APPLICATION OF RASPBERRY PI HARDWARE AND PYTHON PROGRAMMING LANGUAGE IN SMALL PRODUCTION PLANTS

Boris LIČINA
Dejan VIDUKA
Luka ILIĆ

Abstract: *The paper is dedicated to the use of a very simple and accessible device that was created with the idea to serve as a platform for training computer science students. The Raspberry Pi did not claim to become a high-circulation product, but due to its performance, it became one. Primarily due to its low price followed by a wide range of possible practical applications. This device became very popular and enabled the development of IoT that are applied in various production processes. This paper presents a case study where this device is used in small production plants. The Python programming language is used to run this device, which is very simple but also very efficient. This combination of hardware and software has performed well in small, non-hardware-intensive manufacturing facilities. In addition to the incomparably more affordable price compared to commercial PLCs, it is often even a superior solution. The advantages are especially the easy integration into existing information systems and uncomplicated extensions and upgrades. It is also possible to integrate it into existing software solutions without major difficulties. The example described in this paper certainly represents a successful implementation that includes production planning, organization and monitoring of the production process (workflow) and worker efficiency.*
Keywords: *Raspberry Pi, microprocessor, microcomputer, Python, Linux.*

INTRODUCTION

Nowadays, smaller production plants are more and more common, in which there are unique, artisanal machines or machines that are improvised in order to adapt to the specific needs of production. Such machines generally meet the basic requirements of production, but not logistics (workflow), the solution of which requires very expensive hardware and software. In this paper we can see a financially favorable, but very acceptable solution for small production plants. (Hakim and Suthar, 2019) Of course there are complex hardware solutions offered by technology giants such as Siemens or others, but those are a very expensive solution for small production plants.

The hardware and software presented in this paper is equipment that is also used at the faculties for training young students, (Johnston and Cox, 2017) and thus provides professional staff who are familiar with its capabilities. (Barbosa et al., 2020; Pattichis et al., 2017)

This makes it easier to find a professional trained to design the solution in accordance with the requirements of smaller production facilities. In this way, a positive economic impact can be made through the employment of design, implementation and maintenance experts. The equipment used in this paper is very affordable (House and Cheer, 2019) and can be purchased at almost any specialty store.

The presented solution greatly facilitates the work of the entire staff: planners, storekeepers, employees and workers, and the collected data allow management to more easily monitor the performance of each individual worker, and efficient planning of production work. (Balsam et al., 2018)

The advent of microcomputers such as the Raspberry Pi in early 2012 (Imteaj et al., 2017) led to unimagined opportunities for use (Brand et al., 2017) in small production facilities. When the device was created, the basic idea was its application in education, but today its potential in production and logistics is becoming more and more pronounced. This device is very cheap and thus available to almost everyone. (Patil et al., 2017) When considering the design of a plant, costs are always taken into account, but also the reliability of the business and production process. It is this device that can provide great reliability when counting and packaging the desired amount of finished products. The need for additional control of packaged products is reduced, which reduces the need to engage additional labor in the production itself, but also in planning and warehousing operations for the implementation of the production plant. (Khatri et al., 2017) This device in combination with the Python programming language (Jaskolka et al., 2019) efficiently connects to SQLite or MS Access database, and can be a very powerful tool for small family production plants that generally have a smaller production volume and have less investment.

RASPBERRY PI AS THE AVAILABLE HARDWARE

Raspberry Pi was created in early 2012 (Kamalakannan & Devadharshini, 2019) at the University of Southampton with the idea of serving as a platform for training computer science students, *Figure 1*. (Hajjar and Spears, 2019) Although it did not claim to be a high-circulation product, over half a million copies of this microcomputer were sold in the first half of the year. Today, this number has exceeded 36 million copies sold. In addition to the low price, this system has the ability to interact with the environment by connecting various sensors and actuators to its GPIO port (General Purpose Input Output). In addition to the image in different resolutions, the HDMI output also supports audio transmission. It is equipped with an Ethernet port (RJ45) and USB ports (for connecting a keyboard, mouse, USB flash drive ...) (Ate and Abdelrahim, 2018), while it uses an SD card for the operating system and data storage. (Naik and Sudarshan, 2019)

Figure 1. Raspberry Pi

The causes of its popularity can be found in a wide range of its applications (Pardeshi et al., 2017) as a desktop computer with the use of the Internet, email, LibreOffice packages and all this on various variations of the Linux operating system. The primary operating system for the Raspberry Pi and also its most popular Linux (Hyun et al., 2019) distribution (based on Debian) is Raspbian which contains many integrated features. (Kumar and Jasuja, 2017) Of the programming languages, in addition to Python, (Jaskolka et al., 2019) C, C++, Java... (Carloni et al., 2018; Pereira et al., 2018) are represented, and it is completely open to the development of its own operating system. (Hajjar and Spears, 2019) Therefore, it is not surprising that it is so widely used, popular and widespread. It is interesting to use Raspberry Pi to manage "smart homes" (Ishak et al., 2017) thus raising the energy efficiency of the building, its safety and comfort of life (automation of heating, cooling, fire detection, movement, vibration, noise, light, hazardous gases ...). (Abed and Rahman, 2017)

The Python programming language is intuitive (Kamalakannan & Devadharshini, 2019) and as such designed for the youngest age. At first glance, one would think that as such it is not serious for more demanding applications. However, in every aspect he would be pleasantly surprised. Python is an object-oriented language and as such possesses classes, but it is an interpreter language so we can cite this as a drawback. Not only is it a powerful tool, but it has many modules (libraries, packages), (Abed and Rahman, 2017) allows you to create and write to various files (*.csv, *.txt) or read data from them, as well as share them with other computers. But that's not all, it also allows writing (or reading) directly to a SQLite or MS Access database.

Unlike PLCs (Đurić and Matić, 2019) which are able to perform a task via an interrupt routine (e.g. an asynchronous event of changing the state of a digital input), thus interrupting the execution of the main program using a stack, here it is possible to “listen“ and the processing of the above-mentioned change of the state of the digital input on the GPIO port is performed within a parallel running process, the so-called Thread, allowing the main program to run smoothly. This is essentially multi-tasking, not its (hardware or software) simulation, which is for such a simple (without multiprocessor cores, without memory, very modest performance) and cheap device, simply fascinating.

ADVANTAGES OF USING RASPBERRY PI

Here are some of the more significant benefits of using a Raspberry Pi device (Meachery and Nair, 2018):

- cost-effectiveness - this device is very cheap and easily accessible to everyone, and its use is very simple and well documented.
- high efficiency - it is very efficient, and the price of the device and maintenance is very favorable.
- simplicity - the device can be easily programmed using the Python programming language, as well as easily implemented for some purpose by IoT applications. (Merchant and Ahire, 2017)
- small size - this device has small dimensions that do not affect its performance and planned purpose.
- small energy consumer - this means that the power for its start-up and operation is very small and thus very efficient.

TERMS OF REFERENCE (CASE STUDY)

In one manufactory, there was a need to count products that are assembled by hand and placed in boxes. This need arose for several reasons:

- There are several different products, and each of them has a different number of pieces that need to be packed in one box (since that number is of the order of several hundred, it is impossible for the packing worker to keep records of this number). The current solution involves measuring the weight of the box with finished products, which requires additional time and is not reliable.
- There is also a need to measure the efficiency of each worker on the package (number of packaging products per unit time), or to measure his productivity.
- Good communication with the warehouse of semi-finished products (raw materials) is required, i.e. its "debt relief" (timely delivery of semi-finished products for packaging to an individual worker). It is assumed that it is necessary to record the finished products, i.e. "borrowing" the finished goods warehouse, which is an additional challenge.

As information about the productivity of workers and a specific type of his motivation, there was a need to print the currently packaged quantities on the TV screen.

HARDWARE SOLUTION

Considering the number of packing places (64 places), i.e. the number of necessary digital inputs (64) and digital outputs (64), as well as additional requirements in terms of warehouse operations and the need to inform workers about their current productivity at the shift level, we concluded that for this purpose affordable most acceptable solution with Raspberry Pi microcomputer. Each packing place must be assigned at least one digital input for counting packaged products and one digital output to indicate that the moment of filling the box with the specified number of products is approaching (with flashing LEDs), i.e. that number has been reached (LED lights up). Note that (in order not to increase the number of digital inputs) we have added one counter reset button in parallel to the sensor for counting packaged products when the desired number of products in the box is reached (counter reset is performed by pressing the button during 3 seconds). For information, the number of packaged products is about 10 to 20 per minute, depending on the type of product and the packaging skills of the individual worker (for one package the worker needs between 3 and 6 seconds). Possible ways of connecting LEDs and switches are shown in *Figure 2*.

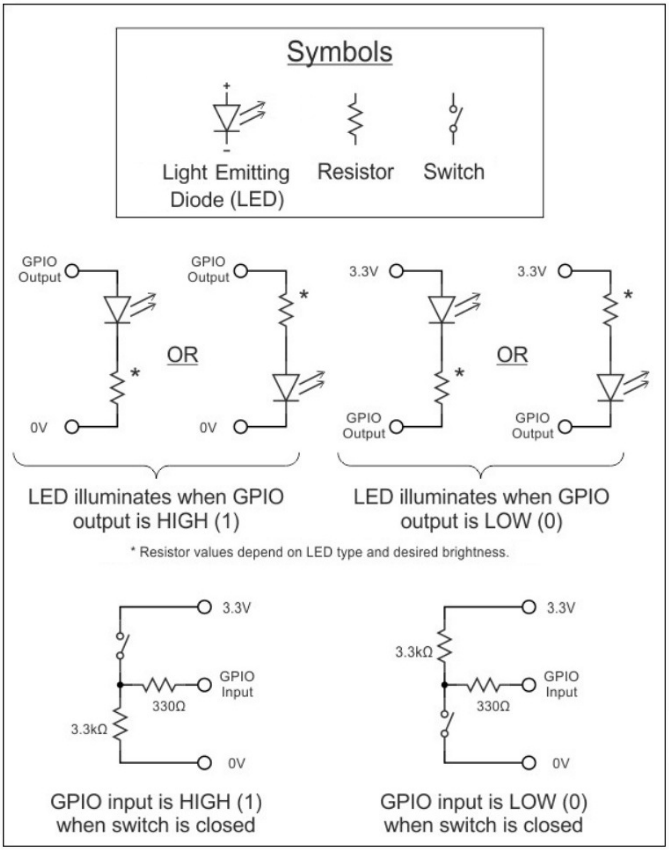

Figure 2. Ways to connect switches and LEDs on the GPIO port

One Raspberry Pi can serve 13 packing places via its GPIO port, which consists of 26 digital input/output pins (GPIO2 - GPIO27), i.e. it can provide as an example 13 digital inputs and 13 digital outputs, as shown in *Figure 3*.

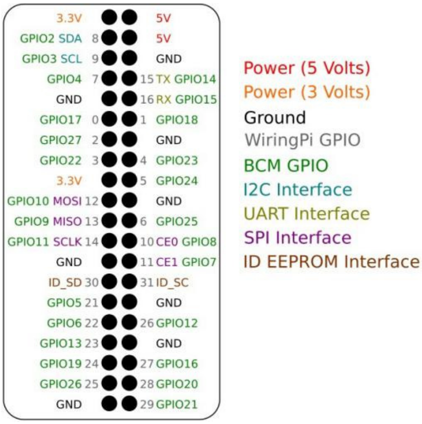

Figure 3. GPIO port

One of the solutions was to use 5 microcomputers for this purpose, which is quite an "awkward" system. Since this counting process is relatively slow, multiplexing of digital inputs (i.e. sensors with a button in parallel) and digital outputs (which turn on LEDs) is imposed as one of the solutions. For this purpose, we use shift (shift) registers - integrated circuits (IC) from the 74HC series (High Speed CMOS). To expand the number of digital outputs, we use SIPO (serial in - parallel out) IC in the designation 74HC595, whose functional diagram is shown in *Figure 4a*. Similarly, to expand the number of digital inputs, we use PISO (parallel in - serial out) IC in the designation 74HC165, whose functional diagram is shown in *Figure 4b*.

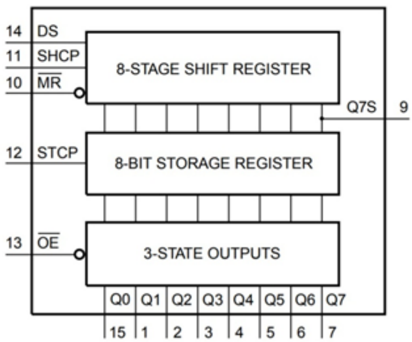

Figure 4a. Functional diagram of 74HC595

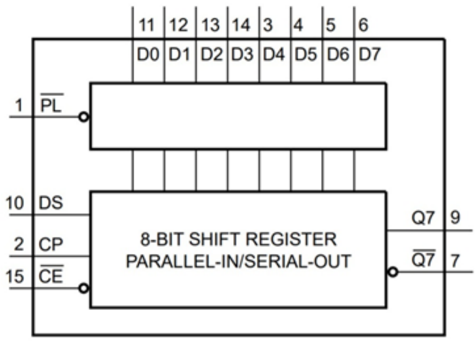

Figure 4b. Functional diagram of 74HC165

Multiplexing of buttons and sensors, as well as LEDs can be reported according to the scheme shown in *Figure 5*. The far right integral circuit 74HC595 performs row multiplexing (circular, from first to last) by setting (at one time) only one of its outputs to logical zero (low level). Of course at that point it is necessary to set to the logical high level the corresponding outputs on the middle IC circuit 74HC595 which corresponds to the LEDs that should light up. The far left IC 74HC165 will then detect a logical zero at the digital input corresponding to the key pressed in the row selected by the far right IR circuit 74HC595. In other cases, in order for this IC circuit not to register logical zero, one pull-up resistor is connected to each digital input, which will hold it on the logical high level when the key is not pressed. Note that the clock is common to all three integrated circuits, and that the IC 74HC595 has two additional control signals (except for the clock), while the IC 74HC165 has (in addition to the clock) three additional control signals. All control signals according to IC 74HC595 on the GPIO port side are digital outputs, as well as the signal by which data is transmitted in series (DS - pin 14). Also, all GPIO port control signals sent to IC 74HC165 are digital outputs, except for the signal that serially receives data (Q7 - pin 9) to the digital input of the GPIO port.

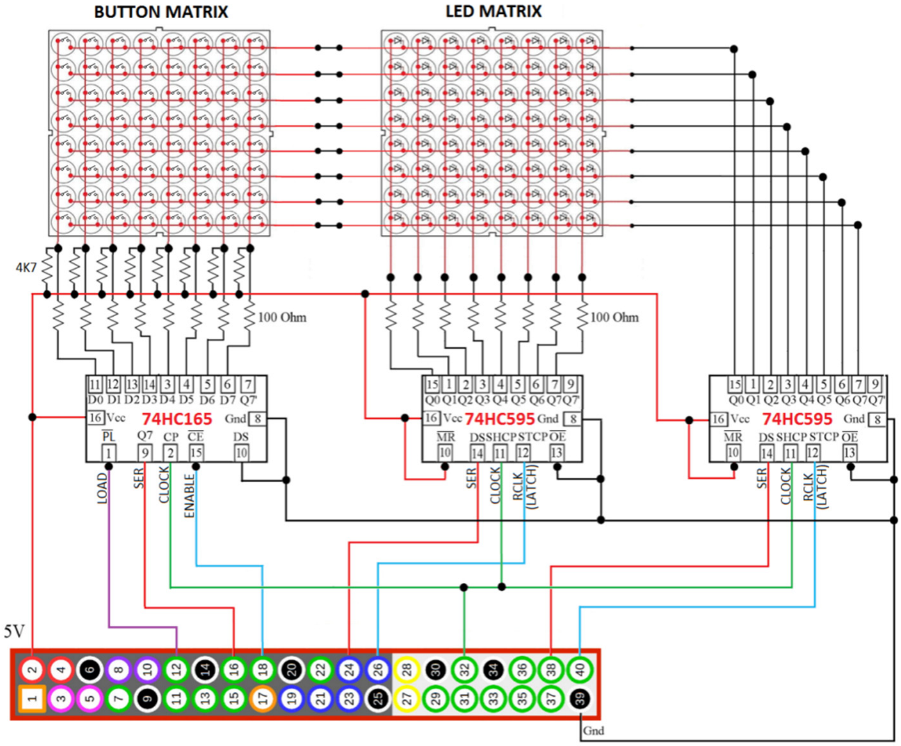

Figure 5. Multiplexing buttons and LEDs via GPIO port

Three pins (SH_CP - clock pin 11; ST_CP - latch pin 12; DS - serial data receive pin 14) are used to control 74HC595. If the latch pin is at logical zero, the outputs of the integrated circuit are disabled. The data is serially transmitted to the integrated circuit via a serial data pin at the clock determined by the clock pin. By placing the latch pin on the logical high level, the loaded data is forwarded to the parallel output (Q0-Q7). In case we want to connect more such integrated circuits in series, it is necessary to connect the output Q7' (pin 9) of the previous integrated circuit to the DS (pin 14) of the next integrated circuit.

Similarly, four pins are used to control the 74HC165 (PL - load pin 1; CP - clock pin 2; Q7 - serial data send pin 9; CE - clock enable pin 15). By setting the load pin, which works in inverted logic, the data from the parallel input to the buffer is loaded to logical zero (after a few microseconds, the load pin is placed on the logical high level). This is followed by the procedure of serial data transmission via a serial data pin. To enable the data sending function, it is necessary to set the clock pin to a logical high level, and the clock enable pin, which works in inverted logic, to a logical zero. The clock for sending data via the serial data pin is determined by the clock pin. After the transfer is completed, it is necessary to set the clock enable pin on the logical high level (until the next sending of data via the serial data pin). For the purpose of connecting several such integrated circuits in series, the output Q7' (pin 7) of the previous integrated circuit must be connected to the DS (pin 10) of the next integrated circuit.

COMMUNICATION WITH THE DATABASE AND DATA EXCHANGE

The control system and the server are connected by a direct Ethernet connection (in this case, IP address sticks are used, but this is not necessary - the system can be configured even if it works with DHCP address assignment). In order for the control system to meet the set conditions, for each of the 64 working positions, information must be available on the required number of packages of a given product in a box. In addition, each work position must have unambiguous information about the worker (name and surname) who works on it (worker ID), so that they can transmit this information (together with the number of packed boxes and the current number of products in the box to be filled) via HDMI to send to the TV screen, as shown in *Figure 6*.

WORKER POSITION	NAME OF WORKERS	NUMBER OF FILLED BOXES	QUANTITY IN BOX ON FILLING
Pos_01.	Mara Marković	14	253
Pos_02.	Mira Mirković	15	57
Pos_03.	Dara Marić	14	135
Pos_04.	Mileva Marić	15	11
Pos_05.	Gordana Perić	14	115
Pos_06.	Danijela Mandić	14	72
Pos_07.	Vera Mitrović	13	251
Pos_08.	Sanja Dakić	14	34
Pos_09.	Zora Zorić	13	245
Pos_10.	Sanja Škorić	13	137

Figure 6. Display of shift performance of an individual worker

On the other hand, the control system must have information on the number of boxes packed and, optionally, on the current number of products in the box to be filled for each working position. Information about the current number of products in the box being filled is not important, because the packaging process defines that each started box must be packed (with a given number of products) by the same worker who started the box packaging operation. For this purpose, we use the possibility of direct access to the SQLite database. Further, this data is processed on a PC and generated (predefined) reports on the efficiency of each individual worker on a daily (shift), weekly and monthly basis. The database on warehouse operations was also implementing (both the state of the input warehouse containing semi-finished products necessary for the packaging process and the state of stocks of finished products). This is necessary for planning the packaging process and creating work orders, but it is not an integral part of the project described here.

CONCLUSION

The aim of this paper is to demonstrate the capabilities of the Raspberry Pi hardware device in combination with the Python programming language in designing a comprehensive production system, including planning, organization and monitoring of production (workflow). There are many reasons for application of this solution, for example reliability, accuracy, simplicity and last but not least price, which is very important for small production plants. This paper shows the possibility that by simply applying available technologies through a relatively small investment, much can be done to automate production and reduce labor costs. This type of automation increases productivity, which ultimately brings higher revenue to the manufacturer.

The proposed system has a favorable price, it is compact and very reliable, and when combined with certain sensors can improve production in many segments. By applying such a system, an almost perfect compromise is achieved between cost and reliability in production. Of course, there are reliable manufacturers of sophisticated equipment whose high prices are an obstacle for small manufacturers or there is no justification for investing in it.

In addition to the hardware itself, the application of the Python programming language is very important and in this synergy it becomes a very powerful tool for implementation in an industrial environment.

The main goal of this research was to identify the key benefits and challenges that the industry can expect in case of implementation of such a solution. The goal is also to encourage other researchers to initiate and dare small producers through their projects to improve the production process and to increase the efficiency, quality and safety of the final product with relatively modest funds, and at the same time create savings.

References

Abdelrahim Ate and Mohamed Abdelrahim. *Controlling the Temperature Reactor Based on Raspberry Pi System Control.* 5th International Conference on Electrical and Electronics Engineering. 2018.

Ahmed Imteaj, Tanveer Rahman, Muhammad Kamrul Hossain, Mohammed Shamsu Alam and Saad Ahmad Rahat. *An IoT based Fire Alarming and Authentication System for Workhouse using Raspberry Pi 3.* International Conference on Electrical, Computer and Communication Engineering (ECCE), February 16-18, 2017, Cox's Bazar, Bangladesh

Andrea Carloni, Federico Baronti, Roberto Di Rienzo, Roberto Roncella and Roberto Saletti. *Open and Flexible Li-ion Battery Tester Based on Python Language and Raspberry Pi.* Electronics 2018, 7, 454; doi: 10.3390/electronics7120454

Ali A. Abed and Sarah A. Rahman. *Python-based Raspberry Pi for Hand Gesture Recognition.* International Journal of Computer Applications (0975 – 8887), Volume 173 – No.4. 2017.

Balsam Qasim, Suha M. Hadi and Faiz F. Mustafa. *Design and Implementation an Industrial Application System by using Internet of Things (IOT).* Innovative Systems Design and Engineering, Vol.9, No.1, 2018

Dobrivoje Đurić and Vladimir Matić. *Computer Vision in Industry, Data Science & Digital Broadcasting Systems.* SINTEZA 2019, Intrnational Scientific Conference on Information Technology and Data Related Research, DOI: 10.15308/Sinteza-2019-439-444. 2019.

Gabriel H. Cassel Barbosa, Marcus Varanis, Kelven M. S. Delgado, Clivaldo de Oliveira. *An acquisition system framework for mechanical measurements with Python, Raspberry-Pi and MEMS sensors.* Revista Brasileira de Ensino de Física, vol. 42, e20200167 (2020), DOI: https://doi.org/10.1590/1806-9126-RBEF-2020-0167

House, Charlie and Cheer, Jordan. *Efficient Raspberry Pi implementation of a SISO active control system using python*, University of Southampton 5pp. 2019.

Isaiah Brand, Josh Roy, Aaron Ray, John Oberlin and Stefanie Tellex. *PiDrone: An Autonomous Educational Drone using Raspberry Pi and Python.* IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2018.

Kamalakannan & Devadharshini. *Controlling the Speed of Conveyor Belt using Python Raspberry Pi 3B+*. Orient. J. Comp. Sci. & Technol., Vol. 12(2) 57-64. 2019.

Karina Jaskolka, Jürgen Seiler, Frank Beyer, Andre Kaup. *A Python-based laboratory course for image and video signal processing on embedded systems.* Heliyon 5. 2019.

Lalitha Navneet Meachery and Rijul Nair. *Raspberry PI Based Cyber-Defensive Industrial Control System With Redundancy And Intrusion Detection.* International Journal of Pure and Applied Mathematics, Volume 118 No. 20. 2018, 4273-4278.

Marios S. Pattichis, Sylvia Celedon-Pattichis and Carlos Lopez Leiva. *Teaching Image and Video Progressing Using Middle-School Mathematics and the Raspberry Pi.* IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2017.

Merchant K. H. and D. D. Ahire. *Industrial Automation using IoT with Raspberry Pi.* International Journal of Computer Applications (0975 – 8887), Volume 168 – No.1. 2017.

Neha Patil, Shrikant Ambatkar and Sandeep Kakde. *IoT Based Smart Surveillance Security System using Raspberry Pi.* International Conference on Communication and Signal Processing, India. 2017.

Ishak S.N., N.N.N.Abd Malik, N.M. Abdul Latiff, N. Effiyana Ghazali, M. A. Baharudin. *Smart Home Garden Irrigation System Using Raspberry Pi.* IEEE 13th Malaysia International Conference on Communications (MICC), 28-30 Nov. 2017, The Puteri Pacific, Johor Bahru, Malaysia.

Punit Khatri, Karunesh Kumar Gupta and Raj Kumar Gupta. *Raspberry Pi-based smart sensing platform for drinking-water quality monitoring system: a Python framework approach.* Drink. Water Eng. Sci., 12, 31–37, https://doi.org/10.5194/dwes-12-31-2019

Renata I.S. Pereira, Ivonne M. Dupont, Paulo C.M. Carvalho, Sandro C.S. Jucá. *IoT embedded linux system based on Raspberry Pi applied to real-time cloud monitoring of a decentralized photovoltaic plant.* Measurement 114 (2018) 286–297

Salam Hajjar, Trenton Spears. *Hardware Microprogramming Education Using Raspberry PI and Arduino Technologies*. International Journal of Intelligent Information Systems. Vol. 8, No. 2, 2019, pp. 47-51. doi: 10.11648/j.ijiis.20190802.12

Seena Naik and E. Sudarshan. *Smart Healthcare Monitoring System Using Raspberry Pi on IoT Platform.* ARPN Journal of Engineering and Applied Sciences, VOL. 14, NO. 4. 2019.

Somansh Kumar and Ashish Jasuja. *Air Quality Monitoring System Based on IoT using Raspberry Pi.* International Conference on Computing, Communication and Automation (ICCCA2017). 2017.

Steven J Johnston and Simon J Cox. *The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams.* Electronics 2017, 6, 51; doi: 10.3390/electronics6030051

Vivek Pardeshi, Saurabh Sagar, Swapnil Murmurwar and Pankaj Hage. *Health Monitoring Systems using IoT and Raspberry Pi – A Review.* International Conference on Innovative Mechanisms for Industry Applications (ICIMIA 2017)

Zishan Hakim and A.C. Suthar. *An IOT based industrial machine monitoring and controlling system using raspberry pi.* Journal of Emerging Technologies and Innovative Research (JETIR), Volume 6, Issue 4. 2019.

Wook Hyun, Mi Young Huh and Ju Young Park. *Implementation of greenhouse service control protocol using Python on Raspberry PI.* International Conference on Advanced Communications Technology (ICACT). 2019.

NOTES ON THE AUTHORS

Boris S. Licina was born in Bihac, former Yugoslavia, in 1967. He received the B.Sc. degree in electrical engineering from the University of Novi Sad, Novi Sad, Serbia, in 1993, and the M.Sc. and Ph.D. degrees from the University of Novi Sad in 2013 and 2020, respectively. From 1993 to 1996 he has been a teaching at Technical High School Mihajlo Pupin in Novi Sad, the courses involving programming and electrical measurements. From 1996 to 2020 he has been working in Apatin brewery pioneering the introduction of automation and control equipment in the production process. He is currently an Assistant Professor with the University Business Academy of Novi Sad. His current research interests include programming, A/D conversion techniques, and applications of statistical signal-processing techniques in the measurement science.

Dejan Viduka was born in Osijek, Croatia in July 30, 1980. Currently he is a Assistant professor of Computer Science Department at the Faculty for Applied Management, Economics and Finance, Belgrade, University Business Academy, Novi Sad, Serbia. Having 23 years of experience at IT sector (hardware, software, SEO, Internet, e-business, e-marketing, operating systems, CMS systems and Open Source systems) and extensive experience in the development of Open Source projects, and is also a member of Elxis Community and provides expert assistance to customers from Croatia and Serbia. As an IT expert, he worked on many projects for well-known clients from Serbia and from abroad.

Luka Ilić was born in 1998 in Bijeljina, Republika Srpska. Since 2014 he has been working as freelancer in information technology. Graduated from the Faculty of applied management, economics and finance (MEF), department of Information technology. He is currently studying for a master's degree in information technology. Works as a teaching associate at MEF faculty and as a freelancer in web technologies. He has an interest in technology, sports and adventure.