REMOTE MACHINING OF PRISMATIC PARTS THROUGH THE INTERNET IN A CNC MACHINE COMPLIANT WITH THE STEP-NC STANDARD

Fábio José Souza
REIVAX S/A Automação e Controle, Rod. José Carlos Daux, 600, Tecnópolis, 88030-904, Florianópolis, SC, Brazil
fabioautoind@gmail.com

João Carlos Espíndola Ferreira
Carlos Alberto Martin
Wagner Fernando Gascho
UFSC – Universidade Federal de Santa Catarina, Departamento de Engenharia Mecanica, GRIMA/GRUCON, Caixa Postal 476, 88040-900, Florianópolis, SC, Brazil
j.c.ferreira@ufsc.br
c.am.martin@ufsc.br
wgascho@bol.com.br

Abstract: *The current standard programming of machine centers is still the (ISO 6983-1) standard, known as G-code, which was developed in the 1950s. This language contains low-level information performing the movement of the tool in the machining process, making it difficult for the operator to take decisions in machining. For this reason, the (ISO 6983-1) standard currently limits significantly the flexibility and interoperability part manufacturing. Given this scenario, in the 1990s a new data model for manufacturing processes was proposed to extend the STEP standard (ISO 10303-1), called STEP-NC (ISO 14649-1 standard), which was developed as a result of numerous projects by companies and universities. In this paper we describe the development and validation of software and hardware that allows the user to command a CNC gantry milling machine so that it will machine parts using the STEP-NC standard (ISO 14649-11). STEP-NC is based on the concept of workingsteps, machining features and machining operations, among other entities described by the standard (ISO 14649-1). The STEP-NC Controller software was developed using the Java programming language, which enabled the creation of a control interface for the CNC milling machining process. Programming compliant with the RS-274 standard (G and M codes) generated by the open source software GRBL (software and control movements on machines) was used to program the Arduino MEGA2560 microcontroller in the C programming language. G-code programming was removed from GRBL in order to make room for use of the STEP-NC standard (ISO 14649-1), thus allowing the physical file p21 (a part of the STEP standard) to be interpreted and sent to the axes of the CNC milling machine. A direct connection is established between the STEP-NC Controller and a board with 8 relays to allow the CNC milling machine to work selectively (either with G and M code or STEP-NC). Remote operation of the machine via the Internet was also implemented, allowing the sending of a p21 file via web, and the machining process can be monitored via webcam.*

Keywords: *Numerical Control, STEP-NC, Milling Machine, Standardization, G-Code, Internet.*

1. INTRODUCTION

For over 50 years computer numerical control (CNC) machines have been using the same programming standard to manufacture parts, which corresponds to the ISO 6983-1 (1982) standard known as G-code (Kemmerer, 2009). It contains low level information that have been used as an interface between CAM (Computer-Aided Manufacturing) and CNC. However, ISO 6983-1 (1982) standard focuses on the programming of the cutting tool path relative to the machine axes rather than the machining process relative to the workpiece (Rosso Jr., 2005).

Moreover, it is infeasible for the CNC control system to run an autonomous and intelligent control that addresses potential needs for changes, such as changing a workpiece surface, causing the loss of much information relating to machining strategies, making it difficult or impossible to share such information (Harbs, 2012).

Given these limitations, a new data model among CAD (Computer-Aided Design), CAM, and CNC systems has been designed internationally, known as the STEP-NC standard (ISO 14649-1, 2003), which includes extensions for milling, turning, drilling, and electro discharge machining (EDM) processes. This standard defines an object-oriented data model for CNC, with a detailed data structure interface that incorporates feature-based programming (Álvares, 2005).

The STEP-NC standard has a high level language that contains extensive information on the machining process, allowing not only the interoperability of CAD/CAM /CNC, as well as sending the STEP-NC file containing the tool paths via Internet to the machine controller, making manufacturing more flexible and rapid when compared with the use of G-code (Benavente *et al.*, 2013).

STEP-NC helps companies to accelerate their decision making related to the manufacturing process due to the merging of CAD, CAM, and CNC into a single software, besides enabling the sharing of process information as a whole, both internally and via Internet (Suh *et al.*, 2008).

Given the advantages of the STEP-NC standard, this paper describes the conversion of a CNC milling machine that understands G-code to work also with the STEP-NC standard, configuring a hardware for the machine to work selectively (either G-code or STEP-NC). The open source GRBL software (GitHub, 2015) was used, compiling it in Arduino Mega2560 microcontroller. The STEP-NC file in the part 21 format of the STEP standard (ISO 10303-21, 2002) is loaded into the CNC, and the implementation is validated by machining a part using STEP-NC.

2. THE G CODE - ISO 6983 STANDARD

The G-code (ISO 10303-1, 1982), also known as RS-274, is the name used to refer to the numerical control (NC) programming language, in which the developer defines the parameters of what to manufacture so that the CNC machine makes the part. However, the reduced flexibility of this standard does not guarantee the exchange of programs between machines (Xu and He, 2004). The programming of NC machines using the ISO 6983 standard results in large and difficult programs to manipulate. Changes and last minute corrections of complex programs are difficult to manage on the shop floor, and control of the program execution on the machine is very limited (Stark *et al.*, 2002).

In order to simplify the programming activities and to make the NC programs more compact, the ISO 6983 standard provides a special set of functions called canned cycles. It has for example a cycle for deep drilling with chip breaking (G83), also known as a “peck drilling”, where the drill moves in and out the part to facilitate chip breakage and removal (Rosso Jr., 2005).

The programming differences in canned cycles and subroutines lead to more than 5000 “dialects” (Stark *et al.*, 2002), resulting in interoperability problems between CNC machines. Therefore, a standardized approach to interoperability in CNC manufacturing is required, and it is for this reason that manufacturing-related standards have evolved (Rosso Jr. and Newman, 2003).

3. THE NIST RS274NGC INTERPRETER

The RS274NGC interpreter reads the G-code program and generates calls to a set of canonical functions to perform machining. The interpreter's output can be used to move three to six axes machining centers (ISO 14649-11, 2003).

A G-code file is used as input to the interpreter, while the output commands can be printed for future use or can be run directly on a CNC machining center. There is also a report that includes a full description of RS274NGC input language and the machining canonical functions called by the interpreter (ISO 14649-11, 2003).

4. STEP-NC - ISO 14649 STANDARD

The CNCs are becoming increasingly complex, requiring a decentralized control together with a flexible and customizable internal structure (Minhat *et al.*, 2009). In this context, STEP-NC (ISO 14649-1, 2003) provides a data model that can be used not only for the complete description of parts, but also the manufacturing process with its parameters and tool requirements, and it can be used to transfer data between CAD, CAM, and CNC systems (Yusof *et al.*, 2009).

STEP-NC shows each step of machining instead of the movement of the tool in relation to the machine, using the object-oriented concept referred to as “workingstep”. A workingstep subdivides each machining operation on steps needed to perform the operation, which can be considered similar to the canned cycles in the traditional G-code. Unlike the G-code, STEP-NC informs the machine “what” to do, instead of “how” to do. Each task is related to an operation and a machining feature. With CNCs compliant with STEP-NC, complete information about the part can be shared, as the 3D model contains design tolerances, manufacturing resources, process sequence, and cutting tool requirements.

The milling process in the STEP-NC standard is based on ISO 14649-10 (2003), ISO 14649-11 (2003), and ISO 14649-111 (2000), which consist of: (a) task description, (b) technology description, (c) cutting tool description, and (d) geometric description. The task description contains the logical sequence of executable tasks such as “machining_workingstep”, “NC_function”, and data types. The details of each workingstep are given in the description of the technology with reference to the description of the tools and the description of the geometry (Suh *et al.*, 2003).

The workingsteps include 2.5D manufacturing features (“two5D_manufacturing_feature”) and 3D (“region”) for general milling operations, and each workingstep has subclasses such as “planar_face”, “pocket”, “step”, “slot”, “round_hole”, and “general_outside_profile”, together with the machining conditions in the p21 file. It should be noted that the tool path in the milling process in STEP-NC is optional, unlike the current CNC programming in which the tool path is the main content of the information necessary to move the machine axes (Suh *et al.*, 2003).

5. EXAMPLE OF A MILLING OPERATION IN THE STEP-NC STANDARD

Figure 1 shows an example of a milling operation contained in the structure of a STEP-NC program (ISO 14649-11, 2003). Each STEP-NC program starts with the entity “PROJECT” (#1), which defines the sequence of machining processes using the attribute “MAIN_WORKPLAN” (#2) and sets the pieces of information using the attribute “WORKPIECE” (#4). The WORKPLAN may contain other workplans, as well as one or more WORKINGSTEPS, which are inserted in a sequence. In the program of Figure 1 five MACHINING_WORKINGSTEPS are executed sequentially. First, a finishing operation for the entity PLANAR_FACE (#10) is executed and then drilling (#11) and boring (#12) operations are performed for the ROUND_ROLE. Finally there is the roughing (#13), the finishing (#14), and operations related to the feature CLOSED_POCKET.

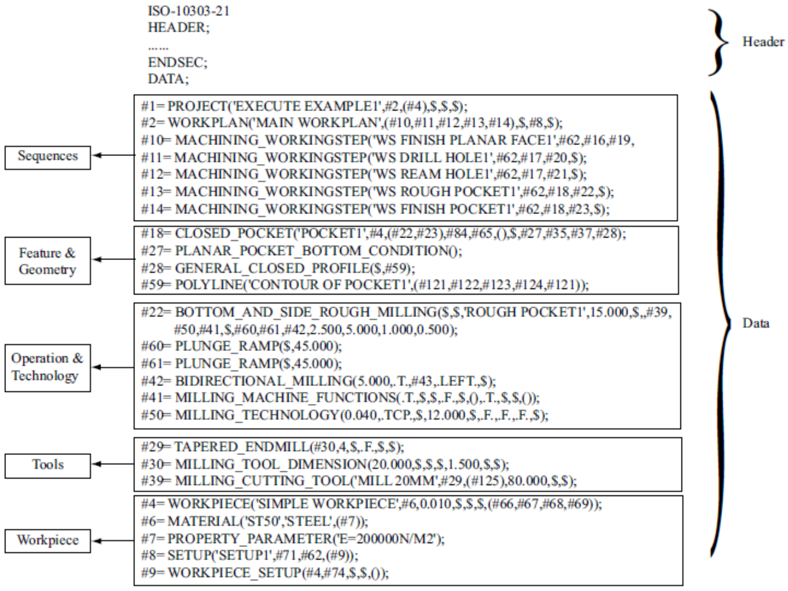

Figure 1. Excerpt of the STEP-NC program of a milling operation (ISO 14649-10, 2003)

The “Feature & Geometry” section contains resource information in the STEP-NC program, in this case the feature CLOSED_POCKET. The bottom of the pocket is defined as PLANAR_POCKET_BOTTOM_CONDITION (#27), while the GENERAL_CLOSED_PROFILE operation (#28), more specifically the POLYLINE (#59) is used for contouring of the CLOSED_POCKET operation.

In the section “Operation & Technology” the method to perform the machining operation is defined. The BIDIRECTIONAL_MILLING entity (#42) sets the direction of the machine, the feed rate along the way, etc. If these values are omitted, the CNC can decide these values autonomously. The MILLING_TECHNOLOGY entity (#50) defines the machining conditions such as the feed and the rotation of the spindle. The feed can be set using the FEEDRATE or FEEDRATE_PER_TOOTH attributes, and the rotation of the spindle can be defined using the SPINDLE or CUT_SPEED attribute. Additional information such as the simultaneous movement of the spindle and feed can be set. The MILLING_MACHINE_FUNCTIONS entity (#41) sets the elements of the machine tool, such as the air pressure, the cutting fluid, chip removal, etc.

With reference to the cutting tools section, a TAPERED_ENDMILL (#29) is used, which defines the diameter (20.0mm), the radius of the cutting edge (1.5mm), the total length (80.0mm), and the number of teeth (4).

Information about the raw material of the part is defined in the STEP-NC workpiece entity. It should be mentioned that the G-code does not provide information on the part, and therefore only the operator knows the information contained in the part and decides the cutting conditions for its machining. Moreover, STEP-NC includes the start and end shapes of the workpiece, the type of material to be machined, and the workpiece location (Figure 1) (Suh *et al.*, 2008).

In view of the advantages contained in the STEP-NC standard, we attempted in this work to convert a three-axis CNC milling machine (router type), originally programmed in G-code, to the STEP-NC standard. The X and Y axes (horizontal) are controlled by servomotors, whereas the Z axis (vertical) is controlled by a stepper motor.

6. CONVERTING A CNC MILLING MACHINE COMPLIANT WITH G-CODE TO STEP-NC

To accomplish the conversion of the CNC milling machine to STEP-NC it was necessary to find a way to separate the signals that enable not only to drive the spindle, but also the pulse, direction, and ground (GND) signals arising from the machine controller, responsible for reading and interpreting the G-code. This separation of the signals was made so as to enable the operator to work with both G-code and STEP-NC in the manufacture of parts.

The solution was to install a board with eight relays as shown in Figure 2, and this installation is relatively simple in the machine control panel without changing the electrical parts. An external 5V power was required, which can be powered by the MK1 plate itself (of the Planet CNC company) (2014). We used a microcontroller with 5V outputs as power supply to external hardware. Each relay has the function of switching and dividing signals, and may contain three signals connected to each of its eight relays, totaling 24 signs.

Each of the eight channels receives a 5V voltage and needs a 15 to 20mA current, besides withstanding high current such as AC250V 10A and DC30V 10A. This board has eight LEDs on each relay in order to indicate the enabled status (5V) or disabled status (0V), besides the normally closed contact (NF), normally open (NA), and common to all. All eight relays on each channel need a 5V supply voltage so they change from NF to NA, providing the redirection of the pulse and direction ways in the CNC machine.

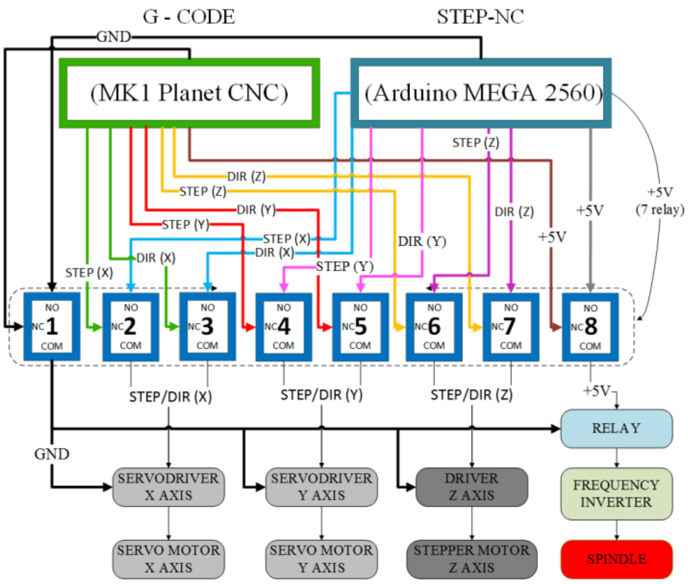

Figure 2. Flowchart of the separation of the pulse, direction, GND, and spindle signals in the eight relays board

6.1. Configuration of the hardware for the CNC to work in a selective manner (G-code or STEP-NC)

The eight relays board provided the selective change between G-code or STEP-NC in the CNC machine. The Arduino Mega2560 microcontroller was used to execute and transmit the STEP-NC tool paths to the CNC machine, in addition to signals that enable the spindle, pulse and direction signals to the X, Y, and Z axes.

The microcontrollers must work separately because each platform (i.e., MK1 and Arduino Mega2560 boards) sends STEP, DIR, and 0V signals to each axis (X, Y, Z), as well as + 5V and 0V signals to the relay, which triggers the frequency inverter for rotating the spindle. These signals must be connected separately so that there are no signal problems of a microcontroller being sent to the other microcontroller (Figure 2).

In order for the pulse, direction, and GND signals of each of the axes of the machine pass the relay board, as well as the signal that triggers the spindle of the MK1 board, each of the seven NF contacts of each relay board was used for transmitting signals. These signals pass through the common contact (COM), which is received by the servomotors on the X and Y axes, and the stepper motor driver on the Z axis. For the same signals coming from the Arduino Mega2560 each of the seven NA contacts in each relay board was used for signal transmission. A command of the software called “STEP-NC Controller” enables NF switching to NA. This software sends a command to the Mega2560 microcontroller, which in its programming a +5V output triggers seven of the eight relays in the signal transmission board. In order to the drive the spindle, the +5V signal coming from either MK1 or Arduino Mega2560 boards triggers another relay that enables the NA contact so that this signal sends a pulse for the frequency inverter to turn the spindle on.

6.2. Interpreter of the STEP-NC file for generating the cutting tool path

In order to interpret the p21 file of the STEP-NC standard, it was necessary to use the CAD/CAPP/CAM system called STEP Modeler, which can be executed on the page http://www.grima.ufsc.br/stepnc_project/ (Benavente *et al.*, 2013). This system allows the modeling of a prismatic part via the Internet based on machining features, and allows the remote user to generate the STEP-NC file, and also provides the visualization and simulation of the tool paths.

In this work we used the open source control software GRBL (GitHub, 2015), which is a low-cost alternative based on parallel port to control movements of CNC machines. It runs on Arduino with the Atmega2560 chip. GRBL was written in C, and uses all the intelligent features of the AVR-chips in order to achieve precise timing and asynchronous operation. The controller can read up to 18 movements ahead of the momentary motion sequence, planning their speeds ahead and in order to maintain the acceleration and smooth curves, preventing the CNC machine from suffering any type of sudden movements compromising its physical performance. Among its files there are libraries in the “.h” format and files in the “.c” format.

6.3. Analysis of the G-code in GRBL

To understand the path taken by pulse and direction signals of each of the three axes of the CNC milling machine, the “serial.c”, “protocol.c”, “gcode.c”, and “stepper.c” files of the GRBL software were analyzed. The path taken by the G-code in GRBL starts at the “serial.c” file, specifically the ISR interrupt (“USUARIO_RX_vect”), which receives the data serially byte by byte. The recorder that stores the received byte is UDR0, and the received bytes are stored in a queue named “rx_buffer”. In parallel to this path, the “serial_read” function is responsible for reading the buffer in order to the bytes and return the value of the byte read.

The “protocol.c” file is responsible for preprocessing lines, executing a line in the input format to the “protocol_process”. It also performs internal commands in GRBL such as settings, start of homing cycle, and changing the switching conditions. This differs from the real time control module running on GRBL when it is ready to execute the next line during a cycle and, if so, the switches turn off the lines of the execution blocks. This option affects only the lines that will be processed later, not necessarily in real time during a cycle, since there are movements already stored in the buffer. However, this delay is not a problem, since these commands are not normally used during a cycle.

In the “gcode.c” file the “gc_execute_line” function is responsible for transforming all G-code lines in internal parameters. In the “protocol.c” file, “protocol_process” calls the “serial_read” function, which reads the byte in a row and returns. These are organized by “protocol_process” in a matrix called “line”. Then “protocol_process” sends the matrix to “protocol_execute_line”, which in turn verifies whether the received line is not an internal parameter. If not, the line is sent to the “gc_execute_line” function where the G-code is interpreted. After processing, the “stepper.c” file aims to send the pulse and direction signals to the X, Y and Z axes.

6.4. Developed software for sending the STEP-NC program to the CNC machine

The STEP-NC Controller software, developed in Java, is responsible not only for sending the STEP-NC file, but also to control the movements of the X, Y and Z axes, as well as the drive of the cutting tool in the CNC milling machine. It can be included in the STEP Modeler software so that it contains activities from product design to manufacturing in an environment fully compliant with the STEP-NC standard. The STEP-NC Controller software was developed so that the Arduino receives, interprets, and executes the GRBL program modified for the STEP-NC standard in the CNC milling machine.

In the STEP-NC Controller software, when a button is clicked, a series of events linked to that button are triggered, running several threads (i.e., executing lines in a software). The main window of STEP-NC Controller is shown in Figure 3, which can be used to access these other windows: Serial Connection, File, Network, Machine Status, Control, and Console.

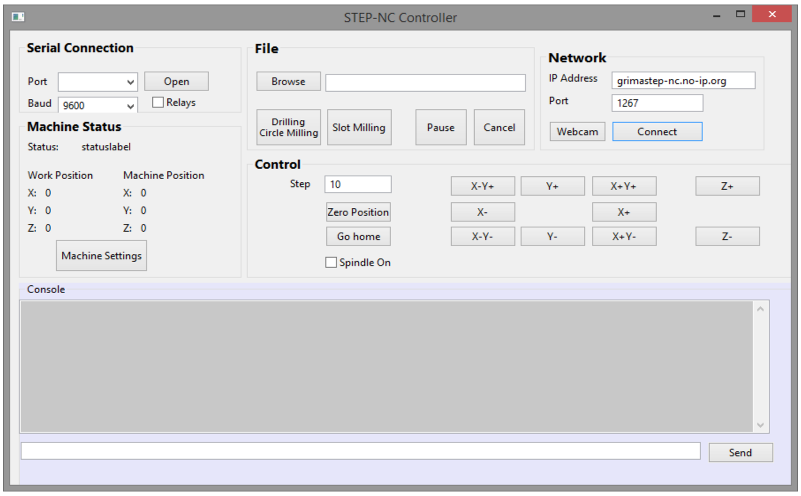

Figure 3. Interface of the STEP-NC Controller software

6.5. Programming the STEP-NC-compliant CNC machine

To develop programming together with the interpreter of the STEP-NC file, including reading and sending tool paths to the CNC machine, the software from Atmel Studio 6 was used, which is an integrated development platform for developing applications for Atmel AVR microcontrollers. It has a simple environment that facilitates writing, compiling, and debugging programs written in C/C++ or assembly languages. It is a license-free software, being integrated with the Atmel Software Framework (ASF), which has a library in its source code. Atmel Studio 6 is responsible for compiling programs in C/C++ languages, and writes “.hex” file into the Atmega2560 microcontroller of Arduino Mega2560.

For programming the STEP-NC standard, new files have been implemented in the “.c” format and “.h” libraries. Among the existing files, “gcode.h” and “gcode.c” were removed, and “parametrosgerais.h”, “step_nc.c”, “step_nc.h”, “step_nc_tools.c”, “step_nc_tools.h”, and “stepnc_ini.h” were added.

6.6. Remote operation of the CNC machine via Internet

The STEP-NC Controller software resides on a server that provides a box that allows the user to access the port used to make the local transmission of the STEP-NC file to the Arduino via serial port. In addition to this box, the server provides two other fields in which the user enters the local IP of the machine where Arduino is located, together with its network port. In that manner, access is granted to the user, who will have full control of the machine via the STEP-NC Controller interface. All the messages from the remote user are redirected via serial port to the Arduino Mega2560 and vice-versa. The following libraries were used: (a) Java.io (InputStream and OutputStream) for receiving data and transmitting via sockets, (b) java.net.Socket using the Transmission Control Protocol (TCP). The server has a function that stores the last setting of ports used so there is always connection between the remote user and the Arduino Mega2560.

To carry out remote communication with the server, two connection options were made available: (a) manually, by entering the current IP of the router using port 1267, (b) automatically by clicking the “Connect” button, where there is no need to enter the IP address manually (Figure 4). This automatic feature was added because every time the IP of the central server is changed, all IPs are renewed, causing the user to perform an IP search on the local machine to have access to the STEP-NC server of the milling machine.

For image capture using the webcam (Figure 5), as part of programming the “com.github.sarxos.webcam.Webcam” library was used, which was added to the Eclipse software (IDE for programming in Java) to connect to the STEP-NC server.

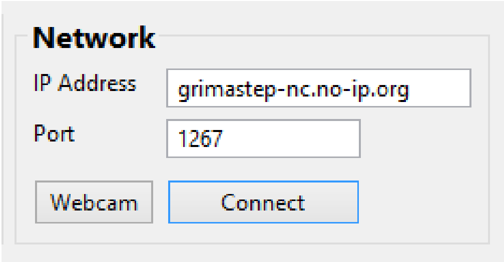

Figure 4. Connection with the STEP-NC server

Figure 5. Picture of the CNC machine taken by the webcam

Figure 6 shows the flowchart of the implemented system, illustrating the modules for the use of G-code and STEP-NC in the CNC milling machine.

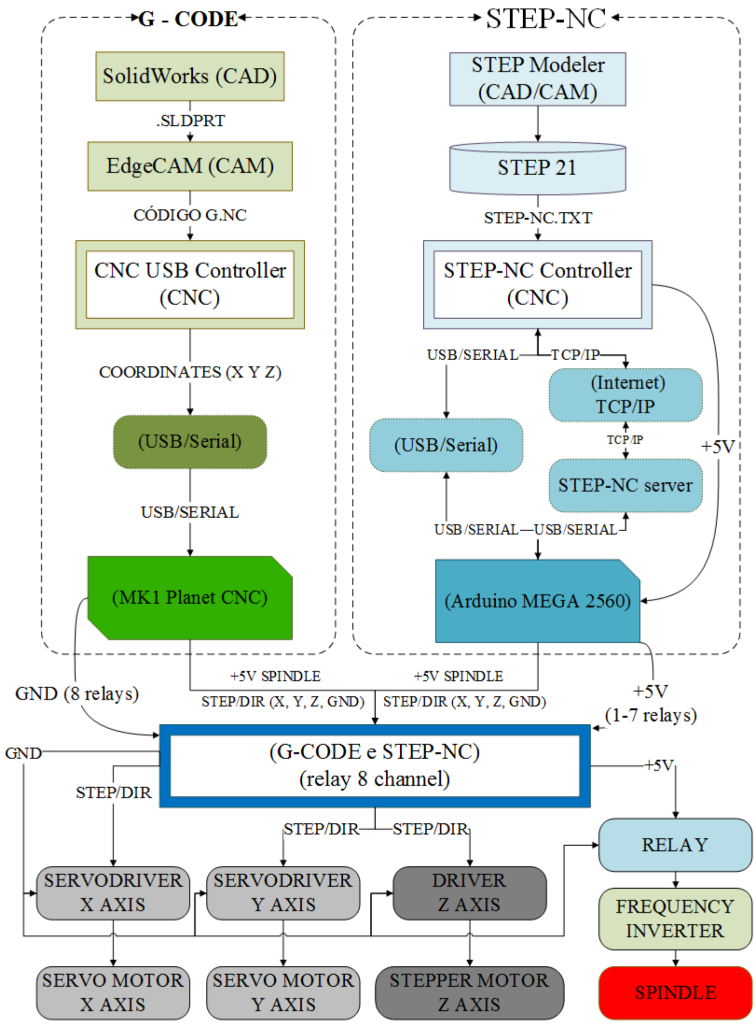

Figure 6. Flowchart of the implemented modules for using G-code and STEP-NC in the CNC machine

7. MACHINING A PART

The implemented system was tested machining a part of PVC containing five holes and a circular slot, which has been modeled in the STEP Modeler system. The STEP-NC file was generated and sent to the CNC machine, and the machined part is shown in Figure 7. The same part was also machined on the same machine using the G-code.

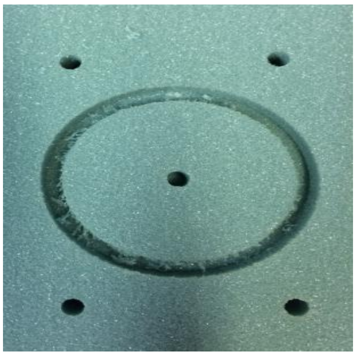

Figure 7. PVC part with five holes and one circular slot that was machined using the STEP-NC standard

8. CONCLUSIONS

In this work it was attempted to make a CNC milling compliant with the STEP-NC standard (ISO 14649-11, 2003). The Arduino Mega2560 microcontroller was used in the CNC milling machine, which executed the STEP-NC files in the part 21 format. In order to achieve this, it was necessary to implement a software to convert the STEP-NC file format to drive the machine axes.

The validation of the conversion software, which is part of the STEP-NC Controller, was performed by machining a part in the CNC milling machine. The same part was also machined using the G-code, so as to compare both implementations.

STEP-NC was implemented using the open source software GRBL, which contains files that have been programmed specifically for G-code according to the standard (ISO 6983-1, 1982). The files contained in GRBL, which perform all mathematical calculations and transformations of the paths into pulse and direction signals for the movement of the machine axes, worked properly for the drilling and milling processes. However, the same performance was not obtained for the STEP-NC implementation, mainly circular milling, in which pauses occur between each line segment, due to the transformation into pulse and direction signals being based on the G-code and not on STEP-NC.

The pauses occurring during the circular milling process in the STEP-NC implementation take place because there is a difference in calling entities between the G-code and STEP-NC. In the G-code, GRBL receives the ".txt" file and performs all the movements from the Arduino memory, whereas in the STEP-NC implementation, instead of reading in the line in the memory, the line is selected and read from a pile in which each entity is assigned to a queue. In G-code each line generates a movement, while in STEP-NC this does not happen because each entity has movements and parameters that are responsible for the movement of the CNC machine axes.

It is considered that the implementation using the STEP-NC standard will achieve an equivalent or better performance than the implementation based on the G-code if the programming is not based on the files responsible for transforming the tool paths into pulse and direction signals contained in the GRBL software. The replacement of the programming language currently in C (structured) to C++ (object-oriented) is a way to improve the STEP-NC standard programming in the CNC milling machine.

9. REFERENCES

Álvares, A.J., 2005. "Uma Metodologia para Integração CAD/CAPP/CAM Voltada para Manufatura Remota de Peças Rotacionais Baseada na Internet", Tese de Doutorado, Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil.

Benavente, J.C.T., Ferreira, J.C.E., Goulart, C.M. and de Oliveira, V.G., 2013. "A STEP-NC compliant system for the remote design and manufacture of mechanical components through the internet", International Journal of Computer Integrated Manufacturing, Vol. 26, No. 5, p. 412-428.

GitHub, 2015. "GRBL", https://github.com/grbl/grbl

Harbs, E., 2012. "Controlador CNC aderente a ISO 14649 com ferramentas de software para construção e integração de uma arquitetura de controle baseada na IEC 61499", Dissertação de Mestrado, Universidade Estadual de Santa Catarina, Joinville, SC.

ISO 6983-1, 1982. "Automation systems and integration -- Numerical control of machines - Program format and definition of address words - Part 1: Data format for positioning, line motion and contouring control systems".

ISO 10303-21, 2002. “Industrial automation systems and integration - Product data representation and exchange - Part 21: Implementation methods: Clear text encoding of the exchange structure”.
ISO 14649-1, 2003. “STEP-NC Project, "STEP-Compliant Data Interface for Numerical Controls”, ESPRIT Project EP 29708, STEP-NC Newsletter, 2003.
ISO 14649-10, 2003. “Industrial automation systems and integration - Physical device control - Data model for computerized numerical controllers-Part 10: General process data”, ISO TC 184/SC 1, May 23.
ISO 14649-11, 2003. “Industrial automation systems and integration- Physical device control-Data model for computerized numerical controllers-Part 11: Process data for milling”, ISO TC 184/SC 1, May 23.
ISO 14649-111, 2000. “Industrial Automation Systems and Integration - Physical Device Control—Data Model for Computerized Numerical Controllers - Part 111: Tools for Milling”, ISO TC 184/SC 1, September 28.
Kemmerer, S.J., 2009. “STEP The Grand Experience”. Manufacturing Engineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, USA.
Minhat, M., Vyatkin, V., Xu, X., Wong, S. and Al-Bayaa, Z., 2009. “A novel open CNC architecture based on STEP-NC data model and IEC 61499 function blocks”, Vol. 25, No. 3, p. 560-569.
Planet CNC, 2014. “CNC USB Controller Mk1 - User manual”, http://www.planet-cnc.com/files/CNCUSBControllerMk1.pdf
Rosso Jr., R.S.U. and Newman, S.T., 2003. “Estrutura de dados para sistemas CAD/CAM aderente à STEP”, VI Congresso Ibero Americano de Engenharia Mecânica, Universidade de Coimbra, Portugal.
Rosso Jr., R.S.U., 2005. “STEP compliant CAD/CAPP/CAM system for rotational asymmetric parts”, Ph.D. Thesis, Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, UK.
Stark, J., Khai, N.V. and Maeder, W., 2002. “Standardisation of the manufacturing process: the IMS STEP-NC project”, ESPRIT project”, available in: http://www.step-nc.org/data/IPLnetWorkshop2002_paper_27082002.pdf.
Suh, S.H., Lee, B.E., Chung, D.H. and Cheon, S.U., 2003. “Architecture and implementation of a shop-floor programming system for STEP-compliant CNC”, Computer-Aided Design , Vol. 35, No. 12, p. 1069-1083.
Suh, S.H., Kang, S.K., Chung, D.H. and Stroud, I., 2008, “Theory and Design of CNC Systems”, Springer Series in Advanced Manufacturing, London: Springer.
Xu, X. and He, Q, 2004. “Striving for a total integration of CAD, CAPP, CAM and CNC”, Robotics and Computer-Integrated Manufacturing, Vol. 20, p. 101-109.
Yusof, Y, Tan, N.Z.Z. and Kassim, N., 2009. “ISO14649 Code Generator for Intelligent Manufacture for STEP-NC Compliant Machining”, Proceedings of the Asia Pacific Industrial Engineering and Management Systems Conference (APIEMS2009), 14-16 December.

10. RESPONSIBILITY NOTICE

The authors are the only responsible for the printed material included in this paper.