

Jozef Hrbček*

ORCID ID: 0000-0002-6984-9470

Emília Bubeníková

ORCID ID: 0000-0003-3053-0822
University of Žilina, **Slovak Republic**

INTRODUCTION

Nowadays, process control, the automatic measurement system and the data processing systems are using programmable logic controllers (PLCs). The architecture and computation power of PLC is the same as PC, but their realization is suitable for use in industrial applications. These are suitable for their determination – resistance to weathering, disturbance and continuous operation (IEC 61131, 2019). Last but not least, the reliability features are also included. The used PLCs from B&R Company can work in redundant connection to increase the control system reliability. The Raspberry Pi is used to process the image and provide the requested information about the controlled system to the PLC via implemented industrial Ethernet POWERLINK (EPL). Our solution requires the real-time communication and computation power for image processing for the purpose of control and monitoring the systems. These functionalities are important for industrial application. Our main goal is creation and using the low cost solution instead of using the vision system offering by a few manufacturers (e.g. Siemens, Cognex, B&R, Keyence, etc.). Computer vision is an important field of research, too. The general steps used in this process are acquiring, processing and analyzing the image and convert it into numerical or symbolic form. Part of the computer vision is image processing, image analysis and machine vision is also closely related to a computer vision. There are a large number of applications in many areas in the world where the image processing is used; for example, in medicine (Koniar et al., 2017), access control systems, transport (Bubeníková, 2015), (Zolotová and Lorenčík, 2018), biometrics and especially in industrial processes (measurement, product quality control, identification, localization), (Mičieta et al., 2015), navigation of robots or MAV (Hruboš et al., 2016), (Pirník et al., 2016).

RESEARCH METHODOLOGY

In our paper, we describe a concept system that processes the acquired image information from the 8MP RPI Camera Board for the Raspberry Pi V2.1 and the image data from it is processed through a Raspberry Pi single-board computer. The camera

* jozef.hrbcek@fel.uniza.sk

can give the image with 3280x2464 pixels resolution and video mode with 1080p30 or 720p60. The image processing unit extracts data from the image and transmits it through the communication interface. The system may also be equipped with additional USB surveillance cameras. More information about the surveillance cameras can be found in (Halgaš & Pirník, 2015). Cooperation between the vision systems and automation solutions based on the high-performance industrial communication protocol open up new perspectives. The PLC system with CPU redundancy provides a layer between image processing unit and superior level (HMI). This concept has been designed to realize the reliable operation of the device under operating conditions.

Our application uses the low cost component with possibility of image processing algorithms. The implementation of the algorithms has been designed to allow testing and validating various object measurement approaches, e.g. distance, radius and speed or detection of different objects such as detecting circles or faces. Our application uses connected embedded system with HD camera. This paper is focused on the system which are the processing unit and the PLC system. Modular PLCs are a system of modules that can be slotted together to build up a system. The basic modules contain a power supply, central processing unit (CPU), communications modules and the input and output modules. This system can be expanded by an amount of input and output terminals. Some PLCs use the open source communications networks and allows us to connect the embedded devices to this modular system – it depends on the system manufacturers. The currently most used PLCs are (sorted alphabetically): B&R, Beckhoff, Mitsubishi, Omron, Rockwell, Schneider-Electric and Siemens. Only the B&R producer prefer the open communication protocol with high speed (the communication cycle less than 400 μ s) and real-time deterministic approach. Moreover, the B&R PLCs support the local and remote visualizations, web server, ftp server and http/https communication with a cloud according to the concept INDUSTRY 4.0. These PLCs also support the third party devices connection without need of any additional modules. Usage of the B&R control system with image processing unit guarantees the successful solution.

INCREASING THE SYSTEM RELIABILITY

A combination of controller and network redundancy maximizes protection against failures across the entire automation system (Ždánky & Rástočný, 2014), (B&R, 2019). The redundancy of the system serves to restore it after failure. The reserve parts of the system could be omitted if other parts of the system work correctly. Redundancy can be categorized into different groups (Hrbček et al., 2014):

According to their using: Power redundancy, CPU redundancy, sensor redundancy, network and media redundancy, redundancy of communication busses.

According to their functionality – Automation redundancy can be done in three ways:

- Hot-standby redundancy provides maximum reliability with outstanding convenience at the same time. Hot Redundancy is used when the process must not go down for even a brief moment under any circumstance.
- Warm – standby is used where time is somewhat critical but a momentary outage is still acceptable. This process is usually automated using a cluster manager.
- Cold – standby makes the secondary node acts as backup of another identical primary system. Cold redundancy is for those processes where the response time

is of minimal interest and may require operator intervention.

- According to the fabrication:
- Safe redundancy provide redundant everything necessary for safe operation, for example: processors are redundant, but signal redundancy is not ensured.
- Full redundancy makes everything redundant: power, wiring, sensors, signals, communications processors.

Presented concept of the redundant control system guarantees maximum availability for entire systems as well as connected systems. It uses warm redundancy according to the Fig. 1. Master redundancy with POWERLINK synchronizes data with microsecond accuracy and can be switched over in a matter of milliseconds. This functionality is integrated in the real time operating system. The interface PCI_5LS182.6-1 provides the configurable ring redundancy for POWERLINK.

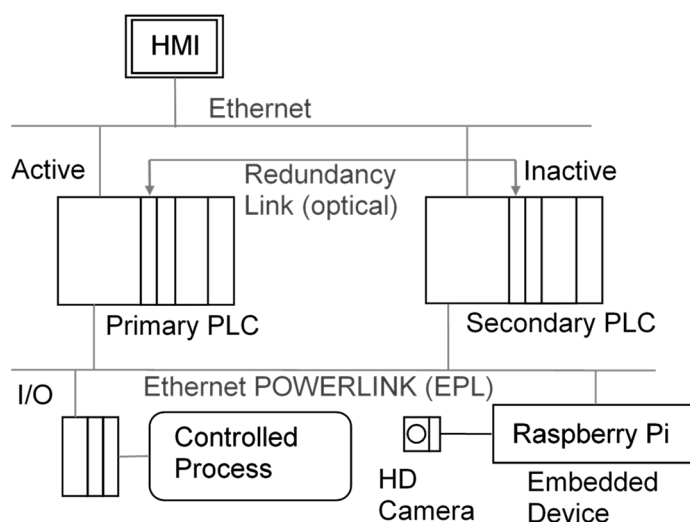


Fig. 1 The realized system with CPU redundancy

Embedded systems can be connected as a node of EPL bus by adding the XDD (XML device description) file from the openPOWERLINK source code stack to the programming tool Automation Studio (Automation Studio, 2019).

COMMUNICATION PRINCIPLE

The deterministic, reliable and fast communication bus is the important requirement. Ethernet POWERLINK (EPL) is the open source real-time Ethernet protocol adopting the CANopen object dictionary concept and its communication mechanisms that provide features like interoperability, flexibility and configurability. The lower layers of the EPL software are specifically designed to follow the EPL cycle schedule and to guarantee fast response times for real-time events on the bus. For comparison, the shortest communication cycle time for EPL is $100\mu\text{s}$ comparing to Profinet's support of $1000\mu\text{s}$. New solution for real-time communication is OPC UA TSN. This networking solution uses IEEE-standard Ethernet with TSN (Time Sensitive Networking) using OPC UA as its communication protocol. The OPC UA TSN is going to supersede various industrial networking based on Ethernet. The prototype with 200 nodes can communicate with $50\mu\text{s}$ communication cycle. In the future $10\mu\text{s}$ cycle time is expected. It depends on the hardware and software used (Bruckner et al., 2018).

Our solution uses industrial network Ethernet POWERLINK. This open source network protocol was implemented to the embedded system.

Implemented EPL stack version 2.7.1 on Raspberry Pi realizes deterministic communication with the redundant control system (Fig. 1). The communication cycle time is now 2000 μ s. Image processing realizes the Raspberry Pi with the operating system Raspbian Jessie kernel 4.14 with implemented RT-PREEMPT. RT-PREEMPT is a popular patch for the Linux kernel to transform Linux into such a real-time operating system. The cyclic test program provides an easy way to assess the maximum latency of a system. The cyclic test basically measures how much it takes to respond to an interrupt generated by the CPU timer.

Table 1
The cyclic test

Operating system	Max latency	Average latency
Raspbian without RT-PREEMPT	502 μ s	14 μ s
Raspbian with RT-PREEMPT	42 μ s	7 μ s

Implemented RT-PREEMPT makes the maximum latency more than 10 times shorter. It makes possibility for faster real-time communication. The tests proved that the maximum communication cycle time could be only 10000 μ s without RT-PREEMPT. It is very important to have a low time delay for receiving and sending commands and data in automated production processes. This can be ensured by Ethernet-based industrial Ethernet networks. One of them is Ethernet POWERLINK, which is supported by B&R devices. OpenPOWERLINK is a complete protocol solution for masters and slaves. This implementation can be ported to any target system programmed in ANSI C.

As a first step we have needed to install *CMake*, *CMake GUI*, *libpcap* and *WiringPi* on the Raspberry Pi. *Libpcap* is a system-independent interface that is designed for capturing packets at the user-level providing a framework for network monitoring at lower levels. By modifying the source code in *linux.cmake*, *main.c*, *app.c*, and *app.h*, we have defined the required data to be transmitted. We have defined the input and output structures and then we have created local and global application variables. These variables are used to transfer information between control system and Raspberry Pi. Then we have set up the configuration files and have compiled the openPOWERLINK source code.

IMAGE PROCESSING

Image processing is a method that performs certain operations to obtain an improved image or to obtain useful information from an image. We use the 8MP camera v2.1 described above to capture images. It is possible to use other external USB camera. Image processing basically consists of three parts:

- acquisition of an image,
- Image analysis and manipulation,
- an output that can be a modified image or an information based on the analysis.

The expansion of image recognition technology effect open-source tools that make programming easier, and make computing more affordable. Open-source frameworks and libraries today make possibilities for companies to benefit from image recognition technology exponentially. For example, libraries as IPL (Image Processing Library)

(Library IPL, 2019), Camellia (Camellia library, 2019), Scikit-image (Scikit-image, 2019), and one of the most popular Open CVs (Librare OpenCV, 2019). Open Source Computer Vision Library (Open CV) is a software library released under the BSD license, which means that it is freely available for both academic and commercial purposes. It can be used in C++, Python, Java and MATLAB interfaces and is supported on many operating systems like Windows, Linux, Mac OS, iOS and Android. It has been designed for computational efficiency with a focus on real - time applications. It is written with optimized C/C ++, which allows it to use multi-core processing. The Open CV library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. It is necessary to install it by typing the *apt-get install python-OpenCV* command into the terminal to use this library. Program implementation includes many algorithms that use the functions of OpenCV library. The algorithms were designed to verify some image processing functions on the platform Raspberry Pi and can be applied especially to industrial processes for measuring objects such distance, radius and speed or detection of different objects such as detection circles or faces. Some examples of algorithms and principles we will describe in the following section.

Thresholding

Thresholding is an image processing operation that converts a grayscale image to binary. In the simplest implementation, the output is a binary image representing segmentation of processed image information. Segmentation is determined by a parameter called the T intensity threshold:

$$g(x, y) = \begin{cases} 0 & \text{for } f(x, y) \geq T \\ 1 & \text{for } f(x, y) < T \end{cases} \quad (1)$$

where:

$f(x, y)$ is the input grayscale matrix,

$g(x, y)$ is the threshold obtained,

T is defined threshold.

When the image is thresholded, each pixel of the input image matrix is compared to the threshold T . If on the other hand the intensity of the pixel is less than a selected threshold T , the value of respective binary pixel is set to 0 (Thresholding, 2019). In the practical application, we used the function to threshold the image: `cv2.threshold(src, thresh, maxValue, cv2.THRESH_BINARY)`; where, the individual arguments are functions: `cv2.threshold(src, thresh, maxValue, cv2.THRESH_BINARY)`; where, the individual arguments are: `src` – source image, `thresh` – value of threshold, `maxValue` – maximal value, `cv2.THRESH_BINARY` – type of thresholding method. It is possible using one of 5 methods of thresholding. For our application we choose the simplest and generally

using type of thresholding. Fig. 2 shows an original and thresholded image with a $T = 120$ threshold.

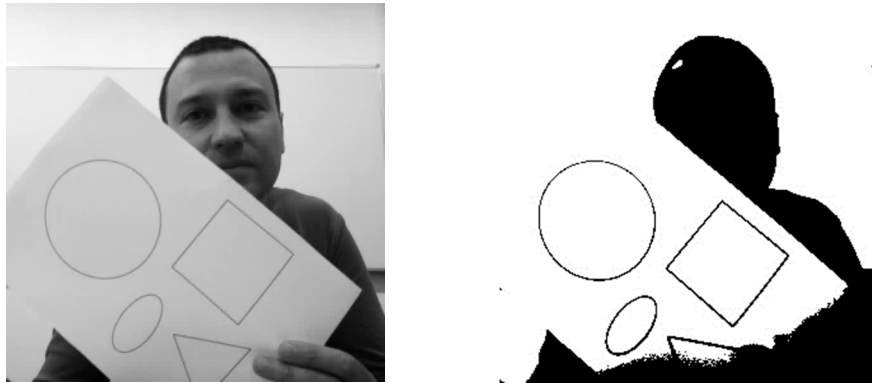


Fig. 2 The input image (left) and image after process of thresholding (right)

Color filtering

In a color image, each pixel is represented by three color values, for a red, green, and blue channel. Each of these values occupies 8 bits, and in the form of the processed data, it takes values from 0 to 255, meaning that in program execution it represents an `integer` data type (RGB color space, 2019). In the created subprogram, this can be used e.g. when is detecting color objects, the algorithm used a comparison and detection of the maximum RGB intensity for each pixel of the obtained image. The color of each pixel in the output image is set by assigning color to the maximum input image intensity.

Face detection using Haar cascades

Face detection is one of the fundamental applications used in face recognition technology. A computer program that decides whether an image is a positive image (with face image) or negative image (without faces image) is called a classifier. A classifier is trained on hundreds of thousands of face and non-face images to learn how to classify a new image correctly. OpenCV provides two types pre-trained and ready to use for face detection classifiers, LBP (Local Binary Pattern) and Haar Cascades. Both of these classifiers process images in gray scales, basically because it don't need color information to decide if a picture has a face or not. A Haar Cascade is basically a classifier which is used to detect particular objects from the source. For example, the `haarcascade_frontalface_default.xml` is a haar cascade designed by OpenCV to detect the frontal face (Face detection using OpenCV and Python, 2019). Object detection using Haar cascade classifiers is an effective method introduced by Paul Viola and Michael Jones in their publication (Face detection using Haar cascades, 2019).

In process image processing, for example, two features are extracting. The first one focuses on the property that the region of the eyes is often darker than the area of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. Most of these calculated functions are irrelevant. So we promptly discard irrelevant features and keep only those relevant with a fancy technique called AdaBoost. AdaBoost (Adaptive Boosting) is a training process for face detection, which selects only those features known to improve the

classification (face/non-face) accuracy of our classifier. In the end, the algorithm considers the fact that generally: most of the region in an image is a non-face region. Considering this, it's a better idea to have a simple method to check if a window is a non-face region, and if it's not, discard it right away and don't process it again. So we can focus mostly on the area where a face is. The concept of a classifier cascade has been proposed for this acceleration. Instead of applying all 6000 features to one window at a time, features are grouped into multiple stages of classifiers and applied one by one. If the window fails at the first stage is discarded if it meets the requirements, it proceeds to the second stage and thus continues the entire process. A window that goes through all stages is a face region. (Face detection using Haar cascades,2019). In the application in our program, at first the cascade classifier using the `cv2.CascadeClassifier()` function have been applied. Then the `detectMultiScale()` function was used. It is a general function for detecting objects, in our case it detects faces because we called it in conjunction with the Haar facial cascade (front view). If it finds faces, it returns a list of their positions in the form `Rect (x, y, w, h)`, which is use for plotting a square around the faces in the resulting image (Fig. 3 on the left) and at the same time it send this information to the control system.

Circle detection

The circle is mathematically defined by equation (2) where (x_center, y_center) are the coordinates of the circle center and r is its radius. The equation implies that three parameters are important in circular detection, which would lead to the use of a 3D accumulator for Hough transformation, which would be very inefficient. Hence, OpenCV uses the Hough gradient method that uses edge gradient information:

$$(x - x_{center})^2 + (y - y_{center})^2 = r^2 \quad (2)$$

Function `cv2.HoughCircles(image, method, dp, minDist)`, which we used in our application `CV_HOUGH_GRADIENT` method to detect circles in images. In Fig. 3 (on the right) we can see results of circle detection.

The output of the function is circles - the output vector of the found circles, each vector is encoded by 3 elements x coordinates (x_center), y coordinates (y_center) and radius (r) (Hough Circle Transform, 2019).

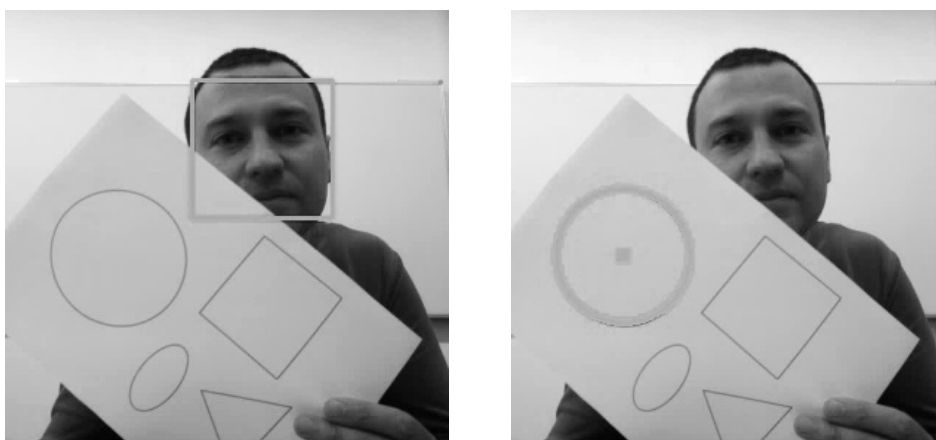


Fig. 3 Face detection using Haar cascade classifier (left) and Circle detection (right)

At the same time, the detected circles are rendered green and their rectangles centered in red. This output vector is send to the control system as the structured variable.

COMMUNICATION BETWEEN PLK STACK AND IMAGE PROCESSING SOFTWARE VIA SOCKET

In our application the socket programming is used to connect two programs with each other. The first program, written in C, realizes deterministic communication with PLC. Second program, written in Python, process the picture from the camera. It is necessary to share data between programs and send them to the PLC for process control. The socket applications are client-server applications, where one side acts as the server and waits for connections from clients. Fig. 4 shows the communication principle. OpenPOWERLINK stack program is the socket server and uses local IP and port 8888.

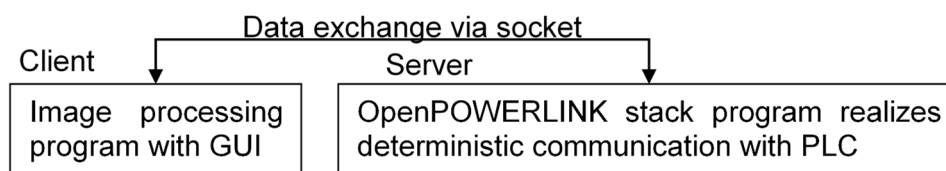


Fig. 4 Data exchange via socket between two programs

We have also created the GUI for testing the developed application with the basic image operations: capturing the original image, color filtration, face recognitions, edge detection, thresholding and circle detection with measurement.

CONCLUSION

In our paper, we described created application which used low-cost components and allows testing functionality of some image processing algorithms. For image processing, it has been used the following items: Raspberry Pi with HD camera and redundant modular B&R control system. The library used for image processing has been openCV and the programming language has been Python and C. The image processing consists of several stages, capture the image, process the image using the Raspberry Pi on the PLC demand and sending back the information. The programmed image processing system can be used in practice for measurement, checking product quality, sorting products according to given specification, or promoting production line safety or access control. This information processing solution was used in CNC Production Stand (as a Controlled Process) for the process control, monitoring the quality and safety functions that is part of the KEGA project. This solution fulfils all requirements specified in the main concept and it is usable for the industrial control systems.

ACKNOWLEDGEMENT

The paper has been written with the support of the project KEGA 016ŽU-4/2018 Modernization of teaching methods of management of industrial processes based on the concept of Industry 4.0.

REFERENCES

- Bruckner, D. (2018). OPC UA TSN - A new solution for industrial communication. B&R Industrial Automation. [online] Available at: https://www.automationworld.com/sites/default/files/opc_ua_tsn_whitepaper_1.pdf [Accessed 10 Apr. 2019].
- Bubeníková, E. (2015), The ways of streamlining digital image processing algorithms used for detection of lines in transport scenes video recording, In: PDES 2015: 13th IFAC and IEEE conference on Programmable devices and embedded systems: Cracow, Poland, pp. 174-179.
- B&R systems documentation. Redundancy. (2019). [online] Available at: <https://www.br-automation.com/en/technologies/redundancy/> [Accessed 10 Apr. 2019].
- Camelia library, C++ (2019), [online] Available at: <http://camellia.sourceforge.net> [Accessed 29 Apr. 2019].
- EPSG (The Ethernet POWERLINK Standardization Group). (2019). An Open Source POWERLINK protocol stack: openPOWERLINK. [online] Available at: <http://openpowerlink.sourceforge.net/web/> [Accessed 10 Apr. 2019].
- Face detection using OpenCV and Python: A beginner's guide. Superdatascience, [online] Available at: <https://www.superdatascience.com/opencv-face-detection/>, [Accessed 29 Apr. 2019].
- Face detection using Haar cascades. OpenCV python tutorials, [online] Available at: http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#face-detection, [Accessed 29 Apr. 2019].
- Halgaš, J., Pirník, R. (2015). Monitoring of parking lot traffic using a video detection. In: Acta Technica Corviniensis - Bulletin of engineering, 8(3), pp. 17-20.
- Hough Circle Transform. OpenCV Python tutorials, [online] Available at: http://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghcircles/py_houghcircles.html#hough-circles, [Accessed 29 Apr. 2019].
- Hrbček, J., Šimák, V., Janota, A. and Pirník, R. (2014). Tunnel central control system enhanced with modern control approaches. In: Archives of Transport System Telematics, 7(3), pp. 3-7.
- IEC 61131. (2019). Standard for programmable controllers. Parts 1-4.
- Janota, A., Šimák, V., Nemeč, D. and Hrbček, J. (2015). Improving the precision and speed of euler angles computation from low-cost rotation sensor data. Sensors.
- Automation Studio. (2019). 3rd-party devices. Bernecker + Rainer Industrie Elektronik GmbH Help Explorer.
- Hrbček, J., Šimák, V., Janota, A. and Pirník, R. (2014). Tunnel central control system enhanced with modern control approaches. In: Archives of Transport System Telematics, 7(3), pp. 3-7.
- Hruboš, M. (2016), Searching for collisions between mobile robot and environment, In: International journal of advanced robotic systems, 13(5), pp. 1-11.
- Koniar, D. (2017). Visual system-based object tracking using image segmentation for biomedical applications. Electrical Engineering, 99(4), pp. 1349-1366.
- Librare Open CV (2019), [online] Available at: <https://opencv.org> [Accessed 29 April. 2019].
- Library IPL-Image processing (2019). [online] Available at: <https://computervisiononline.com/software/1105138501>, [Accessed 29 Apr. 2019].
- Mičieta, B., Edl, M. and Krajčovič, M. (2015), Delegate MASs for coordination and control of one-directional AGV systems: a proof-of-concept, In: The International Journal of Advanced Manufacturing Technology [print], 94(1-4), pp. 415-431.
- OpenCv library, OpenCV, AdaptiveThreshold (2019) [online] Available at: https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#cv.AdaptiveThreshold [Accessed 29 April. 2019].
- Pirník, R. (2016), Integration of Inertial Sensor Data into Control of the Mobile platform. In: Advances in Intelligent Systems and Computing, pp. 271-282.
- RGB color space. Rapidtables. [online] Available at: https://www.rapidtables.com/web/color/RGB_Color.html, [Accessed 29 April. 2019].
- Sacimage - image processing library (2019). Available at: <https://www.tandfonline.com/doi/abs/10.1080/01431169308904451> [Accessed 29 Apr. 2019].
- Scikit-image, Image processing in Python library (2019). [online] Available at: <https://scikit-image.org/> [Accessed 29 Apr. 2019].

- Thresholding [online] Available at: <https://homepages.inf.ed.ac.uk/rbf/HIPR2/threshld.htm>, [Accessed 29 Apr. 2019].
- Zolotová, I. and Lorenčík, D. (2018), Object recognition in traffic monitoring systems, In.: DISA 2018 - IEEE World Symposium on Digital Intelligence for Systems and Machines, Proceedings, Article number 8490634, pp. 277-282, 1st IEEE World Symposium on Digital Intelligence for Systems and Machines, DISA 2018, Slovakia; Category number CFP18P13-ART; Code 141064.
- Ždánsky, J., Rástočný, K. (2014). Influence of Redundancy on Safety Integrity of SRCS with Safety PLC. In: Proceedings of the 10th international conference ELEKTRO 2014, Rajecké Teplice, pp. 508-512.

Abstract: This paper deals with the image processing from the camera for Raspberry Pi connected with real-time communication network to the control system (PLC). The low time delay for receiving and sending commands, data, etc. is very important in the automating production processes. This can be provided by industrial real-time network based on Ethernet. The Ethernet POWERLINK, which is supported on B&R PLCs, is one of them. It is a simple solution for a variety of applications because the POWERLINK is publicly available as the open source. Connecting the PLC and Raspberry Pi with Ethernet POWERLINK opens up many applications in industrial automation. For example, image data obtained using a camera attached to Raspberry Pi can be used to sense image of manufacturing processes and products and evaluate their quality in industrial automation. This article focuses on an image processing unit and the PLC system with CPU redundancy used in the industrial application. Vision systems are often used to improve products quality control, saving costs and time.

Keywords: image processing, PLC, measurement, Raspberry Pi, real-time, embedded system, openPOWERLINK