only a single message. The Arduino Mega 2560 was programmed to output a high at every timer1 overflow interrupt (4 ms) in order to measure the time difference between two sensor nodes after clock synchronization. The difference in the clocks of the sensor node gives the synchronization error and it shows the effectiveness of being able to estimate the nondeterministic delays in the wireless message delivery. Two hundred samples of the difference between the clocks of two sensor nodes were measured using the oscilloscope. An average synchronization error of 2.20 ± 2.44 µs was obtained. This shows that being able to predict the non-deterministic delays allows clocks to synchronize with fewer errors.

The experiment was then repeated using this estimation method to synchronize nodes that are at 5 hop distance away from the reference node with the correct time. Again two hundred samples of the clock errors between the reference node and the end node were recorded. Figure 7 shows the histogram showing the synchronization error.

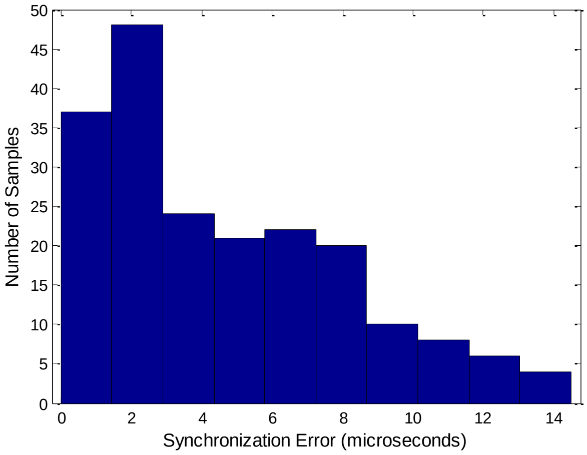


Figure 7. Histogram showing the 5 hop clock synchronization errors.

The absolute average synchronization error obtained is 4.61 µs. For approximately 57% of the times the synchronization error was less than the average synchronization error while 86% of the times the synchronization error was less than 8.69 µs. The best case was the two nodes getting perfectly synchronized and the worst case synchronization error was 14.48 µs. A summary of the multi-hop synchronization errors for the three most representative clock synchronization protocols together with the estimation method is given in Table 1. Comparing the results, the method of estimation proposed in this paper gives a very good result as it uses only a single wireless message per hop for synchronization. The authors of RBS and FTSP also obtained a low synchronization error, however multiple messages are required for synchronization of the sensor nodes. Therefore, the method of estimation uses a low bandwidth and since only a single message is used for synchronization it will consume less energy.

Table 1. Synchronization error over multi-hop.

Method	Error
RBS	3.68 µs on 4 hops
TPSN	20.0 µs on 4 hops
FTSP	3.00 µs on 6 hops
Estimation (proposed)	4.61 µs on 5 hops

5. Conclusions

The non-deterministic delays in the wireless message delivery have been successfully estimated with a high accuracy. Using the estimation method, nodes at one hop distance away require only a single message to synchronize its clock. The estimation method therefore maintains a low communication overhead and consumes very less energy as minimum number of message exchange takes place. The absolute average error of 4.61 µs at 5 hop distance has been achieved.

This estimation method can be applied to other clock synchronization methods in a similar way in order to enhance their performance. Future works in investigating and achieving a similar estimation for the widely used mica2 sensor nodes is recommended.

ACKNOWLEDGEMENTS

This research work was fully supported by Fiji National University through the College of Engineering, Science & Technology College Research Committee (CEST CRC).

REFERENCES

[1] Nitya, Pushpam, and M. Kaur, "An Advanced Wireless Sensor Network for Health Monitoring " International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 2, pp. 882-887, February 2013.

[2] Y. Charlon, N. Fourty, and E. Campo, "A Telemetry System Embedded in Clothes for Indoor Localization and Elderly Health Monitoring," Sensors, vol. 13, pp. 11728-11749, September 2013.

[3] Y. Guo, "Energy attenuation-based time synchronization for mobile underwater sensor networks," The Journal of China Universities of Posts and Telecommunications, vol. 19, pp. 57-64, 2012.

[4] S. Kumar, "Ubiquitous Smart Home System Using Android Application," International Journal of Computer Networks & Communications, vol. 6, pp. 33-43, January 2014.

[5] S. K. Singh, M. P. Singh, and D. K. Singh, "Intrusion Detection Based Security Solution for Cluster-Based Wireless Sensor Networks," International Journal of Advanced Science and Technology, vol. 30, pp. 83-95, May 2011.

[6] A. Terzis, R. Musaloiu-E., J. Cogan, K. Szlavecz, A. Szalay, J. Gray, et al., "Wireless Sensor Networks for Soil Science," International Journal of Sensor Networks, vol. 7, February 2010.

[7] B. Sundararaman, U. Buy, and A. D. Kshemkalyani, "Clock synchronization for wireless sensor networks: a survey," Ad Hoc Networks, vol. 3, pp. 281-323, 2005.

[8] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The Flooding Time Synchronization Protocol," in Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04), 2004, pp. 39-49.

[9] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: a survey," Ad Hoc Networks, vol. 7, pp. 537-568, 2009.

[10] W. Ikram, D. I. Stoianov, and P. N. F. Thornhill, "Towards a Radio-Controlled Synchronized WSN: A work in progress paper," presented at the IEEE EFTA Conference, Bilbao, Spain, 2010.

[11] J. Elson, L. Girod, and D. Estrin, "Fine-Grained Network Time Synchronization using Reference Broadcasts," in ACM SIGOPS Operating Systems Review - OSDI '02: Proceedings of the 5th symposium on Operating systems design and implementation, New York, NY, USA, 2002, pp. 147-163.

[12] M. Maróti, B. Kusy, G. Simon, and Á. Lédeczi, "The Flooding Time Synchronization Protocol," SenSys '04, 2004.

[13] S. Ganeriwal, R. Kumar, and M. B. Srivastava, "Timing-sync Protocol for Sensor Networks," in Proceedings of the 1st international conference on Embedded networked sensor systems (SenSys '03), 2003, pp. 139-149.

[14] K.-L. Noh, E. Serpedin, and K. Qaraqe, "A New Approach for Time Syncronization in Wireless Sensor Networks: Pairwise Broadcast Synchronization," IEEE Transactions on Wireless Communications, vol. 7, p. 5, September 2008.

[15] Z. Zhong, P. Chen, and T. He, "On-Demand Time Synchronization with Predictable Accuracy," in IEEE INFOCOM, 2011.

[16] X. Shen, X. Qian, B. Zhao, Q. Fang, and G. Dai, "Clapping and Broadcasting Synchronization in Wireless Sensor Networks," Tsinghua Science & Technology, vol. 16, pp. 632-639, 2011.

[17] S. Ping, "Delay Measurement Time Synchronization for WSNs," Intel Research, vol. 59, p. 11, 2003.

[18] J. V. Greunen and J. Rabaey, "Lightweight time synchronization for sensor networks," in WSNA '03 Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, New York, USA, 2003, pp. 11-19.

[19] B.-K. Kim, "Energy-efficient and rapid time synchronization for wireless sensor networks," IEEE Transactions on Consumer Electronics, vol. 56, p. 9, 2010.

[20] N. S. ASA, "nRF24L01+ Product Specification: Datasheet," 2008.

Authors

Shiu Kumar received Bachelor of Engineering Technology and Postgraduate Diploma in Electrical and Electronics Engineering from the University of the South Pacific in year 2009 and 2012 respectively. He received his Masters in Electronics from Mokpo National University, South Korea. His research interests include Automation and Control, Wireless Sensor Networks, Embedded Microprocessor Applications, Artificial Intelligence and Digital Electronics.