Algorithm and Data Encoding/decoding Devices Based on Two-dimensional Modular Correction Codes

Vasyl Yatskiv[a], Taras Tsavolyk[a], Nataliya Yatskiv[a], Vasyl Koval[a], Stepan Ivasiev[a]

[a] *West Ukrainian National University, 11 Lvivska str., Ternopil, 46009, Ukraine*

Abstract

In the paper, an error detection and correction algorithm was developed on the basis of modular correcting codes and a two-dimensional scheme. The developed algorithm provides effective correction of error packets, is characterized by high corrective ability and low complexity of decoding device implementation, and accordingly can be applied to increase the reliability of data transmission in wireless sensor networks. Check symbols are calculated separately for rows and columns of the data matrix. Due to the fact that the same mutually prime coefficients are used to calculate checksums along the rows and columns of the matrix, it allows for detecting and correcting error packets that are in the same row or column.

The developed algorithm provides correction of error packets of maximum length $l = 3k - 2$ ($k -$ number of information symbols) when they are placed in two columns and one row or two rows and one column. Possible examples of placing the maximum number of distorted symbols in a data block and methods of their correction are given. The work of the coder for calculating check symbols and forming a data packet is described in the Verilog language and implemented on an FPGA by Altera. The structure of the decoder, which performs error detection based on the error correction syndrome check based on the solution of the system of modular equations, has been developed.

Keywords 1

Wireless sensor networks, modular corrective codes, residual number system, modular arithmetic.

1. Introduction

With the constant development and wide implementation of wireless technologies, improving the reliability of data transmission in wireless sensor networks (WSN) remains an important and urgent task [1]. In order to ensure the necessary level of reliability of data transmission in WSN, unlike wireless computer networks, it is necessary to take into account the limitations imposed by the low computing resources of nodes and the use of autonomous power [2, 3, 4]. In addition, additional complications are caused by the packet relay mode used in most WSN applications. Since WSN often uses methods of reducing traffic redundancy by compressing data and filtering correlated data, the requirements for the reliability of each packet transmission are increasing accordingly. Therefore, for the effective operation of WSN, it is necessary to ensure high reliability and energy efficiency of data transmission in different modes of network operation [16].

The main mechanism for improving the reliability of data transmission in WSN is the use of error control schemes. Their task is to ensure reliable communication in a wireless channel in which errors

IntelITSIS'2023: 4th International Workshop on Intelligent Information Technologies and Systems of Information Security, March 22–24, 2023, Khmelnytskyi, Ukraine
EMAIL: jazkiv@ukr.net (V. Yatskiv); calisto2292@ukr.net (T. Tsavolyk); jatskiv@ukr.net (N. Yatskiv); vko@wunu.edu.ua (V. Koval) ; stepan.ivasiev@gmail.com (S. Ivasiev)
ORCID: 0000-0001-9778-6625 (V. Yatskiv); 0000-0002-1136-5705 (T. Tsavolyk); 0000-0003-2421-4217 (N. Yatskiv) ; 0000-0003-4726-097X (V. Koval) ; 0000-0003-2243-5956 (S. Ivasiev)

occur due to interference, fading, and loss of bit synchronization. This leads to channel errors that affect the integrity of the packets forming the sensor nodes. The low power of WSN transmitters, which is connected with the use of autonomous power, also leads to an increase in errors [4, 5, 6].

Thus, in addition to methods of improving reliability at the physical level, which guarantees the reliability of transmission at the bit level, WSN also requires the application of error control schemes at the channel level to ensure reliability at the level of packet transmission.

The following requirements are put forward for correcting codes in WSN [7, 8, 9, 10]:

- the low complexity of encoding/decoding algorithms;
- low hardware requirements for the implementation of algorithms (microcontroller bit rate and clock frequency, memory volume);
- adaptive change in the number of check symbols when changing the channel parameters.

The conducted analysis showed that many studies prove the advantages of using Reed–Solomon codes in WSN [11, 12, 13]. However, the use of Reed-Solomon codes makes it impossible to use adaptive error control schemes. Because when increasing/decreasing the number of check characters, it is necessary to list all check characters. To choose an effective error control scheme and the type of correction code, it is also necessary to take into account the permissible error probability for a given WSN and the characteristics of the communication channel.

2. Related works

2.1. Modular corrective codes

Nowadays, there is considerable interest in the application of the residual number system (RNS) in new areas, such as increasing the reliability and security of cloud storage, homomorphic encryption, post-quantum cryptography, and others [14, 15, 20].

The residual number system belongs to the non-positional counting systems and is based on the concept of a remainder and the Chinese theorem on remainders. Any positive integer in RNS is presented as a set of the smallest positive remainders from dividing this number by the selected system of mutually prime numbers (modules) $p_1, p_2, \dots, p_n$ [14, 16, 21].

Data processing and transmission in RNS has a number of advantages due to independence, low bit rate and equality of balances, and the possibility of parallel execution of arithmetic operations. In addition to the above-mentioned advantages of RNS for data processing, effective RNS correction codes have also been developed that are capable of detecting and correcting error packets [17, 18, 19].

Modular correcting codes developed in [22, 23] belong to symbol codes characterized by low redundancy and allow efficient detection and correction of single symbol errors.

In modular correction codes, the data packet in the binary code to be transmitted is divided into k blocks of the same length (tetrads or bytes) [24]:

$$a_j^1 \dots a_3^1 a_2^1 a_1^1 a_0^1, a_j^2 \dots a_3^2 a_2^2 a_1^2 a_0^2, a_j^i \dots a_3^i a_2^i a_1^i a_0^i, \dots, a_j^k \dots a_3^k a_2^k a_1^k a_0^k,$$

where a^i – bit of data in binary code, $j = 4,8$.

At the same time, the value of the control bit is equal to

$$x_{k+1} = |v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x_i + \dots + v_k \cdot x_k|_P, \quad (1)$$

where v_i – coefficients, mutually prime with P; x_i– byte of data in a binary or decimal calculation system, moreover

$$x_i = a_7^i \dots a_3^i a_2^i a_1^i a_0^i = a_7 \cdot 2^7 + \dots + a_3 \cdot 2^3 + a_2 \cdot 2^2 + a_1 \cdot 2^1 + a_0 \cdot 2^0.$$

Suppose that an error occurred in one of the data blocks during the transmission (from 1 to 8 binary digits were distorted) and instead of a number X was received X'.

$$A' = (x_1, x_2, \dots, x_i', \dots, x_k, x_{k+1}).$$

The value of the control bit according to the received data is

$$x_{k+1}' = |v_1 \cdot x_1 + v_2 \cdot x_2 + \dots + v_i \cdot x_i' + \dots + v_k \cdot x_k|_P.$$

The minimum code distance of the modular correction code is

$$d_{min} = n - k + 1$$

where n – total code length, k – the number of information blocks. Accordingly, this correcting code detects t or a smaller number of errors under the condition that $d_{min} \geq t + 1$.

In [25] the possibility of constructing two-dimensional correction codes based on modular arithmetic is shown, which allows for detecting and correcting errors in two or more information symbols. However, the proposed method of correcting errors in two symbols based on the extended Euclid algorithm has a high time complexity, which depends on the value of the module P and increases with the gain in the number and bit rate of information symbols, as it requires finding all solutions of the Diophantine equation. Algorithms for correcting error packets also require additional research. Therefore, the development of a two-dimensional coding method based on a modular correcting code and error packet correction algorithms is an urgent scientific task.

3. Proposed solutions

3.1. Two-dimensional code based on modular corrective codes

A two-dimensional code $\gamma = \gamma_1 \times \gamma_2$ two codes γ_1 and γ_2 name of code, code words which are all two-dimensional tables with rows and which are code words γ_1 and columns which are code words either γ_2.

For building a two-dimensional code was chosen, modular corrective codes γ_1 and γ_2 [24, 25]. Moreover, the codes may be the same $\gamma_1 = \gamma_2$, and different $\gamma_1 \neq \gamma_2$.

In the modular corrective codes, the check symbols are calculated:

$$\mathrm{f_{jk}(x)} = \left(\sum_{\mathrm{i=1}}^{\mathrm{k}} \mathrm{x_{ij} v_i}\right) \mathrm{mod P} \tag{2}$$

$$f_{l\,i}(x) = \left(\sum_{j=1}^{l} x_{ji} v_j\right) mod\ P. \tag{3}$$

where x_{ji} – information symbols, v_i, v_j – coefficients (prime numbers), P – module (prime number).

The formation of the two-dimensional modular code is shown in Table 1. At the first, each l row is encoded with the γ_1 and then each k column is encoded with the γ_2.

Table 1

Two-dimensional modular code generation

	Information symbols	Check symbols calculated by rows
	$x_{11} \quad x_{12} \cdots x_{1i} \cdots x_{1k}$	$f_{1\,k}(x)$
	$x_{21} \quad x_{22} \cdots x_{2i} \cdots x_{2k}$	$f_{2\,k}(x)$
	… … … … … …	…
	$x_{j1} \quad x_{j2} \cdots x_{ji} \cdots x_{jk}$	$f_{j\,k}(x)$
	… … … … … …	…
	$x_{l1} \quad x_{l2} \cdots x_{li} \cdots x_{lk}$	$f_{l\,k}(x)$
Check symbols calculated by columns	$f_{l1}(x)\, f_{l2}(x) \cdots f_{li}(x) \cdots f_{lk}(x)$	

The coding process is as follows (Figure 1): data for k symbols are sent to the internal encoder γ_1, where the verification symbols are calculated line by line according to formula (2) (Table 1). From the output of the internal coder γ_1 the code words enter the buffer. From the buffer, the code words are read column by column and enter the external encoder γ_2, where the check symbols are calculated column by column according to formula (3).

Error detection is based on syndrome analysis, which is calculated as the difference between check symbols calculated in the encoder and decoder [26-30].

Syndrome calculation by rows:

$$\delta_i = \begin{pmatrix} \left(f'_{1k}(x) - f_{1k}(x)\right) mod P \\ \left(f'_{2k}(x) - f_{2k}(x)\right) mod\ P \\ \cdots\ \cdots\ \cdots \\ \left(f'_{jk}(x) - f_{jk}(x)\right) mod\ P \\ \cdots\ \cdots\ \cdots \\ \left(f'_{lk}(x) - f_{lk}(x)\right) mod\ P \end{pmatrix}.$$

Syndrome calculation by columns:

$$\delta_j = \begin{pmatrix} \left(f'_{l1}(x) - f_{l1}(x)\right) mod\ P \\ \left(f'_{l2}(x) - f_{l2}(x)\right) mod\ P \\ \cdots\ \cdots\ \cdots \\ \left(f'_{li}(x) - f_{li}(x)\right) mod\ P \\ \cdots\ \cdots\ \cdots \\ \left(f'_{lk}(x) - f_{lk}(x)\right) mod P \end{pmatrix},$$

where, $f'_{jk}(x)$, $f'_{li}(x)$ – check symbols are calculated in the decoder from the received data.

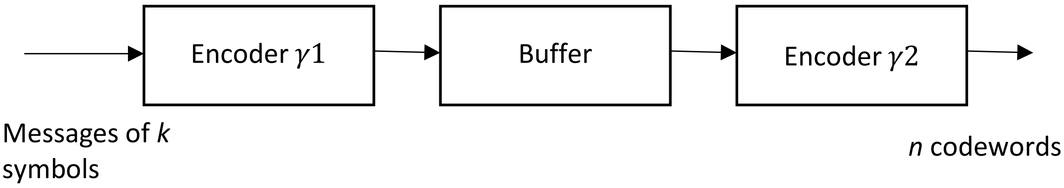

Figure 1: Block diagram of the encoder

If $\delta_i = 0$, $\delta_j = 0$ – errors do not exist. Else if $\delta_i \neq 0$, $\delta_j \neq 0$ – an error exists.

The position of the distorted symbol is determined by the value of the syndromes, which are not equal to zero at the intersection of the row and the column.

To correct an error in one symbol, it is necessary to solve the equation:

$$v_i(x'_i - x_i) = \delta_i(mod\ P). \quad (4)$$

If syndromes $\delta_i \neq 0$, $\delta_j \neq 0$ are not equal to zero in two or more rows or columns, this means that errors occurred in two or more characters. Moreover, if the syndrome $\delta_j \neq 0$ in two columns – it means that two characters in the string are garbled. The positions of non-zero characters correspond to non-zero syndrome values in the column.

Error correction in two information symbols. Analysis of the syndrome by rows and columns allows you to identify the positions of the distorted characters. It is necessary to solve the equation to correct them:

$$v_i(x'_i - x_i) + v_t(x'_t - x_t) = \delta_{ij}(mod\ P); \quad (5)$$
$$v_i x'_i - v_i x_i + v_t x'_t - v_t x_t = \delta_{ij}(mod\ P);$$
$$-v_i x_i - v_t x_t = (\delta_{ij} - v_i x'_i - v_t x'_t)(mod\ P);$$
$$v_i x_i + v_t x_t = (v_i x'_i + v_t x'_t - \delta_{ij})(mod\ P),$$

where i, t – position of garbled symbols.

Let’s change the right side on c:

$$v_i x_i + v_t x_t = \mathrm{c}(mod\ P). \quad (6)$$

It is known if $\gcd(v_i, v_t) = 1$, that is v_i and v_t mutually prime numbers, then equation (6) has a solution in integers.

Let's solve equation (6) with respect to one of the unknowns. Let's express x_i via x_t:

$$x_i = \left(\frac{c - v_t x_t}{v_i}\right)(mod\ P). \quad (7)$$

As $0 \leq x_i < 2^m\ and\ 0 \leq x_t < 2^m$ then by substituting in equation (7) x_t value from 0 to 2^m we find a set of solutions x_i, which only one of the solutions found will be an integer.

With the minimum code distance of codes γ_1 and γ_1 equal d_1 and d_2 in accordance then the minimum code distance of a two-dimensional code γ equal $d_1 \times d_2$. It follows that if codes γ_1 and γ_1

fixed t_1 and t_2 errors, respectively, then the two-dimensional code corrects $2t_1t_2 + t_1 + t_2 + 1$ errors.

Considering that the modular code can correct two errors, if their location is known, then the maximum length of the error packet is $b = 3k - 2$ symbols when the distorted characters are placed in two rows and one column, or in two columns and one row (Figure 2).

*	*	*	*	*	*	*	*
*	*	*	*	*	*	*	*
				*			
				*			
				*			
				*			
				*			
				*			

Figure 2: An example of the maximum length of error packet location

However, this code will not be able to correct a package of errors of six characters, which are located at the intersection of three rows and three columns (Figure 3).

				*	*	*	
				*	*	*	
				*	*	*	

Figure 3: An example of error packet location

A two-dimensional code based on modular correcting codes has been developed, which provides effective correction of error packets of maximum length $b = 3k - 2$ symbols, provided that the distorted characters are placed in two rows and one column, or in two columns and one row.

3.2. An algorithm for the detection and correction of error packages based on modular correction codes

Based on the method of correcting error packets, an algorithm for detecting and correcting error packages using modular correcting codes and a two-dimensional scheme has been developed.

The algorithm works as follows (Figure 4): in block 1, we enter mutually simple coefficients v_i, the value of the module *P*, and information symbols x_i. Then, in block 2 and block 3, syndromes are calculated by rows and columns by formulas (2, 3).

If the value of the syndrome (block 4) $\delta_i = 0,\ \delta_j = 0$, then no error is detected, and if $\delta_i \neq 0,\ \delta_j \neq 0$ or if in at least one of the syndromes $\delta_i = 0,\ \delta_j \neq 0$, then the error is detected and we proceed to its correction.

To correct the error (block 5) in one symbol for each line, we solve the equation

$$v_i(x_i' - x_i) = \delta_i(mod\ P),$$

where v_i – mutually prime coefficients, x_i' – received message and x_i – the message that was already sent.

After finding the solutions of this equation for all rows, we check the value of the syndrome (block 6) when: $\delta_i \neq 0,\ \delta_j = 0$ – there are errors in the columns, which we correct according to the formula (4) (block 8), and in block 9, we substitute the values of the solutions into the matrix; when $\delta_i = 0,\ \delta_j \neq 0$ – there are errors in the lines, which are corrected according to the formula (5) (block 7).

In block 10, we check, if the value of the syndrome $\delta_i = 0$, $\delta_j = 0$, then the errors are corrected, and if the value of the syndrome $\delta_i \neq 0$, $\delta_j \neq 0$ – more errors than the algorithm can correct (block 11).

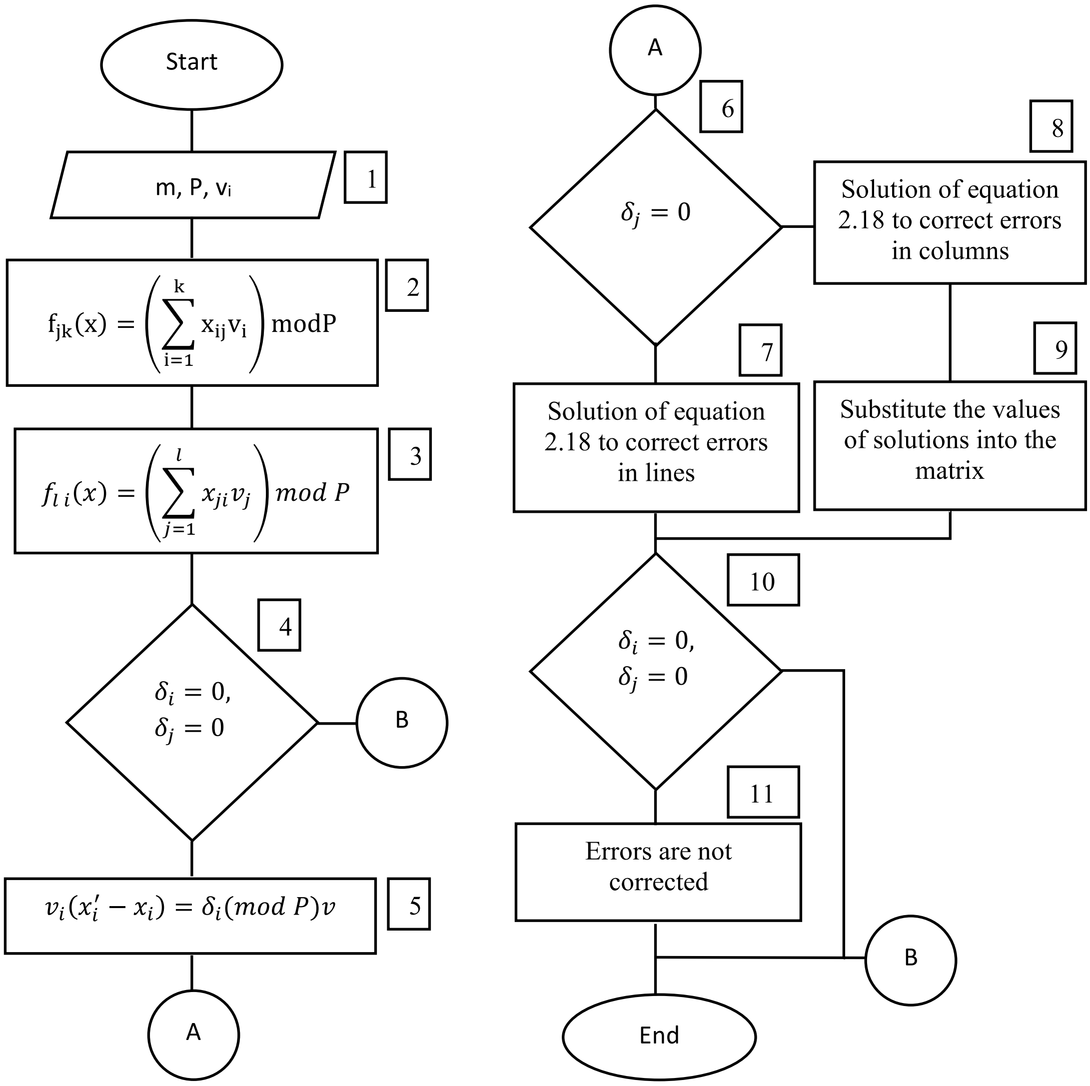

Figure 4: A block diagram of the algorithm for error packet correction and detection

Let's see the example. Let's start to build a two-dimensional code based on a modular correction code with $k = 8$, $l = 8$, bit rate of information symbols $m = 4$, module $P = 1019$, and coefficients: $v_1 = 17$, $v_2 = 19$, $v_3 = 23$, $v_4 = 29$, $v_{15} = 31$, $v_6 = 37$, $v_7 = 43$, $v_8 = 47$ (Table2).

Table 2

A two-dimensional modular code at the output of the transmitter

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Check symbols calculated by rows
10	7	13	10	2	7	9	2	675
13	4	11	3	5	6	15	7	969
5	9	6	6	9	6	9	15	123
3	4	9	14	10	5	7	13	109
11	6	12	9	4	7	10	0	632
7	8	4	11	11	9	5	5	787
9	11	13	1	14	7	0	9	787

	12	14	15	8	6	11	1	3	805
Check symbols calculated by columns	132	93	575	857	3	865	371	583	

Consider the error correction algorithm for the maximum number of distorted symbols (Table 3).

Table 3
A two-dimensional modular code at the input of the receiver

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	Check symbols calculated by rows
	2*	8*	12*	9*	3*	5*	11*	4*	275
	11*	5*	10*	4*	7*	8*	7*	9*	846
	5	9	6	8*	9	6	9	15	181
	3	4	9	12*	10	5	7	13	51
	11	6	12	11*	4	7	10	0	690
	7	8	4	13*	11	9	5	5	845
	9	11	13	3*	14	7	0	9	845
	12	14	15	5*	6	11	1	3	718
Check symbols calculated by columns	977	129	539	928	58	869	253	655	

* – distorted symbols.

The values of the syndromes calculated by the rows and columns of the data matrix are given in Table 4.

Table 4
Syndrome values in rows and columns

Syndrome by rows, δ_i	Syndrome by columns, δ_j
619	845
896	36
58	983
961	71
58	55
58	4
58	901
932	72

Since the syndromes are not equal to zero, the distorted symbols are in all rows and columns of the data matrix (see Table 4).

The error detection and correction algorithm consists of the following steps.

1. We check whether there are lines with one error. To do this, we successively solve equation (4) for each line. The solution results are given in Table 5.

As we can see from Table 5, the solutions found, integers in the given range [0–15], , indicate that the symbol x_3 is distorted in lines 3 – 8.

2. By substituting the calculated values in Table 3, we will get a new table of syndromes (Table 6).

After analyzing Table 6 it shows that there are errors in all columns but only in two rows. That is, all symbols in the first and second lines are distorted. Correction of errors in two symbols is carried out using equation (5).

Table 5

The results of the solution to equation 2

№ of row	Information symbols							
	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
1	25,529	29,052	29,391	22,793	15,903	15,810	20,302	12,511
2	18,235	11,473	15,347	8,241	10,967	11,324	9,860	11,617
3	1,588	5,947	3,478	6	7,129	4,432	7,651	13,765
4	6,411	7,052	11,521	14	11,870	6,567	8,348	14,234
5	7,588	2,947	9,478	9	2,129	5,432	8,651	20,447
6	3,588	4,947	1,478	11	9,129	7,432	3,651	3,766
7	5,588	7,947	10,478	1	12,129	5,432	22,348	7,765
8	17,117	18,578	18,782	8	8,806	13,351	3,023	4,851

Table 6

A table of syndromes after partial error detection

Syndrome calculated by rows, δ_i	Syndrome calculated by columns, δ_j
987	845
896	36
0	983
0	2
0	55
0	4
0	901
0	72

To correct errors in the symbols placed in the first column, if we substitute numerical values in equation (5), we will get:

$$17(2 - x_1) + 19(11 - x_2) = 845(mod\ 1019);$$
$$17x_1 + 19x_2 = 417(mod\ 1019); \tag{8}$$

After solving equation (8) for each column of the matrix, we find the correct values of the information symbols (Table 7).

As can we can see from Table 7, the found solutions of equation (8) unambiguously ensure the correction of errors in 22 symbols of the incoming message. Therefore, the developed algorithm allows correcting error packages with a maximum length of $b = 3k - 2$ symbols, when the distorted characters are placed in two rows and one column, or in two columns and one row. At the same time, it provides the speed of the code $R_1 = \frac{k_1}{n_1} = 0.62$, at $k = 8$, $m_1 = 4$, P = 1021, $k_1 = k * k * m_1 = 256$, $r_1 = 2 * k *]log_2 P[= 160$, $n_1 = k_1 + r_1 = 416$ and $R_2 = \frac{k_2}{n_2} = 0.67$ at $m_2 = 8$, $k_2 = 512$, $n_2 = 768$. The approximate redundancy of the code is equal to: $r = (1 - R_1) * 100\% = 38$ %.

An algorithm for detecting and correcting random single errors and error packets with a maximum length of $b = 3k - 2$, at a code speed of $R = 0.67$, has been developed.

Considering the low complexity of the decoding algorithm implementation, this code is planned to be used to increase the reliability of data transmission in wireless sensor networks.

Table 7
Solutions to equation 8

	Information symbols							
x_2	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}
0	24,53	11,47	25,29	13,35	7,59	13,71	25,76	9,82
1	23,41	10,35	24,18	12,24	6,47	12,59	24,65	8,71
2	22,29	9,24	23,06	11,12	5,35	11,47	23,53	7,59
3	21,18	8,12	21,94	**10,00**	4,24	10,35	22,41	6,47
4	20,06	**7,00**	20,82	8,88	3,12	9,24	21,29	5,35
5	18,94	5,88	19,71	7,76	**2,00**	8,12	20,18	4,24
6	17,82	4,76	18,59	6,65	0,88	**7,00**	19,06	3,12
7	16,71	3,65	17,47	5,53	-0,24	5,88	17,94	**2,00**
8	15,59	2,53	16,35	4,41	-1,35	4,76	16,82	0,88
9	14,47	1,41	15,24	3,29	-2,47	3,65	15,71	-0,24
10	13,35	0,29	14,12	2,18	-3,59	2,53	14,59	-1,35
11	12,24	-0,82	**13,00**	1,06	-4,71	1,41	13,47	-2,47
12	11,12	-1,94	11,88	-0,06	-5,82	0,29	12,35	-3,59
13	**10,00**	-3,06	10,76	-1,18	-6,94	-0,82	11,24	-4,71
14	8,88	-4,18	9,65	-2,29	-8,06	-1,94	10,12	-5,82
15	7,76	-5,29	8,53	-3,41	-9,18	-3,06	**9,00**	-6,94

3.3. An interference-resistant data encoding/decoding device based on two-dimensional modular correction codes

3.3.1. Encoder implementation

The coding process is as follows (Figure 5). A message with k symbols is received at the encoder input. The coder calculates the check symbol and writes the message with the check symbol to the buffer. The process is repeated until k messages are received. A data array is formed from the received messages, from which verification symbols are calculated by the columns of the array.

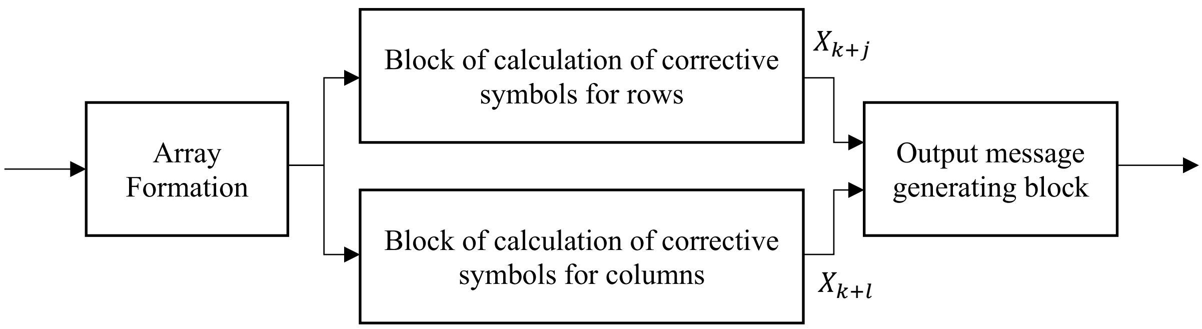

Figure 5: The structure of coder (X_i – incoming message)

After the encoder completes the calculation of check symbols by columns, a new array of the received message with calculated check symbols by rows and columns is formed at the output. The structure of the message at the output of the encoder is presented in Figure 6.

An example of interference-resistant coding based on modular codes and a two-dimensional scheme. Incoming messages formed in the form of a two-dimensional array are shown in Table 8.

To implement the coder, we will choose mutually simple coefficients v_i: $v_1 = 211, v_2 = 257, v_3 = 263, v_4 = 269, v_5 = 271, v_6 = 277, v_7 = 283, v_8 = 288$ and module $P = 131297$.

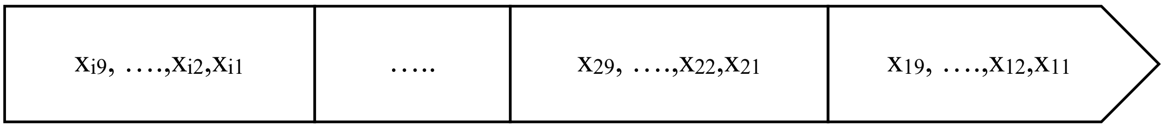

Figure 6: The structure of the incoming message

Table 8
Message representation in a form of an array

	x1	x2	x3	x4	x5	x6	x7	x8
y1	137	35	61	163	2	36	44	60
y2	**71**	102	118	108	205	214	101	39
y3	68	177	77	63	204	236	100	226
y4	211	212	168	240	15	181	46	99
y5	89	109	62	206	133	225	58	19
y6	75	10	13	186	161	210	84	190
y7	182	165	23	113	254	121	110	245
y8	9	41	201	1	33	92	7	103

We calculate the verification symbols by rows according to formula (2) and by columns according to formula (3) (Table 9).

Table 9
Message with check symbols calculated in rows

	x1	x2	x3	x4	x5	x6	x7	x8	x_{k+j}
y1	137	35	61	163	2	36	44	60	121046
y2	71	102	118	108	205	214	101	39	113914
y3	68	177	77	63	204	236	100	226	115208
y4	211	212	168	240	15	181	46	99	13115
y5	89	109	62	206	133	225	58	19	102511
y6	75	10	13	186	161	210	84	190	66638
y7	182	165	23	113	254	121	110	245	119951
y8	9	41	201	1	33	92	7	103	102264
x_{k+l}	89718	96919	62369	20751	10678	91093	14976	3388	

The operation of the coder is described in the Verilog language and synthesized on a FPGA by Altera. The functional diagram of the encoder is presented in Figure 7.

3.3.2. Decoder implementation

The decoder calculates check symbols on the received data and compares them with those that were calculated by the encoder and sent. To correct errors in the data packet, a table of syndromes is formed in the decoder by rows and columns.

The correction of single errors is based on the comparison of the calculated syndrome with the table of syndromes, if the values match, the error has been detected and can be corrected. After that, the table of syndromes is calculated by rows and columns, if the value of the syndrome is zero, then the errors have been corrected.

If the values in the table of syndromes are not equal to zero, it means that errors have been detected and they are placed in two lines. To correct such errors, equation (3) must be solved for each row/column (multiple error correction block) (Figure 8).

The developed device of interference-resistant data decoding based on two-dimensional modular correction codes makes it possible to detect and correct a large number of errors in a data packet.

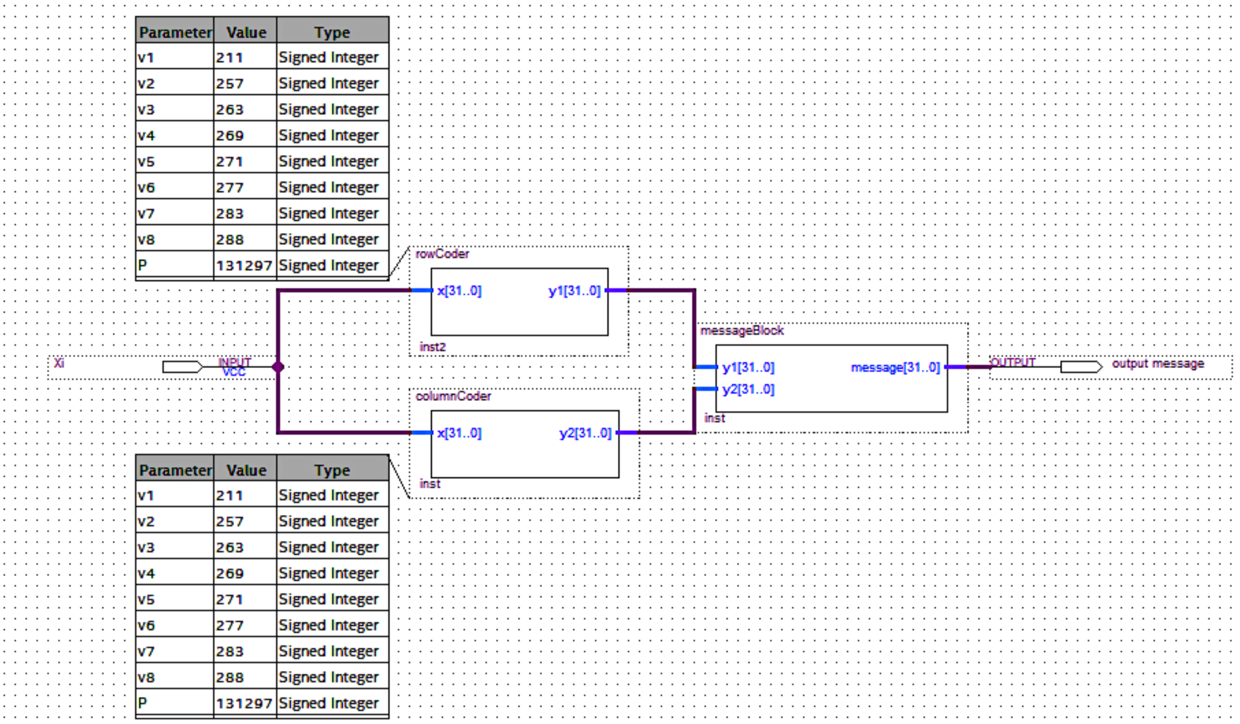

Figure 7: Functional schema of the coder

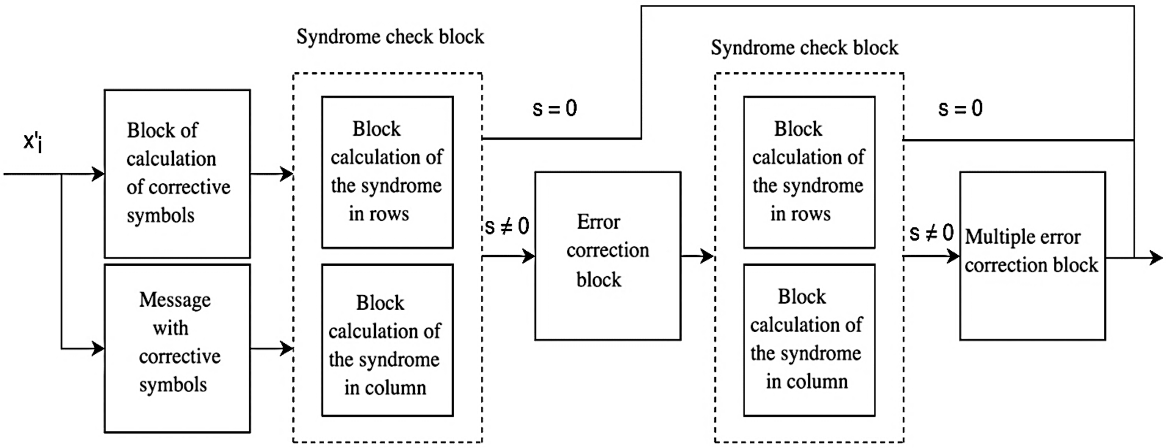

Figure 8: Decoder structure

4. Conclusion

A two-dimensional approach to error correction based on modular correction codes is proposed, which provides effective correction of error packets, is characterized by high correction ability and low complexity of decoding algorithm implementation, and accordingly can be applied to increase the reliability of data transmission in wireless sensor networks.

An algorithm for detecting and correcting error packets based on modular correcting codes and a two-dimensional scheme has been developed. This algorithm provides detection and correction of packets with the k length of errors that are in one row or column of the data matrix of size $k \times k$, or $2 \cdot k$ errors that are in two rows or two columns of the data matrix. Moreover, the algorithm provides correction of the maximum length of error packets that can be $l = 3k - 2$ at speed of code $R = 0.67$, if the distorted characters are in two rows and one column or in two columns and one row.

A structure was developed and an encoder for calculating check symbols and forming a data packet based on two-dimensional modular correction codes was developed and implemented on an

FPGA. To be more precise, the limitations of the conducted research should be attributed to the low bit rate message size of only 8 bits, as well as the fact that only an 8x8 matrix size was considered.

Further research is required on algorithms for finding mutually prime coefficients v_i with a minimum value of the check module P for encoding messages with a bit depth of 8-16.

5. References

[1] M. Bettayeb, S. Ghunaim, N. Mohamed, and Q. Nasir. "Error correction codes in wireless sensor networks: a systematic literature review". In 2019 International Conference on Communications, Signal Processing, and their Applications (ICCSPA). IEEE. (2019): 1-6 doi: 10.1109/ICCSPA.2019.8713725.

[2] Yakubu Abdul-Wahab Nawusu, Alhassan Abdul-Barik, Salifu Abdul-Mumin. Residue Number System-Based Approach to Minimise Energy Consumption in Wireless Sensor Networks. Asian Journal of Research in Computer Science. 14(4), 2022, 46-65. doi:10.9734/ajrcos/2022/v14i4291

[3] S. M. Chowdhury, A. Hossain. Different energy saving schemes in wireless sensor networks: A survey. Wireless Personal Communications, 114(3), 2020, pp. 2043-2062. doi:10.1007/s11277-020-07461-5.

[4] I. Daanoune, A. Baghdad. Reliable and energy-efficient transmission scheme based on error correction codes and clustered routing protocol for WSN. In 2022 8th International Conference on Optimization and Applications (ICOA). 2022, pp. 1-6. IEEE. doi: 10.1109/ICOA55659.2022.9934530.

[5] J. Jana, S. Tripathi, J. Samanta, J. Bhaumik, and S. Barman. An Approach to Reduce Power Consumption and Delay of Single Error Correction Codes in WSNs for IoT Applications. In Computers and Devices for Communication: Proceedings of CODEC 2019, pp. 196-203. doi:10.1007/978-981-15-8366-7_27.

[6] S. Kaur, D. B. Kuttan, and N. Mittal. An Energy-saving Approach for Error control Codes in Wireless Sensor Networks. In 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC) (2021): 313-316. IEEE. doi:10.1109/ICOSEC51865.2021.9591677.

[7] R. Kadel, K. Paudel, D. B. Guruge, and S. J. Halder. Opportunities and challenges for error control schemes for wireless sensor networks: a review. Electronics, 9(3), (2020): 504. doi:10.3390/electronics9030504.

[8] M. A. Magzoub, A. Abd Aziz, M. A. Salem, H. Ab Ghani, A. A. Aziz, and A. Mahmud. Physical layer security and energy efficiency over different error correcting codes in wireless sensor networks. International Journal of Electrical and Computer Engineering (IJECE), 10(6), 2020, pp. 6673-6681. doi:10.11591/ijece.v10i6.pp6673-6681.

[9] S. Razali, K. Mamat, and N. S. K. Bashah. Multiple error correction towards optimisation of energy in sensor network. Indonesian Journal of Electrical Engineering and Computer Science, 13(3), 2019, pp. 1208-1220. doi:10.11591/ijeecs.v13.i3.

[10] S. Tripathi, J. Jana, J. Samanta, and J. Bhaumik. Fast and power efficient sec-ded and sec-ded-daec codes in IoT based wireless sensor networks. In TENCON 2019-2019 IEEE Region 10 Conference (TENCON) (2019): 540-545. IEEE. doi:10.1109/TENCON.2019.8929354.

[11] R. Ma, L. Xing, and Y. Wang. Performance Analysis of Reed-Solomon Codes for Effective Use in Survivable Wireless Sensor Networks. International Journal of Mathematical, Engineering and Management Sciences, 5(1), (2020): 13. doi:10.33889/IJMEMS.2020.5.1.002.

[12] Eric B. Olsen, "Continuous Error Detection and Correction of Arithmetic in a Complement RRNS", 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, pp.0009-0017. doi:10.1109/CCWC47524.2020.9031218.

[13] M. Yigit, P. S. Boluk, and V. C. Gungor. A new efficient error control algorithm for wireless sensor networks in smart grid. Computer standards & interfaces, 63, 2019, pp. 27-42. doi:10.1016/j.csi.2018.11.006.

[14] T. Plantard. Efficient word size modular arithmetic. IEEE Transactions on Emerging Topics in Computing, 9(3), 2021, pp. 1506-1518. doi:10.1109/TETC.2021.3073475.

[15] D. Schoinianakis. Residue arithmetic systems in cryptography: a survey on modern security applications. Journal of Cryptographic Engineering, 10(3), 2020, pp. 249-267. doi:10.1007/s13389-020-00231-w.
[16] Yi Huang, Chunhua Zhao, Baoping Tang, Yaowen Yang, Hao Fu, "Sparse Random Reconstruction of Data Loss With Low Redundancy in Wireless Sensor Networks for Mechanical Vibration Monitoring", IEEE Sensors Journal, vol.22, no.21, (2022): 20328-20335. doi:10.1109/JSEN.2022.3209330.
[17] R. Ye, A. Boukerche, H. Wang, X. Zhou, and B. Yan. RESIDENT: a reliable residue number system-based data transmission mechanism for wireless sensor networks. Wireless Networks, 24, 2018, pp. 597-610. doi:10.1007/s11276-016-1357-1.
[18] PAN Agbedemnab, EY Baagyere, MI Daabo. Single and multiple error detection and correction using redundant residue number system for cryptographic and stenographic schemes. Asian Journal of Research in Computer Science. 4(4), 2020, pp. 1-14. doi:10.9734/ajrcos/2019/v4i430123.
[19] V. Krasnobaev, V.Popenko, T. Kuznetsova, K. Kuznetsova, "Examples of Usage of Method of Data Errors Correction which are Presented by the Residual Classes", 2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT), 2019, pp.45-50. doi:110.1109/ATIT49449.2019.9030512.
[20] Yaw Afriyie, "A Novel Exploitation of Errors in RRNS Architecture", 2019 International Conference on Mechatronics, Remote Sensing, Information Systems and Industrial Information Technologies (ICMRSISIIT), vol.1, (2019): 1-8. doi:10.1109/ICMRSISIIT46373.2020.9405850.
[21] Eswar Deep Pitchika, Shivang Bharadwaj, "Fast Base Extension using Single Redundant Modulus in a Residue Number System", 2019 International Conference on Power Electronics, Control and Automation (ICPECA), (2019): 1-5. doi:10.1109/ICPECA47973.2019.8975450.
[22] J. Chen, V. Yatskiv, A. Sachenko, and J. Su. Wireless sensor networks based on modular arithmetic. Radioelectronics And Communications Systems, 60 (5), 2017, pp. 215-224. doi:10.3103/S073527271705003X.
[23] V. Yatskiv, T. Tsavolyk. Improvement of Data Transmission Reliability in Wireless Sensor Networks on The Basis of Residue Number System Correcting Codes Using the Special Module System. Conference Proceedings, IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON) Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine, (2017): 890-893. doi:10.1109/UKRCON.2017.8100376.
[24] V. Yatskiv, A. Sachenko, N. Yatskiv, P. Bykovyy, and A. Segin. Compression and Transfer of Images in Wireless Sensor Networks Using the Transformation of Residue Number System. In 2019 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) (Vol. 2). (2019): 1111-1114. doi:10.1109/IDAACS.2019.8924372.
[25] V. Yatskiv, T. Tsavolyk, N. Yatskiv. Burst error-correcting codes based on modular correcting codes. Conference Proceedings of 14 th International Conference Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET-2018) (Lviv-Slavske, Ukraine February 20-24 2018). Lviv. 2018, pp. 389-392. doi:10.1109/TCSET.2018.8336388.
[26] Chen, Z.; Zhao, Y.; Lu, J.; Liang, B.; Chen, X.; Li, C. TECED: A Two-Dimensional Error-Correction Codes Based Energy-Efficiency SRAM Design. Electronics 2022, 11, 1638. https://doi.org/10.3390/electronics11101638
[27] Sergii Lysenko, Kira Bobrovnikova, Roman Shchuka, Oleg Savenko. A Cyberattacks Detection Technique Based on Evolutionary Algorithms. 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), 2020. Vol.1. pp. 127-132.
[28] Antony, B.; Divya, S. Modified Hamming Codes with Double Adjacent Error Correction along with Enhanced Adjacent Error Detection. Int. J. Innov. Res. Comput. Commun. Eng. 2015, 3, 7706–7713.
[29] Chen, X.; Lu, Z.; Lei, Y.; Wang, Y.; Chen, S. Multi-bit transient fault control for NoC links using 2D fault coding method. In Proceedings of the 2016 Tenth IEEE/ACM International Symposium on Networks-on-Chip (NOCS), Nara, Japan, 31 August–2 September 2016; pp. 1–8.
[30] Rostand, N.; Hubert, G.; Martinie, S. SEU Prediction for Very Integrated Circuits based on Advanced Physical Considerations. In Proceedings of the 2019 19th European Conference on Radiation and Its Effects on Components and Systems (RADECS), Montpellier, France, 16–20 September 2019; pp. 1–