SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 19, No. 3, October 2022, 303-327

UDC: 004.738.5:004.71 DOI: https://doi.org/10.2298/SJEE2203303N

Modeling and Practical Implementation of the Optimal Wireless Security Gateway for the Industrial Automation Network

Igor Nevliudov[1], Sergiy Novoselov[1], Oksana Sychova[1]

Abstract: Reducing energy consumption in networks of IoT devices is very important. This paper presents the results of experimental studies of the proposed method of reducing power consumption for the protective gateway of the industrial network. Research was conducted for devices built on Lora modules. The LoRa Modem Calculator software from Semtech was used as a modeling tool. The recommended values of the program registers are given to achieve optimal energy consumption parameters for data reception and transmission modes. Information about the prototype is given and its practical implementation is shown. The developed prototype of the gateway implements the "Protocol Transformation" method, which allows removing possible dangerous inserts from the data packets received by the gateway. This can improve the security of packets from the gateway to IoT devices and back, and thus the network as a whole.

Keywords: IoT, LoRaWAN, Internet of things, Gateway, Industry.

1 Introduction

The modern trend in the development of automation systems in production requires the close integration of sensors, executive devices, PLCs and dispatching tools into the general network of the industrial Internet of Things. Using this concept, we get the opportunity to quickly respond to changes in the operation of the equipment. Application of big data processing technology makes it possible to get even more advantages from the use of this concept. For example, it becomes possible to predict equipment failures, or predict product sales volumes depending on external factors [1].

In this concept, in addition to positive features, there are also negative ones. Opening access to automation tools from the outside can lead to undesirable effects on the operation of the technological process by attackers. This can cause

[1]Department of Computer-Integrated Technologies, Automation and Mechatronics Kharkiv National University of Radio Electronics, Nauky Ave. 14, Kharkiv, 61166, Ukraine;
E-mails: igor.nevliudov@nure.ua; sergiy.novoselov@nure.ua; oksana.sychova@nure.ua

interruptions in the work of the enterprise and even man-made disasters. Such cases are already known and it is necessary to protect against them.

Security gateways provide an opportunity not only to limit access to the internal network, but also to perform the tasks of collecting and analyzing information passing through them. Such a function significantly expands the scope of application of gateways and becomes extremely necessary when integrating the existing topology with big data processing tools.

Thus, an important task is the introduction of protective components such as, for example, the IoT Gateway into the IIoT network [2]. The purpose of conducting these studies is to develop and verify the performance of a method for finding such a mode of operation of the IIoT network protection device, in which the main task of analyzing external protocols and organizing information exchange between technical means of the production line will be performed.

The following tasks were solved in the presented work:

– the organization principles study of data packet transmission in order to regulate the total transmission time and power consumption,

– computer simulation was performed to determine the optimal parameters of the gateway configuration,

– developed test models of LoRaWAN gateway and IoT device,

– conducted experimental studies to verify the developed method for determining the optimal mode of operation of the gateway.

2 The Principle of Organizing Messaging in the IoT Network using LoRa Modules

IoT (Internet of Things) network is an ecosystem of various, unconnected devices that do not interfere with each other and exist to collect and transmit a wide range of data [3 – 8]. The main condition for their successful operation is a single data transfer protocol. A set of popular protocols is used to organize the exchange of messages in the IoT network. MQTT, AMQP, COAP and DDS are used as application layer messaging protocols. Channel layer protocols include: BLE, LoRaWAN, SigFox and LTE.

LoRaWAN is one of the most popular standards for data transmission in low power Wide Area Network (LPWAN) devices [3]. The LoRaWAN standard is open. One of its main advantages is the highly competitive market of equipment suppliers and compatibility of devices from different manufacturers (you can connect base stations of several brands to the server, and to include devices of several vendors in the network).

In the LoRaWAN network, end devices (radio modules) send data to the hub (gateway, base station). From the base station, high-speed data packets are transmitted to the server, which gives a specific type of data to the corresponding application server [4, 9 – 11].

In LoRa there are three classes of devices for energy consumption:

A-class is the most economical. Devices of this class are battery-powered for several years, which is achieved through the activity of the device only when transmitting data on a programmed schedule.

The C-Class, vice versa, is constantly in a state of receiving. That is why class C devices do not provide battery power.

B-Class, identical A-Class, is in power-saving mode most of the time, but has some server-side interoperability features that are typical of C-Class.

LoRaWAN uses an unlicensed part of the radio frequency spectrum in the range of 868.0-868.6 MHz (in Ukraine). The standard provides for the presence of base stations and subscriber devices, which, in the case of autonomous power supply, most of the time are in energy saving mode. Devices "wake up" only to communicate with the server.

One of the main tasks in designing an industrial network gateway is to optimize the time of packet transmission from sensors to network sensor and get feedback [12].

Airtime directly affects energy consumption, especially when it comes to mobile devices for converting and transmitting information. To check the performance of the proposed method, we will determine the main characteristics of the data transmission protocol in the LoRaWAN network. We will also consider and select those protocol parameters that have the greatest impact on energy consumption.

Information exchange takes place between devices of the internal segment of the network, which are called end nodes (End Node) and the LoRa protection device (Gateway) [1].

The following steps are performed on the transmitter side:

– receiving a block of data from the upper hardware level (PHYPayload),

– forming a physical packet header (PHDR + PHDR_CRC),

– encoding the physical packet header (PHDR + PHDR_CRC) with a fixed speed of 4/8,

– calculating the checksum of the useful data block PHYPayload (CRC),

– useful data block encoding (PHYPayload + CRC) at a preset speed CR,

– transmission of the preamble by radio channel,

– modulation and transmission by radio of the physical data block.

The following is performed on the side of the receiving device:

- identifying the preamble and determining the beginning of the physical data block,
- signal demodulation,
- decoding the physical packet header (PHDR + PHDR_CRC) and checking its checksum,
- decoding the payload (PHYPayload + CRC) and checking its checksum,
- confirmation of received data (for relevant message types),
- data transfer to the upper level of the module for further use by end devices.

The general view of the LoRa package is shown in Fig. 1 and consists of three elements: preamble, optional header and useful data [1].

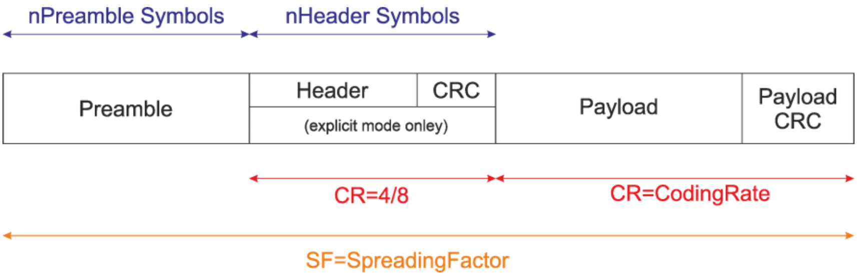


Fig. 1 – *LoRa packet data structure.*

At the beginning of the communication session, the receiver and transmitter must be synchronized. The preamble field is used for this purpose. The size of this field can vary from 6 to 65535 characters. With the help of special registers, you can change the size of this field. Which also affects the time the device is on the air. A field size of 6 characters is very often used, which is sufficient for operation at short distances and with a good transmitter signal level [1].

On the side of the transmitter and the receiver, one condition must be fulfilled - the size of the preamble field must be the same. It is also necessary to include in the algorithm for setting the size of this field the possibility of dynamically increasing or decreasing the size depending on the search mode of the corresponding device.

There are different operating modes of LoRa modules. There are two such modes and the following headings are used for them:

- explicit mode,
- implicit mode.

The header type is selected by the ImplicitHeaderModeOn bit, which is located in the RegModemConfig1 register. Fig. 2 shows an example of a data packet in explicit mode.

In Fig. 2 Preamble is a preamble used to synchronize the receiver with the input stream and determine the beginning of the physical data block. The preamble length for the SX1278 is a programmable value [13].

PHDR is the physical packet header. Present only when using explicit mode and contains the following:

– the length of the data field (Payload) in bytes,

– frequency of error codes correction,

– availability of additional 16-bit CRC for the data field.

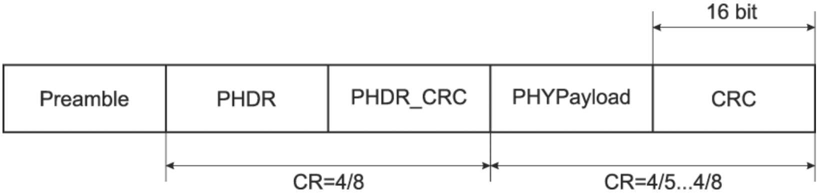


Fig. 2 – *Explicit mode message format.*

In IoT devices, work scenarios are often used when the packet parameters are known in advance and their values are fixed. For example, this applies to encoding speed and the presence of a checksum calculation field. In this case, you can remove the header field. Thus, we will reduce the data packet transmission time. But at the same time, we need to ensure the implementation of the specified actions by other means on the side of the receiver and the transmitter in order to preserve the integrity of the data.

In this case, the PHDR header is encoded with redundant code with a fixed speed of 4/8, and the payload - with a programmable speed. Fig. 3 shows an example of a data packet in implicit mode.

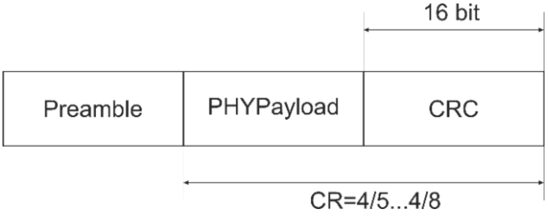


Fig. 3 – *Implicit mode message format.*

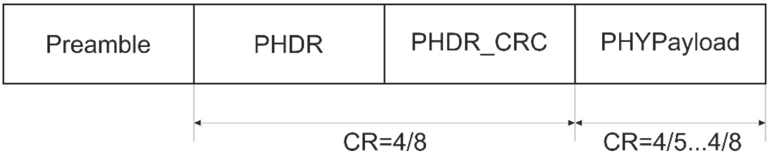


Fig. 4 – *Message format with missing CRC field.*

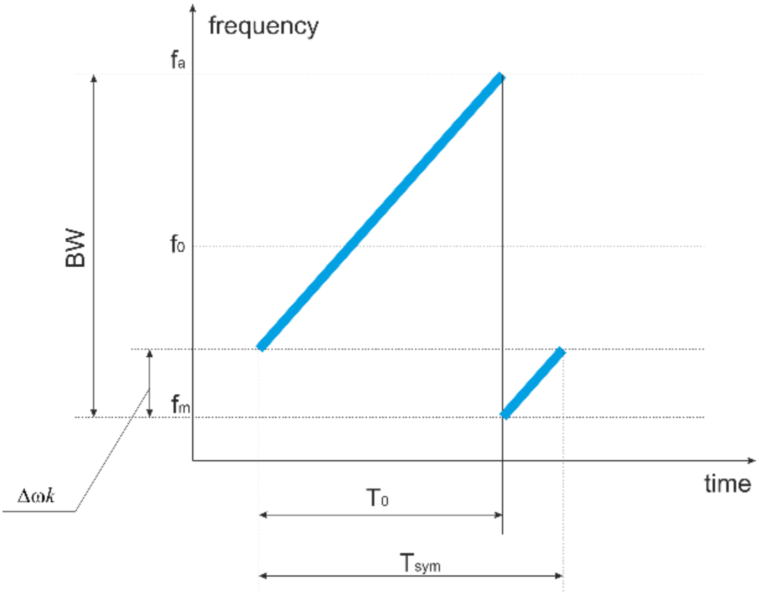


Fig. 5 – *The frequency of the radio signal depends on time.*

When using implicit mode, the physical packet header is not transmitted and the devices work with pre-set parameters.

Fig. 4 shows the case where the checksum field (CRC) is missing.

A feature of the technology used is that the devices work asynchronously. With the help of the preamble field, the task of setting or synchronizing the operating frequency of the transmitter and receiver is solved. This field includes a specified set of symbols and their sequence so that the receiver could determine the fact that the transmitter is on the air and perform the tuning procedure more precisely. Based on the received data, the spectrum broadening factor (SF) is determined. The duration of the preamble can be changed, but it should not be less than

$$\tau_p = T_1 + 2T_2 , \tag{1}$$

where T_1 determines the maximum time the receiver is in the Sleep state, T_2 determines the search time of the preamble receiver.

The principle of transmitting the symbols of the information of the physical layer data block using the broadband radio signal LoRa is the frequency offset $\mathrm{e}^{\mathrm{j}\Delta\omega kt}$ relative to the reference signal $\mathrm{e}^{\mathrm{j}\left(\omega_n t+\mu t^2\right)}$ [1].

Thus, the function $x(t)$ is written as follows:

$$x(t)=\begin{cases} A_0\cos\left(\omega_n t+\Delta\omega kt+\mu t^2/2\right), & 0\le t<T_0 \\ A_0\cos\left(\omega_n t+\Delta\omega kt-BWt+\mu t^2/2\right), & T_0\le t<T_{sym} \end{cases}, \qquad (2)$$

where BW is the spectrum width of the radio signal, $k=0,1,2,...,2SF$ is information symbol with dimension SF bits, $T_{sym}=2SF/BW$ is radio signal duration, $\mu=BW/T_{sym}$ is the rate of change of the frequency of the radio signal, t is unit of data transfer time, ω_n is frequency of radio signal.

An example of the radio signal frequency dependence on time for the data frame is shows in Fig. 5.

3 Computer Simulation and Selection of Optimal Gateway Parameters

The gateway simulation was performed using LoRa Modem Calculator software from Semtech. The input conditions for modeling are the following parameters:

– data field length,

– preamble field length,

– the width of the radio signal spectrum BW,

– propagation coefficient (spectrum expansion factor),

– frequency range,

– transmitter power,

– data packet format.

As a result of modeling, it is necessary to determine the following operating conditions:

– time to transmit the data packet,

– data transmission rate,

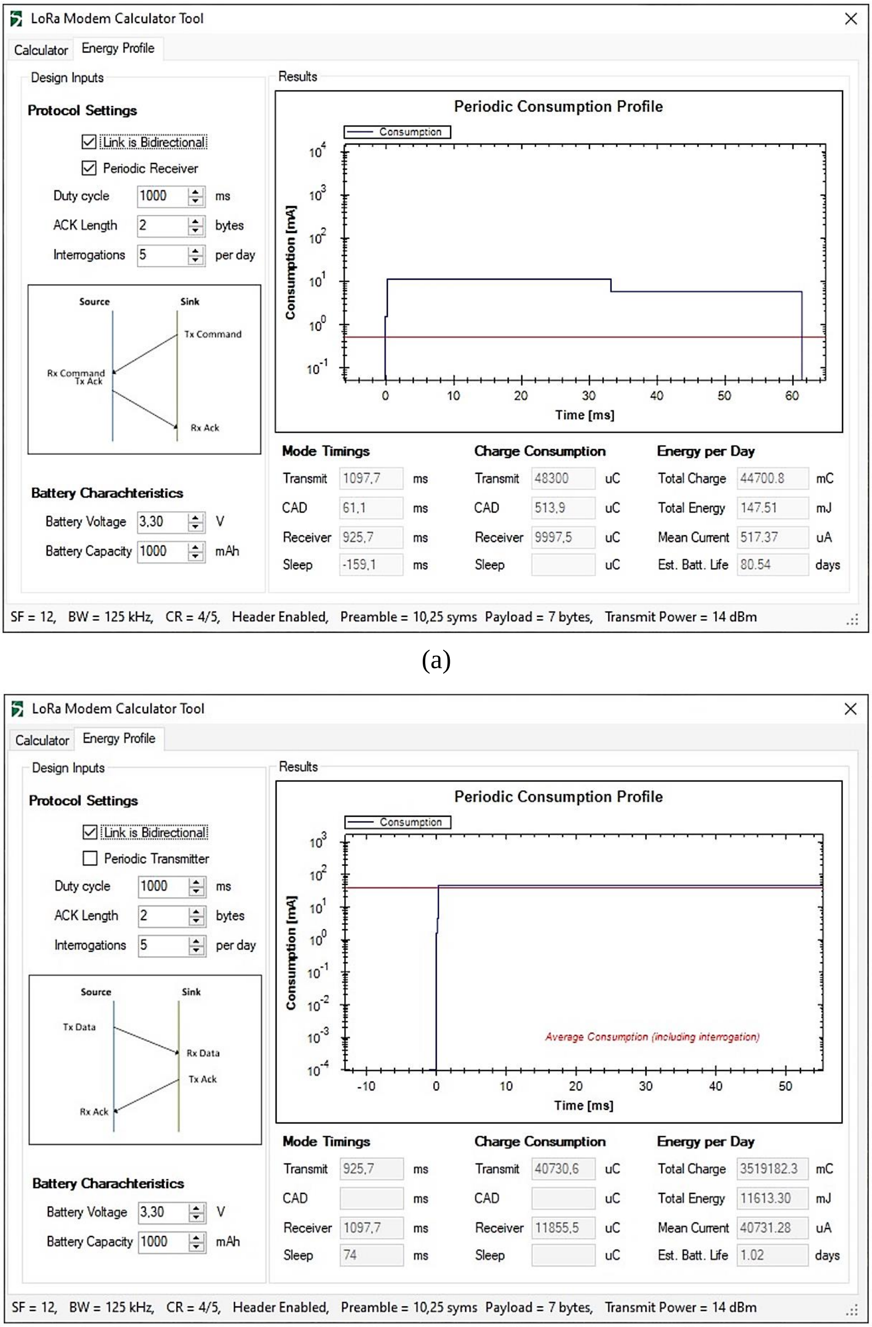
– receiver sensitivity,

– power required for data transmission,

– service life from an autonomous power source.

Initial conditions that are unchanged: the transmission interval is one second, battery capacity 1000 mAh, supply voltage 3.3 V, operating frequency 433 MHz, Payload size can vary between 7-3 bytes. Thus, the device operation is simulated for these parameters. The simulation results are presented in **Table 1**.

Fig. 6 shows the simulation result in the LoRa Modem Calculator program for the Periodic Receiver mode (a) and Periodic Transmitter mode (b).



Design Inputs

Protocol Settings

☑ Link is Bidirectional

☑ Periodic Receiver

Duty cycle 1000 ms

ACK Length 2 bytes

Interrogations 5 per day

Battery Charachteristics

Battery Voltage 3,30 V

Battery Capacity 1000 mAh

Results

Mode Timings			Charge Consumption			Energy per Day		
Transmit	1097,7	ms	Transmit	48300	uC	Total Charge	44700.8	mC
CAD	61,1	ms	CAD	513,9	uC	Total Energy	147.51	mJ
Receiver	925,7	ms	Receiver	9997,5	uC	Mean Current	517.37	uA
Sleep	-159,1	ms	Sleep		uC	Est. Batt. Life	80.54	days

SF = 12, BW = 125 kHz, CR = 4/5, Header Enabled, Preamble = 10,25 syms Payload = 7 bytes, Transmit Power = 14 dBm

(a)

Design Inputs

Protocol Settings

☑ Link is Bidirectional

☐ Periodic Transmitter

Duty cycle 1000 ms

ACK Length 2 bytes

Interrogations 5 per day

Battery Charachteristics

Battery Voltage 3,30 V

Battery Capacity 1000 mAh

Results

Mode Timings			Charge Consumption			Energy per Day		
Transmit	925,7	ms	Transmit	40730,6	uC	Total Charge	3519182.3	mC
CAD		ms	CAD		uC	Total Energy	11613.30	mJ
Receiver	1097,7	ms	Receiver	11855,5	uC	Mean Current	40731.28	uA
Sleep	74	ms	Sleep		uC	Est. Batt. Life	1.02	days

SF = 12, BW = 125 kHz, CR = 4/5, Header Enabled, Preamble = 10,25 syms Payload = 7 bytes, Transmit Power = 14 dBm

(b)

Fig. 6 – *The result of simulation in LoRa Modem Calculator for* (a) *Periodic Receiver mode*; (b) *Periodic Transmitter mode.*

Table 1

The results of modeling the operation of the data receiving/transmission module at different sizes of the Payload field.

Payload, bytes	7	6	5	4	3
SF = 12, BW = 125 kHz, CR = 4/5, Header Enabled, Preamble = 10,25 sym., Payload = 7 bytes, TR Power = 14 dB					
Time on the air, ms	925,7	761,86	761,86	761,86	761,86
Symbols transmission time, ms	32,77	32,77	32,77	32,77	32,77
Current consumption during transmission, mA	44	44	44	44	44
Receiver sensitivity, dB	-138	-138	-138	-138	-138
Approximate battery life, days	80,54 1,02	80,54 1,24	80,54 1,24	80,54 1,24	80,54 1,24
CAD, ms	61,1	61,1	61,1	61,1	61,1

After analyzing the documentation and researching the data transmission protocol in the LoRaWAN network, it is proposed to use the Lora feature to reduce power consumption while reducing the time the transmitter is on the air. This can be achieved by reducing the size of transmitted packets [1].

The first experiment was aimed at reducing the size of the data field to 4 bytes. The experimental data in **Table 1** show that the change in frame length led to a decrease in the time on the air from 925.7 to 761.86 ms, but a significant decrease in the current consumption was determined.

The next experiment was aimed at changing the number of header fields in transmitted packets. The dependence of the consumed current for these parameters is shown in **Table 2**. Studies were also conducted for different sizes of data fields.

Table 2

The results of modeling the data receiving/transmission module operation at different frame sizes.

Payload, bytes	7	7	7	4	4	4
Header Enabled	Yes	No	No	Yes	No	No
CRC Enabled	Yes	Yes	No	Yes	Yes	No
SF = 12, BW = 125 kHz, CR = 4/5, Header Enabled, Preamble = 10,25 sym., Payload = 7 bytes, TR Power = 14 dB						
Time on the air, ms	925, 7	761,86	761,86	761,86	761,86	598,02
Symbols transmission time, ms	32,77	32,77	32,77	32,77	32,77	32,77
Current consumption during transmission, mA	44	44	44	44	44	44
Receiver sensitivity, dB	-138	-138	-138	-138	-138	-138
Approximate battery life, days	80,54 1,02	80,6 1,24	80,6 1,24	80,54 1,24	80,6 1,24	80,6 1,58
CAD, ms	61,1	61,1	61,1	61,1	61,1	61,1

Analyzing the results, we can see that when you turn off one of the fields Header or CRC managed to slightly increase the battery life from 80.54 to 80.6 days. For the transmitter when disabling these fields, the frame managed to increase the battery life from 1.02 to 1.24 days for 7 bytes of data, and from 1.24 to 1.58 for 4 bytes of data. Thus, the increase in battery life of the transmitter is more than 20%. By disabling two fields at once, we also reduced the total time spent on air to 598.02 ms.

To determine whether a signal is present or not, instead of using the received signal strength indicator (RSSI) in the LoRa system, a combined adaptive channel activity detection (CAD) system is used to identify the presence of a signal.

It can distinguish noise and useful LoRa signal. The operation of this system requires two symbols. If the system detects a signal, the interrupt by CAD_Detected will confirm, and in this case, to obtain useful data, the device will remain in receive mode.

Fig. 7 shows the simulation result of Channel Activity Detection. From the figure you can see that for the transmission mode, this parameter is infinitely important for given conditions.

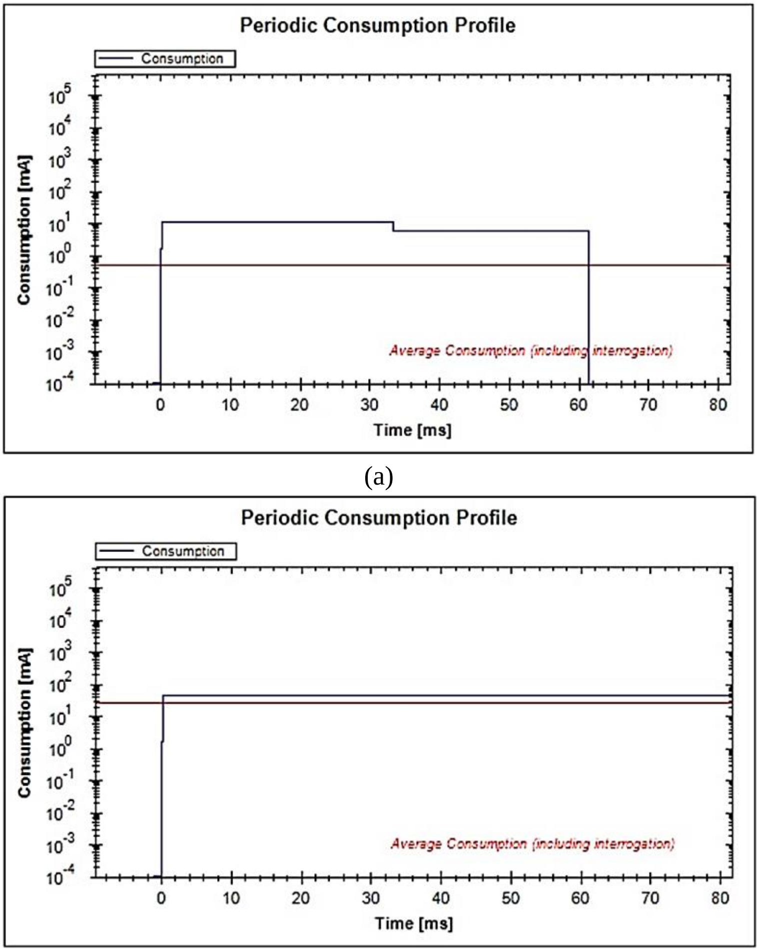
The next parameter that can be corrected is “Coding Type (CR)”. The following values of this parameter are available to the programmer when configuring the device operation: CR = 4/5, 4/6, 4/7, 4/8. The simulation results are presented in **Table 3**.

Table 3

The results of modeling the operation of the module receiving/transmitting data for different values of the CR parameter.

Payload, bytes	7	7	7	7	4	4	4	4
CR	4/5	4/6	4/7	4/8	4/5	4/6	4/7	4/8
SF = 12, BW = 125 kHz, CR = 4/5, Header Enabled, Preamble = 10,25 sym, Payload = 7 bytes, TR Power = 14 dB								
Time on the air, ms	925, 7	991,2	1057	1122	761,9	794,7	827,4	860,16
Symbols transmission time, ms	32,7	32,7	32,7	32,7	32,7	32,77	32,77	32,77
Current consumption during transmission, mA	44	44	44	44	44	44	44	44
Receiver sensitivity, dB	-138	-138	-138	-138	-138	-138	-138	-138
Approximate battery life, days	80,54 1,02	80,52 0,96	80,6 1,24	90.2 0,95	80,54 1,24	80,52 1,19	80,51 1,14	80,4 1,10
CAD, ms	61,1	61,1	61,1	61,1	61,1	61,1	61,1	61,1

According to the results of the experiment, we can see that changing the CR parameter leads to an increase in the required time for information transmit and, accordingly, to a decrease in the battery life of the module.



(a)

(b)

Fig. 7 – (a) *The result of simulation of Channel Activity Detection for the receiver*; (b) *The result of modeling the Channel Activity Detection for the transmitter.*

Another conclusion is that for the values of the parameters CR = 4/7 and CR = 4/8, we go beyond the specified value of the transmission interval of 1000 ms. At the specified values of the CR parameter, the time spent in the air should be 1057 and 1122 ms. This contradicts the initial conditions, so you cannot use these parameters. Thus, for practical use, there are two values of the parameter CR = 4/5 and CR = 4/6.

The considered parameters are peculiar to the organization of the maximum possible signal transmission range. The patented technology of communication organization uses its own method of modulation, which is called “chirp”. This is a nonlinear modulation in which the signal frequency increases linearly from the initial frequency f0 to the final f1. According to the LoRa standard, coding is performed by cyclically shifting the chip relative to the time frame.

The signal parameters in LoRa are Spreading factor (SF) and Bandwidth (BW). The SF parameter is set by default values of SF7–SF12, where 7 is the fastest and 12 is the slowest mode. In Fig. 8 shows an example of different data rates at different SF values [14].

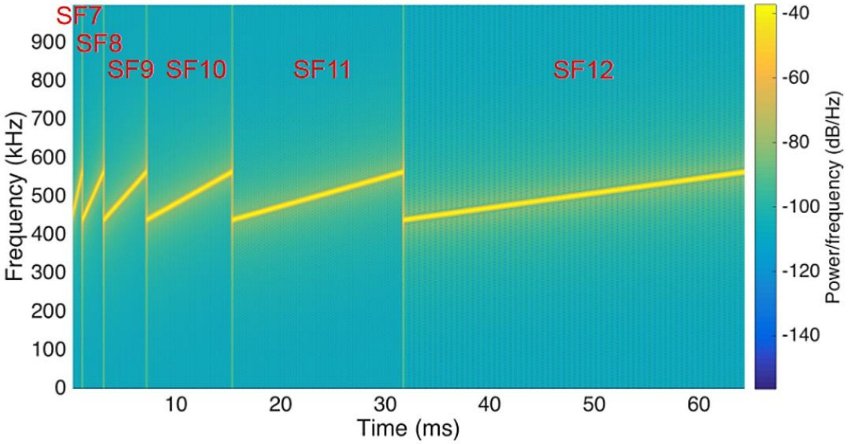


Fig. 8 – *An example of different data rates at SF values.*

The slowest signal corresponds to the longest transmission range. If range is not a criterion to be achieved when designing the device, then the documentation for the module SX1278 can significantly reduce power consumption by changing the parameters SF and BW.

We will simulate such situations using the LoRa Modem Calculator tool. The BW parameter will change as follows: 125 kHz, 250 kHz and 500 kHz. The simulation results are shown in **Table 4**.

As you can see, bandwidth expansion significantly reduces data transmit time and increases overall battery life. At the maximum value of the parameter BW = 500 kHz, the battery life has quadrupled for the data transmit mode, and by 70% in the receive mode.

The following simulation will be performed for different values of the Spreading factor. Will change all possible values of this parameter (SF = {6, 7, 8, 9, 10, 11, 12}). The results of the simulation are presented in **Table 5**.

Having determined the values of the parameters by which the maximum energy efficiency is achieved, we perform the simulation of such a situation using LoRa Modem Calculator. Simulations were performed for two values of the Payload field size (7 and 4 bytes).

Table 4

The results of modeling the operation of the data receiving/transmission module at different values of the parameter BW.

BW, kHz	125	250	500

SF = 12, CR = 4/5, Header Enabled, Preamble = 10,25 sym,
Payload = 7 bytes, TR Power = 14 dB			

Time on the air, ms	925, 7	462,85	231,42
Symbols transmission time, ms	32,7	16,38	8,19
Current consumption during transmission, mA	44	44	44
Receiver sensitivity, dB	-138	-135	-132
Approximate battery life, days	80,54 1,02	110,76 2,05	124,96 4,09
CAD, ms	61,1	44,6	36,3

Table 5

The results of simulation of the SX1278 for different values of the SF parameter.

SF	12	11	10	9	8	7	6
BW = 125 kHz, CR = 4/5, Header Enabled, Preamble = 10,25sym, Payload = 7 bytes, TR Power = 14 dB							
Time on the air, ms	925, 7	462,8	231,42	115,7	68,1	34,05	17,02
Symbols transmission time, ms	32,7	16,38	8,19	4,10	2,05	1,02	0,51
Current consumption during transmission, mA	44	44	44	44	44	44	44
Receiver sensitivity, dB	-138	-135,5	-133	-130	-127	-124	-119
Approximate battery life, days	80,54 1,02	164,12 2,05	332,21 4,09	663,20 8,18	1288 13,90	2376 27,80	4041 55,57
CAD, ms	61,1	29,5	14,3	7	3,5	1,8	1

For the first case of 7-byte packet transmission, the following parameters were used:

– Spreading factor SF = 6,

– data field size Payload = 7,

– header field is missing,

– the checksum field is missing,

– coding type (CR) = 4/5.

The simulation results will be shown for three bandwidth values of 125, 250 and 500 kHz. The simulation results are presented in **Table 6**.

Table 6

Simulation results for optimal values of LoRa module parameters at Payload = 7 bytes.

BW, kHz	125	250	500

SF = 6, CR = 4/5, Header Disable, CRC Disable, Preamble = 10,25 sym,
Payload = 7 bytes, TR Power = 14 dB			

Time on the air, ms	14,46	7,23	3,62
Symbols transmission time, ms	0,51	0,26	0,13
Current consumption during transmission, mA	44	44	44
Receiver sensitivity, dB	-119	-116	-113
Approximate battery life, days	4041,37 65,39	6269 130,64	8321,94 260,66
CAD, ms	1	0,6	0,4

The following parameters are used to simulate the transmission of a 4-byte packet:

– Spreading factor SF = 6,

– data field size Payload = 4,

– header field is missing,

– the checksum field is missing,

– coding type (CR) = 4/5.

The simulation results for the three bandwidth values 125, 250 and 500 kHz are presented in **Table 7**.

Table 7

Simulation results for optimal values of LoRa module parameters at Payload = 4 bytes.

BW, kHz	125	250	500

SF = 6, CR = 4/5, Header Disable, CRC Disable, Preamble = 10,25 sym,
Payload = 4 bytes, TR Power = 14 dB			

Time on the air, ms	11, 9	5,95	2,98
Symbols transmission time, ms	0,51	0,26	0,13
Current consumption during transmission, mA	44	44	44
Receiver sensitivity, dB	-119	-116	-113
Approximate battery life, days	4041,37 79,44	6269 158,65	8321,94 316,40
CAD, ms	1	0,6	0,4

For the 1000 ms data cycle limit we set, the transfer time must be 10 ms.

4 Experimental Study

To test the proposed method of determining the bandwidth of the gateway and test its operation, a test layout was assembled. In Fig. 9 shows a block diagram of the layout.

The layout consists of the following modules:

– microcomputer with operating system installed,

– LoRa module,

– test module for emulating IoT devices.

The operating system on the microcomputer starts the process of polling the LoRa module in order to obtain data from IoT devices that are within the range of the gateway and registered on it [15-18].

The https://www.thethingsnetwork.org/ resource is used as a cloud service.

The SX-1268-based Raspberry Pi LoRa HAT is used as communication module on the side of gateway. This module operates at a frequency of 433 MHz and allows data transmission up to 5 km on a serial port. Compared to usual LoRa modules, the SX-1268 LoRa HAT provides higher data rates, lower power consumption, better security and interference protection [13]. Thus, this module is suitable for various applications, such as industrial equipment management, smart home, data collection, etc. A standard 40PIN GPIO connector is used to connect the LoRa HAT module to the Raspberry Pi. It supports all Raspberry Pi series boards. RSSI signal power definition technology is supported in the LoRa HAT module. This can be used to assess signal quality, configure the network, and conduct experimental studies of the gateway.

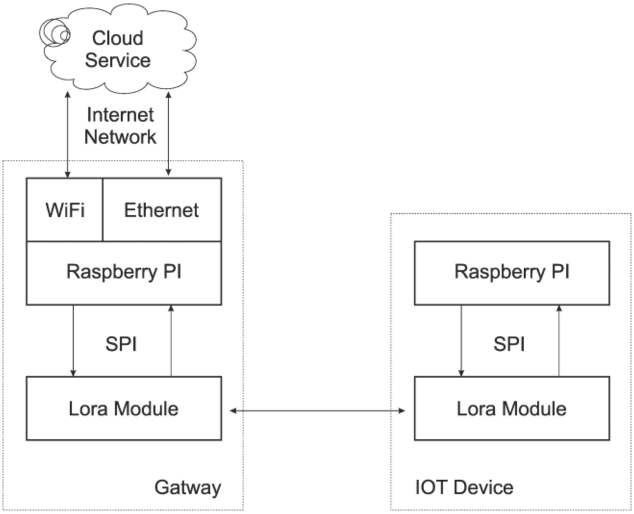


Fig. 9 – *Block diagram of the layout.*

The module has the ability to configure a communication private key, which significantly increases the security of user data through the gateway. It is also possible to configure wireless settings by sending a packet of wireless commands or data remotely over the Internet. In Fig. 10 shows the assembled layout of the LoRa WAN gateway. The LoRa WAN gateway is a mini-PC Raspberry PI (1) and Lora SX-1268 module (2) with an antenna 433 MHz (5) connected to it. The power supply and monitor connections are connected to the connectors (3) and (4).

Another device with a LoRa module is necessary to test the gateway. This module will simulate the operation of the IoT device and exchange data with the gateway. The test module was developed using a microtransmitter Ra-01 LoRa - SX1278 - 433 MHz. Ra-01 is a functionally complete module built on the SX1278 chip. The SPI interface is used to connect to the Raspberry PI microcomputer. The basis of the Ra-01 module is the SX1278 chip, which allows you to programmatically set the operating frequency in a certain range, change the number of the data channel, control the transmitter power, and set the ability to encode the transmitted information. In Fig. 11 shows a test device that is built on the SX1278 module and will interact with the gateway.

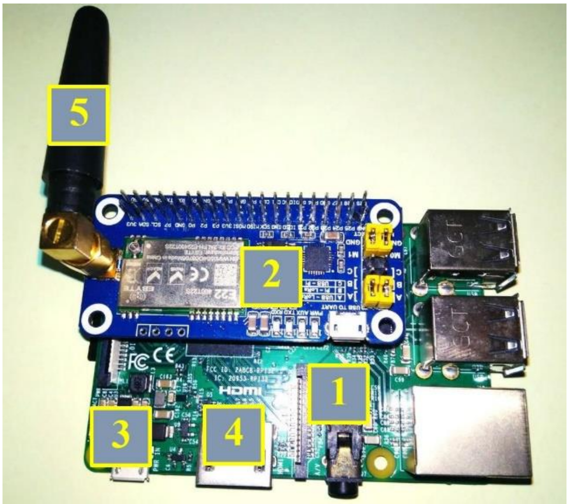


Fig. 10 – *LoRa WAN gateway layout.*

The test device is a mini-PC Raspberry PI (4) and Lora module (3) with an antenna 433 MHz (7) connected to it. Keyboard (5), mouse (6), HDMI monitor (1) are also connected to mini-PC.

To verify the simulation results, the programming and test connection of the module that simulates the operation of the IoT device to the gateway was performed.

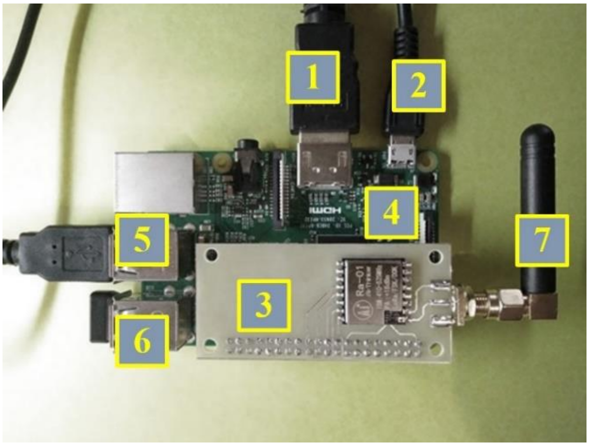


Fig. 11 – *Appearance of the test IoT device.*

The LoRa module configuration options on the gateway and IoT device are as follows:

– operating frequency 433 MHz,

– Payload = 7,

– Spreading factor = 6,

– BW ≥ 250 kHz,

– header field is missing,

– the checksum field is missing,

– coding type (CR) = 4/5.

At the first step, it is necessary to set the frequency of the device in the program. To do this, you must first find out the actual operating frequency of the device. This can be done using the following code snippet:

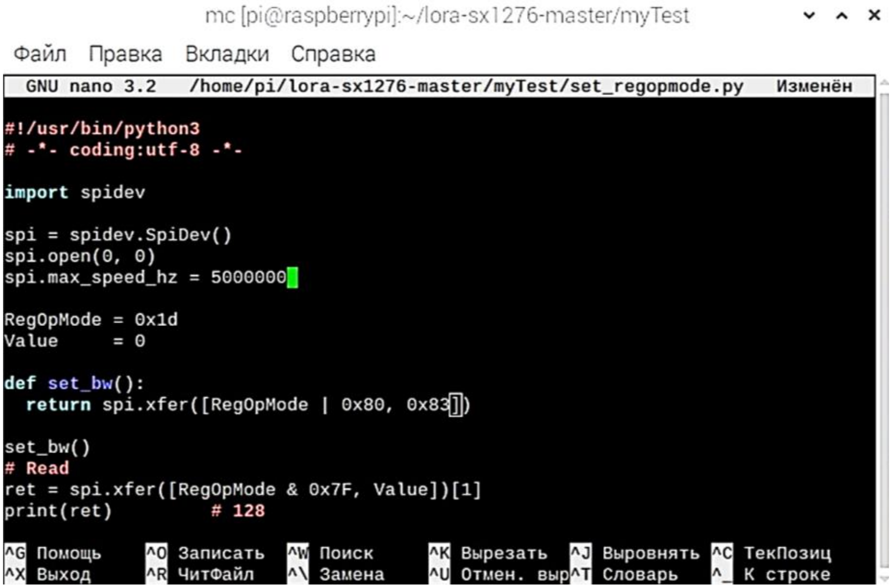
```
import spidev spi = spidev.SpiDev()
spi.open(0, 0)
spi.max_speed_hz = 5000000
RegFrMsb = 0x06
RegFrMid = 0x07
RegFrLsb = 0x08
Value = 0
msb = spi.xfer([RegFrMsb & 0x7F, Value])[1]
mid = spi.xfer([RegFrMid & 0x7F, Value])[1]
lsb = spi.xfer([RegFrLsb & 0x7F, Value])[1]
f = lsb + 256*(mid + 256*msb)
print(f / 16384.0) spi.close()
```

As stated in the documentation [4], the frequency is the sum of the values of the three registers MSB: 0x06, MID: 0x07, and LSB: 0x08. A code snippet was used to change the frequency:

```
RegFrMsb = 0x06
def set_freq(f)
i = int(f * 16384)
msb = i // 65536
i -= msb * 65536
mid = i // 256
i -= mid * 256
lsb = i
return spi.xfer([RegFrMsb | 0x80, msb, mid, lsb])
set_freq(433)
```

Using this code snippet, the operating frequency is set to 433 MHz.

The next step is to set the Spreading Factor to 6. Written a function for reading the specified register, which is responsible for configuring the module to check the actual value of any of the parameters (Fig. 12).


mc [pi@raspberrypi]:~/lora-sx1276-master/myTest
Файл Правка Вкладки Справка
GNU nano 3.2 /home/pi/lora-sx1276-master/myTest/set_regopmode.py Изменён

```
#!/usr/bin/python3
# -*- coding:utf-8 -*-

import spidev

spi = spidev.SpiDev()
spi.open(0, 0)
spi.max_speed_hz = 5000000

RegOpMode = 0x1d
Value     = 0

def set_bw():
  return spi.xfer([RegOpMode | 0x80, 0x83])

set_bw()
# Read
ret = spi.xfer([RegOpMode & 0x7F, Value])[1]
print(ret)         # 128
```

^G Помощь ^O Записать ^W Поиск ^K Вырезать ^J Выровнять ^C ТекПозиц
^X Выход ^R ЧитФайл ^\ Замена ^U Отмен. выр ^T Словарь ^_ К строке

Fig. 12 – *The function of reading the value of the register* 0x1d.

This function operates on the address of the required register and displays its value. Check the value of the 0x1d register, which is responsible for configuring parameters such as:

– radio signal spectrum width (BW),

– coding type (CR),

– the presence of a header (Header).

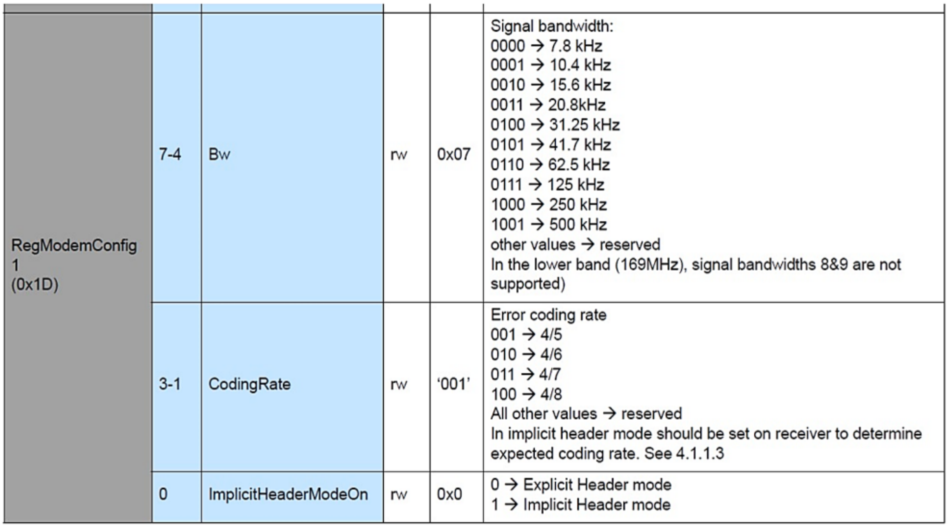
After executing the written program, the value of the register 0x1d will appear on the screen, which in this case is 114. To determine which functions are enabled using this byte, you must use the information in the user manual (Fig. 13).

Thus, the number 114 corresponds to the following configuration:

– 0111 – corresponds to the width of the radio signal spectrum and is 125 kHz,

– 0010 – corresponds to the coding type and is 4/5,

– 0 – corresponds to the explicit header.



RegModemConfig 1 (0x1D)	7-4	Bw	rw	0x07	Signal bandwidth: 0000 → 7.8 kHz 0001 → 10.4 kHz 0010 → 15.6 kHz 0011 → 20.8kHz 0100 → 31.25 kHz 0101 → 41.7 kHz 0110 → 62.5 kHz 0111 → 125 kHz 1000 → 250 kHz 1001 → 500 kHz other values → reserved In the lower band (169MHz), signal bandwidths 8&9 are not supported)
	3-1	CodingRate	rw	'001'	Error coding rate 001 → 4/5 010 → 4/6 011 → 4/7 100 → 4/8 All other values → reserved In implicit header mode should be set on receiver to determine expected coding rate. See 4.1.1.3
	0	ImplicitHeaderModeOn	rw	0x0	0 → Explicit Header mode 1 → Implicit Header mode

Fig. 13 – *The value of the register bits* 0x1d.

To display all the actual parameters of the module, you must use the developed program script. As a result of work of this program script the following data are received:

```
SX127x LoRa registers:
 mode – SLEEP
 freq – 434.000000 MHz
 coding_rate – CR4_5
 bw – BW125
 spreading_factor – 128 chips/symb
```

```
implicit_hdr_mode - OFF
 rx_payload_crc - OFF
 tx_cont_mode - OFF
 preamble - 8
 low_data_rate_opti - OFF
 agc_auto_on -  ON
 symb_timeout - 100
 freq_hop_period - 0
 pkt_snr_value - 64.000000
 pkt_rssi_value - 164
 rssi_value - 164
 fei - 0
 pa_select - RFO
 max_power - 13.200000 dBm
 output_power  - 13.200000 dBm
 ocp - ON
 ocp_trim - 100.000000 mA
 lna_gain - NOT_USED
 lna_boost_lf - 0b0
 lna_boost_hf  - 0b0
 detect_optimize - 0x3
 detection_thresh - 0xa
 sync_word - 0x12
 dio_mapping 0..5   [0, 0, 0, 0, 2, 0]
 tcxo - XTAL
 pa_dac - default
status  -  {'signal_sync':  0,  'signal_detected':  0,
'header_info_valid': 0, 'rx_ongoing': 0, 'modem_clear':
0, 'rx_coding_rate': 0}
 version - 0x12
```

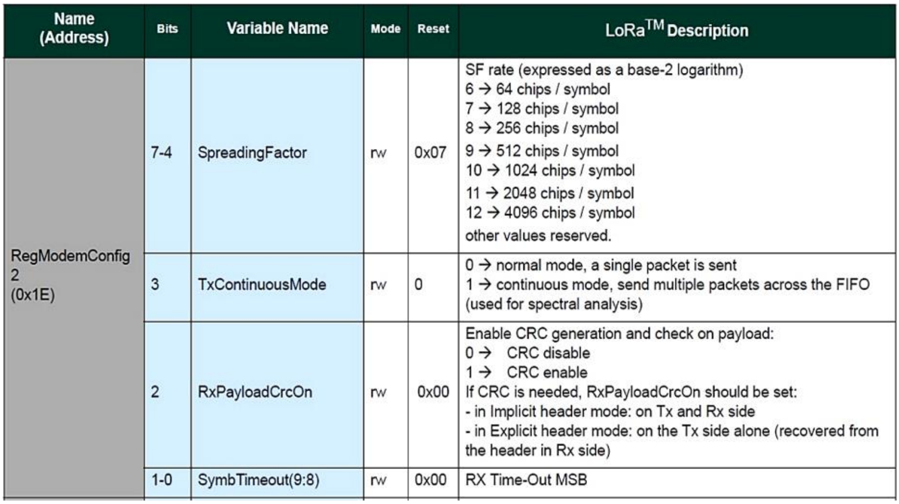
As you can see, the output parameters correspond to the mode of operation Spreading factor = 6 (128 chips/symbol). Thus, the actual values do not correspond to the optimal for a given mode of operation.

Preliminary simulation results have shown that the values of the two registers 0x1d and 0x1e need to be changed to enable the module in the optimal mode. The principle of the configuration of the register 0x1e is shown in Fig. 14.

We will configure the optimal configuration of the LoRa module:

– register value `0x1e:  131 -> 1000 001 1 -> 0x83`,

– register value `0x1d: 96 -> 0110 0000 -> 0x60`.



Name (Address)	Bits	Variable Name	Mode	Reset	LoRa^{TM} Description
RegModemConfig 2 (0x1E)	7-4	SpreadingFactor	rw	0x07	SF rate (expressed as a base-2 logarithm) 6 → 64 chips / symbol 7 → 128 chips / symbol 8 → 256 chips / symbol 9 → 512 chips / symbol 10 → 1024 chips / symbol 11 → 2048 chips / symbol 12 → 4096 chips / symbol other values reserved.
	3	TxContinuousMode	rw	0	0 → normal mode, a single packet is sent 1 → continuous mode, send multiple packets across the FIFO (used for spectral analysis)
	2	RxPayloadCrcOn	rw	0x00	Enable CRC generation and check on payload: 0 → CRC disable 1 → CRC enable If CRC is needed, RxPayloadCrcOn should be set: - in Implicit header mode: on Tx and Rx side - in Explicit header mode: on the Tx side alone (recovered from the header in Rx side)
	1-0	SymbTimeout(9:8)	rw	0x00	RX Time-Out MSB

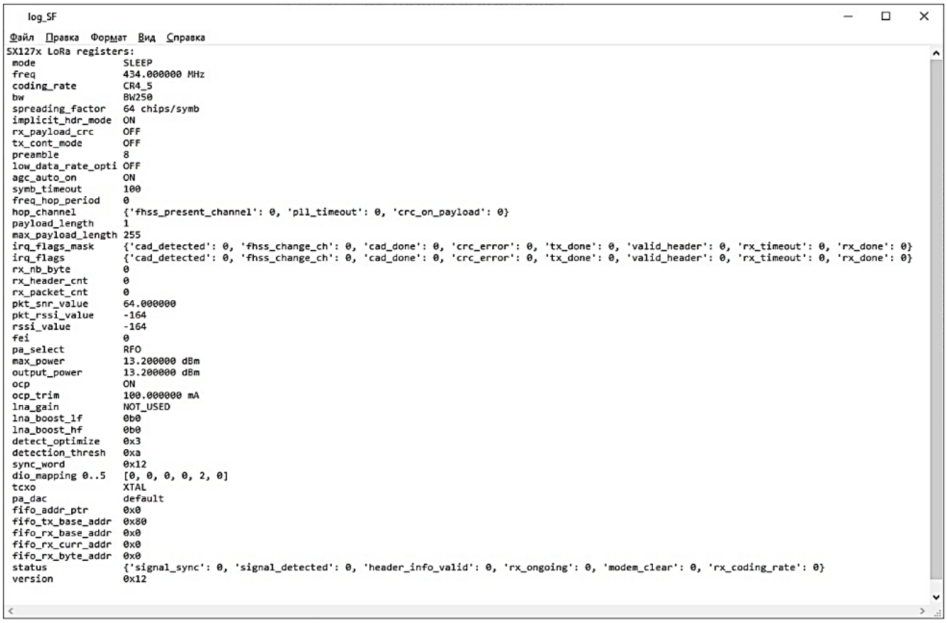
Fig. 14 – *The principle of the register configuration* 0x1e.

The following function has been developed to change the configuration of registers:

```
#!/usr/bin/python3
import spidev
spi = spidev.SpiDev()
spi.open(0, 0)
spi.max_speed_hz = 5000000
RegOpMode = 0x1d
Value     = 0
def set_bw():
  return spi.xfer([RegOpMode | 0x80, 0x72, 0x70])
set_bw()
# Read
ret_1 = spi.xfer([0x1d & 0x7F, 0])[1]
ret_2 = spi.xfer([0x1e & 0x7F, 0])[1]
print(ret_1)         # 128
print(ret_1 >> 7)    # 1
print(ret_2)
print(ret_2 >> 7)
spi.close()
```

After the operation of this program script, we obtained the following configuration data, which are shown in Fig. 15.

Fig. 16 shows the result of modeling the operation of the gateway using a set of optimal parameters.



```
SX127x LoRa registers:
 mode                SLEEP
 freq                434.000000 MHz
 coding_rate         CR4_5
 bw                  BW250
 spreading_factor    64 chips/symb
 implicit_hdr_mode   ON
 rx_payload_crc      OFF
 tx_cont_mode        OFF
 preamble            8
 low_data_rate_opti  OFF
 agc_auto_on         ON
 symb_timeout        100
 freq_hop_period     0
 hop_channel         {'fhss_present_channel': 0, 'pll_timeout': 0, 'crc_on_payload': 0}
 payload_length      1
 max_payload_length  255
 irq_flags_mask      {'cad_detected': 0, 'fhss_change_ch': 0, 'cad_done': 0, 'crc_error': 0, 'tx_done': 0, 'valid_header': 0, 'rx_timeout': 0, 'rx_done': 0}
 irq_flags           {'cad_detected': 0, 'fhss_change_ch': 0, 'cad_done': 0, 'crc_error': 0, 'tx_done': 0, 'valid_header': 0, 'rx_timeout': 0, 'rx_done': 0}
 rx_nb_byte          0
 rx_header_cnt       0
 rx_packet_cnt       0
 pkt_snr_value       64.000000
 pkt_rssi_value      -164
 rssi_value          -164
 fei                 0
 pa_select           RFO
 max_power           13.200000 dBm
 output_power        13.200000 dBm
 ocp                 ON
 ocp_trim            100.000000 mA
 lna_gain            NOT_USED
 lna_boost_lf        0b0
 lna_boost_hf        0b0
 detect_optimize     0x3
 detection_thresh    0xa
 sync_word           0x12
 dio_mapping 0..5    [0, 0, 0, 0, 2, 0]
 tcxo                XTAL
 pa_dac              default
 fifo_addr_ptr       0x0
 fifo_tx_base_addr   0x80
 fifo_rx_base_addr   0x0
 fifo_rx_curr_addr   0x0
 fifo_rx_byte_addr   0x0
 status              {'signal_sync': 0, 'signal_detected': 0, 'header_info_valid': 0, 'rx_ongoing': 0, 'modem_clear': 0, 'rx_coding_rate': 0}
 version             0x12
```

Fig. 15 – *The results of the optimal configuration of the LoRa module are obtained.*

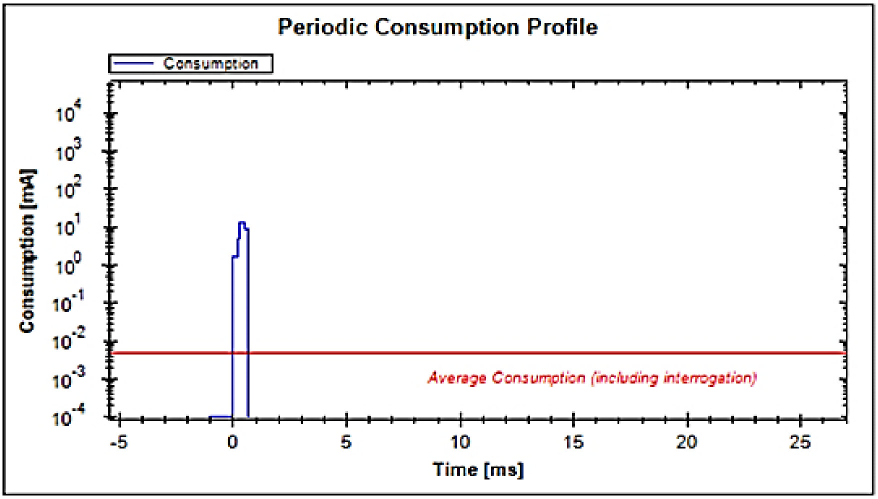
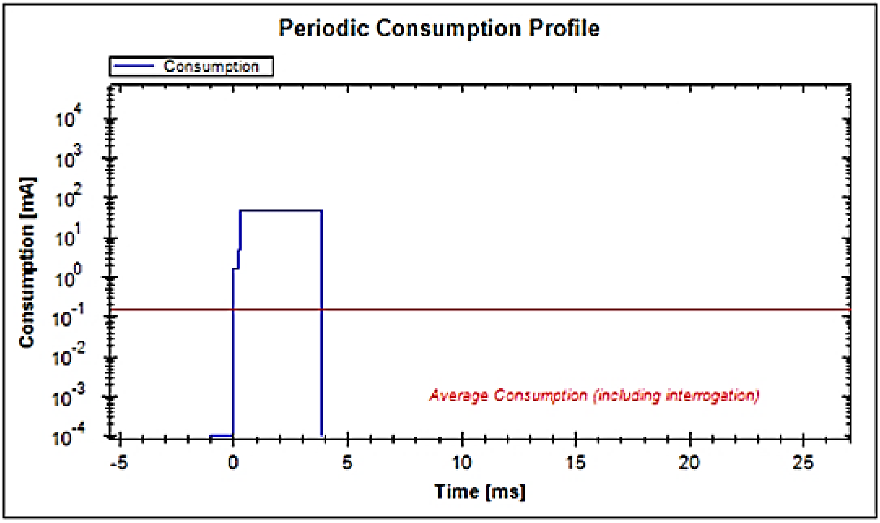


Fig. 16a –*The result of modeling the operation of the gateway using the specified set of optimal parameters for the receiver.*



(b)

Fig. 16b –*The result of modeling the operation of the gateway using the specified set of optimal parameters for the transmitter*.

7 Conclusion

In this work, computer simulation was performed to select the optimal parameters of the IoT gateway using the tools LoRa Modem Calculator, Channel Activity Detection. A method and program scripts for automating the determination of the optimal configuration of IoT gateway modes are proposed.

- The developed prototype of the gateway implements the method of “Protocol Transformation”, which allows you to remove from the data packets received by the gateway, possible dangerous inserts. This can increase the security of packets from the gateway to IoT devices and in reverse, and thus the network as a whole.
- An experimental study of the gateway, which is built on the module SX1278. This device was used to simulate the operation of the gateway with the configuration parameters obtained at the stage of computer simulation.

Experimental studies have shown that the achieved set modes of operation of the LoRa module correspond to the optimal mode of data reception / transmission for the set operating conditions. It should be noted that in order to reduce the energy consumed by the devices and increase the time of their autonomous operation in the framework of the research, it was decided not to use the header and checksum fields.

As a result of modeling and experimental research, the optimal parameters of the gateway operation are determined, provided that up to 1% of the active cycle time is in the air:

– Payload = 7 or 4 bytes,

– Spreading factor = 6,

– BW ≥ 250 kHz,

– header field is missing,

– the checksum field is missing,

– coding type (CR) = 4/5.

8 References

[1] I. Sh. Nevludov, M. A. Omarov, S. P. Novoselov: Modeling and Selection of Optimal Parameters of Security Gateways to Protect Industrial Equipment from Cyberattacks, Journal of Modern Technology and Engineering., Vol. 6, No. 3, 2021, pp. 219 – 229.

[2] S. Novoselov, O. Sychova: Methods of Organizing Communication Between Microcontrollers in the System of Monitoring Energy Consumption, Proceedings of the II International Scientific and Practical Conference – Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs – MC&FPGA-2020, Kharkiv, Ukraine, June 25-26, 2020, pp. 30 – 33.

[3] U. Raza, P. Kulkarni, M. Sooriyabandara: Low Power Wide Area Networks: An Overview, IEEE Communications Surveys & Tutorials, Vol. 19, No. 2, Secondquarter 2017, pp. 855 – 873.

[4] LoRa WAN Technology, Available at: https://itechinfo.ru/

[5] M. Ali Ertürk, M. Ali Aydın, M. T. Buyukakkaslar, H. Evirgen: A Survey on LoRaWAN Architecture, Protocol and Technologies, Future Internet, Vol. 11, No. 10, October 2019, p. 216.

[6] A. J. Wixted, P. Kinnaird, H. Larijani, A. Tait, A. Ahmadinia, N. Strachan: Evaluation of LoRa and LoRaWAN for Wireless Sensor Networks, IEEE Sensors, Orlando, USA, October 2016, pp. 1 – 3.

[7] W. Yang, M. Wang, J. Zhang, J. Zou, M. Hua, T. Xia, X. You: Narrowband Wireless Access for Low-Power Massive Internet of Things: A Bandwidth Perspective, IEEE Wireless Communications, Vol. 24, No. 3, June 2017, pp. 138 – 145.

[8] O. Georgiou, U. Raza: Low Power Wide Area Network Analysis: Can LoRa Scale?, IEEE Wireless Communications Letters, Vol. 6, No. 2, April 2017, pp. 162 – 165.

[9] J. P. Shanmuga Sundaram, W. Du, Z. Zhao: A Survey on LoRa Networking: Research Problems, Current Solutions, and Open Issues, IEEE Communications Surveys & Tutorials, Vol. 22, No. 1, Firstquarter 2020, pp. 371 – 388.

[10] J.- T. Lim, Y. Han: Spreading Factor Allocation for Massive Connectivity in LoRa Systems, IEEE Communications Letters, Vol. 22, No. 4, April 2018, pp. 800 – 803.

[11] G. Zhu, C.- H. Liao, T. Sakdejayont, I.- W. Lai, Y. Narusue, H. Morikawa: Improving the Capacity of a Mesh LoRa Network by Spreading-Factor-Based Network Clustering, IEEE Access, Vol. 7, February 2019, pp. 21584-21596.

[12] How an IoT Edge Ddevice Can be Used as a Gateway, Available at:

https://docs.microsoft.com/en-us/azure/iot-edge/iot-edge-as-gateway?view=iotedge-2020-11

[13] SEMTECH, Wireless, Sensing & Timing, Datasheet (SX1276/77/78/79), Available at: https://cdn-shop.adafruit.com/product-files/3179/sx1276_77_78_79.pdf

[14] J. Courjault, B. Vrigneau, O. Berder, M. R. Bhatnagar: How Robust is a LoRa Communication Against Impulsive Noise?, Proceedings of the IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications, London, UK, August 2020, pp. 1 – 6.

[15] S. Novoselov, O. Sychova: Using Wireless Technology for Managing Distributed Industrial Automation Objects within the Concept of Industry 4.0, Proceedings of the IEEE International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kyiv, Ukraine, October 2019, pp. 580 – 584.

[16] I. Nevludov, O. Sychova, A. Andrusevich, S. Novoselov, D. Mospan, V. Mospan: Simulation of the Sensor Network of Base Stations in a Local Positioning System in Intelligent Industries, Proceedings of the IEEE Problems of Automated Electrodrive. Theory and Practice (PAEP), Kremenchuk, Ukraine, September 2020, pp. 1 – 6.

[17] S. Novoselov, O. Donskov: Distributed Local Positioning System Using DWM1000 Location Chip, Proceedings of the 4th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkov, Ukraine, October 2017, pp. 489 – 492.

[18] S. Novoselov: Wireless Sensor Network for Communication Between Base Stations in the Local Positioning System, Proceedings of the International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkov, Ukraine, October 2018, pp. 383 – 386.