DOI. 10.1109/ROSE.2014.6952988

Low-Level Communication Time Analysis in Real-Time Wireless Sensor Networks

Valentin Stangaciu, Mihai V. Micea, Vladimir I. Cretu
Department of Computer and Software Engineering
Politehnica University of Timisoara, Romania
valys@dsplabs.cs.upt.ro, mihai.micea@cs.upt.ro
vladimir.cretu@cs.upt.ro

***Abstract*—Studying the time components and delays introduced by the lower level communication protocols in real-time wireless sensor networks, as well as taking them into consideration at the design phase of such protocols, still remains an open issue in the field. This paper addresses this problem through a detailed analysis of the communication times and their implications in providing a predictable low-level support for real-time sensor networks. A measurement framework is proposed specifically on this purpose and then used in an extensive set of experiments to validate this timing analysis.**

***Keywords*—wireless sensor network; real-time; communication; low-level protocol; time analysis; measurement framework**

I. INTRODUCTION

The field of wireless sensor networks (WSN) received a significant attention from the researchers during the last decade mainly because of their proved usefulness and importance in modern day applications [1]. Such applications are focused on controlling and monitoring the environment [2], security systems, robotic platforms and also on critical domains like fire detection and extinguishing systems [3] and military applications [4]. These latter types of applications require real-time conditions in order to meet their requirements.

In a real-time system, not only the correctness of the produced result is important but also the amount of time needed by the system to generate the result [5]. Such specifications also apply to wireless sensor networks that are used in critical applications. Therefore, in a real-time WSN, each node has to operate in a real-time manner and, additionally, the whole network must meet its time requirements. The success of achieving real-time constrains in WSN depends on the communication latency. This implies that, besides the successful delivery of a message, the time of its arrival at the destination is also of key importance [6].

II. RELATED WORK

Regarding real-time communication in a wireless sensor network, the network and application layers within the protocol stacks have received a greater attention in research and development works. Significant progress has been made especially in routing algorithms which take the time constraints into consideration [7, 8]. Some of these higher layer protocols have also been evolving towards becoming energy aware, while still maintaining their real time support [9].

One of the main issues still to be addressed is that the lower layers in the communication stacks have either been ignored, from the real time perspective, or the researchers start from the presumption that the medium access layer operates in real time and all deadlines are met by default [10]. Another important issue is that almost all the methods are based on networks that are organized in clusters where special nodes have crucial importance regarding communication. This can seriously affect the reliability of the whole network in case these network coordinators fail [9]. Also many researchers start from the presumption that all the nodes of the network are non-mobile and their position is known at deployment [11].

As stated before, lower level protocols are not so evolved regarding their real-time characteristics. Many of the studies on the higher level protocols and many of the existing wireless sensor networks are based on a standard designed for this kind of networks, the IEEE 802.15.4 [12]. The main issue of this standard is that its real time support is extremely weak [13] and is not feasible for a real-time environment, even if improvements were made [14].

The main studies regarding the medium access layer protocols for WSNs are focused on energy efficiency rather than on the time constrained, critical aspects, even if some of them offer some minimal support for this type of requirements [15, 16].

Many of the current proposals for a MAC protocol are oriented on soft real time requirements [17]. Significant progress has been made in real time MAC communication protocols when using TDMA and FDMA techniques [18] but the disadvantages are that a coordinator node is usually needed, forcing the network to be cluster-organized. On an unorganized ad hoc network with a random deployment of the nodes, as well as on a network where the nodes have mobility capabilities, these techniques are hard to use.

A large number of MAC protocol models are validated in a simulation environment and, for almost all of them, there are not detailed analyses on the time components that affect the predictability of the communication in a wireless sensor network.

III. COMMUNICATION TIME ANALYSIS FROM THE REAL-TIME PERSPECTIVE

From the real-time perspective, the key parameter in wireless communication is the time needed for a message to be delivered from node A to node B. In order to meet the real-time constraints, the components of this time parameter need to be correctly identified and modeled. Such a description of the total time needed to deliver a message from node A to node B is presented in [11] as follows:

$$T(A,B) = T_C + T_t + T_{pp} + T_p + T_q + T_S \tag{1}$$

where T_c is the time needed for node A to obtain the wireless channel (practically the time needed to make a *clear channel* assessment), T_t is the time needed to transmit the packet (which is determined by packet length, modulation, bandwidth), T_{pp} is the propagation time of the message from node A to node B, T_p is the data processing delay, T_q is the queuing delay and T_s is the delay caused by each node's periodic sleep intervals [11]. This model describes the total delivery time from the perspective of the routing layer.

However, regarding the medium access layer another approach is needed to describe the time components involved in the communication between nodes. Before analyzing the delay introduced by the MAC layer, certain assumptions and clarifications have to be made. First of all, from a communication point of view, each node of the network has as main components a host microcontroller and a wireless communication module. We also consider that the wireless module cannot activate the receiver and the transmitter at the same time, as are most of the wireless modules on the market. One of the microcontroller's main functions is to implement, in a real-time manner, the communication protocol stack. Therefore, we have to consider not only the delay introduced by the delivery of a message from node A to node B, but also the MAC layer processing delay, with its main component, the delay introduced by the wireless module driver.

In (2) we propose a more accurate model of the total delay for a transaction between a transmitter node A and a receiving node B.

$$\begin{aligned} T(A,B) = {} & T_{transfer_A} + T_{TXON_A} + T_{TX\_SFD_A} + \\ & + T_{RX\_SFD_B} + T_{RX\_PACK_B} + T_{transfer_B} \end{aligned} \tag{2}$$

The time parameters in (2) are defined as follows:

- $T_{transfer\_A}$ – the time needed for the host processor of node A to transfer the packet to the wireless module. This time component is predictable and calculable. It depends on the packet size and on the wired communication protocol between the host process and the wireless module.
- T_{TXON_A} – the time needed by the wireless module to deactivate the receiver and to activate the transmitter. This parameter is depended on the wireless module characteristics. Usually this parameter is given by the manufacturer of the module in its datasheet, as a maximum value.
- $T_{TX\_SFD\_A}$ – the time needed by the wireless module to transmit the Start of Frame Delimiter (SFD) sequence. This component usually includes the clear channel assessment (CCA) procedure. This parameter is highly dependent on the wireless medium.
- $T_{RX\_SFD\_B}$ – the time needed for the SFD sequence to arrive at its destination node B. This parameter depends on the wireless medium characteristics as well as on the bitrate and modulation used by the wireless modules.
- T_{RXPACK_B} – the amount of time needed by module B to receive the whole packet.
- $T_{transfer\_B}$ – the time needed by the host processor of node B to transfer the newly arrived packet from the wireless module to its internal memory. This parameter has the same characteristics as $T_{transfer\_A}$.

Another important aspect regarding the communication delay is represented by the time needed by the host microcontroller to control the transmission and reception process. This aspect is important when designing the driver for the wireless module in a real-time context. Usually in a WSN with real-time capabilities, a real-time operating system (RTOS) is needed, as the main software component of each node. A driver design for a wireless module within such an RTOS needs to take into consideration several time components. For a transmitter node (A), these time components can be described as follows:

$$\begin{aligned} T(A) = {} & T_{transfer} + T_{TXON} + T_{TX\_SFD} + \\ & + T_{TX\_PACK} + T_{RX\_ON} \end{aligned} \tag{3}$$

The time parameters in (3), like $T_{transfer}$, $T_{transfer\_A}$, T_{TXON}, T_{TX_SFD} are similar to those in (2). The remaining parameters have the following meaning:

- T_{TX_PACK} – the time needed for the module to transmit the entire packet. This component is usually predictable and is dependent on the packet size
- T_{RX_ON} – the time needed by the module to change from the transmitting mode to receiving mode after the packet has been sent. This parameter is usually a characteristic of the wireless module.

The same analysis is performed on the receiver side (B):

$$T(B) = T_{RX\_PACK} + T_{transfer} \tag{4}$$

Here, parameter T_{RX_PACK} can be described as the time from the moment the module signals the arrival of a SFD sequence to the moment the module receives the whole packet. The $T_{transferB}$ parameter is the time needed for the host microcontroller to transfer the received data from the communication modules.

IV. MEASUREMENT FRAMEWORK

This chapter describes a measurement framework proposed and used to validate the above analysis. The measurement framework, described in Fig. 1, has as a main component an Olimex LPC-H2294 board [19] with a NXP LPC2294 microcontroller [20]. The board also has a 1MB SRAM memory which was used to temporary store the measurement data. The microcontroller peripherals can be configured in order to obtain a total of 12 capture inputs: 8 timer capture inputs from the 2 dedicated timers and 4 additional external interrupts. In this configuration one can capture a total of 20 different events (each capture channel can detect 2 events, one for the rising edge and one for the falling edge, and each interrupt can detect 1 event). The timestamp for each event is assigned from one of the dedicated timers which represent the system reference clock. Another important aspect is that the two dedicated timers were synchronized with an error of 1 tick (62.82 ns).

This measurement hardware was designed in order to measure the events generated by two communicating nodes (node A and node B) described as follows:

TABLE I. EVENT DESCRIPTION

Event no	Event description	Capture channel	Falling/ rising edge
1	Module received SFD sequence	External interrupt 0	Rising edge
2	Module received packet. Host processor started packet transfer	Capture channel 0	Rising edge
3	Finished packet transfer to host processor after packet receive	Capture channel 0	Falling edge
4	Host processor started packet transfer from internal memory to the wireless module	Capture channel 1	Rising edge
5	Host processor finished packet transfer to wireless module. Host processor commands wireless module to switch to transmit mode	Capture channel 1	Falling edge
6	Wireless module started the transmission procedure	Capture channel 2	Rising edge
7	Module transmitted the SFD sequence	Capture channel 2	Falling edge
8	Module finished transmitting the entire packet. Host processor commands wireless module to switch in receiving mode	Capture channel 3	Rising edge
9	Module finished switched to receiving mode	Capture channel 3`	Falling edge

The above table presents the meaning of the events and the capture channels that were assigned, as well as the edge multiplexing, regarding module A. A similar event mapping is for module B, using the remaining capture channels. The time parameters described in (1), (2) and (3) can be obtained with these events, as presented in Table II. For example, the time parameter $T_{transfer\_A}$ can be obtained by subtracting the timestamp of event 4 generated by module A (EV_{4_A}) from the timestamp of event 5 generated by module B(EV_{5_A}).

To minimize overhead, the microcontroller saves the events into the 1 MB memory with their respective event ID and timestamp. The timestamp is represented by the actual value of the system timer. All the calculations are made offline on a host PC that downloads the measured data from the measurement board, via an UART interface. A software tool has also been created to automate the entire measurement process. The PC software not only downloads the data from the measurement board but also commands the two nodes in order to generate wireless traffic.

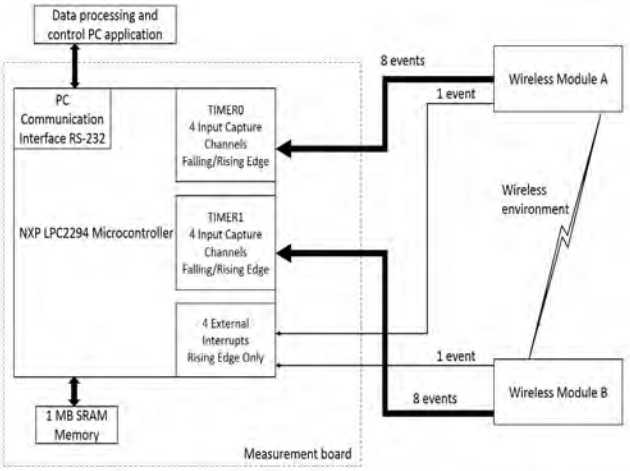

Fig. 1. Measurement platform

TABLE II. TIME PARAMETERS CALCULATION

Time parameter	Formula
$T_{transfer\_A}$	$EV_{5\_A} – EV_{4\_A}$
T_{TXON_A}	$EV_{6\_A} – EV_{5\_A}$
$T_{TX\_SFD\_A}$	$EV_{7\_A} – EV_{6\_A}$
$T_{TX\_PACK\_A}$	$EV_{8\_A} – EV_{7\_A}$
$T_{RX\_ON\_A}$	$EV_{9\_A} – EV_{8\_A}$
$T_{RX\_SFD\_B}$	$EV_{1\_B} – EV_{7\_A}$
$T_{RX\_PACK\_B}$	$EV_{2\_B} – EV_{1\_B}$
$T_{transfer\_B}$	$EV_{3\_B} – EV_{2\_B}$
T(A,B)	$EV_{3\_B} – EV_{4\_A}$

V. EXPERIMENTAL RESULTS

A set of experiments have been conducted to validate the time analysis and measurement framework presented above. The results were collected from measuring the communication delays using two platforms as case studies. The first platform is represented by two modules of the CrossBow MicaZ hardware platform [21] which consists of a host processor and a ChipCon CC2420 wireless module [22]. This platform usually has as its main software component the TinyOS operating system, but within these measurements this system was not used. A low level driver for the wireless module was used instead. The second platform is represented by two Olimex LPC-H2294 boards with a ChipCon CC2500 [23] wireless module attached to it. In each case the platforms were programmed to generate the events described in Table I using the General Purpose Input output (GPIO) pins from the microcontroller.

In both cases the firmware of the platform is represented by a low level driver for the wireless module, capable of generating the needed events.

To properly validate the time components presented earlier, a significant number of measurements were needed. The packet delays were measured in various conditions: high and low transmit gain, low noise medium, high noise medium, etc. The noise was introduced by generating wireless traffic on adjacent channels.

TABLE III. MEASUREMENT COUNT PER PLATFORM

Module	Max packet size	Measurements per packet size in various conditions	Time components measured	Total measurements
CC2420	64 bytes	600	10	384000
CC2500	50 bytes	400	10	200000
Total				**584000**

In Fig. 2 we present the results collected from measuring the time needed by the CC2420 module to deactivate the receiver and activate the transmitter. In Fig. 4 the opposite situation is presented: the CC2420 switches form transmitting mode to receiving mode. An analysis of these plots can conclude that this module may be used in a real time environment.

Another important observation can be made by analyzing the time needed by the CC2500 module to transmit the SFD sequence. The behavior of this delay is presented in Fig. 3, and leads to the conclusion that this component may be a significant issue in a real time protocol. Also, this component is not given as a maximum value by the producers.

VI. CONCLUSIONS

Three major delays have been identified for the communication time at the MAC level in real-time WSNs: the peer-to-peer message transaction delay, and the transmission and reception control delays at host level. A detailed analysis of their components and the results of extensive measurements reveal the impact on the communication predictability of some of the components of these delays, as well as the necessity of taking them into consideration when designing low-level protocols featuring real-time support.

ACKNOWLEDGMENT

This work was partially supported by the strategic grant POSDRU/159/1.5/S/137070 (2014) of the Ministry of National Education, Romania, co-financed by the European Social Fund – Investing in People, within the Sectoral Operational Programme Human Resources Development 2007-2013.

REFERENCES

[1] L. B. Ruiz, J. M. Nogueira, and A. A. F. Loureiro, "Sensor Network Management," in Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, M. Ilyas and I. Mahgoub, Eds., ed Boca Raton, Florida: CRC Press, 2005, ISBN: 0-8493-1968-4, pp. 57-84.

[2] J. V. Capella, A. Bonastre, J. J. Serrano, and R. Ors, "A pollution monitoring system based on an energy efficient Wireless Sensor Networks architecture," in 17th Telecommunications forum TELFOR, Serbia, Belgrade, 2009, pp. 1145-1148.

[3] P. N. N. Reddy, P. I. Basarkod, and S. S. Manvi, "Wireless Sensor Network based Fire Monitoring and Extinguishing System in Real Time

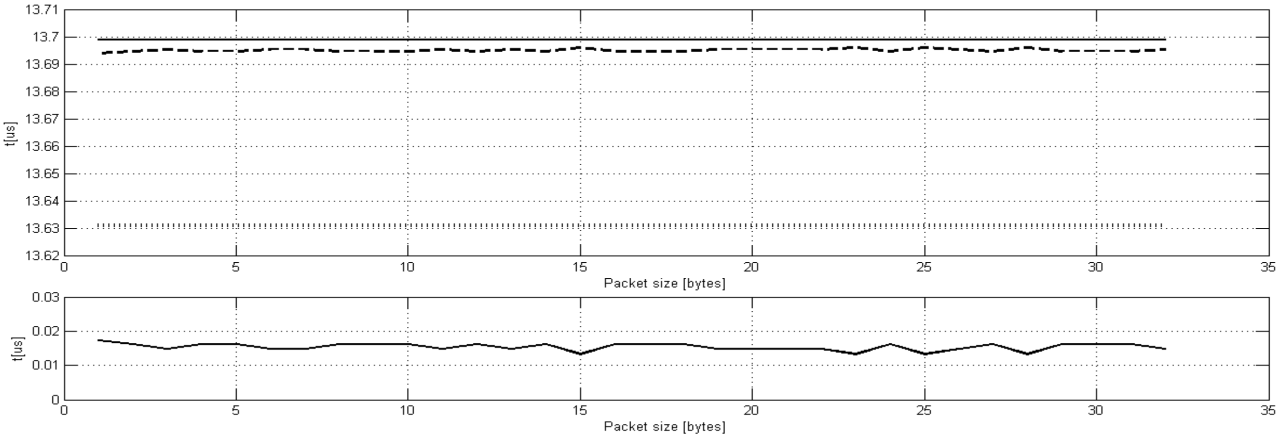

Fig. 2. CC2420 based platform – Delay generated when switching to transmit mode. a) Maximum, average and minimum values, b) Sigma values

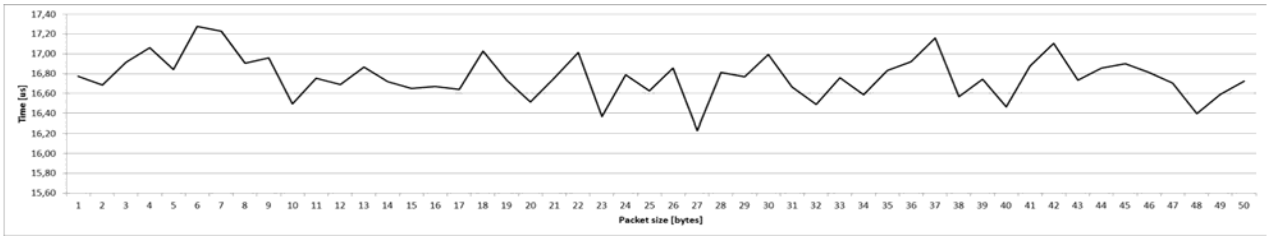

Fig. 3. CC2500 based platform – SFD reception delay

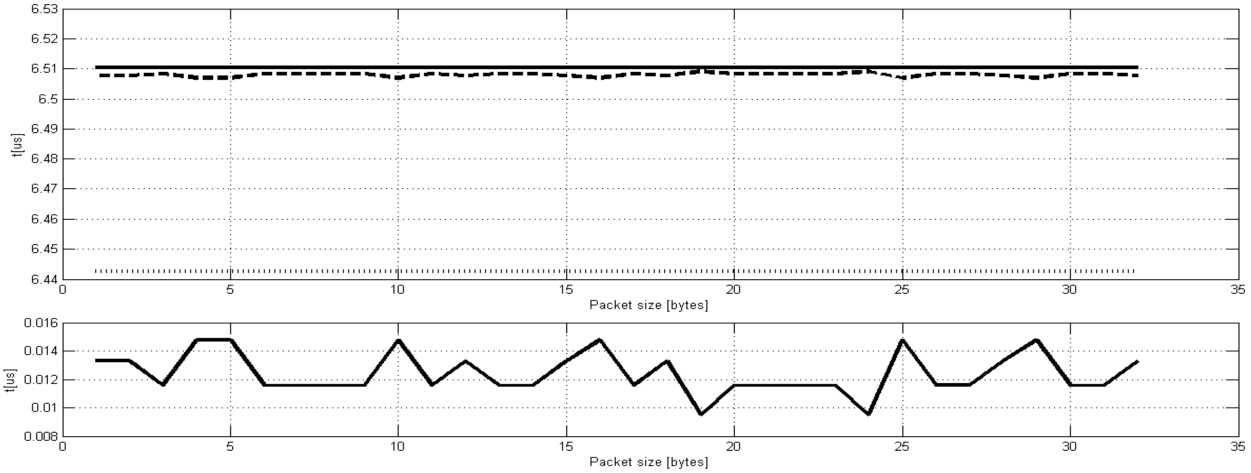

Fig. 4. CC2420 based platform – Delay generated when swithing to receving mode. a) Maximum, average and minimum values, b) Sigma values

Environment," International Journal Advanced Networking And Application, vol. 3, no. 2, pp. 1070-1075, 2011.

[4] M. J. Markowski and A. S. Sethi, "Wireless MAC Protocols for Real-Time Battlefield Communications," in Military Communications Conference MILCOM, Monterey, Canada, 1997.

[5] G. Butazzo and P. Gai, "Efficient EDF Implementation for Small Embedded Systems," in orkshop on Operating Systems Platforms for Embedded Real-Time applications OSPERT, Dresden, Germany, 2006.

[6] K. K. Z. Teng, "A Survey on Real-Time MAC Protocols in Wireless Sensor Networks," Communications and Network, vol. 2, no. 2, pp. 104-112, 2010.

[7] P. Rezayat, M. Mahdavi, M. GhasemZadeh, and M. AghaSarram, "A novel real-time routing protocol in wireless sensor networks," in Current Trends in Information Technology CTIT International Conference on the, 2009, pp. 1-6.

[8] Li-Ming He, "A Novel Real-Time Routing Protocol for Wireless Sensor Networks," in 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing SNPD, 2009, pp. 411-416.

[9] F. J. Atero, J. J. Vinagre, E. Morgado, and M. R. Wilby, "A Low Energy and Adaptive Architecture for Efficient Routing and Robust Mobility Management in Wireless Sensor Networks," in 31st International Conference on Distributed Computing Systems Workshops ICDCSW, Minneapolis, Minnesota USA, 2011, pp. 172-181.

[10] G. S. A. Kumar, G. Manimaran, and Z. Wang, "End-to-End Energy Management in Networked Real-Time Embedded Systems," Parallel and Distributed Systems, IEEE Transactions on, vol. 19, no. 11, pp. 1498-1510, Nov 2008.

[11] A. Ali, L. A. Latiff, M. A. Sarijari, and N. Fisal, "Real-time Routing in Wireless Sensor Networks," in 28th International Conference on Distributed Computing Systems Workshops ICDCS, 2008, pp. 114-119.

[12] IEEE Computer Society, "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless personal Area Networks (LR-WPANs)," in 802.15.4d, ed: Institute of Electrical and Electronics Engineers, Inc., 2009.

[13] G. Anastasi, M. Conti, and M. Di Francesco, "A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802.15.4 Wireless Sensor Networks," Industrial Informatics, IEEE Transactions on, vol. 7, no. 1, pp. 52-65, 2011.

[14] Y. Seong-Eun, C. Poh Kit, K. Daeyoung, D. Yoonmee, P. Minh-Long, C. Eunchang, and H. Jaedoo, "Guaranteeing Real-Time Services for Industrial Wireless Sensor Networks With IEEE 802.15.4," Industrial Electronics, IEEE Transactions on, vol. 57, no. 11, pp. 3868-3876, 2010.

[15] B. Fateh and M. Govindarasu, "Energy-Aware Adaptive MAC Protocol for Real-Time Sensor Networks," in Communications (ICC), 2011 IEEE International Conference on, Kyoto, Japan, 2011, pp. 1-5.

[16] Y. Sun, S. Du, O. Gurewitz, and D. B. Johnson, "DW-MAC: A Low Latency, Energy Efficient Demand-Wakeup MAC Protocol for Wireless Sensor Networks," in 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing MOBIHOC, 2008, pp. 53-62.

[17] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "RAP: a real-time communication architecture for large-scale wireless sensor networks," in Eighth IEEE Real-Time and Embedded Technology and Applications Symposium RTAS, Proceedings. Eighth IEEE, 2002, pp. 55 - 66.

[18] D. Abdeli, S. Zelit, and S. Moussaoui, "RTH-MAC: A real time hybrid MAC protocol for WSN," in Programming and Systems (ISPS), 2013 11th International Symposium on, 2013, pp. 153-162.

[19] Olimex. (2005). LPC-H2294 Get Started Guide. Available: https://www.olimex.com/Products/ARM/NXP/LPC-H2294/resources/H2294.chm

[20] NXP Semiconductors, "LPC2119/2129/2194/2292/2294 User Manual," Koninklijke Philips Electronics N.VMay 2004.

[21] Crossbow, "MPR-MIB Users Manual," Crossbow Technology Inc, Revision B 2006.

[22] ChipCon Products from Texas Instruments, "CC2420 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver," Texas Instruments 2012.

[23] ChipCon Products from Texas Instruments, "CC2500 Low-Cost Low-Power 2.4 GHz RF Transceiver," Texas Instruments 2011.