TIME SYNCHRONIZATION IN LORA WIRELESS NETWORKS

Slađana Đurašević Pešović[1], Uroš Pešović[2]
[1]https://orcid.org/0000-0002-3598-5781, [2] https://orcid.org/0000-0001-8722-6544
Faculty of technical sciences Cacak, University of Kragujevac, Serbia

Abstract

LoRa technology has brought significant improvements in terms of range and resistance to interference, which has significantly expanded the field of application of wireless sensor networks (WSN) to larger geographical areas. Time synchronization is a significant problem in WSN since it is necessary to synchronize the local clocks of a large number of sensor nodes with sufficient accuracy. Since communication in WSN is performed at low data rates and nodes spend a lot of time in a low-power modes, it becomes challenging to maintain time synchronization between nodes under such conditions. This paper presents the method of time synchronization between WSN network nodes using LoRa technology, and based on measurements, the expected synchronization accuracy is determined in the case of crystal resonators and crystal oscillators, which are most often used as clock sources in WSN nodes.

Keywords: LoRa, Wireless sensor network, time synchronization, IEEE 1588 PTP.

INTRODUCTION

Wireless sensor networks consist of a large number of battery-powered nodes that are spatially distributed in specific environments to monitor certain phenomena. These nodes are completely autonomous and are able to work for months and years without human intervention. Therefore, they must be extremely efficient in terms of energy consumption, small enough to fit into any environment and low-cost to be used in large numbers.

The problem of time synchronization is important in certain types of wireless sensor applications that require recording of exact time when the measurement of a certain measurement was performed. Since wireless sensors most often rely on timing using a locally generated clock signal, without time synchronization, the measurement error will increase over time [1]. Oscillator error is expressed in ppm units, which is the deviation of the generated frequency from the specified frequency expressed in parts per million. Depending on the type of oscillator used, the time measurement error ranges from a few tens of thousands of ppm in the case of internal RC oscillators to a few tens of ppm in the case of using passive quartz oscillators. Higher timing accuracy can be achieved by using temperature compensated or temperature controlled oscillators or by using GPS disciplined oscillators, but such solutions are usually impractical for WSN due to larger dimensions, power consumption and high cost [2]. Nodes in WSN generally use crystal oscillators, which cause significant deviations between local clocks over a long period of time, especially in the case of variations in the sensor supply voltage due to battery discharge or changes in the sensor's environment such as temperature, vibration or pressure.

The problem of time synchronization is particularly important in applications where several sensor nodes monitor some phenomenon, where it is necessary for the sensors to have a common time reference in order to be able to implement coordinated actions [3]. A typical example is determining the exact location of the occurrence of a certain phenomenon using the triangulation method. Also, wireless sensor nodes use long sleep intervals to reduce energy consumption, so synchronization mechanisms are necessary to activate sensors only when their participation in information exchange with neighboring nodes is required.

In these applications, it is necessary to have a centralized time synchronization, where the main sensor node, using special synchronization packages, informs the other nodes about the exact time, based on which they correct their local time [4-6].

This paper presents the method of time synchronization between WSN network nodes using LoRa technology, and based on measurements, the expected synchronization accuracy is determined in the case of crystal resonators and crystal oscillators, which are most often used as clock sources in WSN nodes.

LORA TECHNOLOGY

Wireless sensor networks relied on different communication protocols for data transmission between nodes, which enabled transmissions at distances of up to several hundred meters, which significantly limited the coverage area of wireless sensor networks. In recent years, thanks to LoRa technology, which enables a signal range of up to several kilometers, wireless sensor networks can be set up over a wider geographical area with a smaller number of nodes, which has significantly reduced the costs of setting up the network.

LoRa ("Long Range") technology uses the Chirp Spread Spectrum technique, which encodes data using a sinusoidal signal whose frequency continuously increases during the duration of one symbol called a chirp. The range of frequency change is limited by a bandwidth BW that ranges from 7.8 kHz to 500 kHz. The spread spectrum technique makes it possible to send a larger number of pulses, called chips, instead of one bit of data. LoRa technology supports the use of different degrees of spreading (SF - Spreading factor) which range from 6 to 12 and are powers of the number 2. Higher degrees of spreading enable much better signal detection in the case of low values of signal strength in relation to noise and interference in the channel. So in the case of the spreading factor SF=12, one bit is replaced by 4096 chips, which enables signal detection even when the received signal is 100 times (-20dB) weaker than the background noise. The use of a larger spread factor enables communication over greater distances because in this way the sensitivity of the receiver is improved, but the result is a lower transmission speed. As a result, data transmission airtime takes longer, which results in higher energy consumption.

In order to further increase the transmission reliability, LoRa technology uses forward error correction code, where 1 to 4 redundant parity bits are sent with a series of four data bits, which enable the correction of reception errors, without the need for retransmission.

Air time is dependant of number of symbols which are present in the message, and the duration of symbol which is calculated by following formula

$$T_{sym}=2^{SF}/BW \qquad (1)$$

TIME SYNCHRONIZATION

In order to conserve sensor node battery energy, microcontroller core spends long periods of time in low power sleep mode. In this mode, the main oscillator stops working, which also pauses the Arduino TC0 timer which is used by built-in functions for time measurments. TC0 timer is clocked from 16 MHz external oscilator and is preseted to generate interrupts every 1024 µs. Each TC0 overflow interrupt increments global vaiables which are used to measure elapsed time in microseconds and miliseconds using *micros()* and *millis()* functions respectevely. The only option for AVR microcontroller to keep track of time during low power speep mode is to use TC2 timer in asynchronous mode, or watchdog timer or a external real-time clock module.

In Arduino Uno and Nano boards, TC2 timer cannot be used in asynchronous mode with low frequency oscilator, since it shares the same pins which are occupied with external 16 MHz oscilator.

In the case of a master node, it is most practical to use an external real-time clock module with that allows independent time measurement from the state of processor core. Such RTC clocks use precise crystal oscillators that operate at a frequency of 32768 Hz and have an accuracy of ±20ppm, which corresponds to a time measurement error of approximately ±1 minute per month of operation. Such clocks have an output that generates an interrupt signal every second, so the microcontroller can use this information to synchronize its local TC0 high-resolution timer. Real-time information can be exchanged by the master node with slave nodes in the form of a 48-bit value shown in Fig 1. This value consists of two components, a first part of 32 bits that contains the exact time and date data in UNIX format, and a second part of 16 bits that is used to display the precise time with a resolution of 16 μs obtained from the local TC0 oscillator.

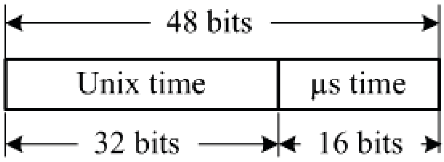

Fig. 1. *Timestamp synchronization message*

When the master node is woken up from sleep mode by the RTC clock, it starts its local oscillator and sets its TC0 timer to start accurate timing in microseconds. Since the time required to process an interrupt and execution of interrupt routine is deterministic, that elapsed time is added at the time of initialization of the global variables used for TC0 high-resolution timer. When going into deep sleep mode, the main microcontroller can set an alarm in the RTC clock that will generate an interrupt to wake up the microcontroller at the desired time.

In the case of slave nodes, they do not need to have RTC clocks, but use the real-time information they receive from the master node in the network. Based on that information, they tune their local TC0 oscillator to measure time in high resolution. In case of going into low power mode, slave node will set their own watchdog timer to wake him from sleep mode after the exiration of watchdog period which is selectable in interval from 16 ms to 8 seconds. Time synchronization requires that the nodes in the WSN network remember the current value of their local clock (timestamp) at the same moment. The local time sampling procedure must be synchronized for all nodes in the network and is done by sending synchronization packets. Since the packet transmission time is very long in the case of LoRa technology, synchronization is performed using interrupts that signal the completion of sending and receiving packets.

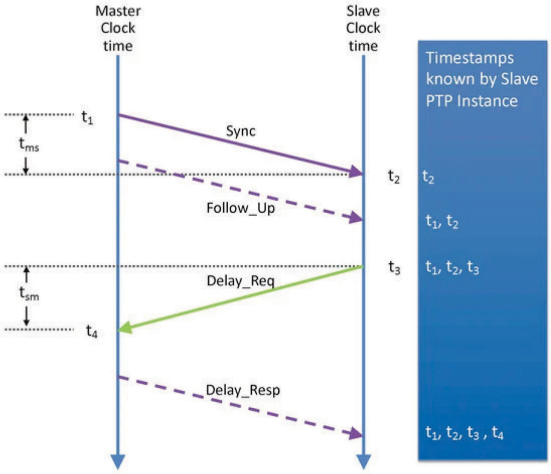

Fig. 2. *IEEE 1588 Precision Time Protocol (PTP)*

When the master node finishes sending a synchronization packet, its radio transceiver generates a TXDone interrupt that triggers an interrupt routine on the microcontroller that remembers the current value of the master node's clock in microseconds. This memorized value will be sent with the next synchronization packet to other nodes in the network. Upon completion of reception of the synchronization packet, the radio transceiver in the slave nodes generates an RXDone interrupt that triggers an interrupt routine on the microcontroller in which the current local clock value is remembered in microseconds. By receiving the synchronization packet, each node receives the sampled value of the master node, based on which it determines the deviation of its local clock, and performs a correction based on the calculated deviation value. Due to the relatively small distances between nodes, the propagation time of radio waves can be ignored or used as a constant if the distance between the slave and the master node is known. For example, if the distance between the master and slave nodes is 3 km, the propagation time will be 10 μs.

Arduino UNO and Nano development boards use different types of local oscillators depending on the board revision. The Arduino Nano 3 development board uses an external crystal resonator with an operating frequency of 16 MHz with a nominal accuracy of ± 5000ppm. The Arduino Uno development system uses an external crystal oscillator with an operating frequency of 16 MHz with a nominal accuracy of ± 50ppm.

In this paper we measured time deviation of two master-slave pairs of Arduino Nano and Arudino Uno boards. The first master-slave pair is based on Arduino Nano board which is clocked by 16MHz crystal resonator. A measurement shows -979 ppm deviation from the nominal frequency for the master node, and -1163 ppm deviation for the slave node. Time deviation between Arduino Nano master and slave node clocks is +184 ppm. Same type of measurements is performed for the second master-slave pair, based on Arduino Uno boards which are clocked by more precise 16 MHz crystal oscillator. Conducted measurements shows frequency deviation of +38.9 ppm for the master node and -24.7 ppm for the slave node, which is much more accurate when compared to crystal resonator clocked Arudino Nano. Time deviation between Arduino Uno master and slave node clocks is +63.6 ppm.

The master node is able to send synchronization messages using LoRa wireless radio to slave nodes but due low data rate, accuracy of time synchronization is limited. In this paper we determined maximum accuracy which can be achieved using LoRa modulation with different spreading factors. Modulation is set to use 125 kHz bandwidth with 4/8 coding rate, and results for different spreading factors are shown in Table 1.

Table. 1. *Clock deviation for different spreading factors of LoRa modulation*

Spreading factor	Airtime (ms)	Arduino Nano clock deviation	Arduino UNO clock deviation
SF7	45.31	8.36 ppm	2.88 ppm
SF8	74.24	10.98 ppm	4.72 ppm
SF9	148.48	27.32 ppm	9.44 ppm
SF10	296.96	54.64 ppm	18.88 ppm
SF11	462.85	85.16 ppm	29.43 ppm
SF12	925.69	170.33 ppm	58.87 ppm

CONCLUSION

In this paper we presented time synchronization of Arduino based wireless sensor nodes which using LoRa modulation. Frequency of time synchronization packet exchange between master and slave node will depend of desired accuracy of target application. Results presented in this paper showed maximum achieved time synchronization accuracy for different spreading factors.

ACKNOWLEDGMENT

The research in this paper is part of the project 451-03-47/2023-01/200132 funded by the Ministry of Science, Technological Development and Innovations of the Republic of Serbia.

REFERENCE

[1] S. Lasassmeh, J. Conrad, *Time Synchronization in Wireless Sensor Networks: A Survey*, Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon)

18-21 March 2010, Concord, NC, USA, ISSN: 1091-0050

[2] KY Koo, D. Hester and S. Kim, *Time Synchronization for Wireless Sensors Using Low-Cost GPS Module and Arduino*, Front. Built Environ. Vol. 4., 2019, DOI: 10.3389/fbuil.2018.00082

[3] P. Marti, M. Velasco, C. Lozoya, J. Fuertes, *Clock Synchronization for Networked Control Systems Using Low-Cost Microcontrollers*, Research Report: ESAII-RR-08-02, Automatic Control Department, Technical University of Catalonia, 2008

[4] M. Perez, E. Carrera, *Time Synchronization in Arduino-based Wireless Sensor Networks*, IEEE Latin America Transactions, Volume: 13, Issue: 2, 2015, pp. 455 - 461, ISSN: 1548-0992

[5] Q. Ye, Y. Zhang, L. Cheng, *A study on the optimal time synchronization accuracy in wireless sensor networks*, Computer Networks Volume 48, Issue 4, 15 July 2005, Pages 549-566, ISSN: 1389-1286

[6] S. Severi, D. Dardari, *Performance Limits of Time Synchronization in Wireless Sensor Networks*, 2008 IEEE International Conference on Communications, 19-23 May 2008, Beijing, China, ISSN: 1550-3607.