UDC 004.41
DOI: 10.15587/1729-4061.2025.326222

DEVISING A METHOD FOR DETECTING AND CORRECTING MULTI-BIT DATA TRANSMISSION ERRORS IN IOT SYSTEMS BASED ON THE GOLAY CODE

Vladyslav Sokolovskyi
Corresponding author
PhD Student, Assistant*
E-mail: falcon.f2b@gmail.com
Eduard Zharikov
Doctor of Technical Sciences, Professor*
Sergii Telenyk
Doctor of Technical Sciences, Professor**
Department of Information Systems and Technologies
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"
Beresteiskyi ave., 37, Kyiv, Ukraine, 03056
Grzegorz Nowakowski
Lecturer**
Department of Automation and Computer Science
Cracow University of Technology
Warszawska str., 24, Krakow, Poland, 31-155
Svitlana Popereshnyak
PhD, Associate Professor*
Mykhailo Marynenko
Chief Technology Officer
O'Shaughnessy Ventures
453 S Spring str., 400, Los Angeles, California, United States, 90013
*Department of Computer Science and Software Engineering
National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"
Beresteiskyi ave., 37, Kyiv, Ukraine, 03056
**Department of Automation and Computer Science
Cracow University of Technology
Warszawska str., 24, Krakow, Poland, 31-155

The object of this study is information systems within the Internet of Things.

The task addressed involves devising an innovative method for detecting and correcting multi-bit data transmission errors in the Internet of Things networks based on Golay codes.

As the main result of the research, a method has been devised for detecting and correcting multi-bit data transmission errors based on Golay codes during byte-by-byte transmission of an information block.

The method devised is distinguished by its coding scheme, which involves calculating 11 control bits and one parity bit for twelve bytes of the original information message with subsequent mixing using shift operations before transmission to the communication channel.

Thus, for twelve bytes of input information, an information block of 24 bytes is formed at the output of the encoder, and the bits of the bytes belong to eight different code words of the extended Golay code (24, 12).

When transmitting an information block, one or more bits of the transmitted byte may be distorted. But after performing the shift operations on the receiving side, the reverse of those performed before transmission, it becomes possible to detect and correct transmission errors using Golay code decoding methods. The transmission errors of a single byte that are subject to detection and correction can reach eight. This is possible because all the bits of a byte transmitted over a communication channel belong to different combinations of the Golay code – each separate combination is formed by bits of different bytes that have the same numbers.

Due to the fact that an information message of 24 bytes consists of eight code combinations of the Golay code (24, 12), it is possible to correct up to 24 bit errors in one message of 24 bytes

Keywords: software engineering, correction codes, Golay codes, Internet of Things, information system, information block

Received 13.01.2025
Received in revised form 04.03.2025
Accepted date 31.03.2025
Published date 29.04.2025

How to Cite: *Sokolovskyi, V., Zharikov, E., Telenyk, S., Nowakowski, G., Popereshnyak, S., Marynenko, M. (2025 Devising a method for detecting and correcting multi-bit data transmission errors in IoT systems based on the golay code. Eastern-European Journal of Enterprise Technologies, 2 (9 (134)), 14–23. https://doi.org/10.15587/1729-4061.2025.326222*

1. Introduction

The trend in the evolution of information networks in recent years is the widespread implementation of Internet of Things technologies. Accordingly, there is a need to implement interference-resistant information networks that use Internet of Things technologies. This is especially true for the segment of industrial Internet of Things networks, given the degree of use of technology for the purposes of production, automation, and monitoring of processes.

And industrial Internet of Things networks located over a large area and within production facilities are exposed to the greatest impact of electromagnetic interference. It is known that intensive industrial interference leads to data transmission errors of various multiplicity. Therefore, devising technologies for enabling interference-resistant information transmission in Internet of Things networks is an urgent current task.

2. Literature review and problem statement

The problem of error-free data transmission has long been known. For the first time, the possibility of error-free data transmission under interference conditions was reported in [1]. The paper proposes a method for enabling interference-resistant data transmission, which involves using correction codes. The advantage of the paper is the fact that the possibility of the existence of correction codes has been proven, but how to construct these codes was not specified.

Paper [2] reports a study on extended binary Golay codes, namely, the study of the constructions of extended binary Golay codes based on the group algebra $F2G$ of the group $G=D24$. The advantage of the work is that a search algorithm has been developed and all elements $v \in F2D24$ that generate the principal ideals that define extended binary Golay codes have been found programmatically. Previously, the extended binary Golay code was built using one element $\in F2D24$, where $v=v$. Also, all 36,864 elements $v \in F2D24$ were found, from which an extended binary Golay code could be constructed, and it was found that 768 of them satisfy the $v=v$ condition.

In [3], the methods of formation and properties of Golay codes are considered. The main advantages of these codes compared to others used for binary carrier phase manipulation are investigated. The advantage of the work is the accessibility of the presentation of the material and its relevance. The main disadvantage is a very concise presentation of the content.

In [4], a method of steganographic information transmission in the syndrome of noise-resistant Hamming codes ($2r$–1, $2r$–r–1) and Golay (23, 12, 7) is considered. The data arrangement algorithm uses additional information about carrier sequences, due to which the information efficiency of steganographic information transmission increases in comparison with algorithms based on the replacement of the least significant bits of the carrier.

The advantage of the work is the correct and accurate presentation of the material. The disadvantage is the fact that the choice of the considered codes is somewhat limited.

In [5], two dual codes are considered, the Golay code G=(23, 12, 7) and the author's proposed algebrogeometric code C, for encoding information in a dual symmetric channel with a width of W=50 KB/s, with an encoder/decoder clock frequency of 1 GHz, and a bit error probability p=0.005 and a probability of successful decoding of the transmitted codeword of at least 0.9999. It is shown that both codes are suitable for these conditions and that the transmission speed of information encoded by the C code over this channel is approximately 1.12 times higher than for information encoded by the G code. The advantage of the work is the methodology devised for comparing different encoding techniques. The main disadvantage is that the practical testing of the results of the work is poorly discussed.

In [6], the adaptation of the polar code decoding technique for decoding the extended Golay code is considered. Owing to the proposed matrix of permutations between the codewords of these two classes of codes, the Golay code can be decoded by any polar code decoding method.

The advantage of this decoding method, in contrast to the sequential denoising list technique, which is characterized by sequential bit evaluation, is the decoding performance with maximum likelihood. The decoding is characterized by low decoding complexity compared to universal linear code decoders, the most well-known in the literature.

The disadvantage of the work is insufficient attention to the practical aspects of using the proposed method.

In [7], a simplified procedure was devised, called the shift search method for decoding three possible errors in the Golay codeword (23, 12, 7). The proposed decoding algorithm is compared with the algebraic decoding algorithm. The results of the comparison based on computer simulation showed that both algorithms are modular, conventional, and suitable for software implementation. Both of these algorithms effectively decode the Golay code (23, 12, 7) and provide error detection with a multiplicity of up to four and error correction with a multiplicity of up to three.

The advantage of the shift search method for decoding three possible errors in the Golay codeword (23, 12, 7) is its simplicity of implementation.

The disadvantage is a slightly lower decoding speed than the speed of the algebraic decoding algorithm.

In [8], a new interpretation of classical binary Golay codes as polar codes with additional internal permutations is proposed. This establishes a new connection between algebraic codes and polar codes. This method allows decoding Golay codes by the type of polar code using sequential decoding of the cancellation list. It is hypothesized that, by using the Golay code as the main one, other algebraic codes can also be represented in a similar way. The advantage of the work is the possibility of using polar code decoding methods for decoding Golay codes. The disadvantage is the insufficient attention to the practical aspects of using the proposed method.

In [9], new simple encoding and decoding methods for Golay codes based on generalized code arrays (GAG) are proposed. The methods make it possible to use both the Golay code (23, 12, 7) and the extended Golay code (24, 12, 8).

The advantage of the proposed encoding and decoding methods is the simplicity of implementation.

The disadvantage is insufficient attention to the practical aspects of using the proposed method.

In [10], the conditions for using error correction codes (ECC) are considered, and it is also described that the use of ECC creates overhead both per bit of memory and for decoding time, which limits the speed. In particular, this is a problem for programs that require powerful error correction capabilities. Advanced ECCs, such as orthogonal Latin squares or difference sets of codes, which can be decoded with relatively low latency, are also recommended for implementation.

The paper states that the rather high decoding speed is achieved due to the fact that the codes in most cases are not optimal in terms of memory overhead and require more parity check bits.

However, codes such as the Golay (24, 12) code, which minimize the number of parity check bits, have a more complex decoding. A compromise solution for Bose-Chaudhury-Gokkwenghem codes has recently been investigated. The idea is to implement a fast decoder to correct the most common error patterns (single and double adjacent) and use a slower sequential decoder for the remaining patterns. This brief description shows that the same scheme can be effectively implemented for the Golay (24, 12) code.

The advantage of the paper is that the proposed decoder is able to correct some triple adjacent errors, thus covering the most common error patterns.

The disadvantage is insufficient attention to the practical aspects of using the proposed method for constructing decoders.

In [11], an innovative method for detecting and correcting data transmission errors based on Hamming codes in the Internet of Things networks is proposed.

The method is distinguished by an original coding scheme, which involves calculating check digits, as well as shuffling bits of an information block by performing bit shift operations.

The main area of application of the method devised is considered to be IoT system networks.

The advantage of the method devised is the ability to detect and correct all transmission errors of a single byte with an error rate from 1 to 8 within an information block of 24 bytes, which was confirmed experimentally.

The disadvantage of such a coding method is redundancy in coding.

3. The aim and objectives of the study

The aim of our work is to improve the noise immunity of data transmission in branched Internet of Things networks by devising a method for detecting and correcting multi-bit errors during byte-by-byte transmission of individual bytes belonging to one information block. This will make it possible to increase the data transmission speed over a communication channel affected by electromagnetic interference. Owing to the use of the proposed method, the number of retransmissions of information blocks with detected uncorrectable transmission errors is significantly reduced.

To achieve the goal, the following tasks were set:

– to develop a coding scheme for detecting and correcting multi-bit errors in one byte based on the Golay code (24, 12);

– to develop coding and decoding algorithms, as well as a program for implementing the method for detecting and correcting multi-bit errors;

– to evaluate the correction capabilities of the proposed method under an automated mode using a software model of the communication channel.

4. The study materials and methods

4. 1. The object and hypothesis of the study

The object of our study is information systems of the Internet of Things.

The main hypothesis of the study assumes that there is a scheme and technique of encoding information based on the Golay code, with which it is possible to detect and correct multi-bit errors in data transmission during byte-by-byte transmission of an information block.

When conducting the study, a number of assumptions and simplifications were adopted in the work:

1. When transmitting an information block, its length remains unchanged.

2. The problem of determining the first byte of an information block is solved at the protocol level.

3. Information transmission was considered in the absence of interruption of data transmission under the influence of interference.

It is known that the main methods for verifying the integrity of data transmitted over noisy communication channels are simple checksums and cyclic redundancy checks. One of the known encoding methods is the use of Golay codes, with which it is possible to detect and correct multi-bit errors in the transmission of a code word without using retransmission.

4. 2. Brief description of the Golay code (23, 11)

The cyclic Golay code (23, 11) was first described in paper [12]. It has been proven that the use of Golay codes can ensure noise immunity of data transmission over a channel with noise caused by external interference.

The code word of the binary Golay code (23, 12) has a total length of 23 bits, contains 12 information bits and 11 check bits. The check bits are obtained by dividing the modulus of two polynomials corresponding to the information part of the code word by the generating polynomial.

Since the code word of the binary Golay code (23, 12) consists of 23 bits, there are a total of 8,388,608 possible binary values. But each set of twelve information bits corresponds to a unique set of eleven check bits. Therefore, there are only 4096 valid codewords of the Golay (23, 12) code.

The valid codewords of the Golay (23, 12) code are spaced by the Hamming distance between them. The code has a minimum distance d of seven. Golay proved that there can be a densely packed binary (23, 12) code that can detect and correct at most 0.5 (d–1)=3 bit errors.

The Golay code is classified as densely packed codes, i. e., codes in which the number of syndromes required to correct errors of a given multiplicity is exactly equal. Due to the uniqueness of such codes, densely packed codes are also termed "perfect" or "optimal".

The Golay code belongs to the class of cyclic codes with dual generating polynomials:

$$G(\boldsymbol{X})=\boldsymbol{X}11+\boldsymbol{X}10+\boldsymbol{X}6+\boldsymbol{X}5+\boldsymbol{X}4+\boldsymbol{X}2+1,$$

and

$$\tilde{G}(X)=X11+X9+X7+X6+X5+X+1.$$

As a generating polynomial of the cyclic Golay code (23, 12), one of the two indicated polynomials can be used.

Unfortunately, in addition to the Hamming codes ($d$min=3, qi=1) and the Golay code (23, 12), no other perfect, densely packed codes have been found yet, the number of syndromes in which exactly corresponds to the required value for guaranteed error correction of a given multiplicity.

Several well-known decoding procedures are known with the ability to correct three possible errors of the Golay code (23, 12):

– the minimum distance method;
– the standard array method;
– the majority logic method;
– the Kasami error interception algorithm;
– the change and search procedure.

Adding to this code a general parity check for all positions increases by one both the total length of the code and the minimum code distance $d$min=8. This code is known as the Golay code (24, 12).

4. 3. Brief description of the Golay code (24, 12)

When devising a method for detecting multi-bit data transmission errors, the extended Golay code (24, 12) was used as the base code. The choice of the Golay code (24, 12) is not random. The extended Golay code (24, 12) has a total length of 24 bits and consists of 12 information bits and 12 control bits. This code has a Hamming distance of eight.

To decode the Golay code (23, 12), which is formed after discarding the parity bit, a permutation decoding method was used. In this case, in order to simplify the error search procedure, the operations of cyclic shift of error syndromes and their comparison with the calculated syndromes (Kasami-Rudolph algorithm) are used.

For the Golay code (23, 12), which has a generating polynomial $g(x)=x11+x9+x7+x6+x5+x+1$, the set of errors whose weight (multiplicity) does not exceed three is covered by three error sequences: $e1(x)=0$, $e17(x)=x16$, $e18(x)=x17$, which have the following syndromes: $S1(x)=0$; $S17(x)=x8+x7+x4+x3+x+1$ та $S18(x)=x9+x8+x5+x4+$ $+x2+x$. For decoding, an error syndrome that differs from $S1(x)$ in no more than three positions is tracked, as well as error syndromes that differ from $S17(x)$ and $S18(x)$ in no more than two positions.

Decoding is performed in two stages. First, the received code word is recorded in the buffer register and the error syndrome is calculated. During the second stage, a cycle of 23 cycles is performed and error search and correction are performed by cyclic shifting of the error syndrome and its comparison with the syndromes in the syndrome analyzer. At the same time, the code word is cyclically shifted in the buffer register. Error positions are detected when any of the inequalities in the syndrome analyzer coincide.

The extended Golay code (24, 12) makes it possible to detect 100 % of errors from one to six bits in any combination, 100 % of odd bit errors of any pattern, and 99.988 % of other errors.

Regarding the error correction property of data transmission, the Golay code (24, 12) makes it possible to correct 100 % of errors with a multiplicity of one to three of any type and also corrects 0.24 % of other errors. The parity bit of the extended Golay code (24, 12) complements its error correction properties, allowing one to detect all combinations of 4-bit errors, but not correct them.

5. Results of investigating the proposed method for detecting and correcting multi-bit errors during the transmission of an information message

5. 1. Information block coding scheme for detecting and correcting multi-bit transmission errors

The coding scheme is constructed in three stages.

At the first stage, an information block was generated, the bit arrangement of which corresponded to that given in Table 1.

At the first stage, twelve check bits are calculated for the bits of the same digits of the first twelve information bytes of the block. One of the check bits is a parity bit. That is, 24 bits of one digit of the 24 bytes of the block form the Golay code (24, 12).

Thus, twelve check bytes are calculated for the twelve information bytes. That is, after the 12 information bytes, there are 12 bytes consisting of the check bits of the Golay code (24, 12).

The next stage was the mixing of data in the information block by applying bitwise shift operations to the information bits and parity bits.

The arrangement of bits in the information block after performing these right shift operations is given in Table 2.

In this process, all the low-order bits (those in bit position 1) of all the bytes in the transmission buffer remain unchanged (no shifting is performed). All the bits of all the bytes in the second-order information block are shifted one bit to the right, etc.

The third and final stage of forming the information block is also the bit shuffling operation. This shuffling was performed by bit shifting operations from low-order to high-order bits.

The finally proposed coding scheme takes the form given in Table 3. The information from the transmission buffer was then transmitted in separate bytes to the data transmission channel.

Table 1

Bit arrangement in the information block

TBUF transfer buffer bytes																							
$n_{i,23}$	$n_{i,22}$	$n_{i,21}$	$n_{i,20}$	$n_{i,19}$	$n_{i,18}$	$n_{i,17}$	$n_{i,16}$	$n_{i,15}$	$n_{i,14}$	$n_{i,13}$	$n_{i,12}$	$n_{i,11}$	$n_{i,10}$	$n_{i,9}$	$n_{i,8}$	$n_{i,7}$	$n_{i,6}$	$n_{i,5}$	$n_{i,4}$	$n_{i,3}$	$n_{i,2}$	$n_{i,1}$	$n_{i,0}$
Information bits												Control bits											
$k_{i,12}$	$k_{i,11}$	$k_{i,10}$	$k_{i,9}$	$k_{i,8}$	$k_{i,7}$	$k_{i,6}$	$k_{i,5}$	$k_{i,4}$	$k_{i,3}$	$k_{i,2}$	$k_{i,1}$	$r_{i,11}$	$r_{i,10}$	$r_{i,9}$	$r_{i,8}$	$r_{i,7}$	$r_{i,6}$	$r_{i,5}$	$r_{i,4}$	$r_{i,3}$	$r_{i,2}$	$r_{i,1}$	p_i
$k_{8,12}$	$k_{8,11}$	$k_{8,10}$	$k_{8,9}$	$k_{8,8}$	$k_{8,7}$	$k_{8,6}$	$k_{8,5}$	$k_{8,4}$	$k_{8,3}$	$k_{8,2}$	$k_{8,1}$	$r_{8,11}$	$r_{8,10}$	$r_{8,9}$	$r_{8,8}$	$r_{8,7}$	$r_{8,6}$	$r_{8,5}$	$r_{8,4}$	$r_{8,3}$	$r_{8,2}$	$r_{8,1}$	p_8
$k_{7,12}$	$k_{7,11}$	$k_{7,10}$	$k_{7,9}$	$k_{7,8}$	$k_{7,7}$	$k_{7,6}$	$k_{7,5}$	$k_{7,4}$	$k_{7,3}$	$k_{7,2}$	$k_{7,1}$	$r_{7,11}$	$r_{7,10}$	$r_{7,9}$	$r_{7,8}$	$r_{7,7}$	$r_{7,6}$	$r_{7,5}$	$r_{7,4}$	$r_{7,3}$	$r_{7,2}$	$r_{7,1}$	p_7
$k_{6,12}$	$k_{6,11}$	$k_{6,10}$	$k_{6,9}$	$k_{6,8}$	$k_{6,7}$	$k_{6,6}$	$k_{6,5}$	$k_{6,4}$	$k_{6,3}$	$k_{6,2}$	$k_{6,1}$	$r_{6,11}$	$r_{6,10}$	$r_{6,9}$	$r_{6,8}$	$r_{6,7}$	$r_{6,6}$	$r_{6,5}$	$r_{6,4}$	$r_{6,3}$	$r_{6,2}$	$r_{6,1}$	p_6
$k_{5,12}$	$k_{5,11}$	$k_{5,10}$	$k_{5,9}$	$k_{5,8}$	$k_{5,7}$	$k_{5,6}$	$k_{5,5}$	$k_{5,4}$	$k_{5,3}$	$k_{5,2}$	$k_{5,1}$	$r_{5,11}$	$r_{5,10}$	$r_{5,9}$	$r_{5,8}$	$r_{5,7}$	$r_{5,6}$	$r_{5,5}$	$r_{5,4}$	$r_{5,3}$	$r_{5,2}$	$r_{5,1}$	p_5
$k_{4,12}$	$k_{4,11}$	$k_{4,10}$	$k_{4,9}$	$k_{4,8}$	$k_{4,7}$	$k_{4,6}$	$k_{4,5}$	$k_{4,4}$	$k_{4,3}$	$k_{4,2}$	$k_{4,1}$	$r_{4,11}$	$r_{4,10}$	$r_{4,9}$	$r_{4,8}$	$r_{4,7}$	$r_{4,6}$	$r_{4,5}$	$r_{4,4}$	$r_{4,3}$	$r_{4,2}$	$r_{4,1}$	p_4
$k_{3,12}$	$k_{3,11}$	$k_{3,10}$	$k_{3,9}$	$k_{3,8}$	$k_{3,7}$	$k_{3,6}$	$k_{3,5}$	$k_{3,4}$	$k_{3,3}$	$k_{3,2}$	$k_{3,1}$	$r_{3,11}$	$r_{3,10}$	$r_{3,9}$	$r_{3,8}$	$r_{3,7}$	$r_{3,6}$	$r_{3,5}$	$r_{3,4}$	$r_{3,3}$	$r_{3,2}$	$r_{3,1}$	p_3
$k_{2,12}$	$k_{2,11}$	$k_{2,10}$	$k_{2,9}$	$k_{2,8}$	$k_{2,7}$	$k_{2,6}$	$k_{2,5}$	$k_{2,4}$	$k_{2,3}$	$k_{2,2}$	$k_{2,1}$	$r_{2,11}$	$r_{2,10}$	$r_{2,9}$	$r_{2,8}$	$r_{2,7}$	$r_{2,6}$	$r_{2,5}$	$r_{2,4}$	$r_{2,3}$	$r_{2,2}$	$r_{2,1}$	p_2
$k_{1,12}$	$k_{1,11}$	$k_{1,10}$	$k_{1,9}$	$k_{1,8}$	$k_{1,7}$	$k_{1,6}$	$k_{1,5}$	$k_{1,4}$	$k_{1,3}$	$k_{1,2}$	$k_{1,1}$	$r_{1,11}$	$r_{1,10}$	$r_{1,9}$	$r_{1,8}$	$r_{1,7}$	$r_{1,6}$	$r_{1,5}$	$r_{1,4}$	$r_{1,3}$	$r_{1,2}$	$r_{1,1}$	p_1

Table 2

Structure of information arrangement in the information block after performing right shift operations

Transmission buffer bytes																							
$n_{i,23}$	$n_{i,22}$	$n_{i,21}$	$n_{i,20}$	$n_{i,19}$	$n_{i,18}$	$n_{i,17}$	$n_{i,16}$	$n_{i,15}$	$n_{i,14}$	$n_{i,13}$	$n_{i,12}$	$n_{i,11}$	$n_{i,10}$	$n_{i,9}$	$n_{i,8}$	$n_{i,7}$	$n_{i,6}$	$n_{i,5}$	$n_{i,4}$	$n_{i,3}$	$n_{i,2}$	$n_{i,1}$	$n_{i,0}$
$r_{8,6}$	$r_{8,5}$	$r_{8,4}$	$r_{8,3}$	$r_{8,2}$	$r_{8,1}$	p_8	$k_{8,12}$	$k_{8,11}$	$k_{8,10}$	$k_{8,9}$	$k_{8,8}$	$k_{8,7}$	$k_{8,6}$	$k_{8,5}$	$k_{8,4}$	$k_{8,3}$	$k_{8,2}$	$k_{8,1}$	$r_{8,11}$	$r_{8,10}$	$r_{8,9}$	$r_{8,8}$	$r_{8,7}$
$r_{7,5}$	$r_{7,4}$	$r_{7,3}$	$r_{7,2}$	$r_{7,1}$	p_7	$k_{7,12}$	$k_{7,11}$	$k_{7,10}$	$k_{7,9}$	$k_{7,8}$	$k_{7,7}$	$k_{7,6}$	$k_{7,5}$	$k_{7,4}$	$k_{7,3}$	$k_{7,2}$	$k_{7,1}$	$r_{7,11}$	$r_{7,10}$	$r_{7,9}$	$r_{7,8}$	$r_{7,7}$	$r_{7,6}$
$r_{6,4}$	$r_{6,3}$	$r_{6,2}$	$r_{6,1}$	p_6	$k_{6,12}$	$k_{6,11}$	$k_{6,10}$	$k_{6,9}$	$k_{6,8}$	$k_{6,7}$	$k_{6,6}$	$k_{6,5}$	$k_{6,4}$	$k_{6,3}$	$k_{6,2}$	$k_{6,1}$	$r_{6,11}$	$r_{6,10}$	$r_{6,9}$	$r_{6,8}$	$r_{6,7}$	$r_{6,6}$	$r_{6,5}$
$r_{5,3}$	$r_{5,2}$	$r_{5,1}$	p_5	$k_{5,12}$	$k_{5,11}$	$k_{5,10}$	$k_{5,9}$	$k_{5,8}$	$k_{5,7}$	$k_{5,6}$	$k_{5,5}$	$k_{5,4}$	$k_{5,3}$	$k_{5,2}$	$k_{5,1}$	$r_{5,11}$	$r_{5,10}$	$r_{5,9}$	$r_{5,8}$	$r_{5,7}$	$r_{5,6}$	$r_{5,5}$	$r_{5,4}$
$r_{4,2}$	$r_{4,1}$	p_4	$k_{4,12}$	$k_{4,11}$	$k_{4,10}$	$k_{4,9}$	$k_{4,8}$	$k_{4,7}$	$k_{4,6}$	$k_{4,5}$	$k_{4,4}$	$k_{4,3}$	$k_{4,2}$	$k_{4,1}$	$r_{4,11}$	$r_{4,10}$	$r_{4,9}$	$r_{4,8}$	$r_{4,7}$	$r_{4,6}$	$r_{4,5}$	$r_{4,4}$	$r_{4,3}$
$r_{3,1}$	p_3	$k_{3,12}$	$k_{3,11}$	$k_{3,10}$	$k_{3,9}$	$k_{3,8}$	$k_{3,7}$	$k_{3,6}$	$k_{3,5}$	$k_{3,4}$	$k_{3,3}$	$k_{3,2}$	$k_{3,1}$	$r_{3,11}$	$r_{3,10}$	$r_{3,9}$	$r_{3,8}$	$r_{3,7}$	$r_{3,6}$	$r_{3,5}$	$r_{3,4}$	$r_{3,3}$	$r_{3,2}$
p_2	$k_{2,12}$	$k_{2,11}$	$k_{2,10}$	$k_{2,9}$	$k_{2,8}$	$k_{2,7}$	$k_{2,6}$	$k_{2,5}$	$k_{2,4}$	$k_{2,3}$	$k_{2,2}$	$k_{2,1}$	$r_{2,11}$	$r_{2,10}$	$r_{2,9}$	$r_{2,8}$	$r_{2,7}$	$r_{2,6}$	$r_{2,5}$	$r_{2,4}$	$r_{2,3}$	$r_{2,2}$	$r_{2,1}$
$k_{1,12}$	$k_{1,11}$	$k_{1,10}$	$k_{1,9}$	$k_{1,8}$	$k_{1,7}$	$k_{1,6}$	$k_{1,5}$	$k_{1,4}$	$k_{1,3}$	$k_{1,2}$	$k_{1,1}$	$r_{1,11}$	$r_{1,10}$	$r_{1,9}$	$r_{1,8}$	$r_{1,7}$	$r_{1,6}$	$r_{1,5}$	$r_{1,4}$	$r_{1,3}$	$r_{1,2}$	$r_{1,1}$	p_1

Table 3

Information encoding scheme in the transmission buffer

Transmission buffer bytes																							
$n_{i,23}$	$n_{i,22}$	$n_{i,21}$	$n_{i,20}$	$n_{i,19}$	$n_{i,18}$	$n_{i,17}$	$n_{i,16}$	$n_{i,15}$	$n_{i,14}$	$n_{i,13}$	$n_{i,12}$	$n_{i,11}$	$n_{i,10}$	$n_{i,9}$	$n_{i,8}$	$n_{i,7}$	$n_{i,6}$	$n_{i,5}$	$n_{i,4}$	$n_{i,3}$	$n_{i,2}$	$n_{i,1}$	$n_{i,0}$
$r_{8,6}$	$r_{7,4}$	$r_{6,2}$	p_5	$k_{4,11}$	$k_{3,9}$	$k_{2,7}$	$k_{1,5}$	$k_{8,11}$	$k_{7,9}$	$k_{6,7}$	$k_{5,5}$	$k_{4,3}$	$k_{3,1}$	$r_{2,10}$	$r_{1,8}$	$k_{8,3}$	$k_{7,1}$	$r_{6,10}$	$r_{5,8}$	$r_{4,6}$	$r_{3,4}$	$r_{2,2}$	p_1
$r_{7,5}$	$r_{6,3}$	$r_{5,1}$	$k_{4,12}$	$k_{3,10}$	$k_{2,8}$	$k_{1,6}$	$k_{8,12}$	$k_{7,10}$	$k_{6,8}$	$k_{5,6}$	$k_{4,4}$	$k_{3,2}$	$r_{2,11}$	$r_{1,9}$	$k_{8,4}$	$k_{7,2}$	$r_{6,11}$	$r_{5,9}$	$r_{4,7}$	$r_{3,5}$	$r_{2,3}$	$r_{1,1}$	$r_{8,7}$
$r_{6,4}$	$r_{5,2}$	p_4	$k_{3,11}$	$k_{2,9}$	$k_{1,7}$	p_8	$k_{7,11}$	$k_{6,9}$	$k_{5,7}$	$k_{4,5}$	$k_{3,3}$	$k_{2,1}$	$r_{1,10}$	$k_{8,5}$	$k_{7,3}$	$k_{6,1}$	$r_{5,10}$	$r_{4,8}$	$r_{3,6}$	$r_{2,4}$	$r_{1,2}$	$r_{8,8}$	$r_{7,6}$
$r_{5,3}$	$r_{4,1}$	$k_{3,12}$	$k_{2,10}$	$k_{1,8}$	$r_{8,1}$	$k_{7,12}$	$k_{6,10}$	$k_{5,8}$	$k_{4,6}$	$k_{3,4}$	$k_{2,2}$	$r_{1,11}$	$k_{8,6}$	$k_{7,4}$	$k_{6,2}$	$r_{5,11}$	$r_{4,9}$	$r_{3,7}$	$r_{2,5}$	$r_{1,3}$	$r_{8,9}$	$r_{7,7}$	$r_{6,5}$
$r_{4,2}$	p_3	$k_{2,11}$	$k_{1,9}$	$r_{8,2}$	p_7	$k_{6,11}$	$k_{5,9}$	$k_{4,7}$	$k_{3,5}$	$k_{2,3}$	$k_{1,1}$	$k_{8,7}$	$k_{7,5}$	$k_{6,3}$	$k_{5,1}$	$r_{4,10}$	$r_{3,8}$	$r_{2,6}$	$r_{1,4}$	$r_{8,10}$	$r_{7,8}$	$r_{6,6}$	$r_{5,4}$
$r_{3,1}$	$k_{2,12}$	$k_{1,10}$	$r_{8,3}$	$r_{7,1}$	$k_{6,12}$	$k_{5,10}$	$k_{4,8}$	$k_{3,6}$	$k_{2,4}$	$k_{1,2}$	$k_{8,8}$	$k_{7,6}$	$k_{6,4}$	$k_{5,2}$	$r_{4,11}$	$r_{3,9}$	$r_{2,7}$	$r_{1,5}$	$r_{8,11}$	$r_{7,9}$	$r_{6,7}$	$r_{5,5}$	$r_{4,3}$
p_2	$k_{1,11}$	$r_{8,4}$	$r_{7,2}$	p_6	$k_{5,11}$	$k_{4,9}$	$k_{3,7}$	$k_{2,5}$	$k_{1,3}$	$k_{8,9}$	$k_{7,7}$	$k_{6,5}$	$k_{5,3}$	$k_{4,1}$	$r_{3,10}$	$r_{2,8}$	$r_{1,6}$	$k_{8,1}$	$r_{7,10}$	$r_{6,8}$	$r_{5,6}$	$r_{4,4}$	$r_{3,2}$
$k_{1,12}$	$r_{8,5}$	$r_{7,3}$	$r_{6,1}$	$k_{5,12}$	$k_{4,10}$	$k_{3,8}$	$k_{2,6}$	$k_{1,4}$	$k_{8,10}$	$k_{7,8}$	$k_{6,6}$	$k_{5,4}$	$k_{4,2}$	$r_{3,11}$	$r_{2,9}$	$r_{1,7}$	$k_{8,2}$	$r_{7,11}$	$r_{6,9}$	$r_{5,7}$	$r_{4,5}$	$r_{3,3}$	$r_{2,1}$

In this form, the information, control bits, and total parity bits of the eight codewords of the Golay code (24, 12) are located in the transmission buffer before the data transmission process begins over the communication channel.

5. 2. Encoding procedure and decoding procedure with detection and correction of multi-bit errors

Initially, an information block is formed from information bits and control bits. To calculate the control bits, the Golay code (24, 12) is used.

The encoding scheme, according to which information and control bits are placed in the transmission buffer, is given above. In this form, the information block is transmitted byte by byte. After the reception of the entire information block is completed (when conducting the experiment, the size is 24 bytes), bit-by-bit shift operations are performed in the opposite direction to the shift operations used during encoding.

As a result, the information received in the reception buffer will be in the same order as given in Table 1, but some bits may be distorted due to physical influences on the communication channel.

Fig. 1 shows the encoding algorithm.

The next step is to perform one of the known decoding methods, as well as detection and correction of data transmission errors for all eight code combinations of the Golay code, which make up the information block received on the receiving side.

The proposed method of transmitting information over communication channels in blocks of 24 bytes allows for diagnostic detection and correction of transmission errors on the receiving side even in the cases of multi-bit transmission errors within one byte belonging to a particular information block.

Fig. 2 shows the decoding algorithm.

In order to test the basic hypothesis, the proposed method for detecting and correcting multi-bit errors was implemented as a program in the Rust language.

But it should be noted that the encoding and decoding algorithms that implement the proposed method for detecting and correcting multi-bit errors can be implemented using different programming languages, for example: Python, C, C++, Java, and others.

5. 3. Evaluation of the noise immunity of the proh posed method by testing in automated mode

The noise immunity of the proposed method for correcting multi-bit errors that occur when transmitting information over a data channel was assessed by automated testing. A software model of the data channel was developed for the evaluation. The structure of the software model of the data channel is shown in Fig. 3.

The specified model was used under an automated mode to assess the performance of the proposed decoding method and the detection and correction of information transmission errors.

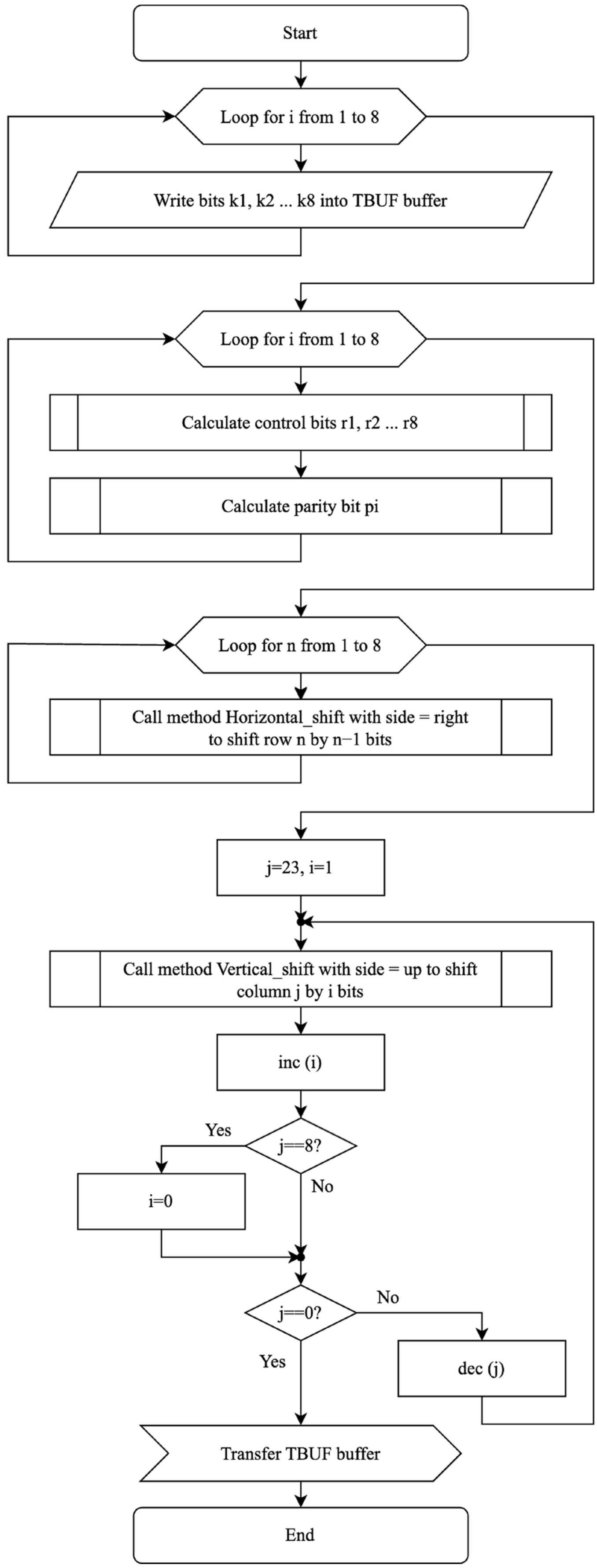

Fig. 1. Coding algorithm

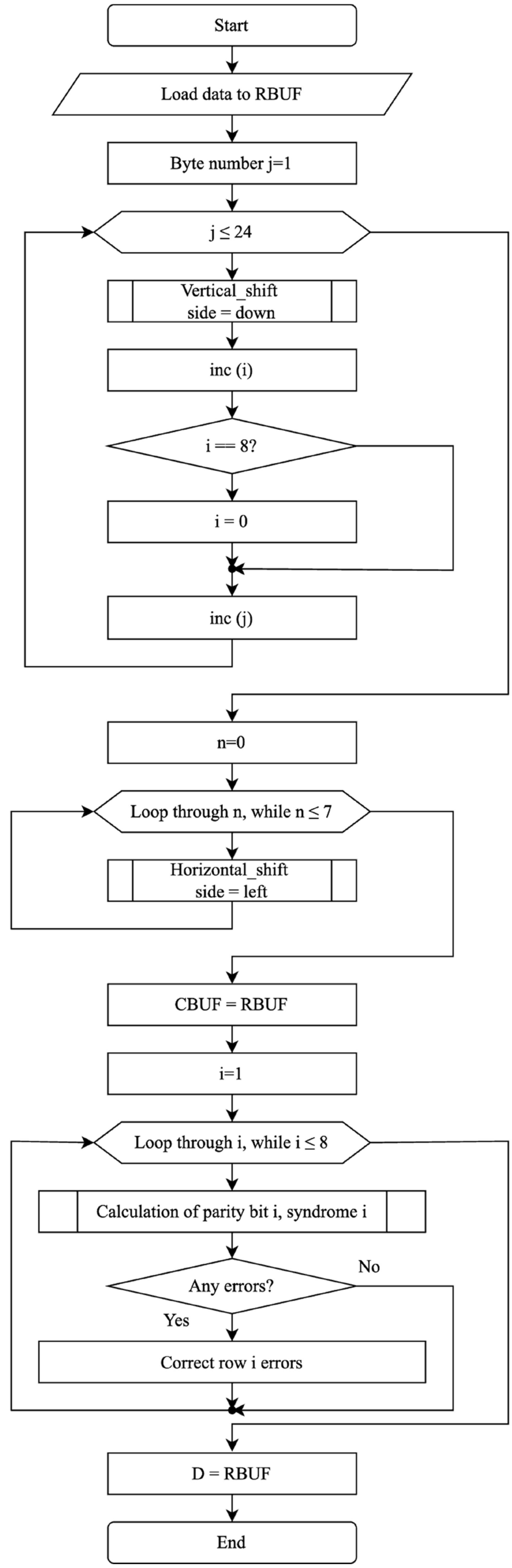

Fig. 2. Decoding algorithm

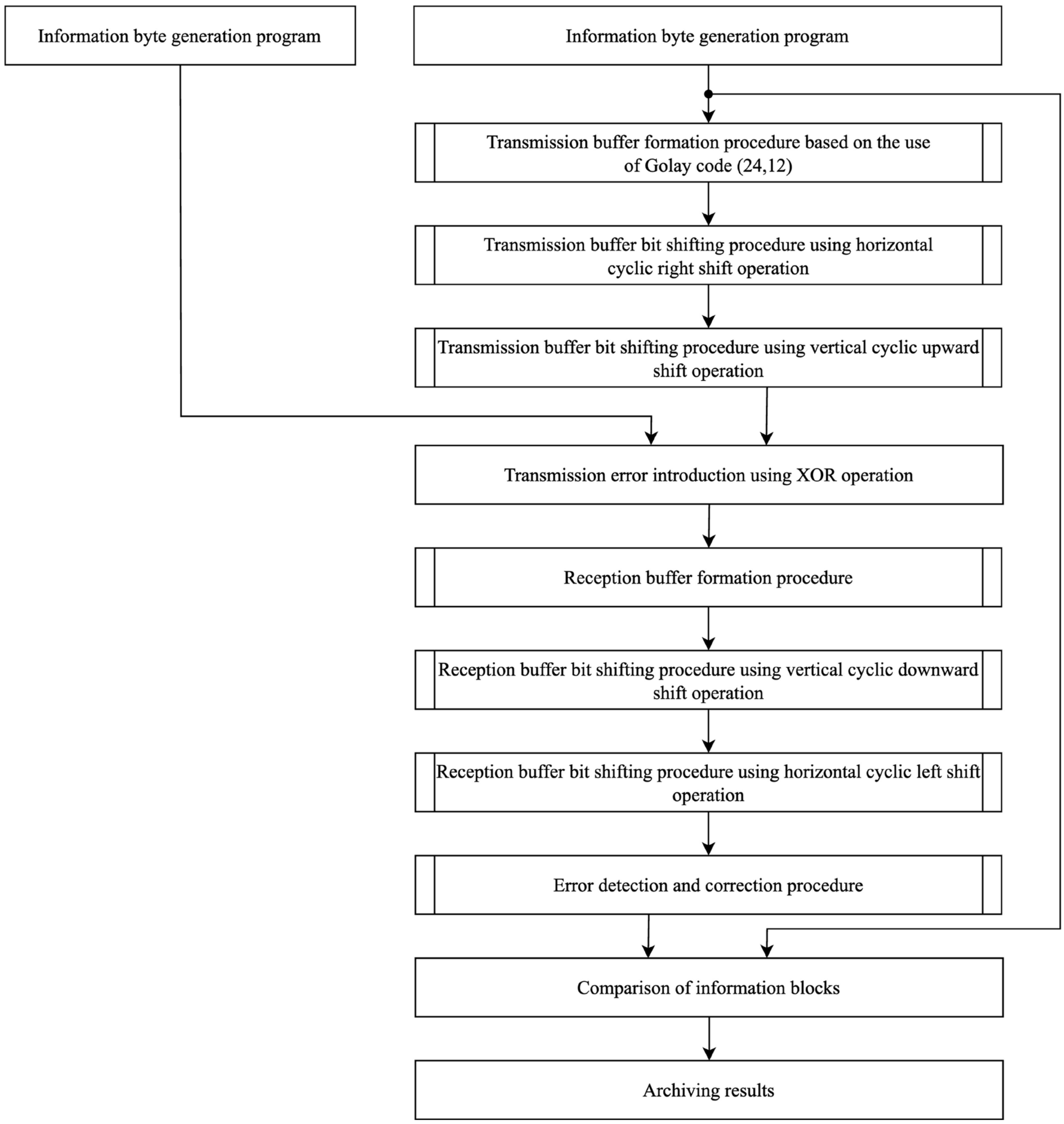

Fig. 3. Structure of the data channel software model

Brief description of the structural scheme. The program that is the information byte generator generates 12 information bytes, which are fed to the inputs of the encoding program and the comparator program. The encoding program performs encoding according to the developed encoding scheme (Table 4). The result of the encoding program is an information block of 24 bytes. The transmission error generator program also forms an information block of 24 bytes, but each bit of the information block can be either zero or one.

Next, the information blocks received from the encoding program and from the transmission error generator program are summed using exclusive OR operations. Thus, we simulate an external influence on the data transmission channel. An information block of 24 bytes, obtained as a result of bitwise summation, is fed to the input of the decoding program. The result of the decoding program, which implements algorithm 2, is 12 restored information bytes. This information is fed to the input of the comparator program. The comparator program compares the information received from the information byte generator program and the restored information from the decoding program. The comparison result is recorded and displayed.

The software model of the data transmission channel used in the testing was written in the Rust language.

The test results are given in Table 4.

The following testing principle was implemented:

1. For each of the 24 bytes prepared for transmission over the communication channel; during testing, artificial errors were sequentially simulated with a multiplicity from one to eight.

2. In this case, each block contained one byte with erroneous bits, and the multiplicity of the error in one byte varied from 1 to 8.

3. Using the proposed methodology and algorithm, error correction and restoration of the original information were performed.

4. After error correction, the resulting data were compared with the original data; the main indicators characterizing the ability and quality of error correction were also calculated.

Table 4

Testing results of the proposed method for detecting and correcting multi-bit errors

Number of errors in one byte	Number of corrupted bytes in the buffer	Number of cases	Number of bytes transferred	Number of damaged bits	Number of bit errors in information bits	number of bits fixed
1	1	192	4,608	192	96	96
1	2	2,208	52,992	4,416	2,208	2,208
1	3	16,192	388,608	48,576	24,288	24,288
2	1	672	16,128	1,344	672	672
2	2	276	6,624	1,104	552	552
2	3	2,024	48,576	12,144	6,072	6,072
3	1	1,344	32,256	4,032	2,016	2,016
3	2	15,456	370,944	92,736	46,368	46,368
3	3	113,344	2,720,256	1,020,096	510,048	510,048
4	1	1,680	40,320	6,720	3,360	3,360
4	2	19,320	463,680	154,560	77280	77280
4	3	38,962,000	935,088,000	467,544,000	233,772,000	233,772,000
5	1	1,344	32,256	6,720	3,360	3,360
5	2	865,536	20,772,864	8,655,360	4,327,680	4,327,680
5	3	113,344	2,720,256	1,700,160	850,080	850,080
6	1	672	16,128	4,032	2,016	2,016
6	2	7,728	185,472	92,736	46,368	46,368
6	3	56,672	1,360,128	1,020,096	510,048	510,048
7	1	192	4,608	1,344	672	672
7	2	2,208	52,992	30,912	15,456	15,456
7	3	16,192	388,608	340,032	170,016	170,016
8	1	24	576	192	96	96
8	2	276	6,624	4,416	2,208	2,208
8	3	2,024	48,576	48,576	24,288	24,288

6. Discussion of results based on investigating the proposed method of noise-resistant data coding in the Internet of Things networks

The main result of our work is the devised method for detecting and correcting multi-bit data transmission errors based on the Golay code (24, 12).

The proposed data transmission method provides detection and correction of up to 24-bit errors during byte-by-byte transmission of a 24-byte message and makes it possible to detect up to three eight-fold errors in individual bytes of the message. No less interesting results of the current work include the fact that we have developed and investigated the following in the process of conducting the experiment: the information block coding scheme, and the coding and decoding algorithms.

To assess the noise resistance of the method for detecting and correcting multi-bit data transmission errors, a software model of the data transmission channel was used, the structure of which is shown in Fig. 3.

Our results are based on application of the original information encoding scheme, which, in turn, involves the use of Golay codes (24, 12). The encoding algorithm is shown in Fig. 1. Information encoding is carried out in three stages. First, twelve control bytes are added to the twelve primary bytes of information using the Golay code (24, 12). The result of this procedure is given in Table 1. Next, the operations of mixing the bits of the transmission buffer are performed using cyclic shift operations. First, horizontal shift operations are performed – the result is given in Table 2, and then vertical shift operations – the result is given in Table 3. In this form, information from the transmission buffer is transmitted byte by byte to the communication channel. On the receiving side, shift operations are performed on the contents of the reception buffer, which are opposite to those given above. The decoding algorithm is shown in Fig. 2.

When using the Golay code (24, 12) in its original form, its main drawback is manifested – the inability to correct multi-bit errors in the transmission of a code word with a multiplicity of more than three.

That is why the use of the original coding scheme leads to the fact that packet transmission errors are distributed between eight code words of the Golay code (24, 12), which leads to a decrease in the multiplicity of errors in the transmission of a separate byte of the information block.

And this in most cases makes it possible to successfully detect and correct errors and makes it possible to correct up to three eight-fold errors in individual bytes belonging to one packet of 24 bytes (Table 4).

It makes sense to compare the method devised with the well-known method [11] of detecting and correcting multiple transmission errors based on the use of the Hamming code (12, 8). The specified method also uses the principle of mixing the code words of the base code. However, the correcting properties of the Hamming code (12, 8) make it possible to correct only one transmission error of the code word, and therefore, when transmitting an information block of 24 bytes, eight-fold transmission errors of a maximum of two bytes can be detected and corrected. In this case, the information block contains 16 code words of the Hamming code.

The proposed method for detecting and correcting multi-bit data transmission errors based on the extended Golay code (24, 12) makes it possible to correct three eight-fold errors of individual bytes in eight code words of the Golay code, which contain the information block. That is, the efficiency of the proposed method for detecting and correcting multi-bit errors based on the Golay code (24, 12) is higher than that of the method reported in [11].

In [13], a large family of error-correcting block codes is considered. The possibility of using different types of block codes for different applications is considered. Also, in [13], a coding scheme for correcting single and double adjacent errors is proposed, which makes it possible to detect 2-bit errors and correct 1-bit errors.

The results of the study allow us to state the following:

1. The proposed method for detecting and correcting multi-bit errors is based on Golay codes (24, 12) and performs the transmission of an information block consisting of 24 bytes, of which 12 bytes contain control bits, as a sequence of individual bytes.

2. The method has relatively high basic indicators for correcting multi-bit transmission errors: when transmitting 12 information bytes, multiple errors of individual bytes can be corrected, the multiplicity of errors can reach eight, and the number of bytes containing such errors can reach three.

3. Permanent errors of one bit of one block are detected and corrected. For example, a permanent error in a specific bit of all 24 bytes.

4. The proposed method makes it possible to increase the noise immunity of the data transmission channel.

The limitation of the method is the real limits of use; specifically, the method can be used in the case when the information is transmitted in blocks of constant length (24 bytes). Moreover, the message must be transmitted byte by byte.

The use of the method is limited to those cases when the length of the information block does not change during the transmission process, but only bit transmission errors occur.

Regarding the conditions of use, it should be noted that the method devised is sensitive to changes in the length of the information block during transmission over the communication channel, that is, to the loss of individual bytes.

The application of the method may have limited use in the event of the need to organize high-speed data transmission.

The main disadvantage of the method is the unreliability of the transmission results in the event of complete loss of individual bytes during transmission of the information block over the information transmission channel.

A significant disadvantage of the proposed method is redundancy. However, this disadvantage, as noted in [14], is not critical from the point of view of the use of noise-resistant coding because in many applications fast encoding and decoding is more important than redundancy.

The disadvantage of the proposed method is that the number of information bytes is equal to the number of bytes containing check and parity bits. That is, the amount of data transmitted is doubled.

It makes sense to use this encoding method in the case when it is necessary to avoid retransmission of data. The Golay code (24, 12) makes it possible to correct errors in exchange for the price of doubling the data.

Further development of our research may involve, on the one hand, the application of information block encoding schemes that are different from the proposed one. On the other hand, the procedure for decoding the code of the extended Golay code (24, 12) after discarding the lower control bit (parity bit) becomes essentially a procedure for detecting and correcting transmission errors of the Golay code (24, 11). And there are more than five such procedures, only the most well-known ones. Therefore, further development of this research may be the use of other procedures for decoding the Golay code (23, 11).

7. Conclusions

1. The scheme that we had built has confirmed the correctness of the decisions made and allowed us to construct a workable software model of the data transmission channel based on the use of the proposed methods for detecting and correcting multi-bit errors. The proposed coding scheme is based on the permutation of bits in the information block before transmitting the block over the communication channel. In the case when data is transmitted in separate code words, this makes it possible to diagnose and correct multiple errors during the transmission of individual bytes belonging to one information block.

2. The coding procedure and the decoding procedure with detection and correction of multi-bit errors, developed on the basis of the coding scheme, allowed us to significantly increase the noise immunity of the data transmission channel, which was confirmed by testing the software model of the channel. The results of the study confirmed that the method devised for the detection and correction of multi-bit data transmission errors based on the Golay code (24, 12) has a special meaning for use in distributed Internet of Things systems operating under conditions of high levels of electromagnetic interference. Coding and decoding algorithms have been developed, which allowed us to implement the proposed method of detection and correction of multi-bit errors in one byte based on the Golay code (24, 12) when transmitting information over a communication channel. The proposed algorithms could be used for transmitting information over serial communication channels, Internet of Things networks, and sensor networks, etc.

3. The evaluation of the proposed method for detecting and correcting multi-bit errors was carried out using the developed software model of the data transmission channel. The software model, developed in the Rust language, allowed us to evaluate the noise immunity of the communication channel under conditions of electromagnetic interference. Testing of the software model of the communication channel was carried out under an automated mode. According to the results of the evaluation, it can be stated that even with an eight-fold error in the transmission of 3 separate bytes, an information block with a length of 24 bytes, fatal errors do not occur.

Conflicts of interest

The authors declare that they have no conflicts of interest in relation to the current study, including financial, personal, authorship, or any other, that could affect the study, as well as the results reported in this paper.

Funding

The study was conducted without financial support.

Data availability

The manuscript has associated data in the data warehouse.

Use of artificial intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Shannon, C. E. (1948). A Mathematical Theory of Communication. Bell System Technical Journal, 27 (4), 623–656. https://doi.org/10.1002/j.1538-7305.1948.tb00917.x
2. Bortos, M., Tylyshchak, A., Khymynets, M. (2024). Extended binary Golay codes by a group algebra. Algebra and Discrete Mathematics, 38 (1), 23–33. https://doi.org/10.12958/adm2241
3. Bychkov, V. E., Mrachkovsky, O. D., Pravda, V. I. (2008). Correlation properties of Golay sequences. Visnyk NTUU KPI Seriya - Radiotekhnika Radioaparatobuduvannia, 37, 28–30. Available at: https://radap.kpi.ua/radiotechnique/article/view/552/0?articlesBySameAuthorPage=3
4. Shishkin, A. V. (2010). Sindromnyy metod formirovaniya cifrovyh vodyanyh znakov i steganograficheskoy peredachi s ispol'zovaniem dopolnitel'noy informacii o nositele. Izvestiya Vysshih Uchebnyh Zavedeniy. Radioelektronika, 53 (1), 12–19. https://doi.org/10.20535/s0021347010010024
5. Shiriaev, P. M. (2015). Comparison of the binary golay code with the algebro-geometric code. Prikladnaya diskretnaya matematika, 30 (4), 77–82. https://doi.org/10.17223/20710410/30/7
6. Khebbou, D., Chana, I., Ben-Azza, H. (2023). Decoding of the extended Golay code by the simplified successive-cancellation list decoder adapted to multi-kernel polar codes. TELKOMNIKA (Telecommunication Computing Electronics and Control), 21 (3), 477. https://doi.org/10.12928/telkomnika.v21i3.23360
7. Reed, I. S., Yin, X., Truong, T. K., Holmes, J. K. (1990). Decoding the (24,12,8) Golay code. IEE Proceedings E Computers and Digital Techniques, 137 (3), 202. https://doi.org/10.1049/ip-e.1990.0025
8. Bioglio, V., Land, I. (2018). Polar-Code Construction of Golay Codes. IEEE Communications Letters, 22 (3), 466–469. https://doi.org/10.1109/lcomm.2018.2793273
9. Honary, B., Markarian, G. (1993). New simple encoder and trellis decoder for Golay codes. Electronics Letters, 29 (25), 2170–2171. https://doi.org/10.1049/el:19931456
10. Reviriego, P., Liu, S., Xiao, L., Maestro, J. A. (2016). An Efficient Single and Double-Adjacent Error Correcting Parallel Decoder for the (24,12) Extended Golay Code. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 24 (4), 1603–1606. https://doi.org/10.1109/tvlsi.2015.2465846
11. Sokolovskyi, V., Zharikov, E., Telenyk, S. (2024). Development of the method of detecting and correcting data transmission errors in IoT systems for monitoring the state of objects. Eastern-European Journal of Enterprise Technologies, 1 (9 (127)), 22–33. https://doi.org/10.15587/1729-4061.2024.298476
12. Golay, M. E. (1949). Notes on Digital Coding. Proceedings of the IRE.
13. Sridevi, N., Jamal, K., Mannem, K. (2021). Implementation of Error Correction Techniques in Memory Applications. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC), 586–595. https://doi.org/10.1109/iccmc51019.2021.9418432
14. Saiz-Adalid, L.-J., Gil, P., Ruiz, J.-C., Gracia-Moran, J., Gil-Tomas, D., Baraza-Calvo, J.-C. (2016). Ultrafast Error Correction Codes for Double Error Detection/Correction. 2016 12th European Dependable Computing Conference (EDCC), 108–119. https://doi.org/10.1109/edcc.2016.28