

G-Code interpreter development using Microsoft Visual Basic for ABL63 Control Systems

Alma G. Rodríguez*

Luis R. Vidal P.**

Juan Miguel Díaz M.***

Cesar Leonardo González Pinzón****

submitted date: November 2013

received date: December 2013

accepted date: February 2014

Abstract

This paper discusses the development process involved in making a user interface translate G-Code (ISO6983) using Microsoft Visual Basic 2010 together with communication protocol OPC. This interface was developed in three modules, the first one verifies syntax and lexicon, the second is a command converter for a Programmable Logic Controller Allen Bradley L63 using a SERCOS interface card and servo drive ULTRA 3000, and the last one executes motion control instructions. The interface is used for a profile milling process, on an XY plane and using G00, G01, G02 and G03 codes generated in a Fanuc postprocessor 3XMILL. The interface shows errors detected in the verification module according to the parameters established and identifies the corresponding line; it also monitors the XY position in real time, The interface allows line-by-line execution or automatic mode. The interface does not require information for cutting speed, tool advance speed, and spindle speed or tool compensation from the NC

* B.Sc. In Industrial Engineering, Universidad Autónoma de Ciudad Juárez (México). E-mail: arodriguez@uacj.mx

** B.Sc. And MSc. in Industrial Engineering, Instituto Tecnológico y de Estudios Superiores de Monterrey (México). Current position: Professor Universidad Autónoma de Ciudad Juárez (México). E-mail: lvidal@uacj.mx

*** B.Sc. In Industrial engineering and MSc. in Industrial Engineering, Instituto Tecnológico y de Estudios Superiores de Monterrey, (México). Current position: Professor Universidad Autónoma de Ciudad Juárez (México). E-mail: jdiaz@uacj.mx

**** B.Sc. In Production engineering, Universidad EAN, Bogotá (Colombia). MSc. In Industrial Automation Engineering, Universidad Nacional de Colombia, (Colombia). Current position: Professor Universidad Autónoma de Ciudad Juárez (México).E-mail: cempaques@hotmail.com, cesar.gonzalez@uacj.mx

program. The interface was also developed using incremental and reuse oriented methodologies.

Key words

Interface, G-Code, Programmable Logic Controller, Milling process, servo drive, communication protocol.

1. Introduction

The increasing demand for quality products introduces the need to develop and improve CNC (Computer Numerical Control) machine tools. CNC systems have a computer that reads and interprets instructions in machine language or G-Codes for guiding a tool [1]. The Numerical Control (NC) program is the one that details the instructions step by step, indicating the machine what path to follow and which operation to execute for the machining of a part. At the beginning, CNC machines were used exclusively by the aerospace industry, now they have been adapted to other industries like wood [2] and plastics [3]. What distinguishes a particular machine for a specific application is a variety of settings like load charges, vibration, cutter, working area, machining velocities and precision requirements. Through the Computer Aided Design and Computed Aided Manufacture (CAD/CAM) it is possible to create different part designs and generate the corresponding NC program with a postprocessor. Postprocessors are translation tools of CAM software that convert a design, or path, in a language that can be understood by the control module of the CNC machine. The commands and information described in the NC program cannot be managed directly, so it is the interpreter that extracts the corresponding data and translates it into machining and motion control commands for the controller.

With the implementation of Programmable Logic Controller (PLC) [4] the motion control functions can be executed, assisted by motion control instructions and motion analysis software. The motion control is an automation field in charge of developing technologies and products that improve precision systems significantly [5]. Servomotors are controlled by the PLC through the use of a servo drive and a communications interface, such as the I/O cards for sensors and other actuators.

In recent years, research on G-Code interpreters has become relevant and intended to develop open systems with universal applications, some cover standards ISO6983 and basic NC program structures [6], [7], [8], [9], some others cover the most recent standard STEP-NC (Standard for the Exchange and Sharing of Product) [10] and [11]. In general, the trend is marked by the interpreter.

2. Methodology

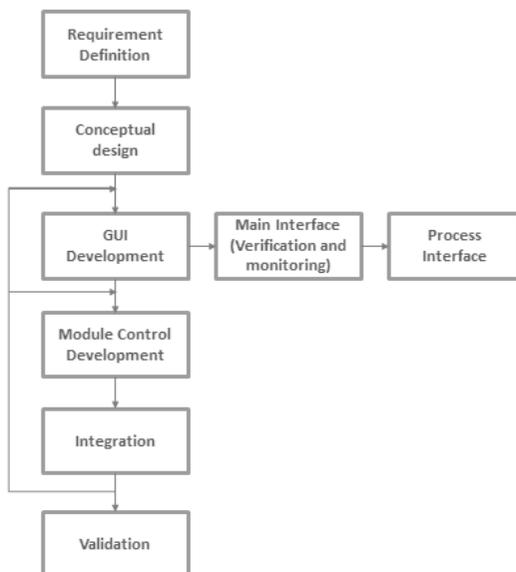
The method used for the development of this study includes six main stages, as shown in Figure 1. In the first stage, requirements are defined, namely the definition of the NC program structure, selection of the post processor used for test and validation, hardware and software requirements. The second stage is called conceptual design and consists in proposing the conceptual model. The third stage is the development of the GUI, where the main module and the process interface are explained. Since the validation of the interface is taken to a physical cutting process, the next stage, called development of the control module, is necessary for the control of the multi-axis system. After the GUI and the control module are finished, integration takes place in order to make everything work together. Finally, the last stage involves validation of the method. Two validations were conducted; the

first validation is through a checklist of the expected operation and performance of the GUI as a software product, and the second validation was performed over the milling process when machining an irregular path.

2.1 Defining requirements

The structure and format of the NC program change from one postprocessor to another, so it was necessary to specify a basic structure with a generic format for the test of the G-Code interpreter prototype. The structure was defined following the research made by Hong et al. [12]. The format considered the default postprocessors included in MasterCAM X4; the one selected was a Fanuc 3XMILL postprocessor.

Figure 1. Method



Source: own elaboration

As a requirement for the interpreter's development, the lexicon and syntax rules are specified for a quick revision of the file, that is, for interpretation so as to check the verification rules and identify errors. The verification is made in the interface and once it is positive, the process may continue. If the verification

of the file is not satisfied, the program will stop and the error should be shown to the user in the interface and saved on an errors file (location shown to the user). Some of the rules defined for the verification function are:

1. XY plane.
2. Metric system.
3. Absolute coordinates.
4. Eliminate commentaries.
5. Each block must be numbered (example, "N100 G00 X3. Y3.").
6. Machining commands G00, G01, G02, and G03 are written once and the following blocks take the same command unless other command is specified.
7. Verify that there is a numeric value after the address character X/Y.
8. Limit travel for X-axis -350 to 350 mm.
9. Limit travel for Y-axis -150 to 150 mm.

An XY multi-axis system was used for the execution of the motion commands of the G-Code. For the DDE between the interface and the PLC, a communication protocol is required. OPC technology was chosen as a solution for the data exchange, making it possible to monitor the process and also to convert machining control commands into motion control instructions in the PLC. Since the main purpose of this research is the interpretation of the machining control commands, turning them into physical motion and therefore a cutting process (milling), it has been set that the cutting parameters are not taken from the NC program but are to remain fixed in the PLC program. The execution of the interpreter is line-by-line and the process may operate between semi-automatic and single block modes.

2.2 Conceptual Design

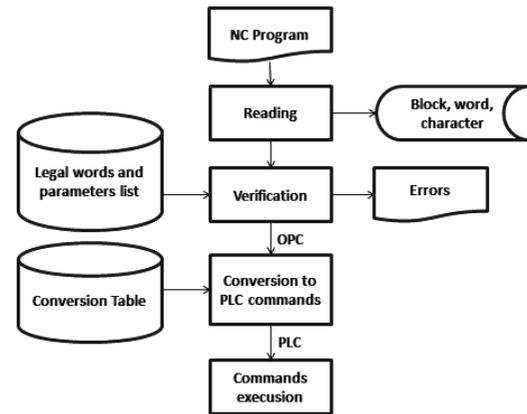
The conceptual design proposed for the G-Code interface consists in a modular design, as shown in Figure 2. The first module corresponds to the reading and storage of the NC program based on a basic structure, where a block is made of a word and the words made of characters; finally these characters are divided in direction and value. After the data is stored, the revision of lexicon and syntax is made in the verification module, where a table of legal words and limit travel parameters is accessed for comparison. During verification, the errors found in the NC program are displayed on the interface with the corresponding block and description. Finally, a .txt file is generated containing an error report. Once the file has been verified, the corresponding G-Codes and its XY parameters are sent as OPC data to the PLC controller for translation into motion command instructions, based on the conversion table shown in Table 1. The motion control instructions are executed by the multi-axis system.

Table 1. Conversion Table

G-Code	Motion Control Instructions (PLC)	OPC data required
G00	MAM (Motion Axis Move)	X, Y
G01	MCLM (Motion Coordinated Linear Move)	X, Y
G02	MCCM (Motion Coordinated Circular Move)	Direction, X, Y, R
G03	MCCM (Motion Coordinated Circular Move)	Direction, X, Y, R

Source: own elaboration

Figure 2. Conceptual design



Source: own elaboration

2.3 Graphical User Interface Design

A GUI assists the interaction between the user and the machine. The software product consists of two GUIs and the controller program. The GUIs correspond to the main interface and the process interface. Two software development process models were followed, namely reused-oriented and incremental.

For the reused-oriented model there were two components, the first component for monitoring a variable by OPC [13] and the second consisting of a VB class with functions for sending and reading OPC data [14]. The incremental model consists of three increments developed for the interfaces; the first for the function that stores the observation. The second, and one of the most important, was for the function that verifies the file chosen, generating an error report if necessary. For the third increment, a function for the preparation of the data that will be sent though OPC was developed.

2.3.1 Main and Process Interfaces

During the process, the main interface includes a user control access, verification file, position monitoring, error handling, and stop-reset.

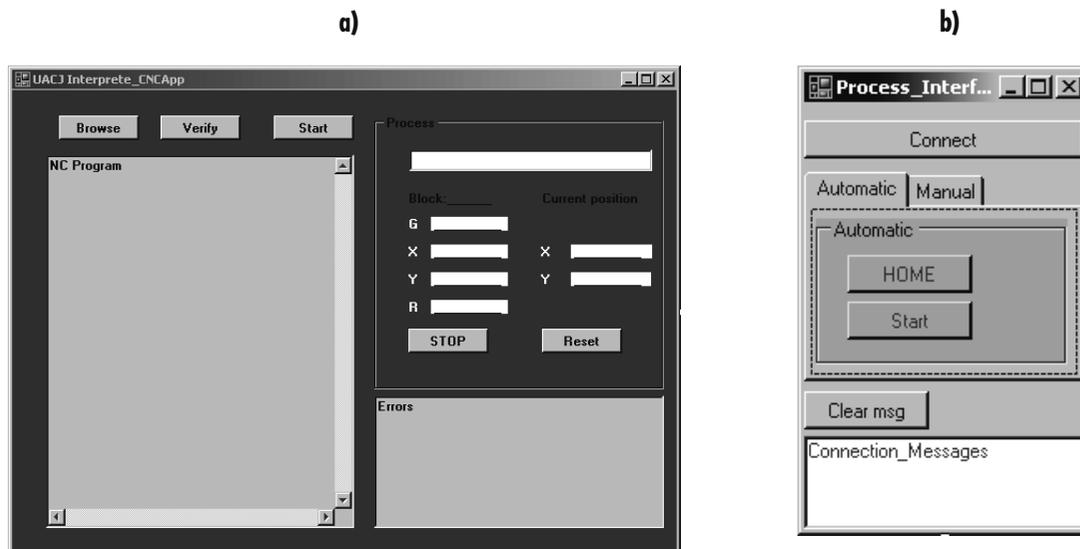
The interface follows sequential programming. The NC program is chosen and verified.

For the process to start, it is necessary to make sure that the OPC connection has been established. The “Connect” button calls for the OPC connection function from the VB class of the second reused-oriented component.

For the semi-automatic mode, the file is read again line by line sending the data extracted from each block. The next line is read until the axes operation is finished. The single block mode also reads line by line, extracting the corresponding data, but the data is sent until the user makes this decision, as long as the previous axes operation has been completed.

When the process is running, either on semi-automatic or single block mode, the main interface displays the current and expected position using monitoring variables created with the first reused-oriented component. The main interface also displays the complete NC program for further observations. The user has the option of stopping and resetting the process in case of emergency or whenever a particular situation arises. The data sent to the PLC in these cases is a priority for the controller program. Once the stop button is activated, the reset button is enabled and it has to be pressed to start another process from the beginning, Figs. 3 a, b.

Figure 3. a) Main interface. b) Process interface



Source:

Overall, 33 OPC items were used for the correct NC program interpretation and functionality of the interface and process. The items exchange data between the interfaces and the PLC, which controls the servo drives and therefore the axis system. The process interface sends the information extracted from the NC

program to the PLC so it has most of the OPC item used. In Figure 4, the items and their corresponding interface relation is shown. It is important for the process to start with all the items on-line; to this end, on the first tests, the items were monitored in the OPC Test Client tool of RSLinx.

Figure 4. OPC items on-line

ItemID	Sub Value	Sub Quality	Sub Updates	Update Rate	Run. Avg
[PRUEBA_OPC]OPC_BIT	0	Good	1	0	0.012483
[Prueba_OPC]OPC_DINT	6	Good	1	0	0.012483
[PRUEBA_OPC]PosicionFeedbackx	238.24	Good	1	0	0.012483
[PRUEBA_OPC]PosicionFeedbacky	-0.00192	Good	53	0	0.661599
[PRUEBA_OPC]Inicio	0	Good	1	0	0.012483
[PRUEBA_OPC]NN	0	Good	1	0	0.012483
[PRUEBA_OPC]posiciony	0	Good	1	0	0.012483
[PRUEBA_OPC]posicionx	0	Good	1	0	0.012483
[PRUEBA_OPC]Inicio_X	0	Good	1	0	0.012483
[PRUEBA_OPC]Inicio_Y	0	Good	1	0	0.012483
[PRUEBA_OPC]offx	0	Good	1	0	0.012483
[PRUEBA_OPC]offy	0	Good	1	0	0.012512
[PRUEBA_OPC]llegox	0	Good	1	0	0.012512
[PRUEBA_OPC]llegoy	0	Good	1	0	0.012512
[PRUEBA_OPC]GGG	0	Good	1	0	0.012512
[PRUEBA_OPC]G01	0	Good	1	0	0.012512
[PRUEBA_OPC]G032	0	Good	1	0	0.012512
[PRUEBA_OPC]Direccion	1	Good	1	0	0.012512
[PRUEBA_OPC]RadioINT	0	Good	1	0	0.012512
[PRUEBA_OPC]llego_XY	0	Good	1	0	0.012512
[PRUEBA_OPC]llego_CXY	0	Good	1	0	0.012512
[PRUEBA_OPC]InicioLATCH	0	Good	1	0	0.012512
[PRUEBA_OPC]llegaron	0	Good	1	0	0.012512
[PRUEBA_OPC]NN_reset	0	Good	1	0	0.012512
[PRUEBA_OPC]sy0	0	Good	1	0	0.012512
[PRUEBA_OPC]finHOME	0	Good	1	0	0.012512
[PRUEBA_OPC]sx0	0	Good	1	0	0.012512
[PRUEBA_OPC]x_encendido	0	Good	1	0	0.012512
[PRUEBA_OPC]y_encendido	0	Good	1	0	0.012512
[PRUEBA_OPC]IniciarHOME	0	Good	1	0	0.012512
[PRUEBA_OPC]ShutDown	0	Good	1	0	0.012512
[PRUEBA_OPC]ShutDownReset	0	Good	1	0	0.012512
[PRUEBA_OPC]paroINTERPRETE	0	Good	1	0	0.012512

Source: own elaboration

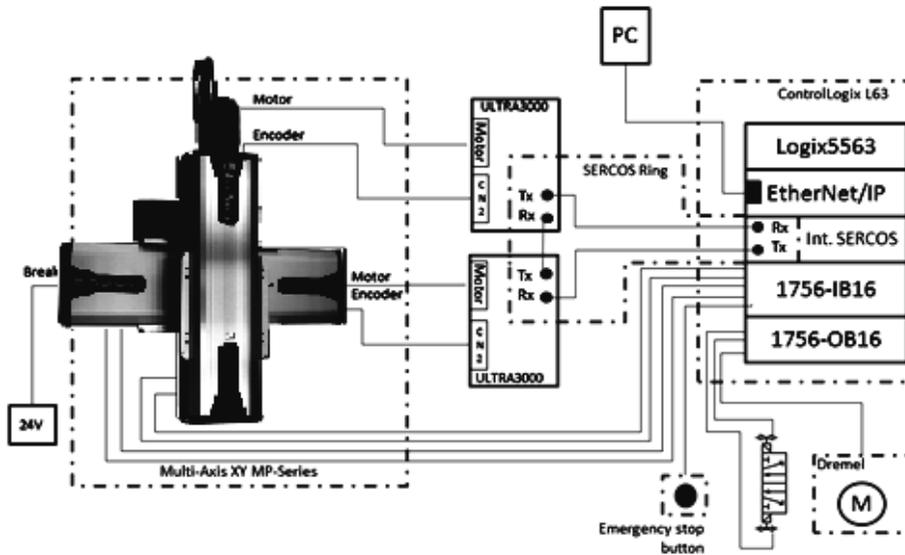
2.4 Control Module

When the interpreter has sent the corresponding data to the motion control instructions, the control module sets these data into the PLC program. The routines programmed in the PLC make the translation into motion control instructions possible as well as the execution of these commands. The program includes ladder and structured text program-

ming in RSLogix 5000. The commands are run block by block as the interpreter translates the program.

The connections and cabling of the system required are shown in Figure 5. The configuration of Ethernet and the OPC Server was made in RS-Linx Classic. The axes were configured following the Rockwell Automation manuals [15], [16].

Figure 5. Hardware diagram



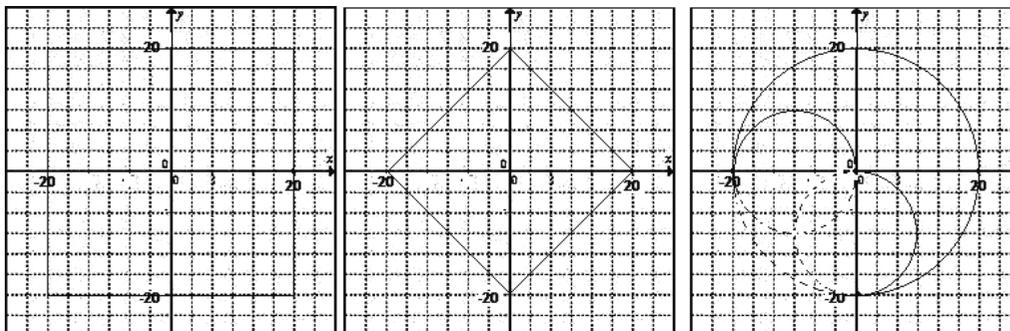
Source: own elaboration

2.5 Integration

For the integration of the interfaces and the control module three routines were programmed for each of the motion control instructions used. The first tests had all the PLC data preset for basic figures such as squares, diamonds and moon-like shapes, and also for testing the circular interpolations, as shown in Figure 6. Once the PLC program was ready for the incoming OPC data, the function of the process

interface started sending data of the same basic shapes, now described in a NC program file. This data-sending process was monitored using the OPC Test Client. Finally, data reading from the PLC to the interface for the position monitoring was added to both programs completing the software product proposed. The tests for the integration were not conducted, simulating the machining process controlling the axes in semi-automatic and single block modes.

Figure 6. Testing paths



Source: own elaboration

2.6 Validation

For the validation of the interpreter, the program was run for irregular paths. The Z parameter was identified during the process and a prompt window was displayed indicating the user when to turn the drill on/off and when to drill down or lift it. The validation of the interpreter was also made by using a requirement revision that included checking the correct functionality of the control user, browsing the file, programming the file displayed, verification, error handling, homing sequence, position monitoring, etc. The second validation was carried out through the machining process of the irregular path generated.

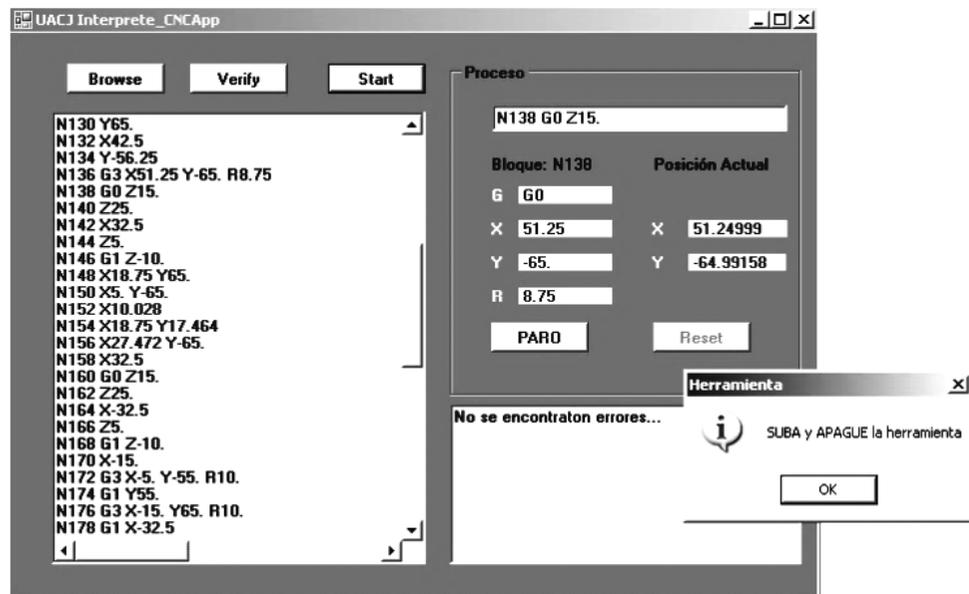
3. Results

The G-Code interpreter prototype was developed and successfully implemented using Visual Basic and a control module. The current version achieved all the functions proposed for the research. The program was made to be tested in a FANUC postprocessor 3XMILL NC

program as an input. The result of the main interface during a process is shown in Figure 7. The verification, error handling and monitoring worked as expected. If an error occurs during the NC program verification, the process will not continue and the errors will be displayed and saved in an error report file (the file location is indicated on a prompt window). Regarding the position monitoring, the current and expected position (corresponding to the current block) are displayed in the interface, as well as the block in process. This is possible due to the OPC data transference between the interface and the control module (PLC).

The first tests are shown in Figure 8; the results were as expected since the machined path followed the programmed routine. The milling process was done with a 1/16" cutter, a 2mm depth at a 7.08 in/min speed. The routines verified the motion machining commands (G00, G01, G02, and G03) conversion into motion control commands

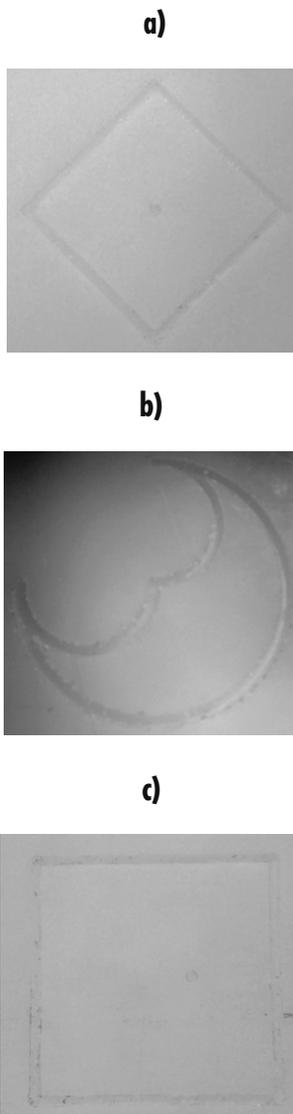
Figure 7. Main interface during process



Source: own elaboration

The equipment used for the control module included an L63 PLC, a Multi-Axis MPS-Series and two servos drives ULTRA3000, brand Allen Bradley. The control module program consists of seven routines: Startup, Shutdown, Homing, Restart, Positioning Motion, Linear Motion, and Circular Motion.

Figure 8. Motion control-routine test result.
a) Diamond, b) Moon, c) Square



Source: own elaboration

For the first run with the whole system integrated, a 1/8" cutter was used at a 4mm depth and a speed of 35.43in/min in a delrin sheet of 1ft 2.1 of thickness. For a better surface finish, the rest of the validation tests were run at a speed of 7.08 in/min with a 2mm depth. The path designed in MasterCAM X4, used for the validation, is shown in Figure 10. The final machining is shown in Figure 11.

Figure 10. "UACJ" paths CAM design.

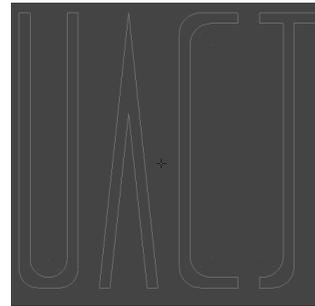
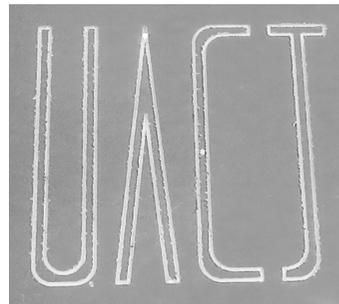


Figure 11. "UACJ" Machining results.



3. Conclusion

A prototype of G-Code interpreter was proposed. The interpreter was successfully implemented through a semi-automatic and single-block CNC milling process. The machining process used for the validation was possible through the control of a three-axis system. The control was PLC-based and the data was sent using OPC, making the communication bet-

ween the user interface and the control module possible. The conversion into motion control instructions made it possible to construct the interpreter from a physical point of view. Since the purpose of this research was to validate the interpretation of the motion machining control commands, the non-motion machining control commands remained fixed during the milling process. Compared to previous works, the present interpretation of the motion machining commands of the G-Code was taken into a control module to actually carry out a machining process (milling), [17], [18].

In future work, the lexicon and syntax rules will be incremented to have a complete dictionary data structure as well as the control of the Z-Axis for a complete automatic control of the CNC milling prototype machine. The GUI will also be improved by adding an edition tab and a simulator for path and machining.

References

- [1] Albert. "Understanding CNC Routers. Canada", Canadá: FP Innovations. 2011.
- [2] Fusaomi Nagata, Yukihiko Kusumoto & Keigo Watanabe. "Intelligent machining system for the artistic design of wooden paint rollers. Robotics and Computer-Integrated Manufacturing", 2009, pp. 680-688. doi:10.1016/j.rcim.2008.05.001:
- [3] Hui Zhang, Changsheng Ai, Fang Zhao & Xiangbo Ze, "The CNC Development of the Sawing and Milling Machining Center for Plastic Door and Window PVC Profile Based on Fanuc0. 7th World Congress on Intelligent Control and Automation (WCICA)", DOI:10.1109/WCICA.2008.4593912, 2008, pp. 9362 - 9367
- [4] W. Bolton, "Programmable Logic Controllers". USA: Newnes, 2009.
- [5] Rockwell Automation, "Literature Library", 2013, April, disponible en: <http://ab.rockwellautomation.com/>
- [6] Long Chen, Dong Yu, Han Zhang, Cong Geng & Lei Dong. "Design and implementation of a modularized CNC interpreter based on the integration of tool path planning module". IEEE International Conference on Computer Science and Automation Engineering (CSAE), vol. 3, DOI: 10.1109/CSAE.2012.6273027, 2012, pp. 613-616
- [7] Gao Qi & Zu Yingli (2011). Research on universal NC program processor for CNC system. 2nd International Conference on Digital Manufacturing & Automation. 2011, pp. 1046- 1049.
- [8] Yadong Liu, Xingui Guo, Wei Li, Kazuo Yamazaki, Keizo Kashihara & Makoto Fujishim, "An intelligent NC program processor for CNC system of machine tool. Robotics and Computer-Integrated Manufacturing". 2007, pp.160-169
- [9] Xingui Guo, Yadong Liu, Daoshan Du, Kazuo Yamazaki & Makoto Fujishima, "A universal NC program processor design and prototype implementation for CNC systems". The International Journal of Advanced Manufacturing Technology, 2012, volumen 60, pp. 561-575, DOI: 1.1007/s00170-011-3618-6
- [10] H. Lan, R. Liu & C. Zhang, "A multi-agent-based intelligent STEPNC controller for CNC machine tools". International Journal of Production Research. , vol. 46, no. 14, pp. 3887-3907, Julio 2008
- [11] W. X. Chen (2010). Research on STEP-Compliant CNC System for Integration of CAD/CAPP/CAM. Mechanical Manufacturing and Automation, pp. 529-551, 2010.

- [12] Haitao Hong, Dong Yu, Xiaohui Zhang & Long Chen (2010), Research on a new model of numerical control program interpreter. 2nd International Conference on Advanced Computer Control (ICACC), vol. 2, 467-472. doi: 10.1109/ICACC.2010.5486632
- [13] Guo Huijuan, Guo Huijuan & Zhi Derui. Design of several OPC servers communication system. International Conference on Electric Information and Control Engineering (ICEICE), pp. 317-319, 2011.
- [14] Yuxiang Wu & Christine W. Chan, "A data analysis decision support system for the carbon dioxide capture process. Expert Systems with Applications", pp. 9949-9960, 2009
- [15] Rockwell Automation (2010, November). MP-Series Integrated Multi-axis Linear Stages. Allen Bradley, Rockwell Software, Rockwell Automation, pp. 1-136, 2010, November
- [16] Rockwell Automation (2011, November). SERCOS and Analog Motion Configuration and Startup. Allen Bradley, Rockwell Software, Rockwell Automation, 1-318, MOTION-UM001B-EN-P.
- [17] QIAN Xinhua, JIA Shiyang, SU Xing, CHEN Yue, WANG Kefeng & YAO Pingjing (2010). Adaptive
- [18] On-line Operation Guide for Dry Gas-to-ethylbenzene Reactor. Chinese Journal of Chemical Engineering, vol.18, no.3, 419-424.