

RTAI in CAN bus based Network Control of CNC for plasma cutting

Fodrek Peter

Foltin Martin

Department of Information and Communication systems,
Institute of Automatic Control and Industrial Informatics,
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology

Ilkovicova 3, 812 19 Bratislava, Slovak republic
{peter.fodrek ; martin.foltin}@stuba.sk

Blaho Michal

Farkas Ludovit

Poliacik Michal

Department of systems and Signals,
Institute of Automatic Control and Industrial Informatics,
Faculty of Electrical Engineering and Information Technology
Slovak University of Technology

Ilkovicova 3, 812 19 Bratislava, Slovak republic
{michal.blaho ; ludovit.farkas ; michlal.poliacik}@stuba.sk

Murgas Tomas

Systemy priemyselnej informatiky, s.r.o.
Kopcianska 14, 851 01 Bratislava, Slovak republic
tomas.murgas@syprin.sk

Chren Vladimir

Prva zvaracka, a.s.
Kopcianska 14, 851 01 Bratislava, Slovak republic
vladimir.chren@pzvar.sk

Abstract

Paper deals of RTAI usage in cutting machine of metal materials with plasma beam controlled via CANOpen protocol on CAN BUS interface. System uses JAVA/JNI for GUI design and communication with kernel modules dealing with RTAI hard realtime capabilities. The machine prototype is well running at Prva Zvaracka, a.s. Paper describes main parts of the system from control and software point of view.

1 Introduction

We have been developed plasma cutting machine control system for a commercial engineering subject. Paper describes main principles of this project especially of the software design. There is main problem to break engineers to permanently change specifications which degrades original lasagna-ravioli code design in direction to produce only spaghetti code. System contains mechanical portal with one motor on the X-axis but two pinion-rack mechanism with torsion shaft. Y- and Z-axis have also one motor each. Movement of each axis is provided by Permanent magnet synchronous motor with corresponding drive with CAN-Open interface, There are also binary and analogue input/output modules with CAN interface for technology. Another parts are two CAN-Open operator panel with 3 line display and industrial computer with additional two port CAN interface.

2 Hardware configuration

In this section we describe used hardware and the lowest level of the software which is in lasagna code equal to hardware drivers

2.1 CANbus interface in PC

In the prototype we used Adlink PCI-7841 because of cost and well documented Philips SJA1000 controller. We were mentioned that SJA1000 is capable to achieve bandwidth of 1Mb/s, but on this card it is impossible. We were used this card because of previous RTLinuxFree based project and own driver derived from CanFestival project's code for hardware which is not yet included in the code. In that code there was an remark that driver does not work on some revisions of Adlink PCI-7841. Code does not worked with our card, so we made some extraction and customization of the driver's code as it is not driver but only module with exported symbols which are function names. When we used only BasicCAN functions then driver worked, but when used another features of the SJA1000, system crashed. BasicCAN is mode of compatibility with Philips PCA82C200 which is pin and electrically compatible predecessor of the SJA1000. There is possibility to use SJA1000 in the PCA82C200 board but then SJA1000 can act only as PCA82C200 or only in BasicCAN mode of SJA1000. About half of the cards with PCA82C200 supports only 500Kbit bandwidth although PCA82C200 chips has 1Mbit capability. During research we tested German Peak system

PCAN-PCI dual channel SJA1000 based card. And there difference. It works on RTAI with RTDM driver on the 1Mb/s well. There was RTDM driver for that card from the manufacturer. It worked extremely well except that there are no examples for kernel mode use of the driver and examples were not able to compile when used RT=RTAI option of the make command which causes to compile driver for RTAI. There were misspelled prefix of t_ for transmit instead of correct r_ for receive in receive function or vice versa. When corrected all examples compiled correctly with even with RT=RTAI option. Just now there is newer version of that driver available on the Peak system website. We have to test it, yet.

Because of listed we have found that there is reason to look at the boards and find differences. Although PCB for PCAN is much smaller than that for PCI-7841, number of ICs is higher for PCAN. All ICs on the PCI-7841 are present on the PCAN but there are additional ICs on the PCAN. As of this we found that our revision of Adlink card is board for PCA82C200 which is not in production for a long time and it is replace by SJA1000 as card is sold incorrectly as SJA1000 card.

Therefor we will use PCAN card in the future version as well as we are to test PCAN-USB opto-decoupled PCAN interface for USB in the future. We used this USB dongle with Peak system Windows only software to monitor CANbus during development for system's unpredictable behavior debugging before testing PCAN-PCI. But because of incompatible interface with current driver it is needed to rewrite lowest level of the application. Here is shown why there is needed lasagna code strategy of development instead of pure ravioli code which is best code strategy for code maintenance.

2.2 Motors and drives

We use 3 motors from German vendor Lenze with drives with CANopen interface from same vendor. There are drives with or without PLC. We used version without PLC because we do not need this function when limit switches are processed as well in non PLC version. Drive also handles over-current, over-voltage over-heating and maximum velocity or position error exceed. We can transmit wished position and velocity to the drive and read current values of velocity position and limit switches statuses with period of 2nd milliseconds instead of planned 1 millisecond with. This achieve wished accuracy of 0.1 mm with speed of 50mm/s when normal speed is 30 mm/s and maximum is 80 mm/s. Accuracy is not dependent on the size of arc as in previous using solutions. If in the previous solution circle has diameter 1 m

then accuracy in previous DOS based solution has dynamic accuracy of 15.7 mm when diameter of and circle is 1 cm accuracy of 0.157 mm. If arc has 90 degrees then accuracy 3.93 mm for arc with diameter 1m and with 1cm is it 0.0393mm for speed of 157.08 mm/s but accuracy of 0,12 with velocity 50 mm/s and 0.21 mm with velocity of 30 mm/s. theses velocities.

Our new solution has dynamic accuracy of 0.32 mm at 160 mm/s; 0.16 mm at 80 mm/s; 0.1 mm at 50 mm/s, 0.06 mm at 30 mm independently of the diameter of the arc. Only parameter new solution depend on is velocity because time interval is fixed. Accuracy is counted as time interval of 1 ms times velocity. Opposite is old solution where two factors limits accuracy. First of them is maximum count of interpolation points of the arc which is 200. Second is minimum time period between points of 1ms.

Therefor new solution has canceled one of two limits of the old solution but because of badly chosen PC interface for CAN and drives with minimum refresh wish value 1-2ms depending on used CANbus bandwidth the second limitation has been worsen. This can be solved by replace CAN interface from Adlink to Peak system card. Only digital drives that are capable to refresh wish value in period lower than 1 millisecond are drives of product line Sinumerik from german vendor Siemens. Sinumerik is only drive product line that is specialized for high precision high speed CNC and therefor their cost is at least two times higher than drives from Lenze, ABB, Rockwell Automation/Allen-Bradley, Control Technique, Rexroth or Invertek which are only companies in second league in drives behind only first league Siemens.

2.3 Input/Output modules

On the CANbus there are also several binary inputs output for example for toxic gas exhaust section select and start/stop of the exhauster. Machine is divided into exhaust section when only one section is enabled because then you do not need to use that high power exhauster as needed by just one section. Using several sections then make machine more green as it consumes lower electricity. Exhauster is needed because cutting metal material by plasma is based on the sublimation of the metal. Produced metal gas reacts with components of the air with product of toxic gases. Exhaust has also cooling effect which is needed to achieve minimal cutting gap. Another binary I/O are used to start/stop plasma and check plasma generator's state because there are part of rapid move between cutting trajectories. Analog output is used to set power of plasma which is pa-

rameter depending on thickness of cutter material. Analog input is used to measure deviation voltage of electric arc of plasma which indirectly measures deviation of hight of cutting head from the material which must be zeroed for best cutting process and minimal cutting gap. There are also binary inputs from operator's panels buttons and binary outputs to change color of the button, to inform operator which buttons are allows and which not. It is possibility not to under-light button with any color.

2.4 CAN bus

All control system components are connected by field-bus from German company Bosch named Controller Area Network which is primarily used in automotive industry. We chose this fieldbus because in 2007 we were world's number one car maker per capita, when car production in Slovakia reached a record 571,071 that means the average of 105.7 cars manufactured for every thousand inhabitants[1]. Therefor knowledge of the CAN bus adds capability to achieve job after crisis. After crisis is it because our top car bodywork designer Jozef Kabana chief-designed Bugatti Veyron, and was a chief bodywork designer of Audi in time they made latest facelifts of Audi A4/A6/A8. Bratislava's Volkswagen produces Audi Q7, Porsche Cayene and VW Tuareg. Our project made major success in area of CAN bus kernel framework as listed in software part. Volkswagen team has been beaten by us in this part of kernel but there are different approaches by doing this.

2.5 Industrial PC

For the GUI and control we are using industrial panel PC with resistive touchscreen without mouse or keyboard. PC is in moveable rack, which could be moved across the machine. the only cables from an to rack are power cable a two four wire CAN bus cables. industrial PC has only two slots for additional cards one PCI and second PCI Express x1. Now we use PCI, for PCI-7841, peak system card is as well for PCI Express x1 as for PCI, so we can have four CAN ports. Processor is Pentium IV based Celeron on 3Ghz. There are two Ethernet adapters one for 1 Gbit/s and one for 100 Mbit/s for use to remote monitoring via Internet or by executives of the company, where machine will be working after sold. As well for planned automatic software updates. Graphic adapter is integrated on the board from Silicon Integrated Systems. for the future we plan use integrated GMA3000/3500 or Radeon HD4200 for much faster graphics and OpenCL usage.

3 Software

In this section we will discuss software. It has three modules working on different computers. First is design of what to cut by engineers in their own CAD system. This leaves unchanged. Second is system to transform CAD image into NC or CNC program into text file with commands in language that machine understands. This was upgraded from DOS application to another multi-platform solution. Third part is control system running at the industrial PC. This basic architecture is as show ravioli architecture of the code because each part of system is independent and communication is strictly defined by DXF(Drawing Exchange Format) format between designing CAD and transforming CAD/CAM software to the own NC program language REZ. NC program is by Prva Zvaracka called Cutting plan but any other firm call it NC program. Machine must be able to read text file with REZ language command and interpret them. It is not ravioli code because independent entities are not as small as it can be. But there is not design of code but design of architecture.

3.1 CAD

CAD could be chosen by customer only condition is that he export drawing into DXF which is format designed to exchange CAD drawings between incompatible CADs. Therefore customer can use amateur's CAD like QCAD, half-professional CADs like Autodesk AutoCAD, which is only available for windows and is most used in Slovakia or as well professional CADs like Bentley Microstation, IBM/Dassault Systemes CATIA or Parametric Technology Corporation Pro/Engineer. All of the professional CAD are able or was able in previous versions to run on Linux or other Unixes as well as majority of amateur CADs. Half-professional CADs are rare for GNU/Linux but there are several noticeable as Weber Systems Synergy, VariCAD or U.S. Army Ballistic Research Laboratory BRL-CAD. There is open source project openCASCADE which produces platform to build multi-platform CAD especially CADs for GNU/Linux. It is main project of Google summer of code of previous and this year with potential to be finished in 2010 as framework that has all function of AutoCAD except DWG file reading. This is because openDWG library from Open Design Alliance[2] which supports DWG is paid, but last year there was a change that Adobe opened specification of the DWG via Open Design Alliance for free and there are several CADs for linux that supports Au-

todesk's AutoCAD native format DWG but it is still no perfect. Project to build free Linux library similar to openDWG, which is free for anyone for Microsoft Windows but paid for Linux is another key project of Google Summer of code 2009. openDWG is technical free Linux, but Linux version is only available for members of Open Design Alliance but membership has obligated yearly fee. To download Windows version of openDWG membership of Open Design Alliance is not required, but to download any Unix version of openDWG is membership required.

3.2 CAM - conversion to the REZ

In this chapter we will discuss three offered solutions of converting DXF to NC program

3.2.1 DCPC

Old version of the software is Slovak language based Design CAD for PC. It has two parts Imsisoft Design CAD for Drawing and Basic script which produces NC programs from the drawings named DCPC (Design CAD for Processing Converter). DCPC work only on DOS and only of specified version of the DesignCAD and only with its Czech localization.

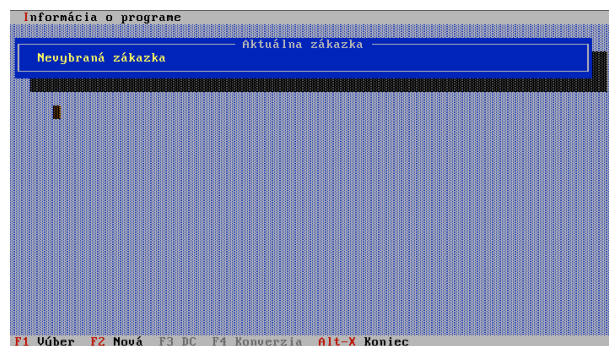


FIGURE 1: *DesignCAD for PC main screen*

As seen of the Figure 1 GUI of this software is based on DesignCAD and looks like traditional Borland's IDE for Pascal or C language. It works on DOS or MS Windows 98 or older but not with NT kernel. For newer Windows it does not work. So there is time to replace it, but some engineers want not to change known GUI. For them there is solution with Sun VirtualBox and virtual FDD for mounting *.img as Windows disk which enable transfer data to and from the DOS machine. Main and only problem is with CODEPAGE for localization. FreeDOS does not support national codepages (national character sets and number formats) at least via needed

country.sys file in stable FreeDOS release. In development releases country.sys is present. But there are free to download supplemental files for MS-DOS 6.0 including country.sys at Microsoft website[3]. There is a problem that FreeDOS is fully compatible with DOS 6.22 not 6.0 and for DOS 6.22 Microsoft does not offer country.sys. Therefore we need to use MS-DOS to run this program with correct display of the national characters. There is no technical barrier to use MS-DOS in VirtualBOX but this is only a legal problem. There is only a solution via Microsoft's license legal guide to downgrade rights[4] there is written: "Downgrade rights grant the right to use prior versions of Microsoft software. Microsoft Volume Licensing programs include specific downgrade rights. The OEM License Terms for most OEM versions of systems software do not grant downgrade rights. The exception is the OEM License Terms for the WindowsXP Professional operating system and the Windows Vista Business and Windows Vista Ultimate operating systems, which grant downgrade rights." This does not mention replacement of the newer system when used older as usually mentioned. This is only Microsoft's that do not mention Windows for downgrade, so if we have Windows License we can use DOS in Virtual BOX legally, when do not use other Windows application simultaneously. You just to have legal copy of the DOS installation disks or copies from subject that owns it legally. This is true for us. Prva zavaracka has Windows licenses as well one license for MS-DOS licenses from the past. They also have valid DCPC and DesignCAD multi-license, but only one computer capable to run MS-DOS.

3.2.2 QCAD/CAM Expert

Ribbonsoft QCAD is GNU GPL licensed multi-platform 2D CAD. CAM expert is proprietary addition to QCAD for transforming DXF to standard or user defined NC program language. GUI is shown on the Figure 2. Software is fully localized due to usage of QCAD code. Q in the name means usage of Trolltech's multi-platform Library Qt. After Nokia has bought Trolltech the name was changed to Qt software. CAM Expert costs 149.00 EUR per single seat as shown on the Zurich, Switzerland located Ribbonsoft home page[5]. CAM Expert is tested on several OS's Windows: 98, ME, XP, 2000, Mac OS X: Mac OS X Tiger (PPC, 10.4), Panther (PPC, 10.3) and from Linux: SuSE 8.0, 8.1, 8.2, 9.0, 9.1, 9.2, 10.0; Redhat 7.2, 8.0, 9.0; Fedora 3; Knoppix 3.3, 3.4, 3.8; Xandros 2; Mandrake 9.0, 9.2, 10.0, 10.1; Linspire 4.5, 5.0; Slackware 8.0; Puppy 1.0.5; UHU-Linux 1.2 and Turbolinux 7.0, 8.0. We have tested it on Slackware 9.0 and 12.0, Mandrake 9.0, Open-

SUSE 11.0 and 11.1 and MacOS X 10.5.7 and 10.5.8 (Leopard) on Intel based Mac. We tested MAC OS 10.4 version on that MAC.

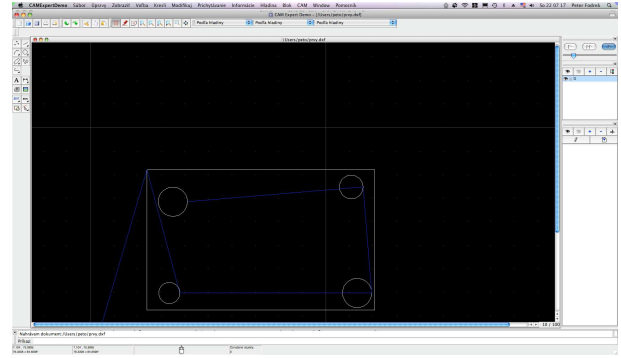


FIGURE 2: CAM Expert GUI

In the CAMExpert are located all menus to select language except CAM with CAM Export and Reorder entries items. First of the is used to export drawing to the NC program with selected language. Simple language definition files are in the machines directory of the directory where CAM Expert is installed. Files are named language.cxm where cxm means Cam eXpert Machine. You can select several languages included G-code which is ISO 6983 norm currently redefining by drafts and approved parts of ISO 14649 series. Part 14 named ISO 14649-14 for laser and EDM e.g. plasma beam technologies process control commands are not yet approved. Preparing stage of ISO 14649-14 was finished this year and on June the 25th, 2009 was started approval process which belong at least half a year. ISO 14649 draft is not currently supported by CAM Expert, but in Q4 2004 there will be new generation of CAM Expert introduced to the market or you can add it via creating own cxm file.

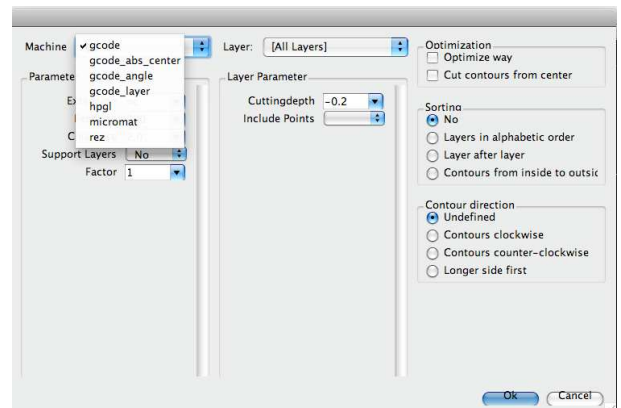


FIGURE 3: CAM Expert CAM Window

CAM dialog is shown on the Figure 3 where language choice is upon Extension, Feed rate and Clear-

ance parameters. Main advantage of CAM Expert is its capability to have user interface in 17 languages including English, German, French, Italian, Czech, Slovak, Hungarian, Polish and Russian but it is far less than 42 languages in QCAD. Where untraditional languages like Bask, Catalan / Valencian, Breton or Irish is possible to set. Other advantages are that program run on many OS-es including Solaris, BSD and listed before OSs as well as simplicity to add own NC language. Main disadvantages are that CAM menu is only in English as well as that it is not capable at current version to automatically order arcs and lines in other order then order the arc and line they were drawn or saved by another CAD. You must reorder arcs and lines manually which means to click on the each part of the image in order you wish to get as a result. Only exception is that next parts of image are in right order then you must not click on that parts. Another disadvantage is that you must add arcs or lines for coming the trajectory a leaving out of trajectory before and after trajectory is cut. These additional parts are usually used for accelerating and decelerating from the cutting speed as they are must for plasma and other EDM technologies but not needed for milling or turning. CAM Expert can do simulation of the cutting to determine order of cutting. You can also choose speed of cutting, but when you use play button then you only see order of the lines and arc and only when lowest speed is chosen, because system is so fast. When you use fast forward button instead of play button you will take point by point simulation which is slower than part simulation via play button.

3.2.3 HeeksCAD/HeeksCNC

HeeksCAD[6] is multi-platform 2.5D CAD at range of low end half-professional CAD that is based on OpenCascade library covered by new BSD license. Technically it is an GUI for openCASCADE, made partially in C and partially in python. HeeksCNC[7] is python library written in C for HeeksCNC which act as an extension of the capabilities of the HeeksCAD. Extension adds one menu item and one toolbar to the HeeksCAD GUI you have to go to the File menu and choose plugin and add shared library (shared object) and then you need to restart HeeksCAD to start installed plugin. For Microsoft Windows there is not separate installation of HeeksCNC because if you install HeeksCNC you will install HeeksCAD with integrated HeeksCNC plugin. Both HeeksCAD and HeeksCNC are projects maintained by Dan Heeks and it is one of the projects under Google summer of Code 2009. Main legal advantage of the BSD license is that you can take all my work and use it

for your own commercial application. That means we can close modified source code for a customer. Main advantage is that HeeksCNC adds automatically accelerating and decelerating additional part of the trajectory, but only as arcs. This is no problem although DCPC uses arc or lines as such parts, but arcs are more suitable for the mechanics of the machine. Arcs has only 90 degrees so it is longer than similar linear item of the path. This slows cutting a bit compared to linear acceleration deceleration item. You can also choose a point of middle of the arc for starting and as well as finishing trajectory or more precisely point near it. Program has several modes of the transfer as shown on the Figure 4 on the 0.7.x series of the HeeksCNC.

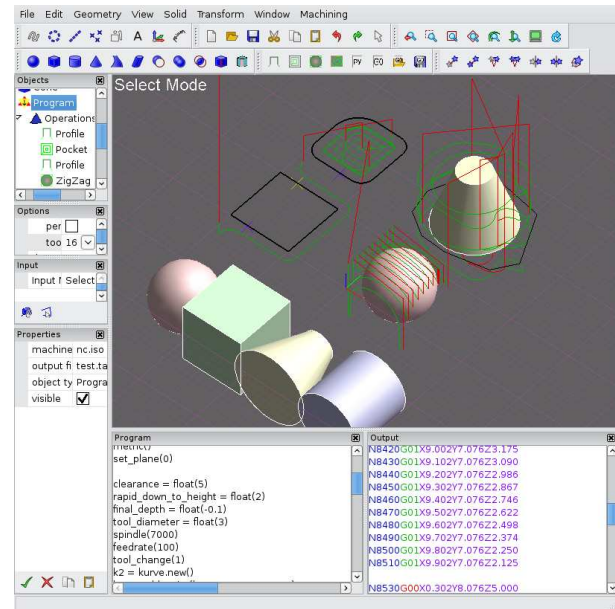


FIGURE 4: Heeks CNC 0.7.x

Current version is 0.8.1 for Linux and source code version and 0.8.2 for Windows. HeeksCNC menu bar item is Machining and toolbar for HeeksCNC starts under scissors and finished with diskette with "GO" label. First four buttons of this toolbar sets profile mode (in HeeksCAD called operation) for current trajectory or 3D object here known as sketch. For sketch we can set clockwise or counterclockwise direction as main of the parameters. For profile created from the sketch we can set correction for tool dimension like left or right, which means on which side of the trajectory when cutting selected direction is correction applied as well as diameter of the tool. There is applied correction as half of the tool diameter which is logic because we control position of the middle of the tool. As of 0.8.x series there is possibility to choose 0 mm diameter. For the profile we can select position

of the start and stop of cutting as shown on the Figure 4 with two crosses near the square. Yellow is for start and blue is for the end. When you choose the at the same point only finish is shown as also visible on the Figure 4 on the object right of the square. "Py" labeled button generates python script from the drawing and "Go" labeled button executes it. From the 0.8.x version there is no "Py" button but "Go" button acts as sequence of "Py" and "Go" buttons. Function of "Py" button is available from the Machining menu in the 0.7.x as well as in 0.8.x series. Script is shown in the Program sub-window. Script contains several import directives but one is dynamic and generated dependent of the the property named machine of the Program object as seen on figure 4. In the 0.7.x there is relative path to the library file containing functions needed by python script to generate code in proper language. In the python there is / in the path replaced by "." and final .py or .pyc in the name of the library is omitted. So nc.iso means path ./nc/iso.py or ./nc/iso.pyc where .py is used for source code of the library in the PYthon language and .pyc means PYthon compiled. compilation of the .py to .pyc is done by first call of the .py file or using import directive which includes library. when nothing changes to the .py file then .pyc is not regenerated and pyc is used. Therefor we can distribute binary python library without source code for generating library of the REZ language. 0.8.x series adds to the nc directory text file index.txt where is written for iso following "iso G-CODE iso6983". This causes that user does not write name of the library file but chooses from the list possible values in the machine parameter string "G-CODE iso6983" for HeeksCNC to add nc.iso the import directive. After run script correctly there is generated file /tmp/test.tap for Unixes and C:\Windows\TEMP\test.tap or similar for Windows. In that file there is generated code but not for user. After this HeeksCNC calls another file that has same name as generating script but with additional suffix _read so if NC code is generated via rez.pyc then there is to call rez_read.py or rez_read.pyc. This file reads /tmp/test.tap file which is sent to the script as parameter and could be changes via property of the HeeksCNC Program object for both generating and reading script. Read script must read line by line a parse output file. In includes library with five main functions add_text that is suitable to add process commands. Similar is add_comment which adds text but this text is shown in the output window with blue color instead of black for add_text. Because REZ language differs remarks and process commands we used both. Next function is add_rapid which adds rapid move between Profiles, this can be only linear move and command

and path of the machine in the display window and output windows is red for part added by add_rapid. Function add_feed adds linear cutting move which is displayed in both output is green. Both add_rapid and add_feed has parameters of final point in 3D in mm. Last command is add_arc which adds arc to the trajectory. Arc is in graphical window green but arc command is cyan in the output list. Arc has different parameters for HeeksCNC and REZ language. REZ language has 3 parameters X,Y coordinates of the arc center and signed angle of the arch, where sign means direction but HeeksCNC had 6 parameters and the direction. First 3 parameters of heeksCNC is 3D vector from current position of start arc to the centre of the arc and second three parameters are coordinates of the final point of the arc. It exists simple conversion from parameters in REZ to the parameters of HeeksCNC by vice versa conversion is much more complicated. The read script produces /tmp/test.tap.nc.xml file which is pure XML a this file imports HeeksCNC after script finished. This file is used to display colored real trajectory of the Cutting head center. Open folder icon baton reads NC command file and displays it via read script. Diskette saves displayed NC program to the specified file not in the working directory.

Main disadvantages are that building new languages via two Python scripts is much more difficult then language definition in the CAM Expert, that system in in beta testing state and that windows version crashes when saving to the Heeks own format where all CNC settings is saved. Saving into DXF or using Linux version has no problem. Last problem is that there is no simulation like in CAM Expert but when you click on the lines of code in the output windows then corresponding arc or line will be bold in the image window. This can replace simulation but it is less practical. Sometimes drawing is not refreshed correctly so you need to change size of program windows two time to see all image. CAM Expert and HeeksCNC has same bug that when you save file. File Save dialog displays only *.nc and *.cnc files but saves into chosen endings of the name(in Windows Extension). HeeksCNC is better because when you choose *.rez file and same file exists HeeksCNC ask if you allow replace file but CAMExpert do no ask and rewrite file without notice. It is surprise because CAMExpert's language definition file has parameter default extension and HeeksCNC has no such parameter. Surprise doubled when you find that CAMExpert saves file with default extension instead of displayed extension. Therefor is HeeksCNC best solution for the future although author of new language can not find if is presently cutting on the left or right part of the trajectory. this function will be helpful

for us but engineers from Prva zvaracska uses only one side and they determine if contour is in or out via clockwise or counter-clockwise direction just now. It runs in Windows and openSUSE 11.1 form rpms in version 0.7.5 and from compiled source of version 0.8.1. on the MAC OS X 10.5.7 and 10.5.8 there is problem with incompatible vxGTK and wxWidgets libraries C language SDK a therefor we can not prove if it works.

3.3 Control system on the machine

In this section we mentioned how is machine software designed. At the top level there is ravioli design. There are three objects: Control application, Reading REZ format and transform it to set of Position and velocity wish values in time and the last one is Module of standard shapes.

3.4 Rez interpretation

Rez format interpretation is C language program that takes parameters as filename of the REZ language program, cutting velocity repeating and spaces between repeatings of the file in each axis. When cutting multiple plans and correction for size of plasma beam which is same as in HeeksCNC but without left or right direction which is written to the REZ file. Next parameters how long before start cutting is cooling gas open so gas comes in the place where will be cut started and how long after shutting down plasma will be gas opened. Last parameter is how long will be plasma beam on before start moving to crosscut thick material. Inputs are well defined and output is file with copy of memory of defined structures of values. Content of this structure has been changed during development and we produced this part based on previous two solutions but done by Mr. Blaho who does none of previous solutions. Rewriting code and redesigning structure made much faster code and better maintainable code but there was needed to add new feature of correction and this code needs to know line that will be interpreted by program in the future as well as current a previous non comment line of the REZ file. This was possible done only by multiple pointers because of bad definition of the REZ language back in 1990's and 1990's. Using REZ language is must because of old system working machines at the Prva zvaracska customer fabrics plants and their need to upgrade to the compatible system. Three different pointers and rewriting working code with new feature has shifted code bit toward to the spaghetti code but no more then it was needed as we have not enough time to

write completely new code with better architecture for adding new features. Before starting adding new features code was based on the lasagna code principals.

3.5 Module of standard shapes

This module was need by Prva zvaracska to allow worker test machine by cutting predefined shapes with parameterization of any aspect of the shape. It is made in Java and it allow user to choose shape, then to write parameters into it, view how parameterized shape looks like with correction of hazardous or not allowed combination of the parameters and then generates program in REA language in predefined directory with predefined name. GUI JAVA program calls this full screen application when chosen to open it and after program finished GUI must start Rez interpretation module. Main problem is that industrial machine has no keyboard nor mouse. It has only touch screen and user that is unable to use general purpose virtual keyboard like xvkbd. There was designed own virtualkeyboards that is able to use worker. Code is in java with strict ravioli internal structure of code.

3.6 Control application

Control application is designed upon lasagna code principles which is must because one part is in kernel space and another in user space. Each of this is different ply (of lasagna) and there is strictly defined which ply is lower level and thus has priority of execution because kernel space part is RTAI nanokernel application which run on same ply of system architecture as kernel. As we have found it run in Ring 0 of the processor. Realtime part is written in C and GUI user space part is written in JAVA using JNI(JAVA native interface) to access RT-FIFO pipes files in /dev. These pipes acts as communication channels between parts of application

3.6.1 Hard Realtime part

This part is designed originally as lasagna code with several layers first is communication with CAN card, then there is an layer of the CANOpen protocol, then there is layer of the devices which is designed as ravioli code where objects (ravioli) are communication with Lenze drives over CANOpen protocol, then there are communication objects for digital inputs, digital outputs, analog inputs and analog output. All these modules reads and writes values to the mem-

ory area of the process values. Next layer is layer of the control where are position controllers and mode of operation tasks as RTAI threads as well as communication via five realtime FIFOs. This layer is too complex and allow to produce spaghetti code directly from design because control depends on communication and thereof GUI could produce problems in the realtime parts. Very special problem is that GUI could ask to change mode of operation of realtime code. There is basic control which does not allow to accept request in any time, but we can not prove that it is enough. This is worst part of the realtime application. Best part is part of CANopen protocol. This part was developed at that level which allow one of the tasks Vladimir Chren's master thesis to be internationally accepted in CANFestival project[8] with "Wow" reaction in January/February 2008. It was the first working CANbus protocol in kernel space. CANFestival is CANopen framework originally from France and Vladimir was maintainer of the kernel port of the framework. But he was busy working on this project which not allow enough time for maintaining code last year. Vladimir overcame five years development of five developers in Volkswagen in Wolfsburg which developed similar CAN protocol kernel framework but VW superseded Vladimir in introduction to the kernel as VW code was introduced in 2.6.25 kernel in April 2008. VW code is still not working well as of 2.6.30 kernel but may be better in 2.6.31. To be correct VW's code used network stack instead of serial/tty stack which was used by Vladimir. As correct lasagna code each layer is written as individual kernel module and therefore modprobe for loading all is used in rc.local file.

3.6.2 GUI part

GUI is written in JAVA so it is ravioli code. As all code of the project it is written in Eclipse and by manual writing of SWT using code because of very bad Visual editor in the eclipse especially three years ago, when project started. The biggest problem is that JAVA does not allow to access /dev files and therefore there is needed C code compiled with JAVA Native interface Libraries and JNI classes for the code. Main problem was that C and JAVA use different ENDIANs so correction of the byte order is needed. JNI has endian correction functions. Application Loading before kdm or as only shell of automatically logged in user allows users to simply shutdown computer and developers are allowed to test new version within usage of old version.

3.6.3 RT FIFO

Biggest problem of control application is communication between parts. It is done by five RT-FIFO of the RTAI 3.6 on the Slackware 12.0. There is one unsolved problem here. We transfer data as structures put in to the one direction pipes. But when size of the fifo where data to display like velocity and positions are sent or wish values to the control module is not equal of integer multiply of the size of send data at once system crashes. As well as system may crash when GUI waits for confirmation of the state change from kernel space and realtime part waits for ask for state change. Both parts are tested individually via C application reading and sending data to the pipe as anticipated when everything works well but when they are tested together it sometimes fails. This is sign of the spaghetti code between layers. Spaghetti code is well known as badly written code where independent parts depends on each other. Spaghetti code is caused by principal misfires of the project/team leader as Peter Fodrek is for us. Team-leader's task is not to allow customer to take not full or not to correct specifications at the start and to change specification during programming. Failure was that specifications were not written and signed before project starts by customer. It was because friendship between company executives. Another task of the team-leader is to be available for programmers to help them with problems and check their codes and third party codes as well.

4 Conclusions

We have designed function working application with stability issue needed to check and repair. Working application will be presented via video on the conference. There is need to change team-leader or at least to force him to believe in programmer analysts which coded Control application as much as it was in this project. And check their work on communication part and not allow them to be forced by continuous specification change to produce spaghetti code.

References

- [1] http://www.industryweek.com/articles/slovak_car_industry_production_almost_doubled_in_2007_16083.aspx
- [2] <http://www.opendesign.com>
- [3] <http://support.microsoft.com/kb/124382>

- [4] <http://www.microsoft.com/downloads/details.aspx?FamilyID=4F4B3CFD-7F4E-46CB-8117-8275F7683D3F&displaylang=en>
- [5] http://www.ribbonsoft.com/store/index.php?main_page=product_info&cPath=69&products_id=193&zenid=37a23ee796e0a6edce7c6a933f3be043
- [6] <http://code.google.com/p/heelscad/>
- [7] <http://code.google.com/p/heelscnc/>
- [8] http://sourceforge.net/mailarchive/forum.php?thread_name=200801191554.49767.vlado.chren%40gmail.com&forum_name=canfestival-devel
- [9] http://kernelnewbies.org/Linux_2_6_25#head-0957d99bd917f82924f1eab1f3f17945e1c4dc90

This paper is supported by grants from APVV and VEGA which solution finished in 2008 or first half of 2009.