REFURBISHMENT OF USED MACHINE ELEMENTS FOR MAKING 2½ AXES MILLING MACHINE PROTOTYPE

Bagus Arthaya[1], Ali Sadiyoko[2],
Oke Setiawan[3], Filemon N.T.[4], Sebastian T.[5]

[1,2]Industrial Engineering Dept., Parahyangan Catholic University
[3,4,5]Alumni of Industrial Engineering Dept., Parahyangan Catholic University
[1]bagusmooi@gmail.com, [2]alfa51@unpar.ac.id

ABSTRACT

CNC machines are operated based on the program inputted to the machine control unit. This program contains a set of commands that controls the movement of the mesin table known as NC program. Developing a machine tool prototype very useful in order to gain alot of knowledge about designing, computer programming, machine tool controller and so on. This research aims at developing a prototye of CNC 2½ axes milling machine that can be used to execute error free NC programs. It is designed to execute basic movements such as rapid traverse, linear or circular interpolations, etc. For a 2½ axes milling machine, cutting a workpiece is done layer by layer. This machine uses two stepping motors and two sets of linear guide to move the table horizontally and the last ½ axis utilizes the third stepping motor. This machine is controlled using two set of microcontroller and three motor driver units. An interface is designed to enable communication between the computer and the machine control unit. The machine motions were tested and the experimental results show the movement capability of this machine. A graphical user interface was also developed to easy the machine operator in controlling and monitoring motions of the machine.

***Keywords**: CNC Machine, NC program, machine prototype, axis controller, microcontroller.*

1. INTRODUCTION

In the future factory, modern manufacturing processes will involve a lot of CNC machines and other smart machines that are very clever to produce any kind of technical parts. To run a CNC machine, operator has to prepare an NC (Numerical Control) program for a typical machine tool. Quality of the machined product depends a lot on the selection of cutting parameters and the way the product processed.

As preparing an NC program is monotonous and time consuming tasks, our production laboratory team has developed a software tool that is capable of checking, validating and simulating NC programs for milling and turning operations (Arthaya et al., 2007 and 2010). This tool helps to avoid incorrect programming that leads to wasting raw material, shortening of cutter life time, wasting production time and so on. It also can check the syntax of the program, show the path of cutter during the machining operation, the change of original product shape into finished machined product. After ensuring that all logical and syntax errors of the NC program are fixed and the final shape is correct, then the program is ready to be fed to an NC machine.

Products typically made in a machine tool can be of any form. Figure 1 depicts some product forms, e.g. (a) is mainly processed in a lathe and (b, c) are processed in a milling machine. Products in Figure 1 (b) should be processed at least by 2½ axes vertical milling machine while in (c) can be processed in vertical as well as in horizontal milling machine.

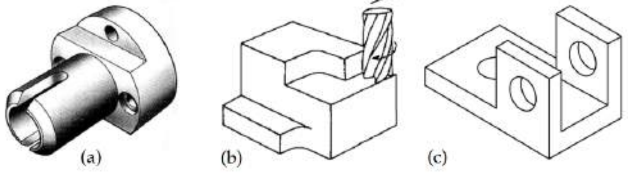


Figure 1. Product shapes typically resulted from milling machine operations.

CNC machine is not a cheap device and traditional industries in Indonesia mostly cannot afford them. As they can not by any build up CNC machine, then this program leads to an encouragement in speeding up

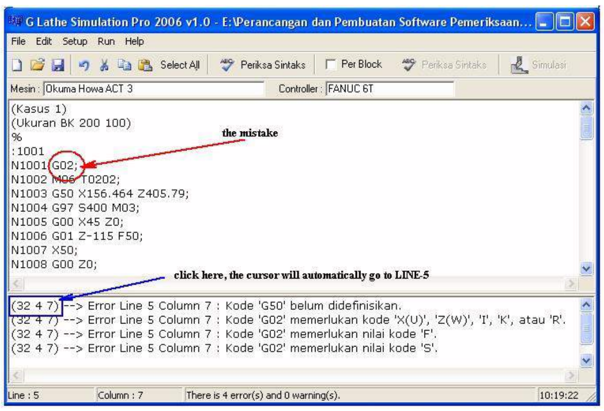
the ability to develop our own machine tools industry.
Starting with the success of making an NC program checker and simulator, small and medium industries can take advantages of it. It just starts with first checking and simulating the NC program carefully before sending it to a CNC machine. At this stage, one can check out their operations on CNC machines without having to hire expensive operators and can prepare the NC program by themselve.
This research starts with preparing the machine components which were collected from used machine elements such as lead screw, linear guides, stepping motors and so on. A pair of linear drive assembly has been built to realize 2-D (2-axes) motion. The last ½ axis is taken care by a stepping motor to see the synchronization between 2-D motion and vertical motion. An interface is then developed to convert the NC program that contains G-Codes into machine codes to be understood by the machine control unit.

2. CNC MILLING OPERATION

Similar to manual machining, in CNC machining an operator has to define the movements of cutting tools over the part to be machined. Especially in milling operations, cutting tool always moves relative to the work piece which is firmly clamped on the machine table. These relative motions consist of x, y and z motions. The two dimensional x-y motions are performed on an x-y table that moves linearly. The final motion of the cutting tool can be a linear, a circular motion or any combination of them.
In this work the motion along last axis is performed incrementally relative to the motion of the table. This means that after every complete motion on x-y plane, a certain motion in z-direction is carried out. That is why this kind of operation is called 2½ axes motion.

2.1. NC Program Preparation

Before running a machining operation, operator needs to prepare the NC program. This program is also called G-Code program where a certain rules apply in preparing it (Groover, 2001). After mastering the basic knowledge about machining, operator can go to the G-Code checker and simulator software as depicted in Figure 2.


G Lathe Simulation Pro 2006 v1.0 - E:\Perancangan dan Pembuatan Software Pemeriksaan...
File Edit Setup Run Help
Select All
Periksa Sintaks
Per Block
Periksa Sintaks
Simulasi
the mistake
click here, the cursor will automatically go to LINE-5

Mesin : Okuma Howa ACT 3   Controller : FANUC 6T

```
(Kasus 1)
(Ukuran BK 200 100)
%
:1001
N1001 G02;
N1002 M06 T0202;
N1003 G50 X156.464 Z405.79;
N1004 G97 S400 M03;
N1005 G00 X45 Z0;
N1006 G01 Z-115 F50;
N1007 X50;
N1008 G00 Z0;
```

```
(32 4 7) --> Error Line 5 Column 7 : Kode 'G50' belum didefinisikan.
(32 4 7) --> Error Line 5 Column 7 : Kode 'G02' memerlukan kode 'X(U)', 'Z(W)', 'I', 'K', atau 'R'.
(32 4 7) --> Error Line 5 Column 7 : Kode 'G02' memerlukan nilai kode 'F'.
(32 4 7) --> Error Line 5 Column 7 : Kode 'G02' memerlukan nilai kode 'S'.
```

Line : 5   Column : 7   There is 4 error(s) and 0 warning(s).   10:19:22

Figure 2. The software tool used to check and simulate milling operation (Arthaya et al., 2007).

This tool helps the user to find any possible mistakes in G-Code program and suggest the best solution. This software version is still limited to one type of CNC controller i.e. Fanuc oi – MC controller. After successfully making an error free program, user can click the “Simulasi” icon and the cutting simulation can be carried out in 3-Isometric views. If user is already sure about the cutting operation, this program can later be fed into the machine controller.

2.2. Milling Process Parameters

In preparing G-Code program, the operator also has to define some parameters related to machining operation such as depth of cut, feed and cutting speed or the spindle rotation. These parameters should be defined separately and determining the correct magnitude is the responsibility of the programmer. As this machine has limited working envelop, user has also to define the dimension of work piece and its position relative the machine zero point, and the dimension of the cutting tool as well (DeGarmo, 2001).
Eventually all this information has to be sent to the milling machine controller. The controller will translate information related to movement of cutting tool relative to the work piece in term of position information as well as the velocity. These movements should be realized by the machine by controlling the motor drivers in an accurate and suitable ways. In this preliminary research, only the

feed and the maximum axis speed are considered in the design.

3. MILLING MACHINE WITH 2½ AXES

To perform cutting operation on a rectangular product having 3-D contour feature one has to use 3-axes milling machine. In some condition, for a simpler shape, 2½ axes milling still can be employed. An example of cutting 3-D feature can be done using this machine as shown in Figure 3. A volume is divided into several layers and each layer is cut using end-mill cutting tool. The tool has to finish cutting the first layer completely and continued to the next layers until total volume is completely removed.

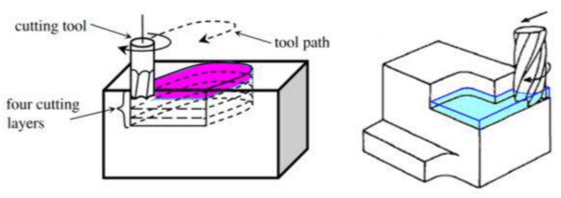


Figure 3. Middle pocket and step are being cut by introducing layer-by-layer cutting operation.

As the consequence, 2-axes milling machine with intermittent movement perpendicular to the machined surface can be used to carry out this type of machining. A prototype of this machine has been developed which enables x-y movement of machine table and intermittent vertical movement.

3.1. Milling Process Parameters

In order to realize x-y movement of machine table, there are three frame construction alternatives were proposed as depicted in Figure 4 (a, b, c) (Filemon et.al, 2009). In Figure 4a, the table moves fort and back driven by the first motor which is mounted firmly on the base. Motions in the other two perpendicular directions are driven by other two motors. The base dimension becomes twice the other alternatives. As the consequence, this structure needs the largest space for the table movement. The second alternative lets the table move freely in x-y direction which is driven by two motors as shown in Figure 4b. This structure gives the smallest dimension of the base. In Figure 4c, machine table is fixed at the base frame, while 2 motors move the axes fort-back and left-right. The tool is mounted on the top structure and is able to move up and down.

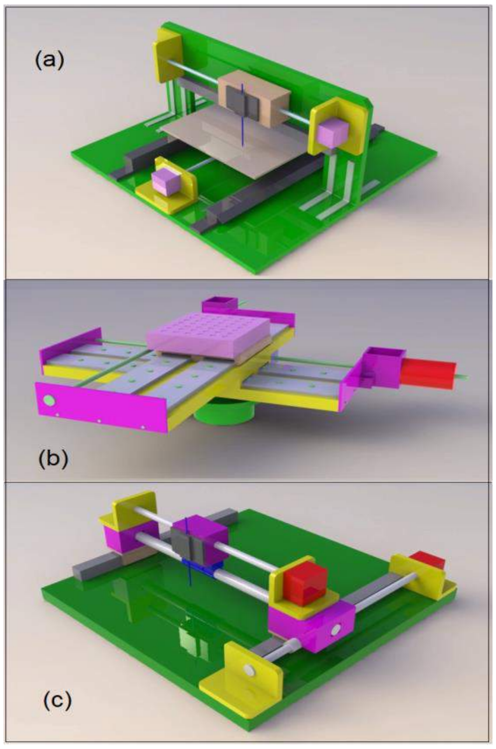


Figure 4. Three alternatives design with fixed position machine table.

After choosing one alternative (alternative 4-b was chosen), we end up with the following machine specification as listed in Table 1. This construction is mainly assembled from some used machine elements such as: linear guide, lead screw and nut, slider blocks and stepper motor. Two pairs of linear motion assembly were prepared apart and one assembly is mounted on the other.
Putting all the main parts together, the basic construction of this milling machine prototype is presented in Figure 5. This frame still can be assembled firmly so it moves securely along each axis. On top of the slider block and the lead screw nut, machine table is mounted. It gives a working area of about 230mm x 190mm on x-y plane. This frame is supported by a heavy cylindrical and square metal base that provides a good stability.

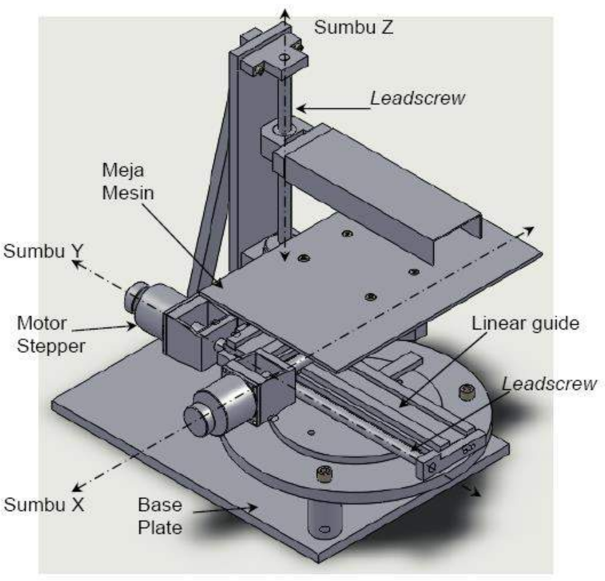


Figure 5. Basic construction of the machine prototype.

3.2. Machine Controller and Driver Units

To control the cutting tool motion over the work piece, two units of microcontroller were used. Each of it is responsible for controlling the command to every motor driver. These microcontrollers receive commands from the control computer which first interpret the G-Code. Codes are standardized symbols that corresponds the type of cutting motion.

Types of code explored here are strait line movement (G00) and linier interpolation (G01). These two codes will move the cutting tool from one point to another but in different modes. G00 forces each axis moving at their maximum speed therefore the two axes may finish the motion at different time. Meanwhile G01 forces the axes to move in a synchronized manner so that they will stop exactly at the same time with a predetermined speed.

Schematically the communication between control computer, machine controller and the motor drivers are depicted in Figure 6 (Sebastian et.al, 2009). Computer sends the command through serial communication (RS232) line to the first and the second microcontroller at once. Based on the encoded symbol, each of microcontrollers will understand which one is sent for itself. The driver units take care of realizing the motor rotations based on code received by the controller.

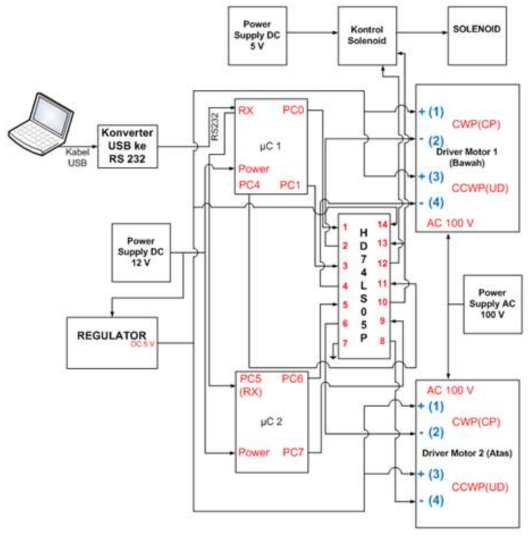


Figure 6. Communication line between computer control, machine controller and driver units.

Each figure should have a caption below the figure. The caption of a table should appear at the top of the table. The words in each caption should be written in the lower case except the first character of the first word. All figures should be positioned at the top of the page where possible. All figures should be numbered consecutively and captioned.

To make the microcontroller understand the command from the computer, some settings must be done both for the computer and the microcontroller. As the communication is done in serial mode, then in Visual Basic 6.0 programming language, "MSComm" word has to define to allow this communication as listed in Table 2 (Razaq, 2004).

The following is an example of command for MSComm in Visual Basic 6.0:

- MSComm1.CommPort = 1; it uses port no.#1 used as comunication port,
- MSComm1.Settings = "9600,N,8,1" ; it sets the baud rate of 9600 bps (bits per second), with No parity, uses data of 8 bits, and 1 stop bit,
- MSComm1.PortOpen = True; it functions to control the communication port at the computer to be opened.

On the other hand, setup should also be done to the microcontroller. To do it, programming language BASCOM-AVR is used. This language is a BASIC based compiler and the setting is shown in Table 3.

The following is an example of command for BASCOM-AVR Compiler:

- $regfile = "m8535.dat" ; it shows that the microcontroller used is ATMega 8535,
- $crystal = 4000000 ; it tells that the crystal frequency used is 4 MHz (which normally used for baud rate 9600 bps),

$baud = 9600; it shows that the used baud rate is 9600 bps and it should match the setup at the computer side.

3.3. G-Code Extraction

G-Code contains complete information of the motion of cutting tool relative the work piece. This information is not directly executable for the microcontroller. To transfer this information to the microcontroller, a G-Code extractor software tool has been developed. It will extract data related to axis motion such as G00 and G01, nominal number of movement in X and Y axis, and the feed as well.

Example of NC program containing G-code is as the followings:

```
(any comments)
2000
N05  G92  X-1.000  Y1.000 Z1.000
N10  G20  G90
N15  M06  T01
N20  S2000  M03
N25  G00  X0  Y0  Z.100
N30  X.375  Y-.375
N35  G01  Z-.125  F10
N40  M05
N45  M30
%
```

The algorithm to extract G-Code at row-x to become a string command (“commandString”) is explained as follows:

1. Declaring the value of dx, dy, and Vf = 0.
2. Take parameter values from G-Code at raw-x.
3. Take character at the 7th and the 9th position from G-Code words to become characterCommand variable.
4. Define x = character position G-Code word, and prnth[x] = the xth character of G-Code words. Set x = 11.
5. Clean up variable temp, set temp = nil.
6. If prnth[x] = ‘X’ then append prnth[x] into variable temp, set x = x + 1, and repeat until prnth[x] = ‘;’ (semicolon) or ‘ ’ (space). Afterward save the content of variable temp into variable dx;
   ElseIf prnth[x] = ‘Y’ then append prnth[x] into variable temp, set x = x + 1, and repeat until prnth[x] = ‘;’ (semicolon) or ‘ ’ (space). Afterward save the content of variable temp into variable dy;
   ElseIf prnth[x] = ‘F’ then append prnth[x] into variable temp, set x = x + 1, and repeat until prnth[x] = ‘;’ (semicolon) or ‘ ’ (space). Afterward save the content of variable temp into variable Vf.
7. set x = x + 1.
8. In value of x has not exceeded the number of character within G-Code words, then go to step 5, else go to step 9.
9. If characterCommand = G00, then Vf = max.speed of stepper motor. If characterCommand <> G00, then call “Conversion Algorithm” to determine commandString based on the value of dx, dy, and Vf.

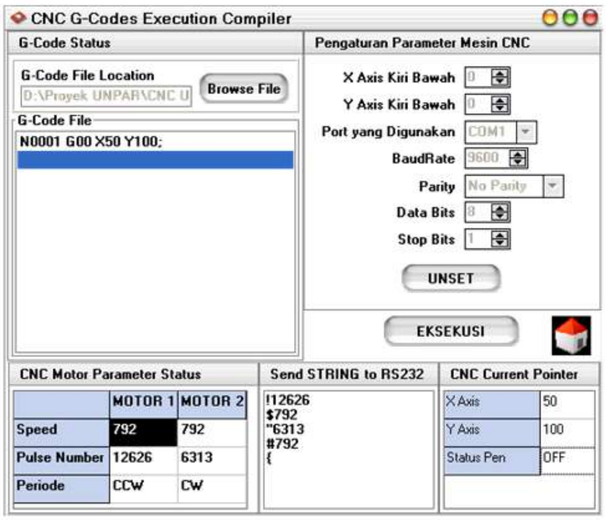
The next algorithm is converting the values of dx, dy, and Vf to become commandString that understood by the microcontroller. This algorithm calculate the suitable feed (Vf) which should be correlated with the distance to be traveled (dx and dy). The algorithm works as the following:

1. Take variables dx, dy, and Vf which are sent from the computer in the form of parameters.
2. If value of dx < 0, than rotationMotor2 = $24. If dx ≥ 0, then rotationMotor2 = $23.
3. Set noStep2 = dx/0.00792
4. If value of dy < 0, than rotationMotor1 = $23. If dy ≥ 0, then rotationMotor1 = $24.
5. Set noStep1 = dy/0.00792
6. If characterCommand = G00, then go to step 7. If characterCommand = G01, then go to step 8.
7. Calculate Period1 = 6 x 10^7/Vf. Calculate Period2 = 6 x 10^7/Vf. Set Pen = $7B. Continue to step 9.
8. Calculate α = $\tan^{-1}$(dy/dx). Calculate Period1 = 6 x 10^7/ (Vf x sin(α)). Calculate Period2 = 6 x 10^7/(Vf x cos(α)). Set Pen = $7D.
9. If Period1 and Period2 > 600, then go to step 10. If Period1 and Period2 ≤ 600, then “declare Extraction failed” and Stop algorithm.
10. If noStep1 > 0, then statMotor1 = $21. If nolStep1 ≤ 0, then statMotor1 = $25.
11. If noStep2 > 0, then statMotor2 = $22. If noStep2 ≤ 0, then statMotor2 = $26.
12. commandString = CHR(statMotor1) + IntToStr(noStep1) + CHR(13) + CHR(10) + CHR(rotationMotor1) + IntToStr(Period1) + CHR(13) + CHR(10) + CHR(statMotor2) + IntToStr(noStep2) + CHR(13) + CHR(10) + CHR(rotationMotor2) + IntToStr(Period2)

+ CHR(13) + CHR(10) + CHR(Pen) + CHR(13) + CHR(10).

In Figure 9, an example of simple G-Code is retrieved from a directory call "D:\Proyek Unpar\CNC Untuk Test", and being loaded to this program. The G-Code is very simple command i.e.: "N0001 G00 X50 Y100". This code tells the machine to move rapidly in x and y directions until its position increased 50 and 100 length units to each axis respectively.

Information about the movement has been converted and sent as shown in "Send STRING" window in Figure 7. This string says motor-Y on (=!) to position 50 (=12626) while motor-X on (=") to position 100 (=6313) and signal for pen is off (={).


CNC G-Codes Execution Compiler

G-Code Status

G-Code File Location: D:\Proyek UNPAR\CNC U

Browse File

G-Code File:
```
N0001 G00 X50 Y100;
```

Pengaturan Parameter Mesin CNC

X Axis Kiri Bawah	0
Y Axis Kiri Bawah	0
Port yang Digunakan	COM1
BaudRate	9600
Parity	No Parity
Data Bits	8
Stop Bits	1

UNSET

EKSEKUSI

CNC Motor Parameter Status

	MOTOR 1	MOTOR 2
Speed	792	792
Pulse Number	12626	6313
Periode	CCW	CW

Send STRING to RS232
```
!12626
$792
"6313
#792
{
```

CNC Current Pointer

X Axis	50
Y Axis	100
Status Pen	OFF

Figure 7. Software interface for extracting motion information from a C program.

4. CASE STUDY ON MILLING MACHINE

The machine prototype is first assembled so that all main components are functioning properly. In the early stage, a pen driven by solenoid is attached above machine table to simulate the Z-axis and it is controlled on and off. It can draw line when it is on and the machine appearance is shown in Figure 8.

Some simple motion tasks were carried out to validate the performance of this machine prototype. The first test was calibrating each of the stepper motor to move in their respective directions and the results are shown in Figure 9. Afterward, some other interpolating motions were also carried out as depicted in Figure 10. G-Code G00 and G01 show different paths of cutting tools as the first takes the maximum speed of both motors while the second one has to take into account the speed Vf. The other motions are comparisons of some different cutting speed (Vf).

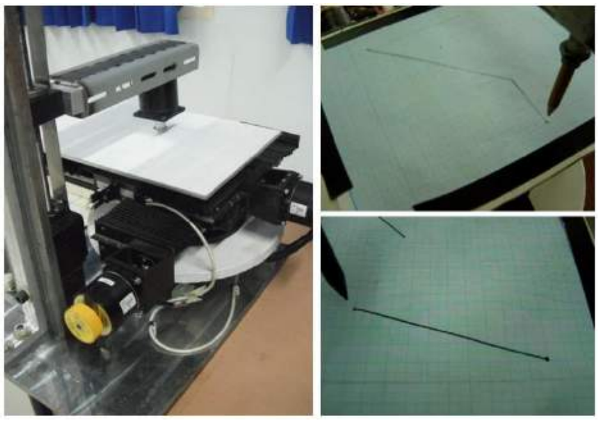


Figure 8. Complete view of the machine prototype.

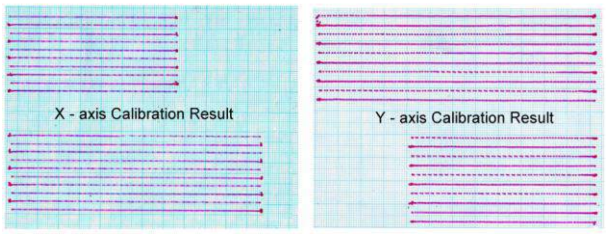


Figure 9. X- and Y-axes calibration results.

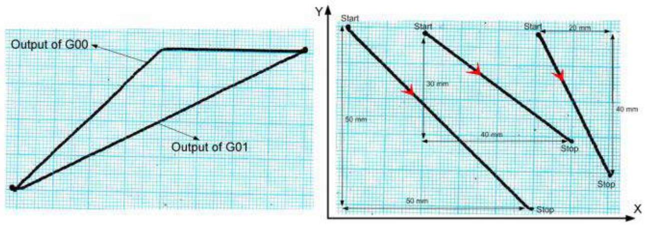
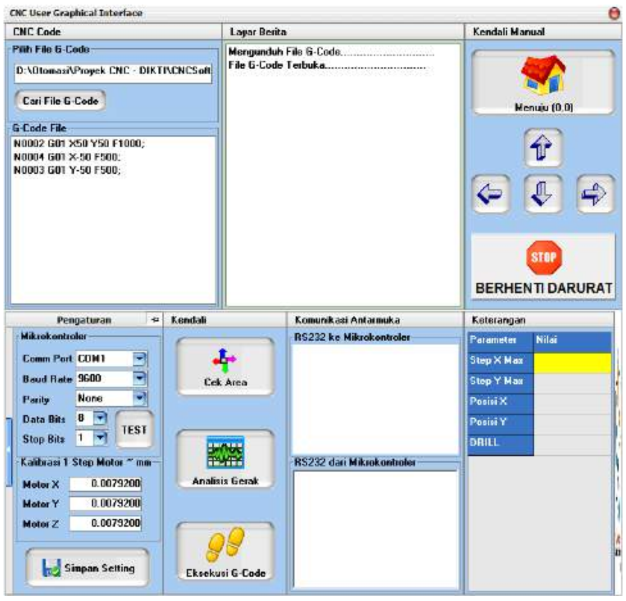


Figure 10. Cutting tool path of different G-Code.

As complexity of NC-program increases, the interface is then improved by adding some features, e.g. for control action, checking area of the machine, checking the type of motions, and so on. This help the operator to easily monitor and anticipate any emergency situation on the machine as depicted in Figure 11. Another improvement made to this early design is modification of the communication scheme between control computer and the micro-controller. The new scheme uses only one microcontroller to extract the NC-program, classifying the type of motion to be executed and I/O control to some external devices and so on as shown in Figure 12.



Pilih File G-Code: D:\Otomasi\Proyek CNC - DIKTI\CNCSoft

G-Code File:

```
N0002 G01 X50 Y50 F1000;
N0004 G01 X-50 F500;
N0003 G01 Y-50 F500;
```

```
Mengunduh File G-Code...............................
File G-Code Terbuka...............................
```

Mikrokontroler

Comm Port	COM1
Baud Rate	9600
Parity	None
Data Bits	8
Stop Bits	1

Kalibrasi 1 Step Motor ~ mm

Motor X	0.0079200
Motor Y	0.0079200
Motor Z	0.0079200

Parameter	Nilai
Step X Max	🟨
Step Y Max	
Posisi X	
Posisi Y	
DRILL	

Figure 11. Improved User Interface.

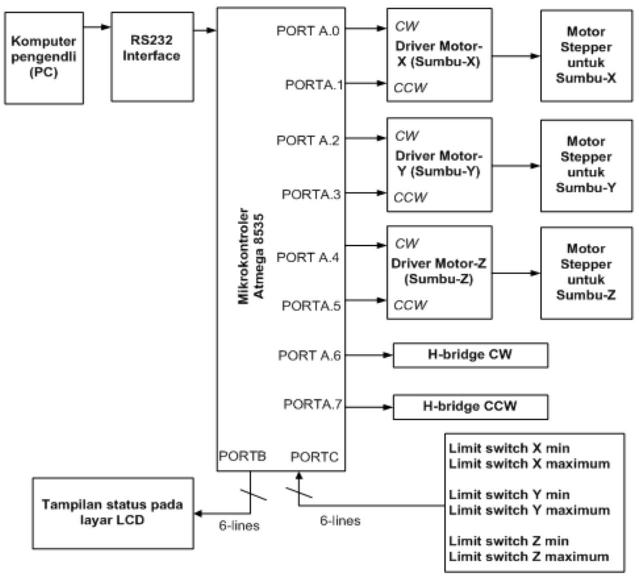


Figure 12. New communication scheme between PC and micro-controller to control the machine prototype.

5. CONCLUSION

From the design and experimental works done in this project, some conclusions can be drawn as follows:

- Simple but accurate enough machine construction can be assembled form used machine elements,
- Simple motions reflecting G-Code command are shown in the form of cutting tool paths that show linear interpolations at different direction,
- Knowledge about constructing 2 axes mechanism is gained during the design and assembling the prototype,
- Microcontrollers are successfully implemented in converting G-Code commands and controlling the motion of the machine.
- Interface program has been developed to able the communication between user and the machine, although some improvements still have to be done.

6. REFERENCES

(a) Arthaya B., Setiawan A., Setiawan A. (2007), The Design and Development of G-Code Checker and Simulation of ISO G-Code for CNC Milling, *Presented on The 1st Asia-Pacific Conference on Manufacturing System, Bali, November 5-6, 2007.*

(b) *Arthaya B.,* Setiawan A., and Sunardi S. (2010), The Design and Development of G-Code Checker and Cutting Simulator for CNC Turning Operation, *printed in Journal of Mechanical Engineering Research*.

(c) DeGarmo, E. P., Black, J. T., and Kohser R. (1988), *Materials and Processes in Manufacturing*. 7th ed. Macmillan Publishing Company. New York.

(d) Filemon N. T., Arthaya B., Setiawan A., (2009), Pembuatan Perangkat Lunak Pengendali Untuk Aplikasi Erakan-Gerakan Dasar G Code Pada Model Mesin Cnc Milling 2,5 Axis, Laporan Internal, Jurusan Teknik Industri, Unpar.

(e) Groover, M.P., (2001), *Automation, Production Systems, and Computer-Integrated Manufacturing*, Prentice-Hall. Inc., New Jersey.

(f) Histand, M.B., Alciatore, D.G., (1999*), Introduction to Mechatronics and Measurement Systems*, McGraw-Hill, Int'l Edition, Singapore.

(g) Nanfara F., Uccello, T., Murphy, D. (2002) *The CNC Workshop: A multimedia introduction to CNC*. SDC Publucation. Kansas

(h) Petruzella, F.D., (1996), *Industrial Electronics*, Glencoe McGraw-Hill, NY.

(i) Razaq, A., (2004). *Quick Learning of Visal Basic 6.0* (in Indonesian), Indah Publisher, Surabaya.

(j) Sebastian T., Arthaya B., Setiawan A., (2009), Pembuatan Model Mesin *Milling*

Cnc 2,5 Axis, Laporan Internal, Jurusan Teknik Industri, Unpar.
(k) Stenerson, J., Curran, K. (1997). *Computer Numerical Control*. Prentice Hall Inc. New Jersey

7. ACKNOWLEDGEMENT

This research was funded by Direktorat Jenderal Pendidikan Tinggi, Kementerian Pendidikan Nasional, based on the Surat Perjanjian Pelaksanaan Kegiatan Penelitian Hibah Bersaing 2010, Nomor: III/LPPM/2010-05/028-P, dated: 24 Mei 2010

AUTHOR BIOGRAPHIES

Bagus Arthaya is a lecturer at Department of Industrial Engineering, Faculty of Industrial Technology, Parahyangan Catholic University, Bandung. He received his doctoral degree in Robotics from Katholieke Universiteit Leuven in 1995. His research interests are in the area of Robotic and Automation, Image Processing, Product Design and CAD/CAM. He is a member of PEI and JSME Indonesia Chapter, APIEMS board. His email address is <bagusmooi@gmail.com>

Ali Sadiyoko is a lecturer at Department of Industrial Engineering, Faculty of Industrial Technology, Parahyangan Catholic University, Bandung He right now is pursuing his doctorate program at Institut Teknologi Bandung in the area about Crowded Robotic His email address is <alfa51@unpar.ac.id >

Oke Setiawan, Filemen N.T and Sebastian T. are alumnis of Department of Industrial Engineering, Faculty of Industrial Technology, Parahyangan Catholic University, Bandung.

Table 1. Detail of technical specification of the 2½ axes milling machine prototype

No.	Specifications	Comments
1	Max. movement of X-axis	190 mm from the origin point of X-axis (on = &H21 or !)
2	Max. movement of Y-axis	230 mm from the origin point of Y-axis (on = &H22 or ")
3	X and Y - movers	Two units of stepping motor (accuracy of 0.72^{O}/step, 500 pulses/rot, min. period of 600 µs, leadscrew 0.00792 mm/step)
4	Driver	Two units of ST12 type driver, one for each of driver motor
5	Cutting tool motion driver	Using stepper motor or solenoid (on = &H7D or } and off = &H7B or {)
6	Clockwise rotation at X-axis motor	Resulting linear motion in positive X-axis (&H23 = #)
7	Counterclockwise rotation at X-axis motor	Resulting linear motion in negative X-axis(&H24 = $)
8	CW rotation at Y-axis motor	Resulting linear motion in negative Y-axis (&H23 = #)
9	CCW rotation at Y-axis motor	Resulting linear motion in positive Y-axis (&H24 = $)
10	Controller	Microcontroller ATMEGA 8535 2 units, each for X- and Y-axes
11	Microcontroller programming	AVR 910
12	Data communication	Serial RS232
13	Control mode	Open loop

Table 2: Command for MSComm in VB

Setting	Comments
CommPort	To control communication port number
Settings	To control the *baud rate*, *parity*, *data bits*, and *stop bits*
PortOpen	To control communication port status
Input	Command to receive data
Output	Command to send data

Table 3: Setup for BASCOM-AVR

Setting	Comments
$regfile	Direction to identify what kind of micro controller is being used
$crystal	Direction to identify what crystal frequency of micro controller is being used
$baud	To match the baud rate used with the one used by the control computer