Industrial Robotic Automation with Raspberry PI using Image Processing

Roland Szabó

Applied Electronics Department
Fac. of Electronics and Telecom., Polithenica Univ.
Timisoara, Romania
roland.szabo@etc.upt.ro

Aurel Gontean

Applied Electronics Department
Fac. of Electronics and Telecom., Polithenica Univ.
Timisoara, Romania
aurel.gontean@upt.ro

Abstract – **The aim of this paper is to present a system powered by a Raspberry PI and an industrial robotic arm, which can execute numerous task in a factory. The used robotic arm and imaging devices are industrial type, but the control system it's a simple development microprocessor board. The robotic arm it's the SCORBOT-ER III and the imagining devices are IP cameras. The goal it's to demonstrate that even complex industrial systems can be run with modest and small card-size pocket computing devices. Final scope is to prove, that with only one microchip, smart autonomous robotic systems, can be created.**

Keywords-actuator; automation; camera; end effectors; image processing; industrial control; Raspberry PI; robotic arm.

I. INTRODUCTION

In the industry a big amount of robotic automation is needed due to the fact that the products are very different, because of many configuration options [1]. The dynamics of the industrial line has grown in the past year, this way many production lines have to be reused and reconfigured [2]. It takes o lot of time and effort to do this manually and to reconfigure robotic arms to the new process. If a video system is used robotic arms can reconfigure themselves for the new production lines, this way it will be no problem when changes are involved in the industrial process [3].

The goal is to include as much automation a possible in the production line, with a computing system with the lowest possible resources and still have no compromise in the results. The production line is an industrial production line with industrial robotic arms and industrial IP cameras [4]. The only simple system is the control computing device which eventually tends to be integrated on a single chip design [5].

The production line it's made from a SCORBOT-ER III educational/industrial robotic arm and the control cameras are industrial IP cameras. The task is to pick with the help of the robotic arm objects with random shape, color, size or position in space. These objects than has to be placed in a random place, where the recipient is placed. The robotic arm follows the objects, even if it's moved from its position and makes the sorting in the specific recipient even if the recipient it's moved.

This is quite new in a production line, because the size, shape, color and place of the objects are specified before and the recipient where it's placed its preset [6]. In a production line if a minor change is made, or the system ships due to numerous repetitions, the whole line has to be reprogrammed. The camera controlled production line reprograms in real-time during execution [7]. At each execution the position of the joints are reprogrammed in order to reach the target position are precise as possible [12]. The control is made with stereo cameras in order to be able to measure distances too with stereo triangulation [13]. The whole system is made in a way that as a human would be in place of the production line. If the place of the object is changed, the human can readapt and be able to pick it and place it correctly in the recipient [14]. The eye system (the two cameras) are place in the side of the production line (the robotic arm it's seen from the side) and not at the gripper of the robotic arm, because the eyes of the human operator are not at his fingers, the hand it's seen by the side, the eyes are in the head [15].

The human body it's the perfect system, which has to be a model for the robots. The human body has to be always an inspiration to the robot builders. The rate of the similarity of the robotic system with the human body will measure the rate of how versatile is the robotic system.

II. PROBLEM FORMULATION

The goal of the system is to create a small industrial environment, where a robotic arm can be controlled only based on the data received from the cameras. This data is processed and the robotic arm can move accordingly.

The system has to be implemented in a loop, so no human intervention is allowed. This means that the robotic arm can readapt to the changes of the environment and to the changes of the objects shape, size, color and position in space. The robotic arm can auto just itself and compensate its offset in order to move always accurate and precise.

The whole control device of the system it's a small Raspberry PI. This computing system it's used due to the fact to prove that the whole industrial system can be automated just with a card size computing system or a microchip.

III. System Implementation

A. Harware Setup

The used robotic arm it's the SCORBOT-ER III.

The video system which it's used is the TP-LINK TL-SC3230 type IP camera.

The SCORBOT-ER III robotic arm can be controlled using SCPI commands on the RS-232 serial port. There is needed a conversion from 25 pin to 9 pin serial cable, but no other adaptation is needed, due to the fact that the used protocol it's the standard RS-232 protocol. It can be used an USB to serial converter too if the control computer has no serial port, which is the case of the Raspberry PI. The Digitus USB to serial converter it's a good and tested device. It's recognized by the Raspberry PI; therefore n driver installation is needed. For further enhancements Bluetooth communication can be used too, and the extra circuitry is not so complex, due to the fact that there it's no need to change the transmission protocol.

On Fig. 1 there is the block diagram of the test setup. There is the Raspberry PI model B board, but the Raspberry PI V2 board can be used too. The second board would be better due to the fact that it has more USB ports and a better CPU, 900 MHz quad-core ARM Cortex-A7and it has 1 GB RAM. The fact that it's used the old Raspberry PI it's because at the time of the implementation the Raspberry PI V2 board was not available. The usage of the old Raspberry PI, with the 700 MHz single-core ARM1176JZF-S CPU and 512 MB RAM, it's a good proof that even on this system, with modest resources, the control of the robotic arm, with image processing, it's possible. To the Raspberry PI it's connected a display on HDMI interface. This can be any type of display, if it's digital and hasn't got HDMI input a simple HDMI to DVI convertor can be used. If the display is analogical and

ahs only VGA input, there is also possible to use them with an active HDMI to VGA convertor, one like this it's the Pi-View. To the Raspberry PI it needs to be connected a mouse and a keyboard, which can be almost any type of peripheral device on USB port. The Raspberry PI model B has only two USB ports, so for more peripheral devices a USB HUB has to be used, one good tested USB HUB it's the D-Link H4, but the main idea is to have an externally powered USB HUB, due to the fact that enough current is drained by the Raspberry PI and there is not a good idea to power too many devices on the USB ports. The Raspberry PI V2 has 4 USB ports, so the usage of a USB Hub it's optional. One of the main devices would be the Digitus USB to serial converter, which is one of the best solutions due to the fact that this converter is compatible with many operating systems and for the Raspbian Linux used by the Raspberry PI there is no need to install any drivers, the converter it's recognized on the fly. With other USB to serial converters there were some problems, but this one works very well. The Raspberry PI has also UART communication on the GPIO pins. Those are not so simple to use, due to the fact that they need level shifters, because the logic levels are on 3.3 V, so if level shifters are not used, the GPIO pins can be damaged if used on 5 V TTL level.

The TP-LINK TL-SC3230 IP cameras are connected via a D-Link DGS-1024D Ethernet switch to the Ethernet port of the Raspberry PI. For this connection any Ethernet switch can be used and if a PC is the control computing system, than a configuration with more network cards it's viable too.

The last and main element of the setup it's the SCORBOT-ER III educational/industrial robotic arm which it's connected to the Digitus USB to serial converter through the RS-232 converter from 25 pins to 9 pins.
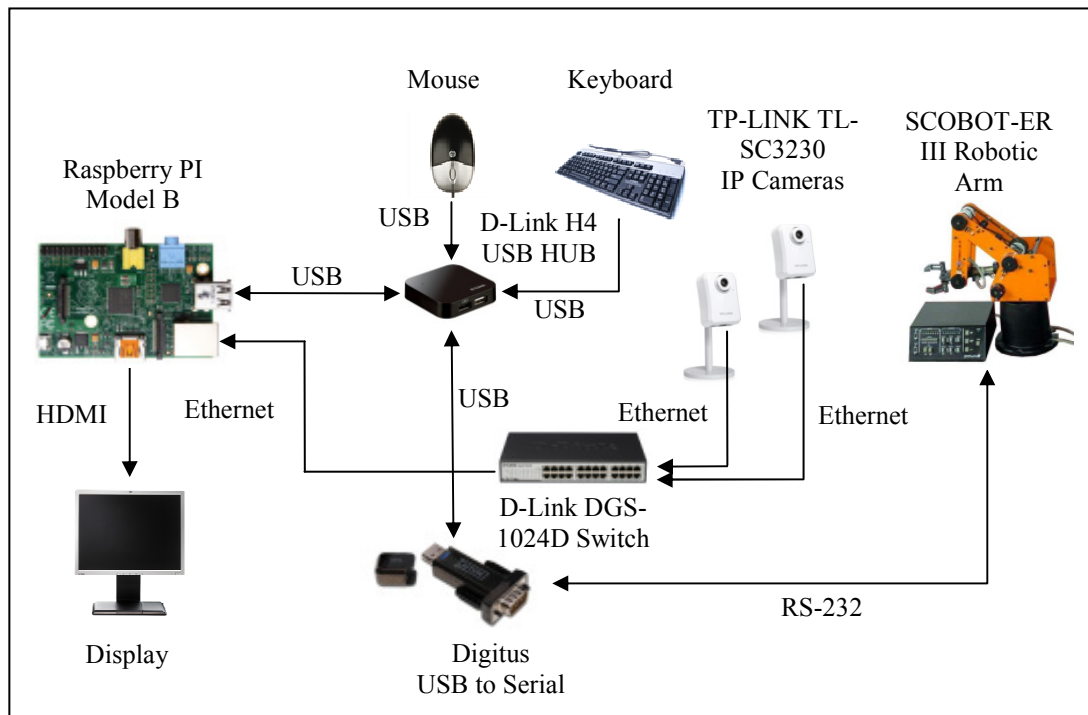

Figure 1.   Block Diagram of the test setup.

B. Theoretical Background

The difference between vectors we calculate the following way.

$$v_x = x_2 - x_0 \tag{1}$$

$$v_y = y_2 - y_1 \tag{2}$$

The vector length we calculate with the following formula.

$$\|l\| = \sqrt{x^2 + y^2} \tag{3}$$

We can compute the orthogonal vector in the following way.

$$temporary = x \tag{4}$$

$$x_0 = y \tag{5}$$

$$y_0 = -temporary \tag{6}$$

$$x_4 = v_{x_{0x}} - 2 \cdot v_{x_{0x}} + x_2 \tag{7}$$

$$y_4 = v_{x_{0y}} - 2 \cdot v_{x_{0y}} + y_2 \tag{8}$$

$$x_5 = v_{y_{0x}} - 2 \cdot v_{y_{0x}} + x_2 \tag{9}$$

$$y_5 = v_{y_{0y}} - 2 \cdot v_{y_{0y}} + y_2 \tag{10}$$

After this only the parallelogram computation is needed.

C. Software Implementation

On Fig. 4 there is the implementation of the SCORBOT-ER III robotic arm control tool made in Python and running on a Raspberry PI. The operating system is the Raspbian 7.0 "Wheezy". The image processing algorithms are made with the use of the OpenCV libraries.

The first image is the real-time acquisitioned image, with the overlays of the circles and lines, which are guidelines for computing angles and distances for system to know how much to move the robotic arm in the target position. The second image is the HSV (Hue, Saturation, and Value) filtering. The next four images are the color filtering windows for each bottle stopper glued to the robotic arm's joints. There is the blue bottle stopper glued to the base, there is the yellow bottle stopper glued to the elbow, there is the red bottle stopper glued to the gripper and there is the green bottle stopper at the target object. The color filtering windows are made for setting the HSV minimal and maximal values, this will filter the specific colored bottle stopper. These values have to be calibrated for every lighting condition. If histogram equalization filters are used, than the HSV values have to be configured only once, after this these values can be used in various lighting conditions. The calibration has to be done only once, when the system it's set up.

On Fig. 5 there is the zoomed image of the initial robotic arm image with the line a circle overlay for computing the movement distance of the robotic arm. For stereo image the tool has to be duplicated.
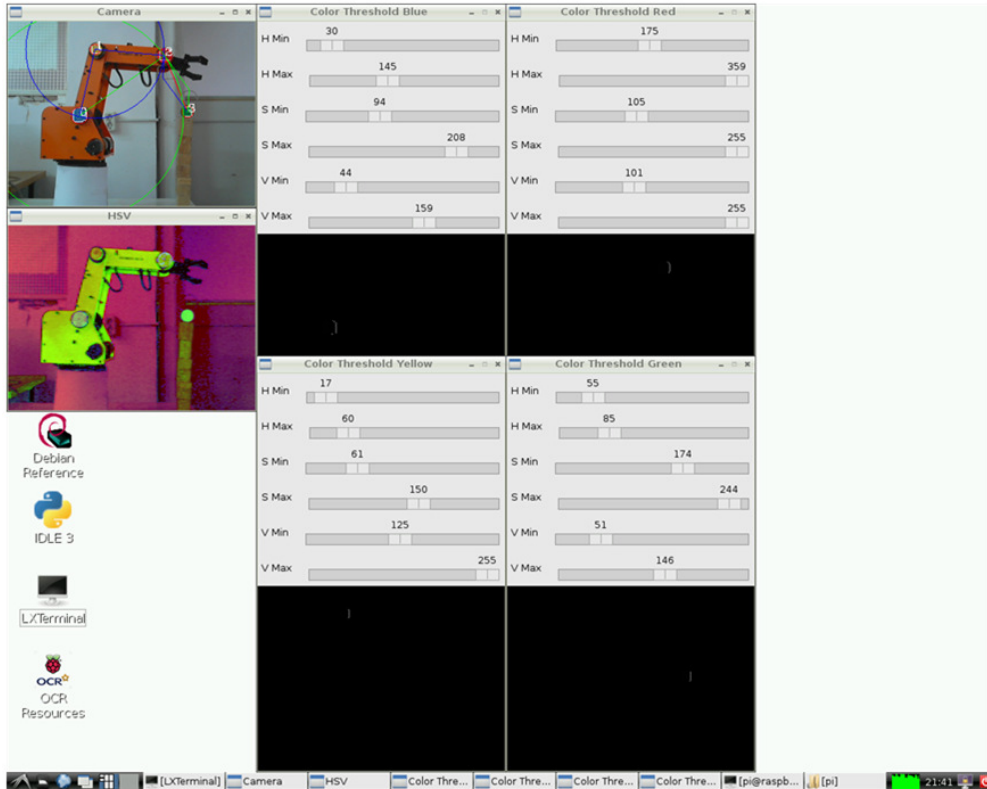

Figure 2.   SCORBOT-ER III robotic arm control GUI made in Python and running on a Raspberry PI.
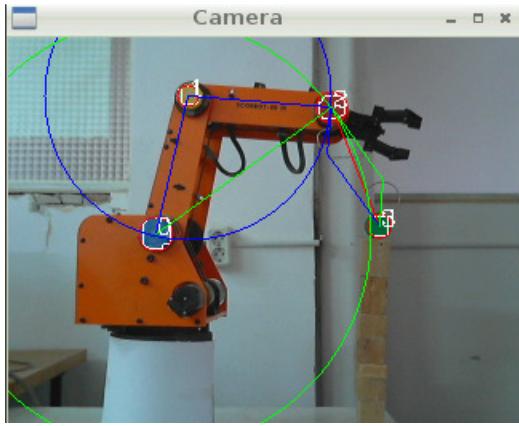
Figure 3. Zoomed image of the SCORBOT-ER III robotic arm with the overlays on the image after detection of the joints.

IV. CONCLUSION

As it was presented, an industrial robotic arm control system was created, using the SCORBOT-ER III educational/industrial robotic arm and industrial IP cameras. The control system it's a simple Raspberry PI, which is the whole control system for this automated environment. The robotic arm can hook the object marked with the green bottle stopper and place it in the correct recipient, regardless of its position. After tests it was proved that the Raspberry PI V2 it's more suitable for multi camera control systems, because of the increased core number and frequency of its CPU, while the Raspberry PI model B it's suitable mostly for one camera systems and 2D control, due to its modest CPU. Even though it was proved that such a simple system can control and industrial environment and make a full automation.

Further enhancement would be to port the system on FPGA boards like the ZedBoard of ZYBO, which can host an Ubuntu Linux operating system, this way the porting of the system it's straight forward.

The porting on the FPGA boards it's just an intermediary step form the final goal, to create an ASIC form the system, so the complicated automated industrial control can be packed din a single chip. With this tool chain, almost any tool can be ported, on a single chip.

REFERENCES

[1] W. G. Hao, Y. Y. Leck, L. C. Hun, "6-DOF PC-Based Robotic Arm (PC-ROBOARM) with efficient trajectory planning and speed control," 4th International Conference On Mechatronics, Kuala Lumpur, 2011, pp. 1–7.

[2] W. Yang, J. H. Bae, Y. Oh, N. Y. Chong, B. J. You, S. R. Oh, "CPG based self-adapting multi-DOF robotic arm control," International Conference on Intelligent Robots and Systems, Taipei, 2010, pp. 4236–4243.

[3] E. Oyama, T. Maeda, J. Q. Gan, E. M. Rosales, K. F. MacDorman, S. Tachi, A. Agah, "Inverse kinematics learning for robotic arms with fewer degrees of freedom by modular neural network systems," International Conference on Intelligent Robots and Systems, 2005, pp. 1791–1798.

[4] N. Ahuja, U. S. Banerjee, V. A. Darbhe, T. N. Mapara, A. D. Matkar, R.K. Nirmal, S. Balagopalan, "Computer controlled robotic arm," 16th IEEE Symposium on Computer-Based Medical Systems, New York, 2003, pp. 361–-366.

[5] M. H. Liyanage, N. Krouglicof, R. Gosine, "Design and control of a high performance SCARA type robotic arm with rotary hydraulic actuators," Canadian Conference on Electrical and Computer Engineering, St. John's, CA, 2009, pp. 827–832.

[6] M. Mariappan, T. Ganesan, M. Iftikhar, V. Ramu, B. Khoo, "A design methodology of a flexible robotic arm vision system for OTOROB," International Conference on Mechanical and Electrical Technology, Singapore, 2010, pp. 161–164.

[7] H. Guo-Shing, C. Xi-Sheng, C. Chung-Liang, "Development of dual robotic arm system based on binocular vision," International Automatic Control Conference, Nantou, 2013, pp. 97–102.

[8] R. Szabó, A. Gontean, "Controlling a Robotic Arm in the 3D Space with Stereo Vision," 21th Telecommunications Forum, Belgrade, 2013, pp. 916–919.

[9] R. Szabó, A. Gontean, "Robotic arm control in 3D space using stereo distance calculation," International Conference on Development and Application Systems, Suceava, 2014, pp. 50–56.

[10] R. Szabó, A. Gontean, "Remotely Commanding the Lynxmotion AL5 Type Robotic Arms," 21th Telecommunications Forum, Belgrade, 2013, pp. 889–892.

[11] R. Szabó, A. Gontean, "Creating a Programming Language for the AL5 Type Robotic Arms," 36th International Conference on Telecommunications and Signal Processing, Rome, 2013, pp. 62–65.

[12] M. Seelinger, E. Gonzalez-Galvan, M. Robinson, S. Skaar, "Towards a robotic plasma spraying operation using vision," IEEE Robotics & Automation Magazine, vol. 5, issue 4, 1998, pp. 33–38, 49.

[13] R. Kelly, R. Carelli, O. Nasisi, B. Kuchen, F. Reyes, "Stable visual servoing of camera-in-hand robotic systems," IEEE/ASME Transactions on Mechatronics, vol. 5, issue 1, 2000, pp. 39–48.

[14] V. Lippiello, F. Ruggiero, B. Siciliano, L. Villani, "Visual Grasp Planning for Unknown Objects Using a Multifingered Robotic Hand", IEEE/ASME Transactions on Mechatronics, vol. 18, issue 3, 2013, pp. 1050–1059.

[15] M. Kazemi, K. K. Gupta, M. Mehrandezh, "Randomized Kinodynamic Planning for Robust Visual Servoing", IEEE Transactions on Robotics, vol. 29, issue 5, 2013, pp. 1197–1211.

[16] R. T. Fomena, O. Tahri, F. Chaumette, "Distance-Based and Orientation-Based Visual Servoing From Three Points", IEEE Transactions on Robotics, vol. 27, issue 2, 2011, pp. 256–267.

[17] N. C. Orger, T. B. Karyot, "A symmetrical robotic arm design approach with stereo-vision ability for CubeSats," 6th International Conference on Recent Advances in Space Technologies, Istanbul, 2013, pp. 961–965.

[18] F. Medina, B. Nono, H. Banda, A. Rosales, "Classification of Solid Objects with Defined Shapes Using Stereoscopic Vision and a Robotic Arm," Andean Region International Conference, Cuenca, 2012, pp. 226.

[19] M. Puheim, M. Bundzel, L. Madarasz, "Forward control of robotic arm using the information from stereo-vision tracking system," 14th International Symposium on Computational Intelligence and Informatics, Budapest, 2013, pp. 57–62.

[20] T. P. Cabre, M. T. Cairol, D. F. Calafell, M. T. Ribes, J. P. Roca, "Project-Based Learning Example: Controlling an Educational Robotic Arm With Computer Vision," IEEE Revista Iberoamericana de Tecnologias del Aprendizaje, vol. 8, issue 3, 2013, pp. 135–142.

[21] G. S. Gupta, S. C. Mukhopadhyay, M. Finnie, "WiFi-based control of a robotic arm with remote vision," Instrumentation and Measurement Technology Conference, Singapore, 2009, pp. 557–562.

[22] L. Haoting, W. Wei, G. Feng, L. Zhaoyang, S. Yuan, L. Zhenlin, "Development of Space Photographic Robotic Arm based on binocular vision servo," Sixth International Conference on Advanced Computational Intelligence, Hangzhou, 2013, pp. 345–349.