Production Systems and Information Engineering
Volume 13 (1), pp. 23-35
doi: https://doi.org/10.32968/psaie.2025.1.3

23

A SERIAL COMMUNICATION METHOD FOR ILLUMINATION CONTROL IN A LINE SCAN CAMERA-BASED MEASUREMENT SYSTEM

ZSÓFIA FORGÁCS
University of Miskolc, Hungary
Institute of Automation and
Infocommunication
zsofia.forgacs@uni-miskolc.hu

Abstract. In the field of embedded systems, communication protocols play a crucial role in enabling seamless data exchange between different hardware components. The existing RS-422-based communication method used in this application is often associated with limitations in terms of flexibility, scalability, and the ability to adapt to evolving requirements. To address these challenges, this paper outlines the development of a communication protocol that leverages the physical, the data link and the application layers of the OSI model. This communication protocol aims to provide a more versatile and efficient solution for transmitting control commands from a Raspberry Pi 4 microprocessor-based development board to an Altera Cyclone II-based FPGA development board in a line scan camera-based measurement system.

Keywords: serial communication, FPGA, LED control, line scan camera

1. Introduction

A line scan camera-based measurement system was implemented previously for examining high-speed rotating components [1]. In the development of such systems, the quality of illumination assumes a pivotal role as it directly impacts the accuracy and reliability of captures. Particularly in line scan camera applications, ensuring proper illumination stands as paramount [2], given the camera's continuous scanning of objects, where even minor lighting inconsistencies can distort or blur images [3]. Thus, ensuring optimal illumination emerges as a crucial prerequisite for achieving precise and consistent measurements.

Line scan cameras acquire only a single (or a few) lines of pixels during a sampling cycle, and their sensor area is considerably smaller compared to traditional area scan cameras. Consequently, in the design of illumination, the geometry of the illuminating unit is not the predominant design factor; a simple line light positioned adequately in front of the examined object suffices. Of greater significance is the necessity for the camera to attain higher light intensity [4] [5] due to the brief exposure time required to gather adequate light for image acquisition.

In industrial machine vision systems, LED light sources are commonly employed due to their extended lifespan, low power consumption [6] [7], and rapid response time. LED light sources can be controlled using either constant voltage or pulse sequences. The advantages of constant voltage control include simplicity and reliability, as it eliminates the need for specialized control or timing. In this mode, the brightness of the LED is directly proportional to the applied voltage, facilitating straightforward adjustment of light intensity. However, constant voltage operation may lead to a shorter LED lifespan due to continuous stress, resulting in heat generation.

In pulse control, the LED light source receives short yet intense light flashes. The pulse sequence must be synchronized with the camera's sampling frequency to ensure proper illumination detection. The primary benefit of pulse control is the extension of LED lifespan, as the LED only emits light for brief periods, thereby reducing heat generation. Pulse control allows precise regulation of the illumination duration during the camera's exposure period, potentially yielding sharper and clearer images. Moreover, by controlling LED light sources with pulse sequences, it is possible to adjust light intensity by varying pulse width or frequency.

The frequency of illumination will be crucial for producing high-quality images. Line scan cameras typically perform sampling at high line refresh rates, necessitating a correspondingly high illumination frequency to ensure adequate light reaches the sensors with each capture. Therefore, the illumination frequency should match or exceed the camera's line refresh frequency for proper synchronization and clear image formation.

Summarizing, in the realm of image capture using line scan cameras, precise measurements are heavily dependent on adequate and consistent illumination. The quality of illumination is of paramount importance, as it directly impacts the effectiveness of the camera's scanning abilities. Typically, illumination levels are managed by the camera's I/O (Input/Output) lines, which communicate using the RS-422 protocol in this case to control lighting. The camera's output reflects the active exposure period, and this exposure often has a low value, resulting in low duty cycle pulses. These low duty cycle pulses lead to lower brightness levels on the LED due to the reduced on-time, even though the pulses act as an input to a MOSFET-based LED driver circuit that provides the desired output voltage. For sufficient brightness, the LED must be driven at higher voltages, which the low duty cycle fails to achieve.

To address this challenge, a communication protocol has been developed specifically for enhanced control of lighting components. This protocol is necessary to generate the desired pulse sequences for LED control using an FPGA (Field-Programmable Gate Array), which requires the transmission of numerical data from a microprocessor. Given that the FPGA only has digital I/O, control commands are sent to it in a serial communication format. The developed protocol aims to provide more precise control over LED illumination, allowing varied voltage levels and improving overall image quality with a more tailored lighting system.

2. Illumination Control via Camera I/O Lines

Most industrial cameras can indicate the active exposure time during image capture. This information enables illumination control synchronized with exposure. When exposure is active, illumination can be initiated or adjusted at the appropriate moment to ensure the captured images are of adequate quality [8] [9]. The camera used in this research has 3 digital inputs and 2 digital outputs, providing symmetrical transmission lines compliant with the RS-422 standard.

RS-422 is a technical standard that specifies the electrical characteristics of a digital signaling circuit. This standard is used for serial communication between devices in a system. RS-422 uses differential signaling where each bit of data is represented by the difference in voltage between two wires (positive and negative).

This method provides greater immunity to electromagnetic interference compared to single-ended signaling systems. The protocol supports long-distance communication, capable of achieving reliable data transmission over distances up to 1200 meters depending on the data rate and quality of the cable.

RS-422 allows for multiple receivers (up to 10) to be connected to a single transmitter in a multi-drop configuration, enabling one communication channel to serve several devices. It facilitates higher data rates than RS-232, supporting up to 10 Mbps at short distances (within 10 meters or so), with the rate decreasing as the distance increases.

In the context of line scan cameras, RS-422 can be used to send control signals to the lighting system, handling instructions like switching on or off, modulating intensity, and synchronization with scanning. Thanks to balanced signaling, RS-422 is robust against electrical noise, which is crucial in industrial and noisy environments where running long cables is common. The block diagram of the camera used in this research is shown in Figure 1.

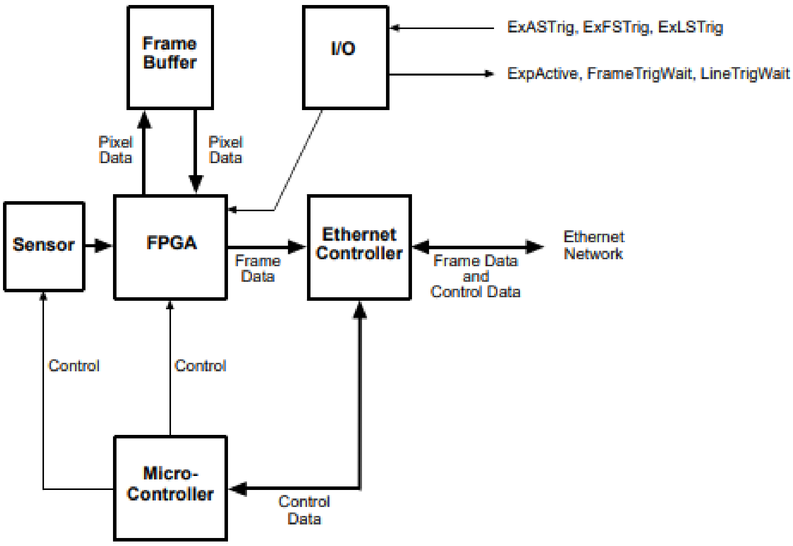

Figure 1. Block diagram of the camera [10]

The use of the camera's digital output to control illumination involves receiving the output signal by a control circuit, which uses this information to regulate LED light sources. The RS-422 controlled circuit governs the LED illumination and synchronizes it with the camera's exposure period. This approach offers various benefits. Firstly, using the camera's digital output allows for precise synchronization with the camera's exposure period, thereby improving image processing accuracy and reliability. Additionally, the RS-422 standard provides highly reliable, differential serial communication that facilitates dependable signal transmission over longer distances, especially beneficial in industrial settings.

For illumination control via I/O lines, the "ExpActive" output signal depicted in Figure 1 of the camera was employed, which transitions to a high logic level at the beginning of each line scan and returns to a low logic level at its conclusion.

A linear illumination consisting of three 10 W LEDs connected serially was employed as the lighting unit. To control it, a MOSFET circuit was created, as depicted in Figure 2. It is important to note that the RS-422 data needs to be converted into a unipolar binary sequence before being used. For this purpose, a converter module with essential communication circuit elements including 120 Ω termination resistors was employed.

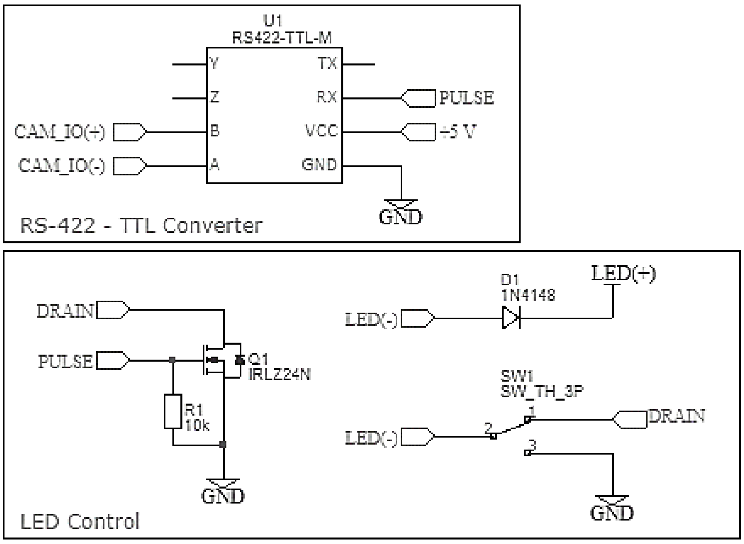

Figure 2. Circuit for LED control via the camera's I/O lines

The MOSFET circuit was also tested, a measurement of which is shown in Figure 3.

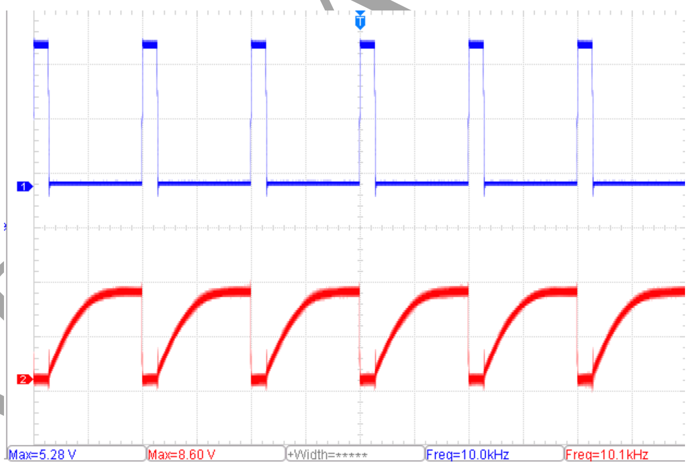

Figure 3. MOSFET circuit test

The circuit is appropriate for illumination control. However, the traditional lighting control method using camera input/output lines has the limitation that it generates the pulse sequence required to control the light sources based solely on the exposure signal, without any feedback from the recorded and processed image data. Additionally, utilizing RS-422 lines necessitates the installation of an extra signal conversion circuit. Therefore, an alternative lighting method was planned to develop, which provides the possibility to feed back the image data, thereby controlling the control signal of the LED light sources.

3. Requirements for the Illumination System

The detailed approach to illumination is designed to ensure that the lighting control unit produces a pulse sequence frequency based on the camera's line rate and processed image data, preventing dark or poorly illuminated pixel rows in captured images. It is important not to excessively increase the pulse sequence frequency as this can introduce significant noise into the system, affecting processing accuracy. The pulse frequency should at least match the maximum line scan frequency of the camera. The largest adjustable line scan frequency in the line scan camera used in the research work is 51 kHz, targeting the generation of pulse sequences in the order of 100 kHz magnitude for the LED control. As a result, specific subtasks were defined for developing the illumination method:

- Reading the line rate set on the camera.
- Analyzing the recordings for the success of the illumination.
- Generating a pulse sequence of possibly 100 kHz magnitude based on the analysis results.
- Controlling LED light sources with a power of 10-30 W using the generated pulse sequence.

To read the line rate and process the recorded images, it is necessary to implement a camera control application that accesses the device description file in the camera via its communication interface (e.g., USB3.0, GigE Vision) and can read the registers within. The specific camera used in this research is a Basler racer raL2048-48gm line scan camera with a GigE Vision communication interface. Its properties are determined by the camera itself, supported by GenICam technology which enables dynamic access. This involves having a GenICam XML description file stored in the camera, which is utilized to generate an application programming interface called GenAPI. Functions and attributes of the camera, such as pixel format or exposure time, can be accessed by reading from and writing to designated registers. Noteworthy is that according to GigE Vision standards, cameras with a GigE interface must provide this XML description file.

To read the camera properties and process the recorded images, a device with the appropriate communication interface for accessing camera data is required. The Raspberry Pi 4 microprocessor-based development board with a Gigabit Ethernet port is suitable for accessing GigE cameras. To evaluate the lighting technique, an application was implemented for controlling the camera on this platform.

One of the listed tasks involves creating a pulse sequence with a frequency of 100 kHz magnitude. However, using a microprocessor-based control unit for this purpose is not ideal because its operating systems cannot guarantee real-time operation due to task scheduling dependencies. Response times may vary depending on the priorities of the operating system and other background processes running. To explore this issue, a test application was implemented that runs on a microprocessor and generates pulse sequences with different frequencies having a 50% duty cycle. The signal output of the device was analyzed using an oscilloscope and compared the calculated characteristics of the generated signal with measured values. The measurements were conducted across signals ranging from 0.5 Hz to 50 kHz, and Figure 4 illustrates the percentage deviation in half-periods between measured and expected values.

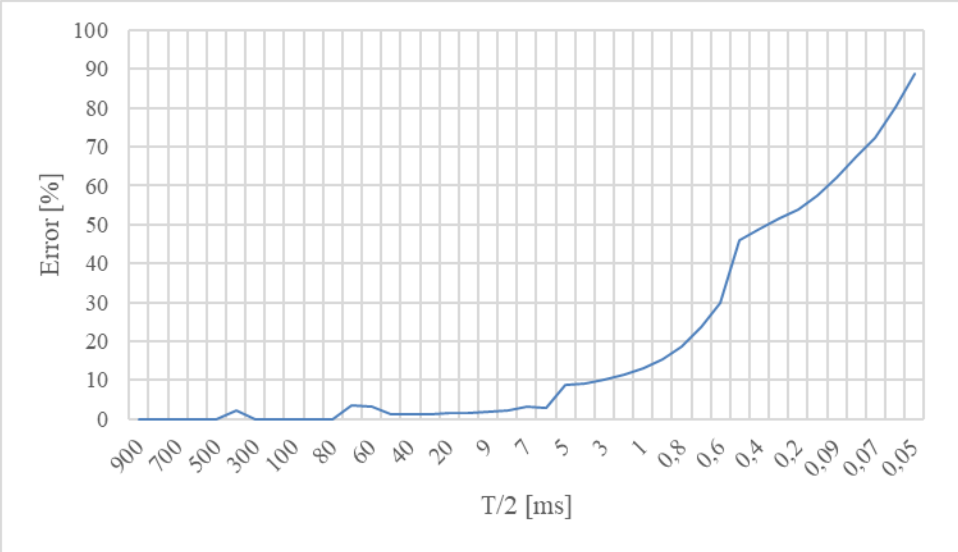

Figure 4. Frequency error values of microprocessor-generated signals

Based on the measurement results, we can conclude that the generation of pulse sequences in the order of 100 kHz magnitude required for LED control is not achievable using this microprocessor-based development tool. An FPGA offers a more accurate solution for this purpose. The internal structure of FPGAs reduces latency by directly linking the logic elements and incorporates specialized hardware resources for specific operations, including DSP blocks, memory blocks, and PLL (Phase-Locked Loops) units capable of processing data swiftly and precisely. In this research, a development board featuring an Altera Cyclone II EP2C2T5 FPGA chip was utilized. This device operates with a 50 MHz external clock frequency and includes 2 PLLs. As there's no guarantee regarding a processing frequency of 50 MHz, the clock generated by a PLL ranging from 10 to 100 MHz was analyzed while verifying the frequency of produced pulse sequences. Based on these measurements, it was confirmed that the necessary frequency for lighting control can be delivered by the device.

Based on the given requirements, it is essential to have a microprocessor-based control unit for carrying out camera control and image processing tasks. Additionally, an FPGA-based development board is required to generate pulse sequences necessary for controlling the LED light sources. However, a question arises as to how to transmit numerical data required for frequency regulation to the FPGA, which only has digital I/O lines? One potential solution involves sending optimized bit sequences in a serial format to the lighting control unit for this purpose. The protocol format designed specifically for this task will be detailed in the following chapter.

4. Communication Protocol for Illumination Control

The data transmission process between the microprocessor and FPGA involves modifying the frequency setpoint for controlling illumination, as well as increasing or decreasing it according to the control algorithm. The communication method is characterized based on the OSI reference model's layered perspective. In industrial communication systems, a simplified OSI model with three main layers [11] - physical, data link, and application - is commonly used.

4.1. Physical Layer

The physical layer defines the mechanical, electrical, and functional parameters of the communication system. The signal levels from the FPGA and microprocessor determine the voltage levels of the transmitter and receiver units, with standard values set at 0 V for logical "0" and 3.3 V for logical "1". Transmission occurs in an asymmetric manner, with signal voltages referenced to a common point. The establishment and breakdown of the connection are achieved through continuous listening by the transmitter, monitoring of start and stop bits. The connection setup and termination are achieved by continuous monitoring carried out by the transmitter unit which focuses on start and stop bits. A simplex system was implemented due to feedback within this control loop being provided by light pulses emitted by an LED. In future iterations, there is potential for further development towards half- or full-duplex transmission modes such as transmitting acknowledgment messages back to the transmitter unit.

The representation of bits is also determined by the physical layer. A solution that offers an advantage in terms of synchronization was chosen, which is crucial to address timing errors previously demonstrated with the microprocessor. Due to significant and unpredictable delays, bit times can shift, making it extremely difficult for error-free processing, especially in the case of traditional NRZ (Non-Return-to-Zero) codes and numerous consecutive identical bit values, making error-free processing nearly impossible above 500 Hz. To address this, a signaling procedure was chosen where a signal transition occurs in every bit time since the rising and falling edges can be precisely detected by the receiver unit despite the uncertain timing conditions of the transmitter. The Manchester encoding presents such a solution: representing logical "1" as a rising edge, and logical "0" as a falling edge. While most industrial communication systems involve polarity changes with signal transitions, a simple unipolar code was implemented in this case, due to the operating voltages of the communication devices.

The data transmission procedure was developed to be compatible with the physical medium used in FPGA- and microprocessor-based development boards. Accordingly, the physical medium used for data communication consists of two twisted wires (the signal wire required for asymmetric transmission and the common ground wire). Only data transmission is required on the physical medium; there is no need for power supply provision.

4.2. Data Link Layer

The data link layer is responsible for various functions, including telegram structure, medium access control (MAC), and error handling during transmission, as well as addressing models of communication participants or flow control. Among the listed functions, the developed telegram structure will be presented first.

4.2.1. The Structure of Messages

The data exchange between the microprocessor and FPGA serves to adjust the setpoint for control pulse sequences and modify its frequency based on a control algorithm. To accomplish this, three options were devised for modifying parameters: adjusting the frequency setpoint, increasing frequency and decreasing it. These modifications are carried out through parameter setting and control commands within the developed protocol. The goal is to enable continuous adjustments of frequency with specific values; therefore, shorter messages are recommended due to their frequent occurrence. Messages responsible for parameter setting will be transmitted less frequently in a measurement cycle, allowing for longer message transmission when necessary. The identified types of telegrams include:

A. Batched setting (setting all three parameters in one message),
B. Base signal frequency setting,
C. Increment unit setting,
D. Decrement unit setting,
E. Frequency increase command,
F. Frequency decrease command.

In the initial phase of the designing process, the potential use of a coding method was examined to decrease message size. Huffman coding emerged as a significant algorithm for data compression [11]. It is a lossless compression technique that efficiently assigns variable-length codes to symbols based on their frequency of occurrence, allowing more frequent symbols to have shorter codes, and leading to overall data compression. Huffman coding finds application in various domains such as text, data, image, and video compression, file archiving, and network protocols.

Although applying Huffman coding to numerical parameter values does not yield benefits; its application for redundant data was explored. Let's assume that the probabilities of occurrence for the messages are as follows: A: 0.01, B: 0.03, C: 0.08, D: 0.05, E: 0.4, F: 0.4. One possible way to build the Huffman tree and the new codewords generated by the Huffman algorithm for the messages is shown in Figure 5.

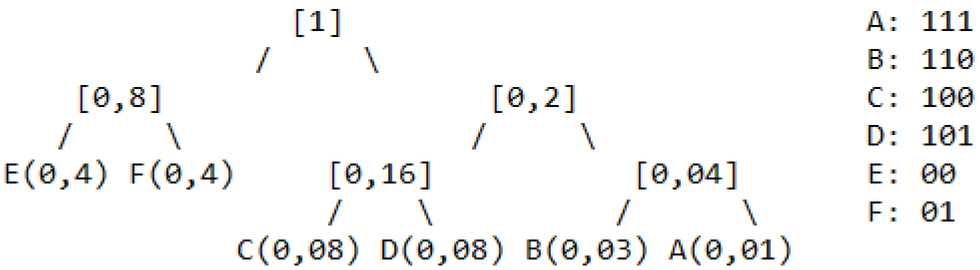

Figure 5. A way to build the Huffman tree for function codes

The most frequently occurring message identifiers can be represented in a 1-bit shorter field, but this greatly raises the intricacy of deciphering the messages compared to the level of compression attained. Additionally, if there's a bit error at the initial position of the two-bit identifiers, it could easily lead to inaccurate processing. Rectifying such bit errors would entail identifying where the malfunction occurred, leading to a substantial rise in redundancy. Due to these reasons, data compression will not be implemented for the message fields.

The following general structure has been applied to the message frames:

`START | FC | DATA | EC | STOP`

The content and length of each field are as follows:

- `START`: start bit, logical "0" (1 bit)
- `FC`: type of telegram (3 bits)
- `DATA`: value of the parameter setting (variable length based on message type)
- `EC`: error detection (3 bits)

- `STOP`: stop bit, logical "1" (1 bit)

The possible values of the FC field, responsible for message identification, and their meanings are detailed in Table 1.

Table 1. Function codes in messages

FC	Function
000	Invalid
001	Initialization (batched) setting
010	Frequency setpoint setting
011	Increment unit setting
100	Decrement unit setting
101	Frequency increment command
110	Frequency decrement command
111	Invalid

The length of the DATA field can vary based on the type of telegram. The length and content of the data payload are summarized in Table 2:

Table 2. Length and structure of data field

FC	Length	DATA field structure
001	36-54 bits	20-bit frequency setpoint 2-bit control information 7/16-bit increment unit 7/16-bit decrement unit
010	20 bits	Frequency setpoint
011	7-19 bits	3-bit control information 4/16-bit increment unit
100	7-19 bits	3-bit control information 4/16-bit decrement unit
101	-	Frequency increase/decrease commands don’t contain data field
110	-	

The frequency setpoint is 20 bits in length to represent all values up to several 100 kHz (2^{20} = 1 048 575). In the case of the ‘001’ (initialization) telegram, the 2-bit control information determines how the increment (msb) or decrement (lsb) unit is specified:

- "0": 3-bit magnitude + 4-bit multiplier digit for values divisible without remainder by powers of 10,
- "1": 16-bit integer otherwise.

The magnitude values in the control information can be represented using the codewords listed in Table 3:

Table 3. Magnitude codewords of increment- and decrement units

Magnitude codewords	
000	Invalid
001	1
010	10
011	100
100	1000
101	10000
110	100000
111	Invalid

The magnitude digits are assigned values ranging from 1 to 9, depending on the

binary representation of the field value. The control information consisting of 3 bits in messages identified as 011 and 100 (for incremental/decremental unit settings) may include magnitude code words from Table 3, except for the invalid code word 111. This particular code word signifies that the frequency increase/decrease unit value is not a multiple of powers of 10, thus it is specified with a 16-bit integer.
The error checking (EC) field contains three parity bits that perform parity-checking on the function code and message data. The most significant bit calculates an odd parity bit for the entire message, while the middle bit calculates one for even positional bits, and the least significant bit computes one for odd positional bits.

4.2.2. Data Access and Error Handling

Beyond the telegram structure, the data link layer has duties related to entrance processes, identifying participants, and ensuring error detection. Regarding access procedures, a conventional method used in industrial communication systems wasn't employed. As a two-point simplex transmission was implemented, a passive and an active device participate in communication. The passive participant awaits commands sent by the active one and remains idle until a message arrives. Therefore, the data transmission procedure was designed for establishing a point-to-point connection, hence, no addressing model is defined. The passive participant monitors designated I/O ports for incoming telegrams.
During transmission, data errors can occur for several reasons. Although encryption is effective in guarding against deliberate attacks, it was not considered because of the experimental nature of the research. Synchronization failures can be mitigated by employing special line coding, such as Manchester encoding in this case. Moreover, employing START and STOP bits to delimit message frames assists in preserving synchronization. To counteract data corruption caused by channel noise, we can analyze the received bit sequence and utilize error detection and error correction coding as a defense mechanism. The developed communication technique can perform the following checks:

- Function Code (FC) check: detecting invalid code words '000' and '111'.
- Increment/Decrement unit check: during batch setting, detecting invalid code words '000' and '111'; during unit setting, detecting invalid code word '000'.
- Data unit check with 3 parity bits: one for the entire data block and two additional ones for the even- and odd positional bits of the data field.

4.3. Application Layer

The application layer is responsible for managing user interactions and accessing network services. An application has been created specifically for this task, providing a user interface through which data can be transmitted and received utilizing the developed communication protocol. This application enables users to transmit data, execute commands, and configure relevant parameters. The package diagram of the Python application responsible for communication control can be seen in Figure 6, its class diagram is shown in Figure 7.

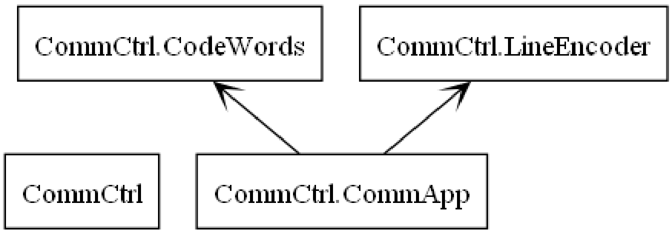

Figure 6. Package diagram of the encoding application

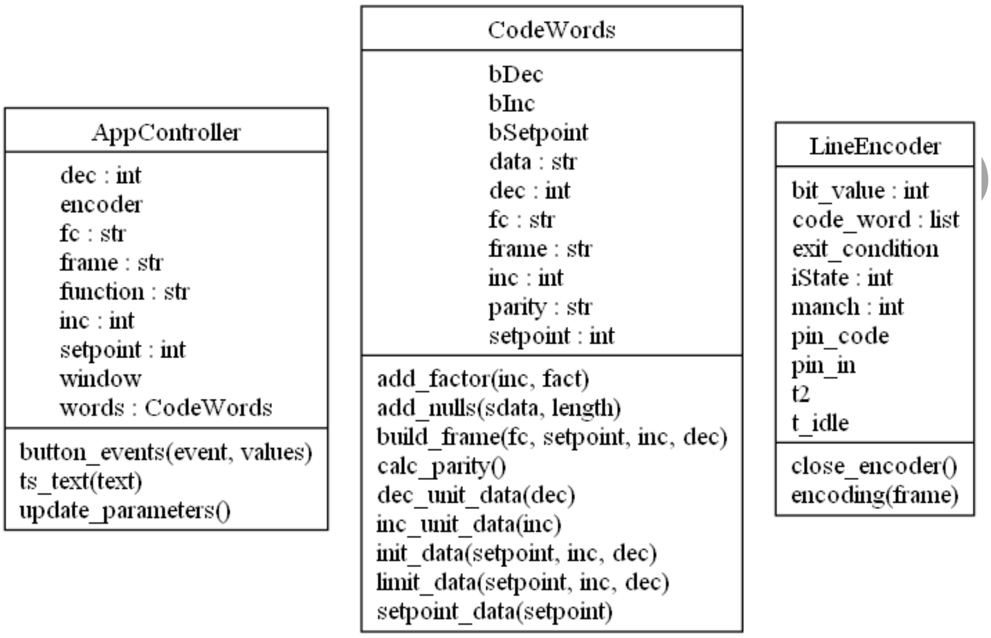

Figure 7. Class diagram of the application

5. Results

The presented communication method was implemented and tested. The wiring diagram of the I/O points of lighting unit can be seen in Figure 8. The DATA_OUT port transmits the sequence of bits needed for LED control, following the specified protocol, to the FPGA's DATA_IN input. The FPGA then generates the LED control pulse series and outputs it on the PULSE port. The MOSFET-based LED driver circuit is not presented again, as it is identical to the one depicted in Figure 2.

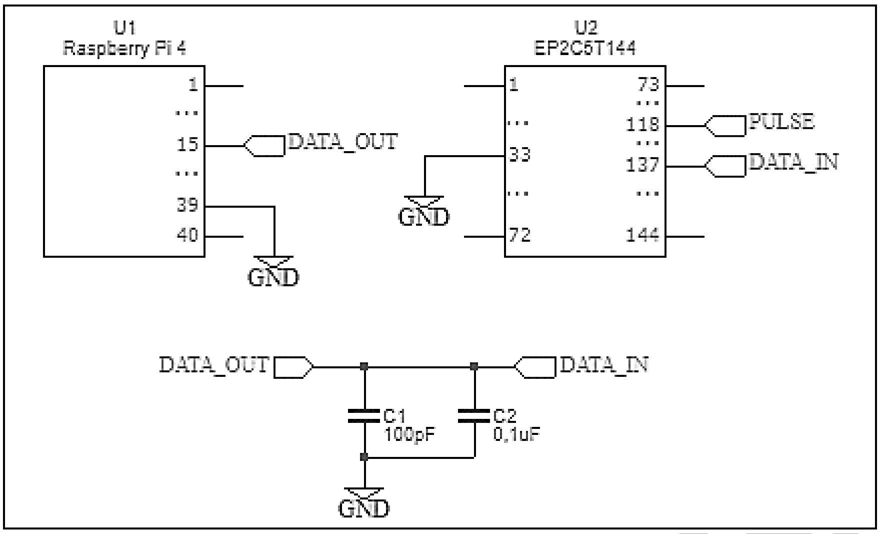

Figure 8. I/O points for LED control

The functionality of the Python application running on the Raspberry Pi 4 and tasked with encoding messages for the lighting control system was evaluated. The application successfully generated the necessary digital data sequences according to the specified communication protocol, and the results were found to be in line with expectations. An example of a generated frequency incrementing command, which is used to adjust the brightness of the LED lighting, is presented in Figure 9.

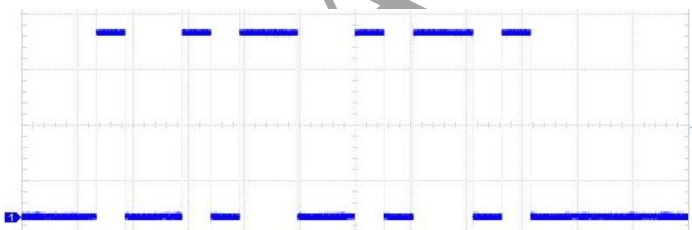

Figure 9. Frequency incrementing command

The decoding of the messages was implemented on the FPGA board according to the developed protocol. This allowed the system to interpret and act upon the serial digital data being transmitted. Figure 10 shows a detail of the related measurements taken to verify the proper decoding and interpretation of the messages by the FPGA.

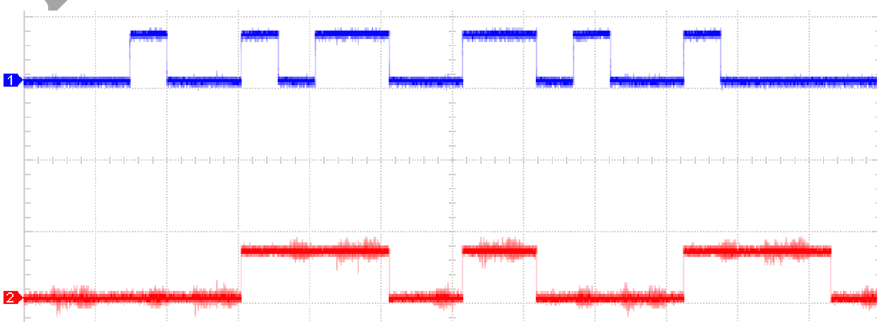

Figure 10. Encoded and decoded

messages

The generated signals demonstrated their suitability for effectively processing the messages and controlling the LED lighting system as intended.

6. Conclusion

The operation of machine vision systems is significantly influenced by the lighting technique employed. Most industrial cameras can indicate the active exposure time during image capture, enabling the synchronization of lighting with the exposure. For instance, a pulse-controlled LED lighting system can be implemented. The advantage of the pulse mode is the extension of the LED's lifespan, as the LED illuminates for a shorter duration per unit of time, thereby producing less heat.

Since traditional lighting control using the camera output signal lacks feedback on the recorded and processed image data, an alternative lighting method was aimed to develop. An independent lighting control unit provides the opportunity to feed back the image data, enabling the regulation of the control signal for the LED light sources or, in the case of displacement measurement, the selective activation and deactivation of the lighting units along the displacement axis.

The implementation requires a microprocessor-based control unit to perform camera control and image processing operations. Furthermore, an FPGA-based development platform is necessary to generate the pulse sequences required for controlling the LED light sources. The fundamental challenge is how to transmit the numerical data necessary for frequency control to an FPGA with limited digital I/O capabilities. One viable approach is to send optimized serial data streams, or telegram structures, to the lighting control unit. This paper presented a serial communication protocol that can be easily implemented on an FPGA and is suitable for transmitting high-frequency signals.

The proposed communication protocol and associated circuit were evaluated and performed as expected. Currently, the illumination method compensating the frequency difference is under development, which, combined with the protocol presented in this article, is expected to yield future research results.

References

[1] Z. Forgacs and A. Trohak, “A Measurement Method for Determining Speed of High-Speed Rotating Parts,” in 2022 23rd International Carpathian Control Conference (ICCC), IEEE, May 2022, pp. 203–207. https://doi.org/10.1109/ICCC54292.2022.9805948

[2] Lopes, G.; Ribeiro, A.F.; Sillero, N.; Gonçalves-Seco, L.; Silva, C.; Franch, M.; Trigueiros, P. High Resolution Trichromatic Road Surface Scanning with a Line Scan Camera and Light Emitting Diode Lighting for Road-Kill Detection. Sensors 2016, 16, 558. https://doi.org/10.3390/s16040558

[3] Qin, J.; Kim, M.S.; Chao, K.; Chan, D.E.; Delwiche, S.R.; Cho, B.-K. Line-Scan Hyperspectral Imaging Techniques for Food Safety and Quality Applications. Appl. Sci. 2017, 7, 125. https://doi.org/10.3390/app7020125

[4] Wendel, A.; Underwood, J. Extrinsic Parameter Calibration for Line Scanning Cameras on Ground Vehicles with Navigation Systems Using a Calibration Pattern. Sensors 2017, 17, 2491. https://doi.org/10.3390/s17112491

[5] Steger, C., Ulrich, M. A Multi-view Camera Model for Line-Scan Cameras with Telecentric Lenses. J Math Imaging Vis 64, 105–130 (2022). https://doi.org/10.1007/s10851-021-01055-x

[6] Ren, Z., Fang, F., Yan, N. et al. State of the Art in Defect Detection Based on Machine Vision. Int. J. of Precis. Eng. and Manuf.-Green Tech. 9, 661–691 (2022).

https://doi.org/10.1007/s40684-021-00343-6

[7] Satorres Martínez, S.; Martínez Gila, D.M.; Rico, S.I.; Teba Camacho, D. Machine Vision System for Automatic Adjustment of Optical Components in LED Modules for Automotive Lighting. Sensors 2023, 23, 8988. https://doi.org/10.3390/s23218988

[8] Wang, Y., Chen, H., Zhang, S. et al. Automated camera-exposure control for robust localization in varying illumination environments. Auton Robot 46, 515–534 (2022). https://doi.org/10.1007/s10514-022-10036-x

[9] Zhang, X.; Tang, X.; Yu, L.; Pan, B. Automated Camera Exposure Control for Accuracy-Enhanced Stereo-Digital Image Correlation Measurement. Sensors 2022, 22, 9641. https://doi.org/10.3390/s22249641

[10] Basler AG, Basler racer USER'S MANUAL FOR GigE VISION CAMERAS, 2019.

[11] Ajtonyi, I., Ipari kommunikációs rendszerek I., AUT-INFO Kft., Miskolc, 2008.