AKADÉMIAI KIADÓ

UNIVERSITY of DEBRECEN

International Review of Applied Sciences and Engineering

14 (2023) 2, 212–219

DOI:
10.1556/1848.2022.00506

ORIGINAL RESEARCH PAPER

Readjustment of a robot for educational purposes

Sándor Imre Szabó and László Rónai*

Faculty of Mechanical Engineering and Informatics, Institute of Machine Tools and Mechatronics, University of Miskolc, Miskolc-Egyetemváros 3515, Hungary

Received: April 11, 2022 • Accepted: June 17, 2022
Published online: December 22, 2022

ABSTRACT

Development of control of a five degrees of freedom robot is discussed in this paper. Only two robots were made, and one of these is in the University of Miskolc. The robot was made in the 80s for educational purposes, the electronic components were obsolete and control software was missing, therefore it became necessary to perform hardware improvements, and develop a new control program. Inverse kinematics problem of the robot is solved by geometric approach to formulate the joint angles, which will form the basis of the control. A braking system containing electromagnets for the robot is constructed to balance it. A printed circuit board is designed to establish the control of the system, the central element is a Cypress PSoC 5LP platform. The development of the control program is performed in software PSoC Creator 4.4. The developed control system of the robot can receive instructions from a computer via a designed special purpose user program, which is written in Python programming language. Thanks to the improvements, the robot has become operational. Thus, the robot can serve educational purposes performing different manipulation tasks. By completing the developments, students can get to know the structure and programming of the robot, its inverse kinematics problem. This will require the development of practice-oriented tasks in the future.

KEYWORDS

robot, inverse kinematics, microcontroller, PSoC platform

1. INTRODUCTION

One of the focus areas of the fourth industrial revolution is the widespread use of robots. Thus, the role of robots in the industry 4.0 concept is very important, e.g., [1] recognizes this aspect, highlighting the role of collaborative robots. Nowadays, various processes in factories, e.g., assembly, workpiece manipulation or technological processes, are increasingly performed by robots. Therefore, there is a big demand from industry for robot programmers as a result teaching robot programming in universities will get more attention. In [2] a "Cyber-physical and intelligent robot systems laboratory" is mentioned, one of its aspects is to make available the programming and handling of robots for students. A self-devised 2 degrees of freedom (DOF) robotic arm and a control system for teaching are discussed in [3]. With a desktop application it can be feasible to implement different controlling algorithms by students. Furthermore, visualisation of the angles of the robot in real time is performed. Not only universities need robotics in the teaching, but also high schools. Literature [4] highlights a 6 DOF robot for educational purposes, the effectiveness of the robotic arm is explored by involvement of 13 high school students.

Robots can appear not only in industry or in education, but also in other areas, e.g., healthcare, media. Thanks to the wide range of available electronic devices and the widespread use of 3D printing, it is now possible for anyone to develop a robot arm for hobby purposes. Thus, the development of robotic arms for different purposes has been the topic of several papers [5–7].

The main aim of this paper is to develop a new control program for an outdated, not functioning robotic arm expanded with hardware improvements. The hardware modifications

*Corresponding author.
E-mail: ronai.laszlo@uni-miskolc.hu

AKJournals

involved the design of braking system and printed circuit board (PCB). The 5 DOF robot was made for educational purposes in the 1980s, but over time, robot control has become outdated. Furthermore, during the exploration process of the electronic system of the robot it has been revealed that now it is not functioning well. The main goal is to make the robot and its controls functional, meeting the requirements of today. The control program is developed for a so-called PSoC 5LP [8] platform, which has four quadrature decoders for the motors to be controlled. The robot is designed to be used with the end-effector held always in vertical position and the 5th DOF is locked, i.e., the end-effector cannot rotate around this vertical axis. The main objective of the robot is to make microcontroller programming available to the students and introduce the field of robotics at beginning level, which contains the solution technique of inverse kinematics problem, definition of coordinate systems, controlling, workspace, etc.

This paper is organized as follows: Section 2 describes the robot and deals with the handling of the inverse kinematics problem of the robot, which is solved by using geometric approach. The necessary hardware developments are discussed in Section 3. A self-devised braking system is made for the robot, and it serves balancing purposes. A printed circuit board (PCB) is designed in EasyEDA software for the control elements. The programming of the robot is described in Section 4. The last Section has the concluding remarks.

2. DESCRIPTION OF THE ROBOT

The robot (see Fig. 1), which was created for educational purposes in the 80's by the Videoton company, can be found at the Institute of Machine and Product Design at the University of Miskolc. The condition check of the system is performed, the electronics need to be completely replaced except for its power supply. The original control software of the robot is missing, and the electrical components are obsolete. Further development of the robot is relevant in order to use it for educational purposes primarily. The development phase is performed at the Robert Bosch Department of Mechatronics. The main goal of the development is to make the robot capable to perform manipulation tasks.

Fig. 1. The photo of the robot

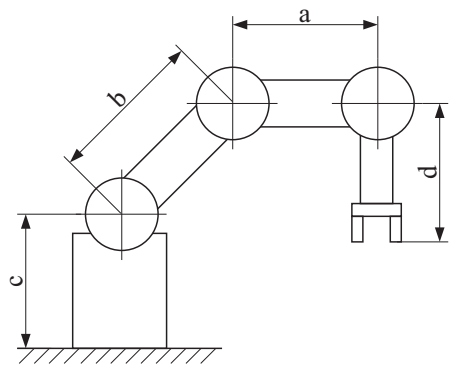

Fig. 2. Links of the robot with its notations

The robot is equipped with DC motors and incremental rotary encoders type HEDS-5000 Series at each joint, which are the basis of the control.

The length of the robot links is measured to solve the inverse kinematics problem. In accordance with the notations of Fig. 2 the length of each link can be defined: $a = 202\ mm$, $b = 220.5\ mm$, $c = 303\ mm$, and $d = 267\ mm$.

2.1. Inverse kinematics problem

The inverse kinematics problem [9] involves the determination of the individual joint angles as a function of the coordinates of the TCP (Tool Center Point). This task can be described in general terms as follows:

$$\overline{q} = f^{-1}(\overline{s}), \tag{1}$$

where $\overline{q}$ is the vector of the joint coordinates of the robot and $\overline{s}$ is the vector of the world coordinates. The inverse kinematics problem can be handled in several ways, depending on the complexity of the robotic structure. It is possible to determine joint angles analytically, numerically or with geometric approach. For simpler robot constructions than the one contained in the current paper, it is worth using the geometric approach [10]. The kinematic scheme of the Videoton robot is illustrated in Fig. 3, which shows the characteristic points and angles. Note that the coordinates of the controlled point, i.e., CP, are used in the kinematic description.

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \begin{bmatrix} X_P \\ Y_P \\ Z_P + d \end{bmatrix}. \tag{2}$$

The task is to write the joint angles as a function of the coordinates of the TCP point [10]. Geometric description method can be used to write the functions of the joint angles using trigonometric identities and functions. The notations used in the formulas are consistent with the notations in Fig. 3.

The joint angle Θ_1 can be written as:

$$\Theta_1 = \text{atan2}(Y_C, X_C), \tag{3}$$

where atan2() is the two-argument arctangent function, it is defined for all range of angles compared to the usual arctangent function, which can return an angle only in the

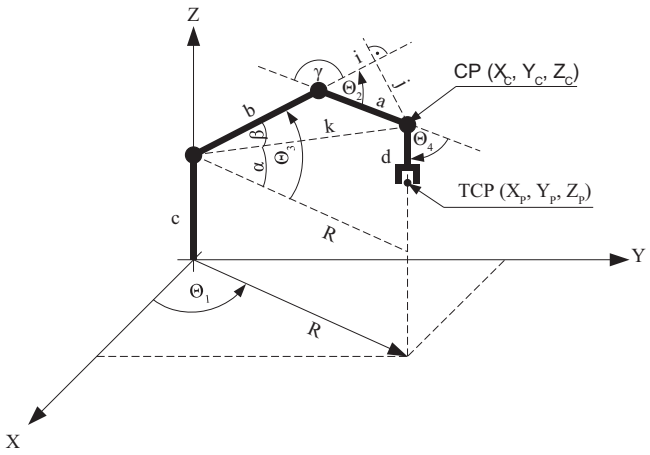

Fig. 3. Scheme of the robotic arm

range of ±90° and does not provide information on which quadrant the angle falls in [10].

The determination of angle γ is important to calculate joint angle Θ_2. The cosine of γ can be formulated as:

$$\cos\gamma = -\frac{k^2 - (a^2 + b^2)}{2ab}. \tag{4}$$

Expressing the length of section k, and using the trigonometric ratio $\cos(180° - \gamma) = -\cos(\Theta_2)$ the cosine of Θ_2 is:

$$\cos\Theta_2 = \frac{R^2 + (Z_C - c)^2 - (a^2 + b^2)}{2ab}, \tag{5}$$

where $R = \sqrt{X_C^2 + Y_C^2}$. The sine of Θ_2 can be written as:

$$\sin\Theta_2 = \pm\sqrt{1 - \cos^2\Theta_2}. \tag{6}$$

The joint angle Θ_2 can be obtained:

$$\Theta_2 = \text{atan2}(\sin\Theta_2, \cos\Theta_2). \tag{7}$$

The calculation of joint angle Θ_3 can be defined as according to Fig. 3:

$$\Theta_3 = \alpha + \beta, \tag{8}$$

where the angles α and β can be determined as follows:

$$\alpha = \text{atan}\left(\frac{Z_C - c}{R}\right), \tag{9}$$

$$\beta = \text{atan}\left(\frac{j}{b + i}\right), \tag{10}$$

where $i = a\cos\Theta_2$ and $j = a\sin\Theta_2$.

The end-effector of the robot during the motion of the robotic arm is kept in vertical position. Joint angle Θ_4 is not independent from the other joint angles, therefore it can be written as follows:

$$\Theta_4 = -\Theta_2 + \Theta_3 + 90°. \tag{11}$$

It is noted that the end-effector of the robot is kept always vertically, therefore point CP can be determined from TCP according to Eq. 2. The distance d is considered in the control software, i.e., the program checks that the target position is reachable or not.

3. HARDWARE DEVELOPMENTS

This section presents the major hardware modifications, expansions of the robotic arm, which are essential to make it functional. A braking system is designed to fix the robot after reaching the target position. A self-devised PCB is created to establish the control of the system. An emergency circuit is made to avoid failures.

3.1. Balancing of the robotic arm

The balancing of robotic arms is addressed in several papers, e.g., [11] presents a teaching manipulator equipped with counterweights that can be used to record the trajectory of a robot. The use of counterweights and springs is discussed in [12], which deals in detail with the issues of balancing robots and other mechatronic structures.

The robot is equipped with a spring as a static balancing feature (Fig. 4) to keep its position fixed. It was revealed that the structure cannot be able to hold its own weight in certain positions. In addition, there have been situations where the spring is already over-tensioned, creating a force that pulls back the robotic arm from the target position. The coils of the motors in the joints can be short-circuited at the desired moment to achieve a braking effect, however, this solution is only suitable for deceleration and not for permanent braking.

Development of a braking system containing electromagnets is performed in order to ensure the precise positioning of the robot. The designed and manufactured brake system in built-in configuration can be seen in Fig. 5.

The electric brake is designed to keep the robotic arm in the target position (see Fig. 5). The solenoids of the brake are actuated only when the robot is moving while the mechanism releases the joint.

The brake mechanism contains two solenoids with a nominal operating voltage of 24V DC. According to the measurements, the resistance of each coil is 20 Ω, with the help of Ohm's law a current of about 1.2 A flows through them, which is higher than the manufacturer's recommendation. By connecting the two coils in series, the current is 600 mA per coil, which is even above the optimal range of

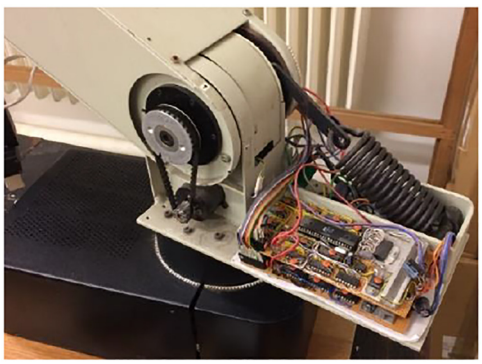

Fig. 4. The robot equipped with the original electronics, and the spring for balancing

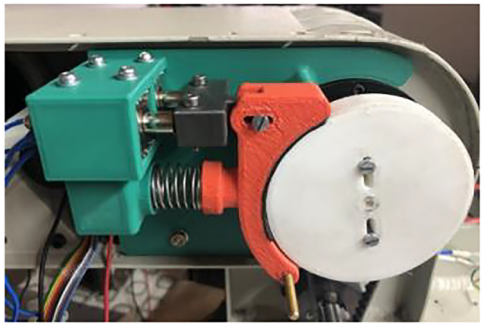

Fig. 5. The integrated brake system

400–450 mA. Thus a 16.6 Ω current limiter resistor with 10 W nominal power is connected in series, which results from two 3.3 Ω and one 10 Ω resistors.

The plastic parts of the brake are made of polylactic acid (PLA), and the brake disc is produced from acrylonitrile butadiene styrene (ABS) using FDM 3D printing technology. An extra rubber surface has been placed on the brake surface to increase the coefficient of friction.

3.2. Self-devised PCB and emergency circuit

During the construction of the structure, there is a demand to make a PCB for the Cypress PSoC platform and for the connected devices. The PCB is designed in the EasyEDA software.

The PCB contains the pins for the connection of the PSoC platform, the various peripherals, and the LM2596 switch-mode power supply with 5 V supply voltage. The power supply is designed to produce the appropriate voltage level for the motor encoders, microcontroller, and motor control ICs. The motor control ICs are BTS7960M type H bridges containing MOSFET-s. During the design of the PCB, due to the limitation of the installation dimensions its size is minimalized, therefore two layers are used. Attention had to be paid to the appropriate thickness of the conductive traces as well as to the bridges between the layers where necessary. The PCB was manufactured in collaboration with a manufacturer called JLCPCB. The upper layer of the PCB is shown in Fig. 6.

An emergency stop button is also mounted onto the robot system. During its operation, the voltage supplied to the motors of the joints, the end-effector and the brake are interrupted, thus preventing their operation, i.e., the power supply unit (PSU) is disconnected from the electric components. The emergency stop button has a pair of normally closed contacts that will be used to switch a relay (see Fig. 7).

The block diagram of the system can be seen in Fig. 8. The robot can receive instructions from a PC via UART

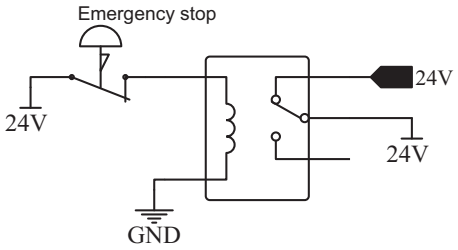

Fig. 7. Emergency stop

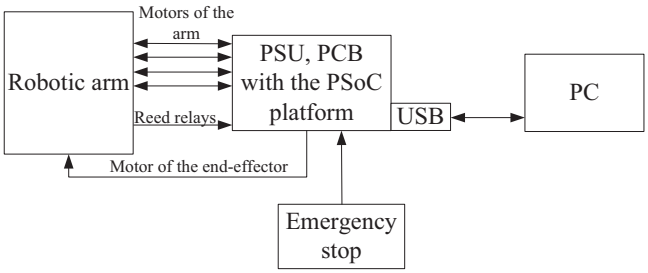

Fig. 8. Block scheme of the system

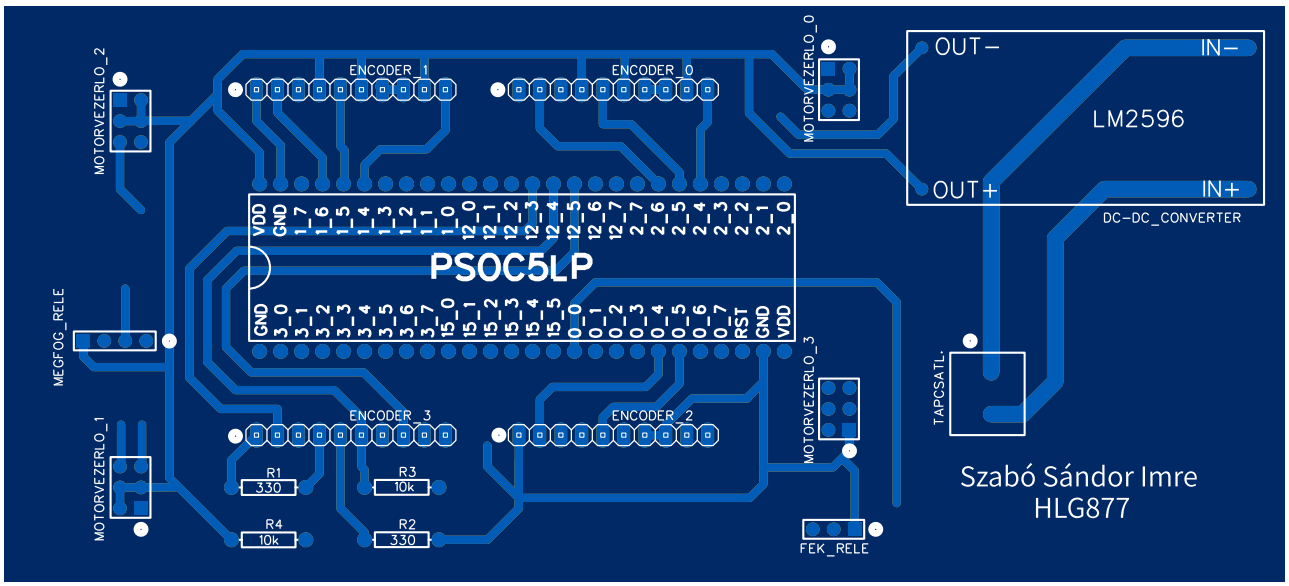

Fig. 6. The designed PCB

communication protocol by USB connection, which will be explained in Section 4.1. The reed relays ensure the availability of the home positioning. Three motors of the robot are used to reach the target position, the fourth one is needed to keep the end-effector in vertical position. The opening and closing process of the end-effector is made by time-limited control to avoid short circuit.

4. CONTROL PROGRAM OF THE SYSTEM

The software side of the robot consists of two major parts: one is the embedded software that runs on the microcontroller built into the robot (Cypress PSoC 5LP) and one user software that runs on a personal computer.

4.1. Embedded software

The PSoC platform consists of a microcontroller based on ARM architecture slightly differing from a traditional microcontroller because it has internal configurable hardware elements, which can be set before programming at the design state. The program consists of a graphical block diagram and text code written in C at the same time.

The graphical block diagram of the program is shown in Fig. 9. Four quadrature decoders (*QuadDec*) are placed to decode the signals from the encoders of the motors. The block requires two input signals that will be the channels of the encoders. From the change of the two input signals, the decoder is capable to convert the rotation of the encoders into a signed number stored in 32 bits. When the robot is moving to a new position, it is necessary to be able to change the rotational speed of the motors in each joint, which required the use of PWM blocks. These blocks contain two PWM channels that operate at 8-bit resolution. Furthermore, an UART block is used for USB communication, and there are output and input pins and clock generators.

When the system is powered on, the program (see Table 1) starts with the *main ()* function, which is located in the *main.c* file. One of the first steps are the initialization of the modules, therefore, the *init ()* function is executed to start the operation of the blocks (Fig. 9), the function can be found in the *init.h* file. An array *string* is declared after the *init ()* function, which stores the instructions received from the user software via UART communication protocol to get the data for the operation of the robot. Four integer variables *joint1-joint3* are defined to store the number of steps that DC motors with rotating encoders must rotate to the target position on each joint, and finally a logic variable is created to prevent the execution of any instruction with the robot until the reference point is reached. Then a *while()* loop is started. For the first time in the loop the *input ()* function is called, which receives the memory address of the 0^{th} element of the character array as the start parameter. The *input ()* function is defined in the *communication.h* header file.

The *input()* function is constantly waiting for incoming characters embedded in an infinite loop, i.e., then the robotic arm is in a standby mode. If a character arrives, it stores it in the next element of the array, continuing until a line feed (LF) character arrives. Thereafter interruption of the execution of the loop of *input()* is performed by the *break* statement. After this, the program will start processing the received instructions. The robot can receive four commands and the last element of the resulting string decides which

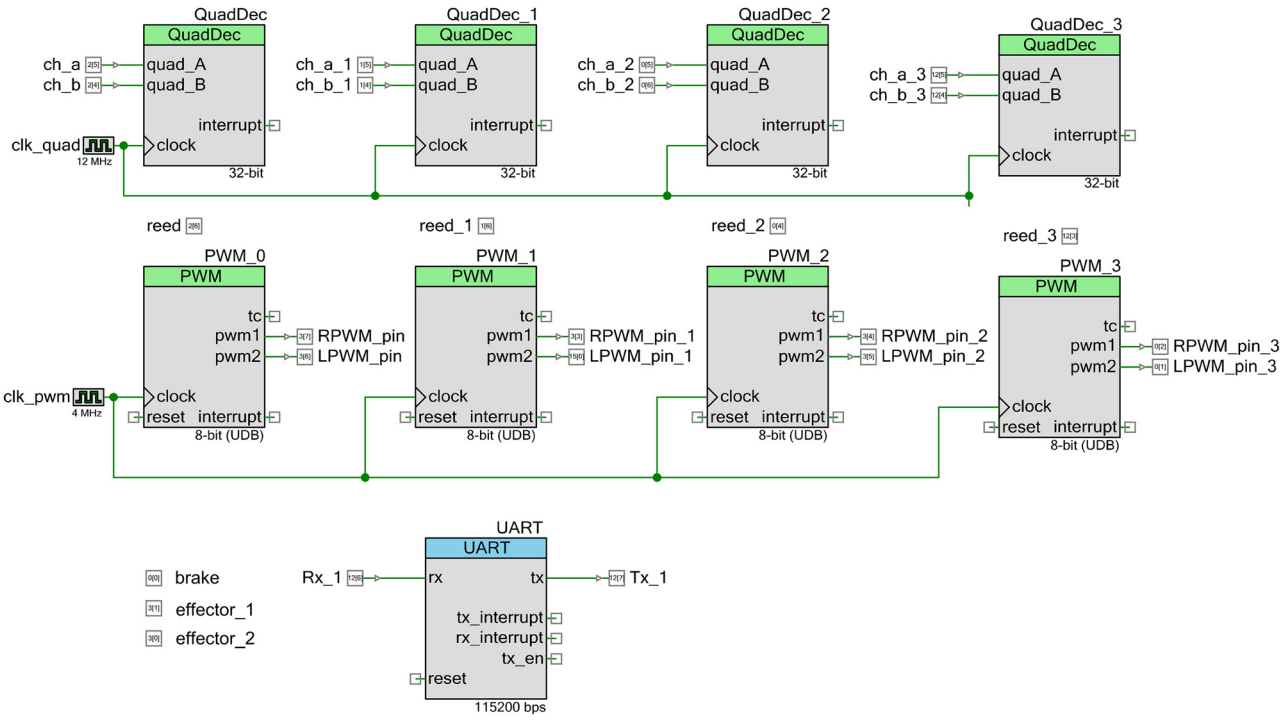

Fig. 9. Blocks of the program

Table 1. The main program code running on PSoC platform

```
#include "init.h"
#include "motordrive.h"
#include "effector.h"
#include "communication.h"
int main(void){
   init();
   char string[40];
   long int joint0, joint1, joint2, joint3;
   bool ref = 0;
   while(1){
    input(string);
    if(string[24] == '1' && ref == 1){
       end_eff_open(); executed(string);
}else if(string[24] == '2' && ref == 1){
       end_eff_close(); executed(string);
}else if(string[24] == '3'){
       reference(); ref = 1; executed(string);
}else if(string[24] == '4' && ref == 1){
       joint0 = 100000*((int)string[0]-48) + 10000*((int)string[1]-48) + 1000*((int)
       string[2]-48) + 100*((int)string[3]-48) + 10*((int)string[4]-48) + ((int)string[5]-48);
       joint1 = 100000*((int)string[6]-48) + 10000*((int)string[7]-48) + 1000*((int)
       string[8]-48) + 100*((int)string[9]-48) + 10*((int)string[10]-48) + ((int)string[11]-48);
       joint2 = 100000*((int)string[12]-48) + 10000*((int)string[13]-48) + 1000*((int)
       string[14]-48) + 100*((int)string[15]-48) + 10*((int)string[16]-48) + ((int)string[17]-48);
       joint3 = 100000*((int)string[18]-48) + 10000*((int)string[19]-48) + 1000*((int)
       string[20]-48) + 100*((int)string[21]-48) + 10*((int)string[22]-48) + ((int)string[23]-48);
       move_to(joint0, joint1, joint2, joint3);
       executed(string);}
string[24] = '0';}}
```

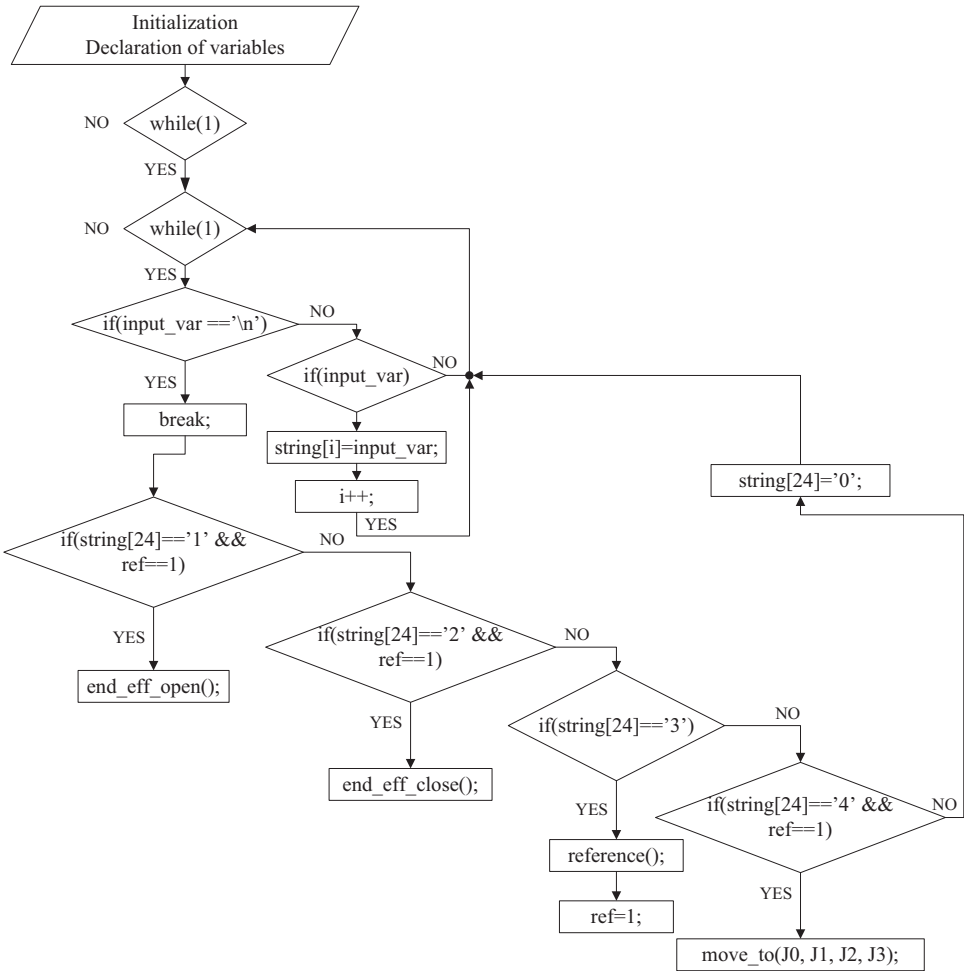

Fig. 10. The flow chart of the operation of the program code

one it needs to perform. These can be opening and closing the end-effector *end_eff_open()* and *end_eff_close()*, recording the reference point *reference()*, and positioning *move_to(joint0, joint1, joint2, joint3)*. Apart from the last element of the character array, the role of the others is only in the positioning of the robotic arm, they do not carry any necessary information for the others. The reaching of the reference point is made by using Reed relays.

After each instruction is executed, the *executed()* function runs sending an acknowledgment to the user software that the execution of the instruction is complete. The flow chart of the program can be seen in Fig. 10.

4.2. User software

The user software is written in Python programming language and can be run from the terminal window on GNU Linux. Its primary purpose is to interpret user instructions and provide the information needed to execute them to the microcontroller using UART. An installer was created for the software, which makes the program run anywhere from the terminal using the *videotonrobot* command. After launching the software, the possible devices are listed, from these, the user must select the desired device by entering the correct serial number. The name of the robot will be "Cypress KitProg - KitProg USBUART" due to the microcontroller. Then the selection of a desired script file is the next step. The script must be written to a text file with a .rbt extension before running the software.

Four special purpose functions are defined: *ref()*, *zar()*, *nyit()*, *mozog(x, y, z)*. These are required to write the script file. The *ref()* function performs the ability to take up the reference point of the robot. The functions *nyit()* and *zar()* actuate the end-effector of the robot. The positioning function is the last one, it has three input parameters, which are the *x*, *y*, and *z* coordinates of the target position in mm. A screenshot of the user software is shown in Fig. 11 in Hungarian language.

Fig. 11. The terminal window of the user software

5. CONCLUSIONS

There is a high demand from industry for engineers with robot programming capabilities. Therefore, the further development of a robotic arm was detailed in this paper, which can be used to demonstrate the field of robotics to our students through manipulation tasks.

The inverse kinematics problem of the structure is solved using geometric approach. The angles of the joints calculated from the target position provide the base of the control. Hardware and software modifications were needed to further develop the robot. A brake system with electromagnets was designed to keep the robot in position. A self-devised PCB was made to merge the elements of the control, and to place the switch-mode power supply unit. Furthermore, the emergency circuit of the robot was created to avoid failures. The working area of the robot was designed with the use of aluminum profile elements. Software developments were also performed. The control program of the PSoC platform was written in C language, the user program was developed under Python programming language.

The user software contains the checking process of the defined target coordinates. When the given coordinates are within the working area, the coordinates can be sent to the microcontroller. The robot was tested with several pick & place tasks successfully.

In the future one of the objectives is to teach microcontroller programing based on the PSoC platform. The developed robotic arm can be integrated into the programing course, therefore creation of robotic tasks can be assigned to students. The other objective is to demonstrate robot programing at beginner level. During the programming course, the students will write their program code containing the inverse kinematics formulae, and positioning instructions, then they test it.

Furthermore, in the future the positioning accuracy of the robotic arm will be measured to increase its accuracy, which can be a new possible improvement.

REFERENCES

[1] M. Gereald and P. Zentay, Industrial Robots Meet Industry 4.0, Hadmérnök (XII) 1V, 2017.

[2] T. I. Erdei and G. Husi, "Singularity measurement in the Cyber-physical and intelligent robot systems laboratory," *Int. Rev. Appl. Sci. Eng.*, vol. 11, no. 2, pp. 82–7, 2020.

[3] M. Hernandez-Ordoñez, M. A. Nuño-Maganda, C. A. Calles-Arriaga, O. Montaño-Rivas, and K. E. B. Hernández, "An

education application for teaching robot arm manipulator concepts using augmented reality," *Mobile Information Systems*, vol. 2018, no. 4, pp. 1–8, 2018.

[4] C. Zeng, H. Zhou, W. Ye, and X. Gu, "iArm: design an educational robotic arm kit for inspiring students' computational thinking," *Sensors*, vol. 22, no. 8, pp. 1–22, 2022.

[5] D. Sáenz Zamarrón, N. I. Arana de las Casas, E. García Grajeda, J. F. Alatorre Ávila, and J. U. Naciff Arroyo, "Educational robot arm platform," *Computación y Sistemas*, vol. 24, no. 4, pp. 1387–401, 2020.

[6] K. Kruthika, B. M. Kiran Kumar, and S. Lakshminarayanan, "Design and development of a robotic arm," *International Conference on Circuits, Controls, Communications and Computing (I4C)*, Bangalore, India, 2016.

[7] A. Elfasakhany, E. Yanez, K. Baylon, and R. Salgado, *Design and Development of a Competitive Low-Cost Robot Arm with Four Degrees of Freedom*, Modern Mechanical Engineering, 2011.

[8] Cypress PSoC 5LP: CY8C58LP Family Datasheet, Cypress Semiconductor Corporation, 198 Champion Court, San Jose, CA 95134-170, Document Number: 001-84932, 2019.

[9] L. Rónai and T. Szabó, "Kinematical investigation and regulation of a 4DOF model robot," *Acta Mechanica Slovaca*, vol. 20, no. 3, pp. 50–6, 2016.

[10] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*, Wiley, 2006.

[11] Z. Fan, Y. You, H. Zheng, G. Zhu, W. Li, S. Chen, K. Deb, and E. Goodman, "Modeling and Multi-objective Optimization of a Kind of Teaching Manipulator," *arXiv:1801.10599v1*, pp. 1–12, 2018, Cornell University.

[12] L. Ciupitu, "Adaptive balancing of robots and mechatronic systems," *Robotics*, vol. 7, no. 4, 2018.