Design and realization of (7,4) Hamming code channel encoder trainer using Arduino Mega 2560

N Karostiani and A B Pantjawati*
Departemen Pendidikan Teknik Elektro, Universitas Pendidikan Indonesia, Jl. Dr. Setiabudhi 207, Bandung 40154, Indonesia

*arjunib@upi.edu

Abstract. Channel encoder is a part of transmitter block in a digital communication system which is capable of protecting information data from errors by adding parity bits from bits of information. The channel encoder receives bits of information from the source encoder and produces a new code called the codeword. Hamming code is one of the coding mechanisms used to protect information data from errors, with the ability to correct an error. This study explains the design of (7.4) Hamming encoder code channel trainer by using Arduino Mega 2560. The purpose of this research is producing a product that synchronizes a theory with the realization of how the channel encoder blocks work. Important parameters in the encoder channel are information bits, matrix generator, and parity bits. The form of the matrix generator affects the parity bits and the generated codeword. Therefore, the trainer is made and designed that can be used in the form of different matrix generator and qualify of (7.4) Hamming code channel encoder. This trainer is made in two stages: the simulation stage using Proteus 8 Professional and the hardware manufacturing stage. The results showed that the trainers worked in accordance with the theory (7.4) Hamming code channel encoder.

1. Introduction
The digital communication system is a system of information exchange from a transmitter (Tx) to a receiver (Rx) through transmission channels using certain procedures, and information data transmitted in the form of digital code as a discrete signal consists of bits 0 and 1. Error is possible to occur in data during transmission from the transmitter to the receiver due to noise factors or environmental disturbances along the transmission channel [1]. Errors will damage the shape of the data and cause differences between the data sent by the transmitter and the data received by the receiver. Digital communication systems must have the ability to control information data sent by the transmitter or received by the receiver to resolve errors. Coding data before transmission is a method used by the transmitter to protect data from errors. The part of the transmitter that has this function is the channel encoder [2].

Channel encoder is a part that forms error control coding as a data protector and for detecting and correcting errors in data received by receivers. The channel encoder converts information data into a codeword, by adding parity bits to the information bits to ensure that the transmitted bits do not change in value after arriving at the receiver [3].

Hamming code is one of the coding mechanisms used by channel encoders, which are easy to use and have the ability to correct the one-bit error and include the linear type of block code. Linear block
code is a code that has a structure \((n, k)\) where \(n\) represents the codeword bit length \((\bar{v})\), \(k\) information bit length \((\bar{u})\), \((n-k)\) the length of the parity bit \((\gamma)\) and \(2^k\) is the number of possible codeword combinations formed. The systemic code of the generated codeword is shown in Figure 1.

\[
\begin{array}{|c|c|}
\hline
n-k & k \\
\hline
\text{Bit parity} & \text{Bit information} \\
\hline
\end{array}
\]

**Figure 1.** Codeword systematics.

The codeword is generated by multiplying the number of bits of information \((\bar{u})\) with a dimension of \(1 \times k\) with a pixel dimension \((G)\) generator whose lines form a linear code \((n, k)\), where the matrix generator and bits of information are 0 and 1 bits. The results of the codeword are indicated by equations \(\bar{v} = \bar{u} \cdot G\) The parameters of the coding are listed in Table 1 [4].

**Table 1.** Dimensions of basic coding parameters.

| No. | Parameters | Dimension |
|-----|------------|-----------|
| 1.  | Information bits \((\bar{u})\) | \(\bar{u} = (u_0, u_1, \ldots, u_{k-1})\) |
|     |            | \(G = \left[ \begin{array}{c|c|c|c|c|c|c|c|}
|     |            | \hline
|     |            | \hline
| 2.  | Matrix Generator \((G)\) | \(p_{0,0} p_{1,0} \ldots p_{0,n-k-1} 1 0 0 \ldots 0\) |
|     |            | \(p_{1,0} p_{1,0} \ldots p_{1,n-k-1} 0 1 0 \ldots 0\) |
|     |            | \vdots \hline
|     |            | \(p_{k-1,0} p_{k-1,1} \ldots p_{k-1,n-k-1} 0 0 0 \ldots 1\) |
|     |            | \hline
|     |            | \left[ \begin{array}{c|c|c|c|c|c|c|c|}
|     |            | \hline
|     |            | \hline
| 3.  | Parity bits \((\gamma)\) | \(\gamma_j = \sum_{n=0}^{k-1} p_{n,j}\) |
|     |            | \hline
| 4.  | Codeword \((\bar{v})\) | \(\bar{v} = (v_0, v_1, \ldots, v_{n-k-1}, v_{n-k}, \ldots, v_{n-1})\) |
|     |            | \(= (\gamma_0, \gamma_1, \ldots, \gamma_{n-k-1}, u_0, u_1, \ldots, u_{k-1})\) |

In this study, the author made a design \((7,4)\) Hamming code channel encoder trainer using Arduino Mega 2560. The trainer designed to do codeword formation on the channel encoder using different matrix generators. The study was conducted by comparing the results of experiments using two forms of matrix generators. Usman, et al. Conducted a similar study entitled Design of \((7,4)\) Hamming Encoder and Decoder using VHDL (Very High integrated circuit hardware Description Language) [5].

2. **Research methods**

The study was conducted using an experimental methodology. The \((7,4)\) Hamming code channel encoder trainer is made using Arduino Mega 2560 through three stages, that is the stage of design, simulation and fabrication. The flow chart of the research procedure is shown in Figure 2.
2.1. Trainer design

The trainer design stage is the first stage of trainer production which consists of determining specifications, designing the work system of the trainer and designing a trainer circuit scheme based on the reference design. The channel encoder trainer that is designed has the specifications of having 4 information bits \( k \), 3 parity bits \( n - k \), 7 bits of the codeword \( n \) and 16 possible combinations of codewords \( 2^7 \). The design of the trainer system is shown in the following Figure 3.

![Flow chart of research procedures](image-url)

Figure 2. Flow chart of research procedures.
The trainer can be designed using a different matrix generator. In this study, two-sample matrix generators were used to show that the trainer is work properly. The type of matrix generator used affects the parity bit algorithm. The parity bit algorithm based on two matrix generator samples is as follows:

2.1.1. Sample 1. The parameters used in sample 1 are listed in Table 2.

**Table 2. Parameters for sample 1.**

| Information Bits | Matrix Generator | Parity Bit Algorithm |
|------------------|------------------|----------------------|
| \( \bar{u} = u_0 \ u_1 \ u_2 \ u_3 \) | \( G = \begin{bmatrix} 101 & 1000 \\ 011 & 0100 \\ 110 & 0010 \\ 111 & 0001 \end{bmatrix} \) | \( \gamma_0 \ \gamma_1 \ \gamma_2 = \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{pmatrix} \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \) |
|                  |                  | \( \gamma_0 = u_0 + u_2 + u_3 \) |
|                  |                  | \( \gamma_1 = u_1 + u_2 + u_3 \) |
|                  |                  | \( \gamma_2 = u_0 + u_1 + u_3 \) |

2.1.2. Sample 2. The parameters used in sample 2 are listed in Table 3.

**Table 3. Parameters for sample 2.**

| Information Bits | Matrix Generator | Parity Bit Algorithm |
|------------------|------------------|----------------------|
| \( \bar{u} = u_0 \ u_1 \ u_2 \ u_3 \) | \( G = \begin{bmatrix} 110 & 1000 \\ 101 & 0100 \\ 011 & 0010 \\ 111 & 0001 \end{bmatrix} \) | \( \gamma_0 \ \gamma_1 \ \gamma_2 = \begin{pmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \end{pmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \) |
|                  |                  | \( \gamma_0 = u_0 + u_1 + u_3 \) |
|                  |                  | \( \gamma_1 = u_0 + u_2 + u_3 \) |
|                  |                  | \( \gamma_2 = u_1 + u_2 + u_3 \) |
2.2. Trainer simulation
At this stage, the circuit scheme was made in the Proteus 8 Professional software and the Arduino
program was built using the Arduino EDA software that was tailored to the trainer system that had been
designed. Modification of circuits and programs is done as a form of optimization if the simulation
results do not match the specifications that have been designed.

2.3. Trainer fabrication
The fabrication stage consists of PCB printing stage by making layouts using Easy EDA with
consideration of the use of Arduino Mega 2560, installation of components according to design, case
installation and testing.

3. Results and discussion

3.1. Simulation results
The scheme of the trainer circuit used for simulation is shown in Figure 4. From the figure, it can be
seen that the circuit for simulation trainers uses rotary switch components to adjust the adder input
according to the type of matrix generator parameters used.

![Figure 4: Schematic for simulation.](image)

The results of the simulation experiments based on a two-sample matrix generator are listed in Table 4
and Table 5. Based on Table 4 and 5 it can be seen that the codeword generated in sample 1 and sample
2 is in accordance with the theory.

3.2. Fabrication results
The fabricated trainer device consists of PCBs which have been installed according to the circuit shown
in Figure 6, jumper and adapter cables. The layout of the PCB is shown in Figure 5 and the physical
shape of the trainer is shown in Figure 7. The fabricated trainer utilizes the header and jumper cable to
adjust the input according to the parameters of the matrix generator used. The results of fabrication trials
based on two-sample matrix generators that have been designed are listed in Table 6 and 7. Based on
Table 6 and 7, it is known that all codewords generated by fabricated trainers use matrix generators
sample 1 and sample 2 is in accordance with the theory.
### Table 4. Simulation result of sample 1.

| No. | Information Bits ($\tilde{u}$) | Parity Bits ($\gamma$) | Codeword ($\tilde{v}$) |
|-----|---------------------------------|------------------------|------------------------|
|     | Theory                          | Simulation             | Theory                 | Simulation             |
| 1   | 0000                            | 000                    | 00000000               | 00000000               |
| 2   | 0001                            | 111                    | 1110001                | 1110001                |
| 3   | 0010                            | 110                    | 1100010                | 1100010                |
| 4   | 0011                            | 001                    | 0010011                | 0010011                |
| 5   | 0100                            | 011                    | 0110100                | 0110100                |
| 6   | 0101                            | 100                    | 1000101                | 1000101                |
| 7   | 0110                            | 101                    | 1010110                | 1010110                |
| 8   | 0111                            | 010                    | 0100111                | 0100111                |
| 9   | 1000                            | 010                    | 0101001                | 0101001                |
| 10  | 1001                            | 011                    | 0110111                | 0110111                |
| 11  | 1010                            | 100                    | 1001111                | 1001111                |
| 12  | 1011                            | 101                    | 1011111                | 1011111                |
| 13  | 1100                            | 110                    | 1101111                | 1101111                |
| 14  | 1101                            | 111                    | 1111111                | 1111111                |
| 15  | 1110                            | 000                    | 0001111                | 0001111                |
| 16  | 1111                            | 111                    | 1111111                | 1111111                |

### Table 5. Simulation result of sample 2.

| No. | Information Bits ($\tilde{u}$) | Parity Bits ($\gamma$) | Codeword ($\tilde{v}$) |
|-----|---------------------------------|------------------------|------------------------|
|     | Theory                          | Simulation             | Theory                 | Simulation             |
| 1   | 0000                            | 000                    | 00000000               | 00000000               |
| 2   | 0001                            | 111                    | 1110001                | 1110001                |
| 3   | 0010                            | 011                    | 0110010                | 0110010                |
| 4   | 0011                            | 100                    | 1000111                | 1000111                |
| 5   | 0100                            | 101                    | 1010100                | 1010100                |
| 6   | 0101                            | 010                    | 0100101                | 0100101                |
| 7   | 0110                            | 110                    | 1100110                | 1100110                |
| 8   | 0111                            | 001                    | 0011110                | 0011110                |
| 9   | 1000                            | 110                    | 1101100                | 1101100                |
| 10  | 1001                            | 001                    | 0011100                | 0011100                |
| 11  | 1010                            | 101                    | 1011010                | 1011010                |
| 12  | 1011                            | 010                    | 0101011                | 0101011                |
| 13  | 1100                            | 011                    | 0111000                | 0111000                |
| 14  | 1101                            | 100                    | 1001101                | 1001101                |
| 15  | 1110                            | 000                    | 0001110                | 0001110                |
| 16  | 1111                            | 111                    | 1111111                | 1111111                |
Figure 5. PCB layout.

Figure 6. Schematic for fabrication.

Figure 7. Physical of trainer.

Table 6. Fabrication result of sample 1.

| No. | Information Bit ($\bar{u}$) | Parity Bits ($\gamma$) | Codeword ($v$) |
|-----|----------------------------|------------------------|----------------|
|     |                           | Theory | Fabrication | Theory | Fabrication |
| 1   | 0000                      | 000    | 000         | 0000000 | 0000000     |
| 2   | 0001                      | 111    | 110         | 1110001 | 1110001     |
| 3   | 0010                      | 110    | 110         | 1100010 | 1100010     |
| 4   | 0011                      | 001    | 001         | 0010011 | 0010011     |
| 5   | 0100                      | 011    | 011         | 0110100 | 0110100     |
| 6   | 0101                      | 100    | 100         | 1000101 | 1000101     |
| 7   | 0110                      | 101    | 101         | 1010110 | 1010110     |
| 8   | 0111                      | 010    | 010         | 0100111 | 0100111     |
| 9   | 1000                      | 101    | 101         | 1011000 | 1011000     |
| 10  | 1001                      | 010    | 010         | 0101001 | 0101001     |
| 11  | 1010                      | 011    | 011         | 0111010 | 0111010     |
| 12  | 1011                      | 100    | 100         | 1001111 | 1001111     |
| 13  | 1100                      | 110    | 110         | 1101100 | 1101100     |
| 14  | 1101                      | 001    | 001         | 0011101 | 0011101     |
| 15  | 1110                      | 000    | 000         | 0001110 | 0001110     |
| 16  | 1111                      | 111    | 111         | 1111111 | 1111111     |
4. Conclusion
It has been designed (7,4) Hamming code channel encoder trainer using Arduino Mega 2560. Simulation and fabrication test results produce 7-bit codewords, 3 bits parity with shapes that match the theory. Then (7,4) Hamming code channel encoder trainer using Arduino Mega 2560 can be used to generate a codeword using different matrix generator.

References
[1] Choudhury D R and Podder K 2015 Design of Hamming Code Encoding and Decoding Circuit Using Transmission Gate Logic International Research Journal of Engineering and Technology 2(7) 1165-1169
[2] Lin S and Costello D J 2010 Error Control Coding 2 edition (Noida, India: Pearson Education)
[3] Rhee M Y 1989 Error-Correcting Coding Theory First Edition (New York: McGraw-Hill)
[4] Nayak V S P, Madhulika C and Pravali U 2017 Design of low power hamming code encoding, decoding and correcting circuits using reversible logic 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) 778-781
[5] Sani U S and Shahono I H 2015 Design of (7, 4) Hamming Encoder and Decoder Using VHDL 1st International Engineering Conference 2015 103–106

---

Table 7. Fabrication result of sample 2.

| No. | Information Bits (u) | Parity Bits (γ) | Codeword (v) |
|-----|----------------------|----------------|--------------|
|     |                      | Theory Fabrication | Theory Fabrication |
| 1.  | 0000                 | 000 000         | 00000000 0000000 |
| 2.  | 0001                 | 111 111         | 1110001 1110001 |
| 3.  | 0010                 | 011 011         | 0110010 0110010 |
| 4.  | 0011                 | 100 100         | 1000011 1000011 |
| 5.  | 0100                 | 101 101         | 1010100 1010100 |
| 6.  | 0101                 | 010 010         | 0100101 0100101 |
| 7.  | 0110                 | 110 110         | 1100110 1100110 |
| 8.  | 0111                 | 001 001         | 0010111 0010111 |
| 9.  | 1000                 | 110 110         | 1101000 1101000 |
| 10. | 1001                 | 011 011         | 0110011 0011001 |
| 11. | 1010                 | 101 101         | 1011010 1011010 |
| 12. | 1011                 | 010 010         | 0101011 0101011 |
| 13. | 1100                 | 011 011         | 0111100 0111100 |
| 14. | 1101                 | 100 100         | 1001101 1001101 |
| 15. | 1110                 | 000 000         | 0001110 0001110 |
| 16. | 1111                 | 111 111         | 1111111 1111111 |