Design and Realization of a STEP-NC Compliant CNC Embedded Controller

Francesco Calabrese
Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio 21, 80125, Naples, Italy
fcalabre@unina.it

Giovanni Celentano
Dipartimento di Informatica e Sistemistica
Università degli Studi di Napoli Federico II
Via Claudio 21, 80125, Naples, Italy
giocelen@unina.it

Abstract

STEP-NC is a new model for data transfer between CAD/CAM and CNC that allows specifying machining process rather that tool motion with respect to the machine axes. STEP-NC CNC controllers are a new breed of CNC controllers that allow using this high-level model for a seamless integration in the CAD-CAM-CNC chain. Various prototypes of STEP-NC CNC controllers have been developed so far, mainly based on industrial PCs. In this paper we instead show how it is possible to realize a STEP-NC CNC controller using a low cost microcontroller, embedded in the CNC machine; this goal has been obtained by accurately parallelizing the tasks involved in the machining process. More precisely, this paper presents the design and realization of an embedded system able to control a CNC machine with two degrees of freedom, to machine STEP-NC part programs and to perform also other high level activities.

1. Introduction

Today's CNC machines have enormous capabilities such as multi-axis control, multi-process manufacture, etc. In the mean time, these capabilities have complicated the programming tasks and made machine tools less adaptable. CNC machines actually use a 50-year old language ISO6983 (also called G-code) [1]. Some limitations derived by the use of such standard are the following: *i)* in order to machine a part designed with a CAD/CAM software using an actual CNC machine it is required to convert, through the use of a post-processing module, the CAM project into a ISO6983 program, executable on the available machine; *ii)* ISO6983 defines the statements syntax but, in some cases, leaves ambiguities in the related semantics [2]; *iii)* ISO6983 allows describing the tool centre path with respect to the machine axes rather than the features to be machined; *iv)* vendors usually supplement the language with further commands to provide new features, but creating machine-specific languages.

To avoid such limitations, the research community which is working in this field is defining the new standard ISO14649, also called STEP-NC [2]-[3], which establishes a unified format for data transfer between CAD, CAM, SFP and CNC, avoiding conversions and post-processing mechanisms.

STEP-NC allows specifying machining process rather that tool motion with respect to the machine axes, using the object-oriented concept of workingstep. In fact, the generation of the tool path is performed at the latter stage of the manufacturing chain by a new breed of intelligent controllers called STEP-NC controllers. These controllers are usually classified in three categories [4]:

I. *conventional CNC control using STEP-NC*: a translation module, implemented on a PC, converts a STEP-NC program into ISO6983 commands, executable on a traditional CNC machine (see e.g. [5] and [6]).
II. *STEP-NC enabled control*: such controller allows processing a STEP-NC file directly into the CNC machine (see e.g. [7], [8] and [9]). However, the controller utilizes only low-level machining information (already available in a ISO6983 program).
III. *STEP-NC enabled intelligent control*: such controller differentiates from the previous ones because it also allows performing high-level activities like: automatic part setup, automatic and optimal tool path generation, accurate machining status and result feedback, adaptive control, etc (e.g. [10], [32]).

The prototypes of controllers realized so far utilize PCs for interpreting the STEP-NC programs and for generating the tool paths. Such PCs communicate with the CNC machines by means of I/O boards. In this paper we instead show how it is possible to realize a STEP-NC compliant CNC controller using a low cost microcontroller, embedded in the CNC machine. To this end, we have characterized and decoupled the different tasks involved in the machining process. Consequently it has been possible to parallelize some tasks with the aim of reducing dead times in the process execution and, at the same time, allowing on-line machining inspection and machining status feedback.

To be concrete, in this paper we present the design and realization of an embedded system able to control a CNC machine with two degrees of freedom, already designed and realized at the Embedded Industrial Microcontrollers Laboratory of the University of Naples Federico II, Italy, and to machine STEP-NC part programs. The developed controller is of the third type and allows performing some high-level activities: feasibility check, on-line machining inspection, accurate machining status and result feedback. The controller's interface makes use of a http server which allows the user to access and remote monitor the CNC machine through PC equipped with web browser and local network connection.

The paper is structured as follows. Sections II gives an overview of the STEP-NC standard. Sections III gives a description of the used CNC machine. Section IV describes the processes of design and implementation of the CNC embedded controller. Section V describes the implemented operation software. A machining test result is presented in section VI. Finally, some conclusions are given in section VII.

2. Overview of STEP-NC

The ISO10303 standard (also called STEP, STandard for the Exchange of Product model data) [11]-[13] provides a mechanism for describing production data along the development chain, for CAD to 3D CAM design, allowing implementing and sharing product databases.

The STEP-NC standard is an extension of STEP and allows connecting CAD/CAM design to CNC commands. The information contained in a STEP-NC file are divided in three subsections (see also Fig. 1):

- workplan executables;
- technology description;
- geometry description.

The *workplan* is characterized by a series of executables whose order is pre-established, or dependent on the machining actual conditions if conditional controls are used. The executables can be of 3 types: workingsteps, NC functions and program structures. The most important executables are the workingsteps which define manufacturing features. The workingsteps represent 2.5-D (two5D_manufacturing_feature) and 3-D (region) machining features. Each workingstep also includes further subfeatures (such as planar_face, pocket, step, slot, round_hole, genereal_outside_profile, etc) together with cutting condition information.

The *technology description* contains a detailed and complete description of all the workingsteps to be executed in the workplan; in particular this description includes data regarding tools, machining strategies, definitions of the workpieces, etc. A complete technology description includes for example: depth of hole to be machined, feed rate, rotational speed, diameter of the tool, etc.

As regards the *geometry description*, all the geometry data used in the various components is described using the ISO 10303 format.

Each STEP-NC program includes a *PROJECT* entity, the top-level entity that indicates the program's starting point. The *PROJECT* entity indicates the workplan to be executed and the workpiece upon which operations have to be performed. The *WORKPLAN* entity indicates the sequence of workingsteps to be machined while the *WORKPIECE* entity describes geometry and material of the part upon which operations have to be performed. Fig. 2 shows a detailed scheme of the described main entities.

Actually two versions of STEP-NC are being developed: Application Reference Model (ARM) (i.e. ISO14649 [2]-[3]) and Application Interpreted Model (AIM) of ISO14649 (i.e. ISO 10303 AP-238 [15]). In the following we will use the ARM model because it is more suitable in such application (see [16] and [17] for more information about the differences).

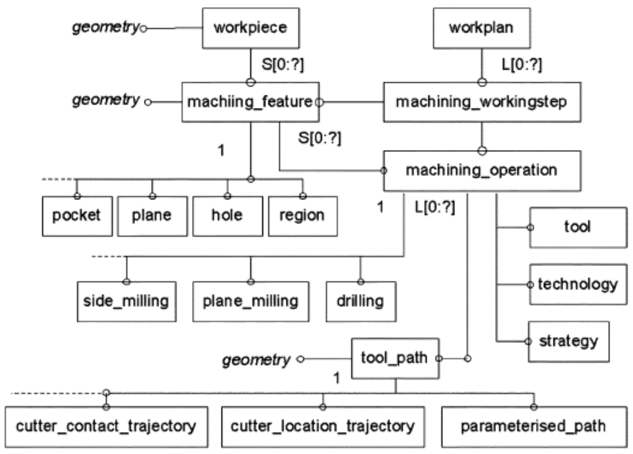


Fig. 1. Feature-based STEP-NC data structure [14]

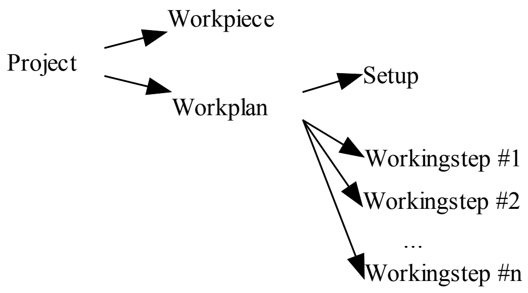


Fig. 2. Interpreted entities

3. Realized CNC machine

The system developed at the Embedded Industrial Microcontrollers Laboratory of the University of Naples Federico II, Italy, has the objective to prove the feasibility of an automatic manufacturing framework. To this end, we realized a two degrees of freedom cartesian robot with a plasma cutting tool (see Fig. 3) [19]. The hardware components utilized to control the robot are the following (see also Fig. 4 for a connections scheme):

- 2 1.8° step motors [20];
- 2 step motor drivers L297 and L298 [21]-[23];
- 4 limit switches;
- 1 relay to command the activation of the tool;
- 1 plasma cutter Cebora Plasma Prof 50;
- 1 embedded microcontroller system based on the Rabbit microprocessor RCM3700 with Ethernet connectivity [24]-[26].

The drivers are used to generate the power signals for the step motors, the frequency of pulses are calculated by an algorithm running on the microprocessor. The motor's torque is then converted to linear force by a rack and pinion. Such thing allows having a resolution of 0.25mm for each axis. The microprocessor's Ethernet connectivity and the embedded web server allow realizing a user interface by means of a http server, accessible through a web browser.

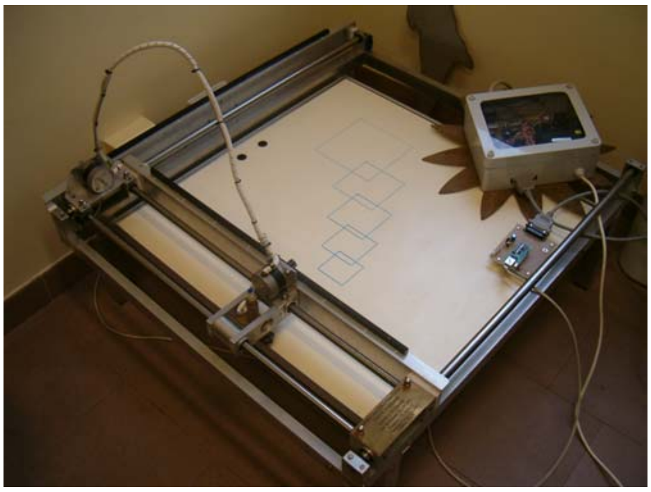


Fig. 3. Realized CNC machine

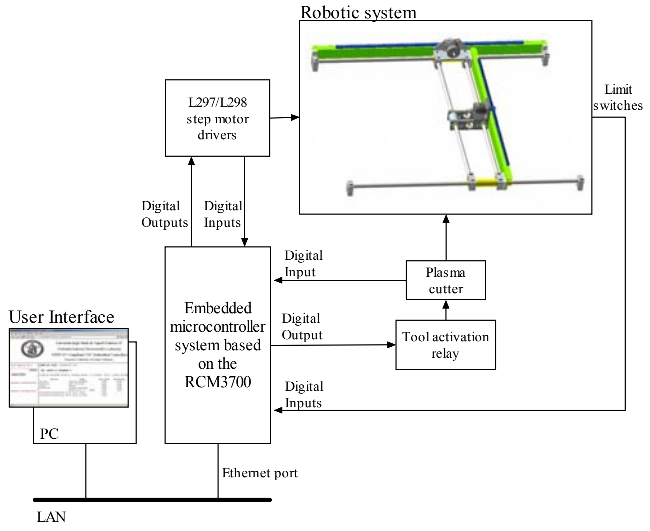


Fig. 4. Hardware connections scheme

4. STEP-NC Compliant Controller

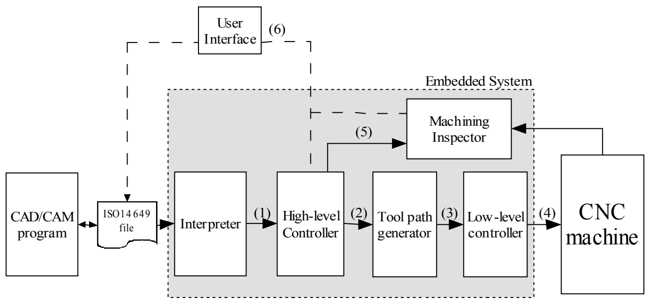
In this section we present the design process of the embedded system able to control the CNC machine presented in section III. In order to realize a STEP-NC compliant controller by means of a low-cost microcontroller, we have characterized and decoupled the different tasks involved in the machining process. Consequently, such tasks have been implemented with different software modules which exchange only specific information and can be executed in parallel.

The activities performed by a STEP-NC compliant CNC controller have been divided in five main tasks (see Fig. 5):

1. Interpreter;
2. High-Level Controller;
3. Tool Path Generator;
4. Low-Level Controller;
5. Machining Inspector.

Such activities substitute the ones performed by an ordinary CNC machine, which is only composed of an interpreter of ISO6983 programs and a low-level controller (see Fig. 6).

The characteristics and data exchanged between the different tasks are analyzed in the following subsections, while section V will describe the operation software.



(1) workingstep description
(2) workingstep parameters
(3) path
(4) motor/tool command signals
(5) workingstep control information
(6) machining status

Fig. 5. Scheme of the designed STEP-NC compliant CNC machine

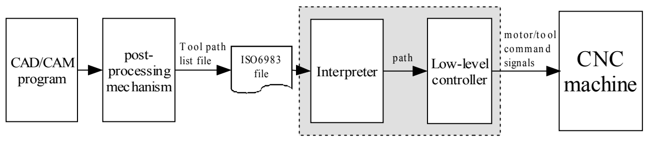


Fig. 6. Scheme of an ordinary CNC machine

4.1. Interpreter

This module interprets the STEP-NC file and memorizes the process information (features to be machined, tools to be used and sequence of operations to be followed) in apposite data structures. Then, for each workingstep, the module calls the High-Level Controller which will evaluate its feasibility and will order the execution of the specified feature.

Compared to this, a classic ISO6983 interpreter has only to traduce the tool centre movement defined by a ISO6983 program into axes movement. Then, such system does not need of a High-Level Controller and Tool Path Generator but only of a Low-Level Controller (see Fig. 6).

The Interpreter defines various structures to store the STEP-NC file information. Two basic structures, used to memorize information about position and placement, are the following:

```
struct point {                struct placement {
    float x;                      point cartesian_point;
    float y;                      point axis;
    float z;                      point ref_direction;
};                            };
```

The *WORKPIECE* entity is memorized in the following structure:

```
struct workpiece {
    Material material;
    STEPGeometry geometry;
    point dimension;
    placement part_placement;
};
```

The *SETUP* entity provides the origin and orientation of the part with respect to the machine's reference system. Such entity is used to describe the reference system to be used in the subsequent workingsteps, and is memorized in a placement element.

For the kind of considered CNC machine (see section III) only the following manufacturing features can be machined: pocket, slot, round hole and toolpath feature. Consequently, for each *WORKINGSTEP* entity the information are stored in the following structure (where some parameters are optional and depend on the kind of machining operation):

```
struct workingstep {
    Feature feature;
    Operation operation;
    float technology;
    float diameter;
    float orthogonal_radius;
    placement feature_placement;
    int qta_points_polyline;
    point wall_boundary [max_points];
    point feature_boundary_polyline [max_points_polyline];
};
```

4.2. High Level Controller

This module analyzes each workingstep description and evaluates its feasibility considering the CNC machine characteristics (workplan dimension, available tool, maximum velocity of each motor). Based on the part material, a cutting speed is determined and compared with the speed described in the technology description. If differences are found, the module communicates the user about the mismatch and suggests a more suitable cutting speed. If the workingstep is feasible, the controller will decide the machining operation to execute and will call the related tool path subroutine. The structure of the algorithm is schematized below.

```
space=checkWorkplanDimesion(workpiece, CNCparameters);
tool=checkTool(workingstep, workpiece);
speed=evaluateToolCuttingSpeed(workingstep, workpiece);

if (speed is not feasible)
        speed=modifySpeed(speed, CNCparameters);

if (space is available and tool is available and speed is feasible) {
        feature=selectMachiningOperation(workingstep, speed);
        workingstep_feasible=true;
}
```

4.3. Tool Path Generator

This module is composed of several subroutines which implement tool path generators for the different kinds of features: pocket, slot, round hole, toolpath feature [18]. As exemplification, in the following we comment the tool path generator for a pocket.

A *pocket* is specified in STEP-NC by the following parameters (see Fig. 7) [2]:

- kind of pocket (close or open);
- pocket bottom condition (in our case only Through pocket);
- orthogonal_radius;
- wall_boundary (in case of open pocket);
- feature_boundary, specified by a polyline (a sequence of cartesian points).

Using these information, the module generates a suitable sequence of segments and arcs the plasma torch has to follow to cover the pocket's boundary (see the dashed line in Fig. 7). Based on the specified cutting speed, the sequence is then converted in a trajectory (see Fig. 8) in the reference system of the pocket. This trajectory is converted to the one in the machine reference system and then sent to the Low-Level Controller.

Observe that each machining workingstep has a well-defined start and end point of the tool motion. If two subsequent workingsteps require different tools, a tool change is required and then the generator also calculates the respective tool paths to and from the tool change position.

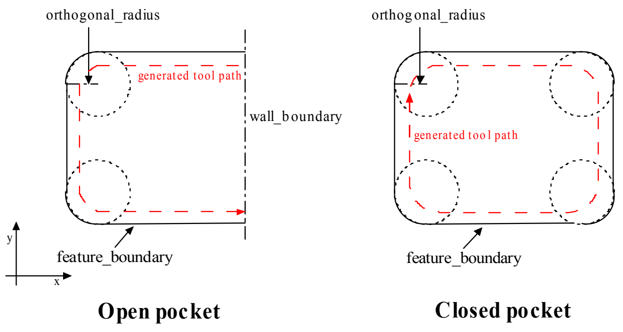


Fig. 7. Parameters of a pocket

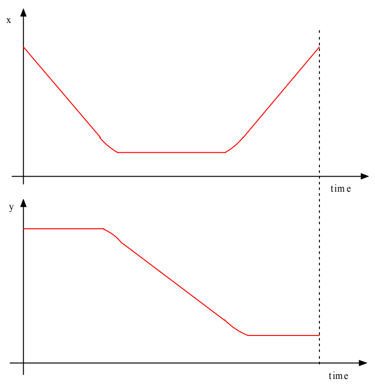


Fig. 8. Trajectory of the tool to realize an open pocket, in the reference system of the pocket

4.4. Low Level Controller

The Low-Level Controller sends suitable signals to the motor drivers and to the tool in order to realize the tracking of the trajectory specified by the Tool Path Generator. In particular, the developed module implements two control strategies:

- an open loop control for the step-motors;
- a control with anticipated reference for the activation of the plasma torch [28].

Details of the realized controller are reported in [26].

4.5. Machining Inspector

This module inspects the machining execution. In particular, the following conditions are considered [2]:

- the machine has to reach the operating conditions specified in the *MACHINING_WORKINGSTEP* entity before the operation of the workingstep commences. If the machine is unable to reach these conditions during the preceding workingstep, a halt must occur before the execution of the workingstep until all parameters are stable;
- it predicts the part geometry change and checks if it matches the geometry change described in the *MACHINING_WORKINGSTEP* entity.

5. Operation software

In this section we describe the operation software that has been developed to implement and parallelize the tasks described in section IV. The aim is to reduce dead times in the machining process execution and, at the same time, allow on-line machining inspection and machining status feedback.

Contrary to other CNC machines that start machining after all the tool path list is made (see for example [10]), the developed intelligent CNC machine simultaneously performs *i)* tool path generation, *ii)* machining and *iii)* machining inspection. This has been obtained utilizing the multitasking feature of the selected microprocessor.

Fig. 9 shows the tasks' execution algorithm, written in a C-like language. Note that the pseudo-command *parallel* indicates parallel execution of the specified tasks. The following four status variables drive the parallel execution of the tasks:

- *workingsteps_to_load*, that indicates if there are still workingsteps to load;
- *workingstep_feasible*, that indicates if the actual workingstep is feasible with the available CNC machine;
- *tool_path_generated*, that indicates if a tool path has been generated and then a machining procedure can start;
- *machining_errors*, that indicates the machining status.

It is easy to recognize that using this algorithm the controller is able to parallelly perform: *i)* low level control, *ii)* machining inspection and *iii)* the sequence of: workingstep interpretation, high level control and tool path generation. This drastically reduces dead times and allows performing both machining and the other implemented high level activities at the same time.

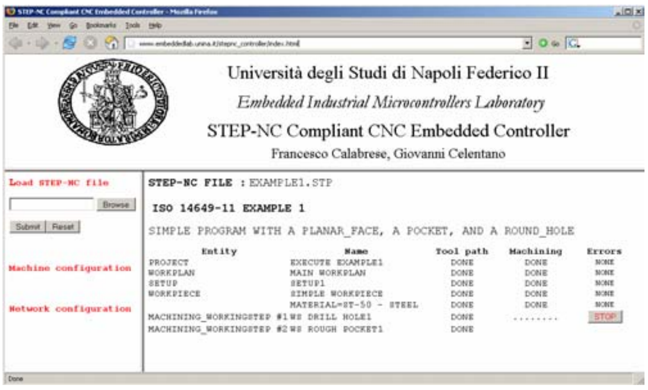
The described algorithm has been implemented and tested on the microprocessor RCM3700, which is now able to perform all the tasks involved in a STEP-NC machining process.

As regards the user interface, it has been realized by means of Macromedia Flash web pages accessible through the embedded http server (see Fig. 10). This allows a user to upload the STEP-NC file, to machine the part and to monitor its execution from any PC connected to the local network.

```
boolean  workingsteps_to_load   = true;
boolean  workingstep_feasible   = false;
boolean  tool_path_generated    = false;
boolean  machining_errors       = false;

InterpreterStartup();   // load PROJECT, WORKPLAN, SETUP
                        //      and WORKPIECE
do {
   parallel {
      if (workingsteps_to_load == true) {
         Interpreter();                 // load WORKINGSTEP
                                        // and update workingsteps_to_load
         HighLevelController();         // update workingstep_feasible
         if (workingstep_feasible == true) {
            ToolPathGenerator();
            tool_path_generated = true;
         }
      }
      if (tool_path_generated == true) {
         parallel {
            LowLevelController();
            MachiningInspector();    // update machining_errors
         }
         tool_path_generated = false;
      }
   }
} while (machining_errors == false AND
      (workingsteps_to_load == true OR tool_path_generated == true) )
```

Fig. 9. Microprocessor tasks' execution algorithm


STEP-NC Compliant CNC Embedded Controller - Mozilla Firefox

www.embeddedlab.unina.it/stepnc_controller/index.html

Università degli Studi di Napoli Federico II

Embedded Industrial Microcontrollers Laboratory

STEP-NC Compliant CNC Embedded Controller

Francesco Calabrese, Giovanni Celentano

Load STEP-NC file

Machine configuration

Network configuration

STEP-NC FILE : EXAMPLE1.STP

ISO 14649-11 EXAMPLE 1

SIMPLE PROGRAM WITH A PLANAR_FACE, A POCKET, AND A ROUND_HOLE

```
Entity                     Name                      Tool path  Machining  Errors
PROJECT                    EXECUTE EXAMPLE1          DONE       DONE       NONE
WORKPLAN                   MAIN WORKPLAN             DONE       DONE       NONE
SETUP                      SETUP1                    DONE       DONE       NONE
WORKPIECE                  SIMPLE WORKPIECE          DONE       DONE       NONE
                           MATERIAL=ST-50 - STEEL    DONE       DONE       NONE
MACHINING_WORKINGSTEP #1WS DRILL HOLE1               DONE       ........   STOP
MACHINING_WORKINGSTEP #2WS ROUGH POCKET1             DONE
```

Fig. 10. User interface for uploading STEP-NC programs and monitoring their execution

6. Machining test

The presented test has been performed with example 1 of ISO14649-11 document (see Fig. 11) [30]. However, the material has been modified to fit the cutting power of the machine. The entities the controller elaborates to machine the part are the following.

1. *PROJECT* entity: it specifies the *WORKPLAN* and the *WORKPIECE* to consider.
2. *WORKPLAN* entity: it specifies the *SETUP* and the *WORKINGSTEPs*.
3. *SETUP* entity: it defines the reference system to be used. The entity indicates that the part origin has coordinates (150.0,90.0)mm and the reference system has usual orientation.
4. *WORKPIECE* entity: it defines the part's characteristics: AISI 304 stainless steel, rectangular shape of dimensions 100.0 x 120.0 mm.
5. *MACHINING_WORKINGSTEP* entity #1 (drill hole 1): it specifies that the feature to be machined is a circular hole. The hole centre is (20.0, 80.0)mm with respect to the reference system specified in the *SETUP* entity. The *diameter* is 22mm.
6. *MACHINING_WORKINGSTEP* entity #2 (rough pocket 1): it specifies that the feature to be machined is a closed pocket. The pocket's first corner has coordinates (45.0, 110.0) mm with respect to the reference system specified in the *SETUP* entity. Moreover, the orientation is rotated 180° clockwise with respect to the one specified in the *SETUP* entity. The *orthogonal_radius* is 10mm while the *feature_boundary* parameter specifies that the pocket's boundary is defined as a polyline which connects the points: (0.0, 0.0)mm; (0.0, 80.0)mm; (50.0, 80.0)mm; (50.0, 0.0)mm; (0.0, 0.0)mm.

Once the STEP-NC file has been uploaded using the web interface (see Fig. 10), the user can start the machining process and on-line monitor its execution. The microprocessor is now in charge of executing the algorithm described in Fig. 9 that generates the machining commands performing the above described workingsteps. Thanks to the developed High Level Controller, the developed CNC machine automatically changes machining data and generates the tool paths according to the available material (stainless steel) and the machining tool (plasma cutter). So, it has been possible to machine the part even if the workingsteps were designed to be executed on a milling machine (with some limitations regarding the machined features). Fig. 12 shows the machining result.

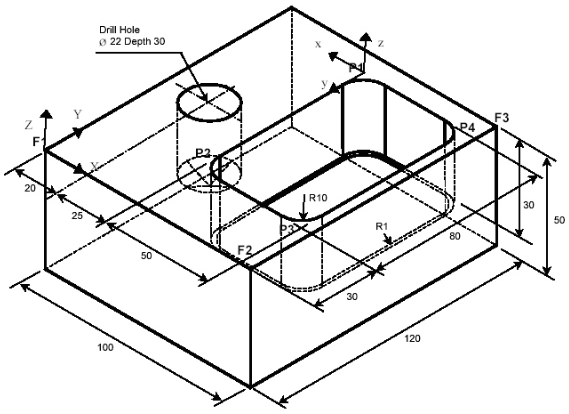


Fig. 11. Machining example 1 of ISO14649-11 [30]

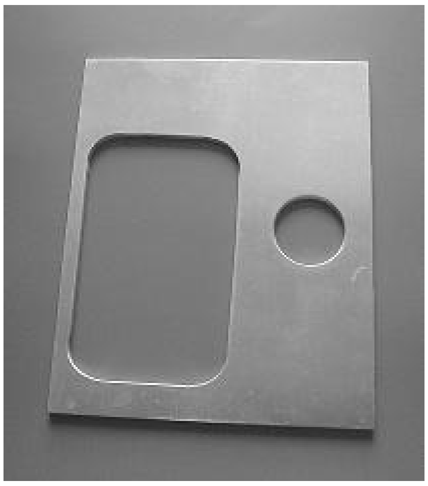


Fig. 12. Result of the machining test using example 1 of ISO14649-11

7. Conclusion

In this paper the design and realization of a microcontroller-based embedded system able to control a CNC machine by means of STEP-NC have been presented. Detailed descriptions of the software modules together with technical details about their implementations have been given.

The realized embedded system was tested with the available CNC machine and the result of the machining using example 1 of the ISO14649-11 has been shown. By means of this test it has been possible to show that such low-cost microcontroller is able to manage all the activities involved in a STEP-NC machining process without using expensive industrial PCs, in opposition with previously developed prototypes.

Future works will address the development of further high-level functionalities on the microcontroller. Moreover, we aim at integrating the controller in a SFP system to create an automatic manufacturing chain.

References

[1] International Standards Organization, *ISO 6983/1. Numerical control of machines—program format and definition of address words—part 1: data format for positioning, line and contouring control systems*, 1st ed. 1982.

[2] International Standards Organization, *ISO 14649-10. Data model for computerized numerical controllers: part 10—general process data*. 2003.

[3] International Standards Organization, *ISO 14649-1. Data model for computerized numerical controllers: part 1—overview and fundamental principles*. 2003.

[4] X.W. Xu and S.T. Newman, "Making CNC machine tools more open, interoperable and intelligent—a review of the technologies," *Computers in Industry*, vol. 57, n. 2, pp. 141-152, February 2006.

[5] http://www.steptools.com, accessed on: 30/07/2004.

[6] Manufacturing Data Systems Inc., *OpenCNC Brochure* www.mdsi2.com/Solutions/CNC_Controls/Brochure/OpenCNCbrochure.pdf, available on: 30/03/2004.

[7] M. Weck, "STEP-NC – A new interface closing the gap between planning and shop-floor", WZL RWTH Aachen, http://www.step-nc.org/, STEP-NC Workshop, Aachen, Germany, February 2003.

[8] X.W. Xu, "Realization of STEP-NC enabled machining," *Robotics and Computer Integrated Manufacturing*, vol. 22, n. 2, pp. 144-153, April 2006.

[9] W. Lee and Y.B. Bang, "Design and implementation of an ISO14649-compliant CNC milling machine," *International Journal of Production Research*, vol. 41, n. 13, pp. 3007–3017, 2003.

[10] S.H. Suh, J.H. Cho, H.D. Hong, "On the architecture of intelligent STEP compliant CNC," *Computer Integrated Manufacturing*, vol.15, n.2, pp. 350-362, 2003.

[11] International Standards Organization, *ISO 10303-1. Industrial automation systems and integration—product data representation and exchange—part 1: overview and fundamental principles*. 1994.

[12] International Standards Organization, *ISO 10303-203. Industrial automation systems and integration—product data and exchange—part 203: application protocol: configuration controlled 3D designs of mechanical parts and assemblies*. 1994.

[13] International Standards Organization, *ISO 10303-21. Industrial automation systems and integration—product data representation and exchange—part 21: implementation methods: clear text encoding of the exchange structure*. 2002.

[14] STEP-NC Newsletters. Issue 3, Nov. 2000.

[15] International Standards Organization, *ISO/DIS 10303-238. Industrial automation systems and integration—product data representation and exchange—part 238: application protocols: application interpreted model for computerized numerical controllers*. 2003.

[16] J. Wolf, "*Requirements in NC machining and use cases for STEP-NC, Analysis of ISO 14649 (ARM) and AP 238 (AIM)*," White Paper, ISO T24 STEP-Manufacturing Meeting, San Diego, USA, March 2003.

[17] A.B. Feeney, T. Kramer, F. Proctor, M. Hardwick, D. Loffredo, "*STEP-NC implementation—ARM or AIM?*," White Paper, ISO T2 STEP-Manufacturing Meeting, San Diego, USA, March 2003.

[18] A.C. Lin, S.-Y. Lin, and S.-B. Cheng, "Extraction of manufacturing feature from a feature-based design model," *International Journal of Production Research*, vol. 35, pp. 3249–3288, 1997.

[19] F. Calabrese, *Design and realization of a robotic system for industrial manufacturing*, Master thesis, Embedded Industrial Microcontrollers Laboratory, University of Naples Federico II, Italy, 2004.

[20] RS Components, *Hybrid stepper motors*, datasheet, 2004.

[21] ST Microelectronics, *L297 Stepper motor controllers*, datasheet, 2004.

[22] ST Microelectronics, *L297 Stepper motor controller*, application note, 2004.

[23] ST Microelectronics, *L298 Dual full-bridge driver*, application note, 2004.

[24] Rabbit Semiconductor, *RCM3700 RabbitCore Data Sheet*, 2006.

[25] Rabbit Semiconductor, *RCM3700 RabbitCore User's Manual*, 2006.

[26] F. Calabrese, G. Celentano, *Realization of an embedded microcontroller system for a 2-dof robot*, Internal report,

Embedded Industrial Microcontrollers Laboratory, University of Naples Federico II, Italy, 2005.

[27] F. Calabrese, G. Celentano, “Optimal Design of Robust Control Systems with Large Band Reference Signals,” *Eurocon 2005 - The International Conference on "Computer as a tool"*, Belgrade, Serbia & Montenegro, 21-24 November 2005.

[28] F. Calabrese, G. Celentano, “*Optimal Design of Filters for the Delayed Estimation and Controllers with Anticipated Reference*,” IEEE Transactions on Automatic Control, submitted for publication, 2006.

[29] G. Ambrosino, G. Celentano and F. Garofalo, “Microprocessor implementation of a robust control law for industrial robots,” *International Journal of Modelling and Simulation*, vol. 4, no. 4, pp. 141-144, 1984.

[30] International Standards Organization, *ISO 14649-11. Data model for computerized numerical controllers: part 11—process data for milling*. 2003.

[31] S.H. Suh, B.E. Lee, D.H. Chung, S.U. Cheon, “Architecture and implementation of a shop-floor programming system for STEP-compliant CNC,” *Computer-Aided Design*, n. 35, pp. 1069–1083, 2003.

[32] S.H. Suh, D.H. Chung, B.E. Lee, J.H. Cho, S.U. Cheon, H.D. Hong and H.S. Lee, “Developing and integrated STEP-Compliant CNC prototype,” *Journal of Manufacturing Systems*, vol. 31, n. 5, 2002.