ANNALS OF THE UNIVERSITY OF ORADEA
Fascicle of Management and Technological Engineering
ISSUE #2, AUGUST 2016, http://www.imtuoradea.ro/auo.fmte/

CONTROL AND POSITION MEMORY OF A TTR ROBOT

Lilian Coutinho Alves MEIGA[1], János TÓTH[2]

[1] Federal University of Itajubá - UNIFEI, Brazil, lilian.c.a.meiga@gmail.com
[2] University of Debrecen, Hungary tothjanos@eng.unideb.hu

Abstract— Document about the controlling of the Translation Translation Rotation (TTR) robot, which can be applied in industries. The control in this article was made by a PLC using an HMI and in this control, besides the movement, the user can teach one position to the TTR robot and after this position is taught, the TTR can return to it as long as it is the last taught position.

Keywords— Positioning, Memory, Robot

I. Introduction

THE recurrent need of improvements, enlargement of the production and operational safety in the early 1900's ,and even now, made necessary the development of better control methods, what makes the Programmable Logic Controller (PLC) very used in most processes inside the industrial environment.

To control it in the site, usually is used a Human-Machine Interface (HMI) , so the user can manipulate the process in a very efficient and friendly way.

To assemble this system together, it's necessary to program this device and connect them, which is one of the main goals of this project besides the position's memory.

II. Theoretical Fundamentation

A. Robots

The idea of an autonomous object originates in the mythology of many cultures around the world and it dates from some centuries before Christ. Along the centuries, a lot of important inventors as Leonardo da Vinci and Nikola Tesla showed interest in this subject. Da Vinci's sketched plans of a humanoid robot were discovered in 1945 and it contained detailed plans of a mechanical knight that would be able to sit, move his arms, head and jaw. [1]

The word robot comes from the Slavic word "robota" which means work, labor, servitude and was introduced by Karel Čapek in his play R.U.R. (Rossum's Universal Robots), published in 1920. In Čapek's play, he describes the robot as androids, because they can be mistaken for human beings. His idea now is just one of the types of robots. Robots can be humanoid, autonomous, mobiles, modular, educational and a lot of other types, including Industrial robots , which was the main goal of this project development.

B. Programmable Logic Controllers (PLC)

Before the invention of this device that revolutionized the industry all controls were made through the use of relays. The principle of functioning of a relay is related to its coil, which if energized, create a magnetic force and pull the switch to the on or off position.[2] But the problem with the relay use was that if you needed a lot of relays and one in the middle of the process started to malfunction you would spend a lot of time to find the problem, because the system wouldn't work, and fix it.

However, with the development of the PLC's even with a very small memory would solve this issue. Even more after the creation of the ladder diagrams, a programming structure that is very similar to the relay's logic, so it was simpler and easier for the workers to adapt to this new inventions.

Fig. 1. PLC used in the project

C. Human-Machine Interface (HMI)

The Human-Machine Inface, also known as User Interface (UI) is the space where the interactions between the machines and the user happen. The goal of the UI designer is to make an easy, self-explanatory, efficient and user-friendly control, so the user can get the desired results.

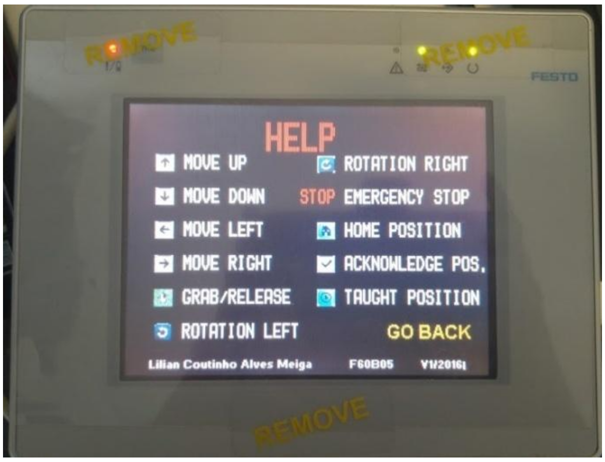

Fig. 2. Project's HELP screen to assist the user during the use of the system

For the development of this project the software used was FED Designer 6, that can be found on Festo's Website.

Fig. 3. Project's HMI screen

Fig. 4. Project's designed HMI screen on FED-500

III. Theoretical and Practical Methodologies

For the realization of the project some material were needed such as:

1) *Festo PLC (FEC-CPX);*
2) *Festo HMI (FED-500);*
3) *Relays;*
4) *24 VDC Motors and 24 V power supply;*
5) *TTR robot previously built with Fischtechnik pieces;*

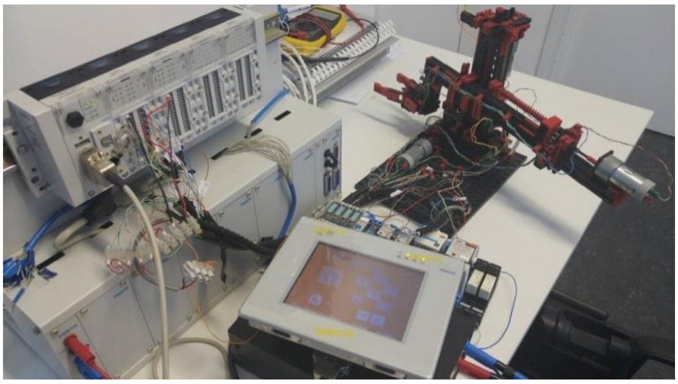

Fig. 5. Picture of my Project

As DC motors were used for the assembly of the project, to change the direction of the current, an H-Bridge was necessary:

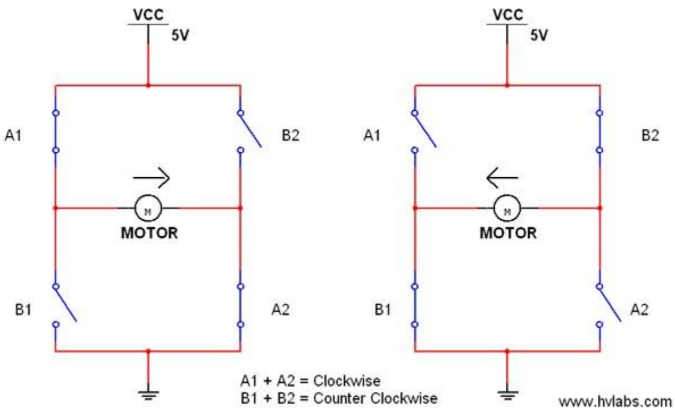

Fig. 6. Schematic of an H-Bridge [4]

But in this case VCC=24 V.

For the PLC programming was used Festo Software Tools (FST) software.

In the PLC programming, the program was developed using statement list, which is one of 5 languages supported by IEC_61131-3, that is an international standard for PLC's.[3]

In this program all the motions of the robot were described and as the input would come from the HMI, instead of using Input variables for the motion, were used flags , which are PLC's internal variables.

```
P 0 (V2) - Robotic arm positioning and grabbing
"" UP MOTION

 IF            ( N  I0.0         'UP END SENSOR
       AND          F0.00        'ARM UP
                              )
       AND      N   F0.01        'ARM DOWN
       AND      N   F0.04        'RIGHT ROTATION
       AND      N   F0.05        'ARM LEFT
       AND      N   F0.06        'ARM RIGHT
       AND      N   F0.03        'LEFT ROTATION

 THEN  SET          O0.0         'UP MOVE
       RESET        O0.1         'DOWN MOVE
       RESET        O0.3         'LEFT MOVE
       RESET        O0.4         'RIGHT MOVE
       RESET        O0.5         'LEFT ROTATION
       RESET        O0.6         'RIGHT ROTATION
       RESET        F0.07        'HOME POSITION
 OTHRW RESET        O0.0         'UP MOVE

"" DOWN MOTION

 IF            ( N  I0.1         'DOWN END SENSOR
       AND          F0.01     )  'ARM DOWN
       AND      N   F0.00        'ARM UP
       AND      N   F0.04        'RIGHT ROTATION
       AND      N   F0.05        'ARM LEFT
       AND      N   F0.06        'ARM RIGHT
       AND      N   F0.03        'LEFT ROTATION

 THEN  SET          O0.1         'DOWN MOVE
       RESET        O0.0         'UP MOVE
       RESET        O0.3         'LEFT MOVE
       RESET        O0.4         'RIGHT MOVE
       RESET        O0.5         'LEFT ROTATION
       RESET        O0.6         'RIGHT ROTATION
       RESET        F0.07        'HOME POSITION
 OTHRW RESET        O0.1         'DOWN MOVE
```

Fig. 7- Project's programming sample on FST software

However, to program the HMI the FED designer was used and to link the HMI information to the PLC without CoDeSys was necessary to create a Tag List and after that, add a Data transfer action, so all the data could be transferred without any problems.

After all the programming done, to upload to the program to the HMI, you firstly have to put the HMI in config mode, which can be done pressing the corner of the screen until the menu appears.

IV. Results and Discussion

For this TTR (Translation Traslation Rotation) robot, a successive number of tests were made to guarantee that the motions were correct, including the grabbing movement, if the end position sensor were protecting the body so it wouldn't break because of excessive forces and/or wrong movements. It was tested as well for the acknowledge position module and taught position modules.

The acknowledge position module works this way: when pressed the button, the robot will move to a base position, which was established by moving the robot until it reaches three of the end-position sensor, and three timers were running meanwhile this movement, so it could register how long it would take to get to that position. These values were saved in registers, so even if you turn off the PLC when you turn it on the last position will still be saved.

```
CMP 0 (V1) - ACKNOWLEDGE POSITION*
STEP ack
"" UP DOWN ACKNOWLEDGE

"" Switch output
 IF              N    I0.1      'DOWN END SENSOR
 THEN  SET            O0.1      'DOWN MOVE
 OTHRW RESET          O0.1      'DOWN MOVE

"" Count time
 IF              N    I0.1      'DOWN END SENSOR
       AND       N    F1.3      'REACTION TIME FLAG 1
 THEN
       SET            F1.3      'REACTION TIME FLAG 1
       LOAD           V0
         TO           R2        'CYCLE COUNTER 1

 IF                   I0.1      'DOWN END SENSOR
       AND            F1.3      'REACTION TIME FLAG 1
 THEN  RESET          F1.3      'REACTION TIME FLAG 1
       LOAD           FW7       'COUNTER POSITION 1
         TO           R2        'CYCLE COUNTER 1
       LOAD           V0
         TO           FW7       'COUNTER POSITION 1

"" Evaluate time
 IF              N    T1        'TIMER 1
       AND       N    I0.1      'DOWN END SENSOR
 THEN  SET            T1        'TIMER 1
         WITH         0.01s
       INC            FW7       'COUNTER POSITION 1


"" LEFT_RIGHT_ACKNOWLEDGE
 IF              N    I0.3      'RIGHT END SENSOR
 THEN  SET            O0.4      'RIGHT MOVE
 OTHRW RESET          O0.4      'RIGHT MOVE

"" Count
 IF              N    I0.3      'RIGHT END SENSOR
       AND       N    F1.4      'REACTION TIME FLAG 2
 THEN  SET            F1.4      'REACTION TIME FLAG 2
       LOAD           V0
         TO           R3        'CYCLE COUNTER 2

 IF                   I0.3      'RIGHT END SENSOR
       AND            F1.4      'REACTION TIME FLAG 2
 THEN  RESET          F1.4      'REACTION TIME FLAG 2
       LOAD           FW5       'COUNTER POSITION 2
         TO           R3        'CYCLE COUNTER 2
       LOAD           V0
         TO           FW5       'COUNTER POSITION 2

"" Evaluate
 IF              N    T2        'TIMER 2
       AND       N    I0.3      'RIGHT END SENSOR
 THEN  SET            T2        'TIMER 2
         WITH         0.01s
       INC            FW5       'COUNTER POSITION 2
```

Fig. 8. Sample of Acknowledge module programmed on FST

The taught position module works in a very similar way than the acknowledge position module. When the button of the taught position is pressed, the TTR goes to the base or home position and from there it makes the reverse move that it had made to go to the base position.

```
CMP 1 (V1) - UP TEACHING POSITION

STEP up
 IF                 F0.10          'TAUGHT POSITION
      AND     N     F1.6           'taught up
      AND     N     F1.8           'CHECK T2
 THEN
      SET           T4             'TIMER 1
        WITH        R2             'CYCLE COUNTER 1
      SET           F1.6           'taught up

 IF                 T4             'TIMER 1
      AND           F1.6           'taught up
 THEN  SET          O0.0           'UP MOVE

 OTHRW RESET        O0.0           'UP MOVE

 IF                 F1.6           'taught up
      AND     N     T4             'TIMER 1
 THEN
      SET           T5             'TIMER 2
        WITH        R3             'CYCLE COUNTER 2
      JMP TO left

STEP left
```

Fig. 9. Sample of Teaching position module programmed on FST

However, as it could be seen during the tests, the weight difference of the axis (that can be seen in Fig. 10.), because of the motors and other variables, cause some small discrepancies between the position, as for one direction the motor would have to make more force to move than to the other one due to the resultant force of the axis.

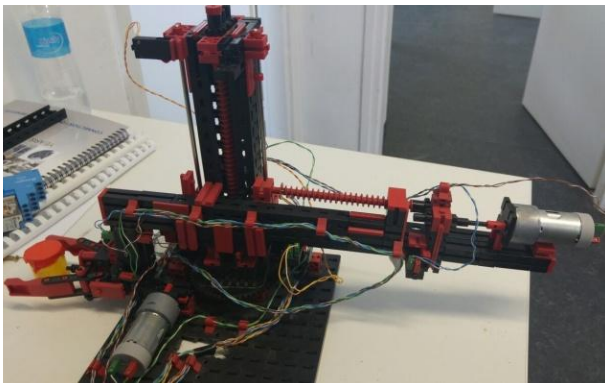

Fig. 10-. - Picture of the TTR robot

Then, another solution would be doing the programming using Counters instead of timers and using the piece is shown in Fig. 11. with some switches to recognize the rotation of the piece and then being able to position the TTR robot to its previous position with more precision, besides not being influenced by the weight of the axis.

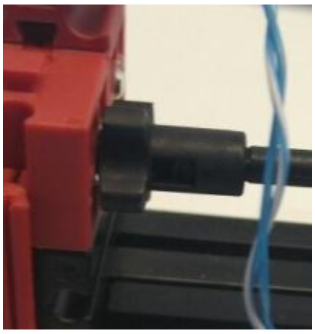

Fig. 11. Piece of the Fishertechnik kit that can be used for the counting solution

V. Conclusion

This project presented a small introduction about programmable logic controllers, human-machine interface and a brief robot's history. It explained the functioning of the program, how the acknowledge and taught position modules were made and justified the small discrepancies that could be seen during the tests.

Acknowledgment

L.M. thanks Ms.Xénia Tóth and Mr. Zoltán Deskó for the help provided during the realization of this project.

References

[1] M. Taddei, "Leonardo da Vinci's Robots". Available at <http://Leonardo3.net> Retrieved 2016-05-10.

[2] Unknown Author, "History of the PLC", available at: <http://library.automationdirect.com/history-of-the-plc/> *Retrieved 2016-05-04.*

[3] W. L. Jr. G. Keller, " A Functional Chart for Sequential Processes", *14th Annual International Programmable Controllers Conference Proceedings*, 1984, p. 71-96.

[4] Unknown Author , "Controlling DC Motors - The H-Bridge Way", available at < http://www.hvlabs.com/hbridge.html> *Retrieved 2016-05-04.*