AREM PRO, s.r.o.
155 00 Praha 5, Na Hvížďalce 23/1702

automation, regulation, measurement
www.arempro.cz

Operation of the Cnc886/Win Control System

General Part

revision 7.

7. revision, 5.2.2021

AREM PRO, s.r.o., Na Hvížďalce 23, 155 00 Praha 5
www.arempro.cz

Content

1. Basic Information

1.1. System variants

Cnc886 control system from AREM PRO, s.r.o. was created in 2000. It was originally designed for the real time operating system QNX. The need for "real time" operating system arises from the need to control the movements of the axes so that they reach the correct position at a precisely determined time. Once one of these conditions is not met, the movement is jerky. The time inaccuracy in the cnc886 system had to be less than 30 microseconds due to the chosen method of communication with the servo systems. The QNX system was able to meet these requirements. The version with OS QNX is no longer distributed at the time of writing.

In 2005, a version for Windows XP was created for numerous customer requirements. The concept of a Windows-based system has many benefits. One of them is the possibility of cooperation with other programs in the computer control system, such as CAD/CAM systems, programs for production control, programs for workshop programming, etc. Another advantage is the functionality of all peripherals, provided by the operating system, the functionality of all Windows network services and, finally, guaranteed growth in the performance of the control system in the future. In addition, Windows brings many interesting programming technologies that Cnc886 / Win uses or plans to use to improve the graphical user interface. The disadvantages of Windows are their unsatisfactory real time properties. However, these could be supplemented by the Windows RTX extension.

In 2012, another version for the Windows operating system began to emerge. This version no longer uses the RTX real-time extension. To ensure the necessary real time properties, it uses an external board (developed by AREM PRO, s.r.o.), with which it communicates via the Ethernet interface. The advantage is financial savings on the part of the computer and the RTX superstructure, as well as the possibility to run Cnc886 on almost any HW and any version of Windows. The RTX version always required careful selection of the control computer for RTX compatibility. Each version of Windows also required its own version of RTX. The disadvantage of the version without RTX is that the real time features are not so good. The external board works with the data that the Cnc886 prepares "forward" to cover up short-term timing irregularities in Windows. This data can no longer be changed. Therefore, any change is placed at the end of the resulting queue. This delays the system's response to external events. Fortunately, practice shows that the total response delay is in the order of 50ms or better, which is rich enough for most applications and does not limit the user in any way.

In 2018, it was necessary to leave the Windows XP operating system and upgrade to a higher version - Windows10. The system was modified to a 64-bit version and at the same time many internal modifications and improvements were made. The old "skeleton", on which new functions have been packed for years, has been replaced by a new one that will better carry all future changes.
For current users of Cnc886 in the version with Windows XP, the good news is that we have retained the possibility to generate the latest version of Cnc886 for Windows XP. After a minor modification, it is possible to switch to the new version of cnc886 without

changing the HW and operating system. However, switching to Windows 10 is a better upgrade option.

1.1.1. Cnc886/Win with RTX

The Cnc886/Win control system is based on an industrial PC, equipped with an additional card for controlling units via the CAN bus. In the course of 2020, we also anticipate communication via EtherCAT.

The Cnc886/Win system runs on Windows 10 or older Windows XP or Windows XP Embedded. A real-time RTX extension in the appropriate version is installed to the system.

1.1.2. Cnc886/Win with EtherCAN

The system is based on a control computer, which can be almost any model. From an industrial computer to a cheap laptop. It can run almost any version of Windows from XP onwards. The control computer is connected by an Ethernet cable to the board (initially it was an ACPU795Sk board, now it is an ACPU-F4 board), which provides communication over two CAN buses and also sends packets on CAN lines to meet the strict timing of servo systems. From this place, i.e. from the CAN connectors of the board, the system no longer differs from previous versions.

1.2. Start of the Cnc886 system

When you turn on your computer, you see the standard Windows user interface. There is a shortcut on the desktop that allows you to run the Cnc886 program. Once launched, the application performs a number of preparatory operations. Their progress largely depends on the state of the Windows system, so it can happen that it takes several tens of seconds to start. Especially shortly after power-up and startup, Windows processes the process of logging on to a network. These processes block the startup of Cnc886, as well as the startup of other applications. All that remains is to wait.

After starting Cnc886, a window will appear on the computer screen, forming the graphical interface of the program. Most operations are performed in the usual way. We move the mouse over the object and press the left mouse button (click) to activate it. Or after clicking in the text box, we can use the keyboard to change the text. Most of these operations are completely standard and are respected by the Cnc886 system.

The graphical user interface of the Cnc886 system is variable. Different types of control applications required different graphics and feature sets easily accessible from the interface. When creating these variants, we set out on the following path. The Cnc886 program has its standard interface, which is further described in this general part of the system operation description. In some applications, the standard interface is supplemented by additional windows or is completely replaced by them. In the case of "full replacement", the standard interface is minimized on the bar and remains ready for use. Less frequent tasks can thus be omitted in the specialized interface, as they remain available within the standard interface. All graphical interfaces can be used at the same time, so there is no need to switch off the existing one before using another.

1.3. System security

There are some risks involved in using the Windows operating system. Thanks to its expansion, the system is the target of computer virus attacks. The system also entices the installation of new and new applications. This can cause the driver to collide with newly installed applications. Therefore, the following rules must be strictly observed.

- No programs, drivers or patches may be installed on the control computer without the knowledge and consent of the control system supplier. Otherwise, the correct function of the control program cannot be guaranteed.
- Programs and scripts that are not locally installed on the control computer must not be run. Therefore, programs must not be run from floppy disks, CDs, the network, and any other media.
- Firewall protection must not be disabled.

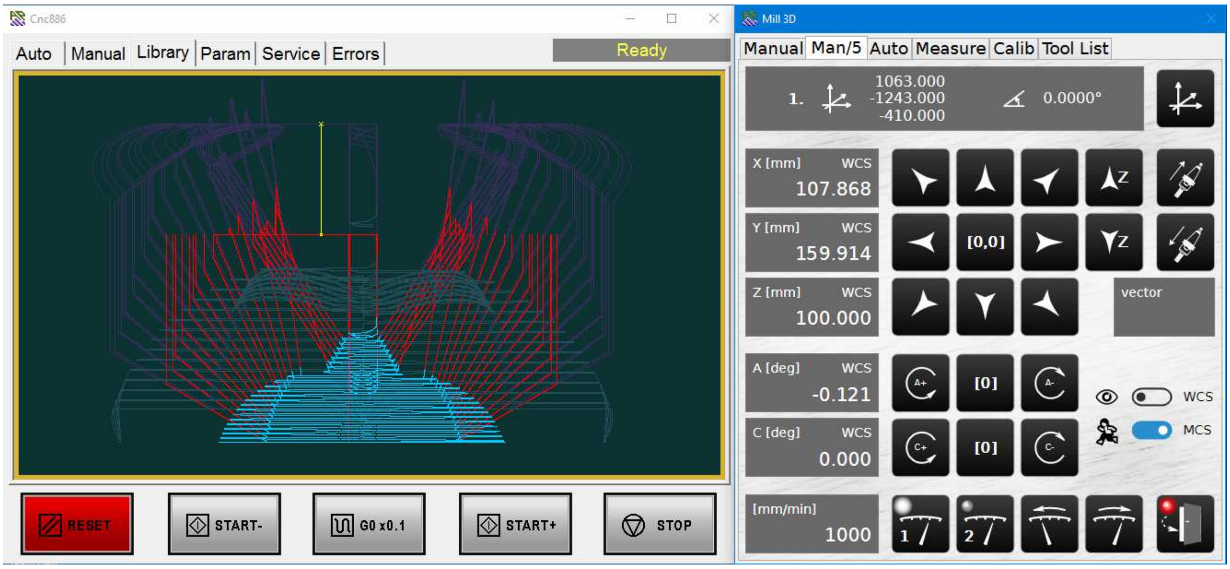
Windows comes with an End User License Agreement (EULA). If you do not agree to this agreement, you are not authorized to use the control system. The warranty conditions of the EULA (clause 8) are naturally reflected in the warranty conditions of the control system as a whole and are therefore the same.

2. Standard system user interface

When preparing the new version of the control system, we were faced with the decision whether to keep its current appearance and division, or to rework it into a more modern form. In the end, we decided to keep the look and transfer any new look to add-ons (plugins), which can gradually completely replace the original GUI. We have already created a separate interface for some applications, while the original one is primarily hidden and can be brought to the foreground by clicking the mouse.

Therefore, there are relatively few changes to the user interface, although Cnc886 has changed significantly on the inside.

An example of a plugin that complements the standard interface is, for example, MILL3D for milling machines:



1. 1063.000 -1243.000 -410.000 0.0000°

X [mm] WCS 107.868

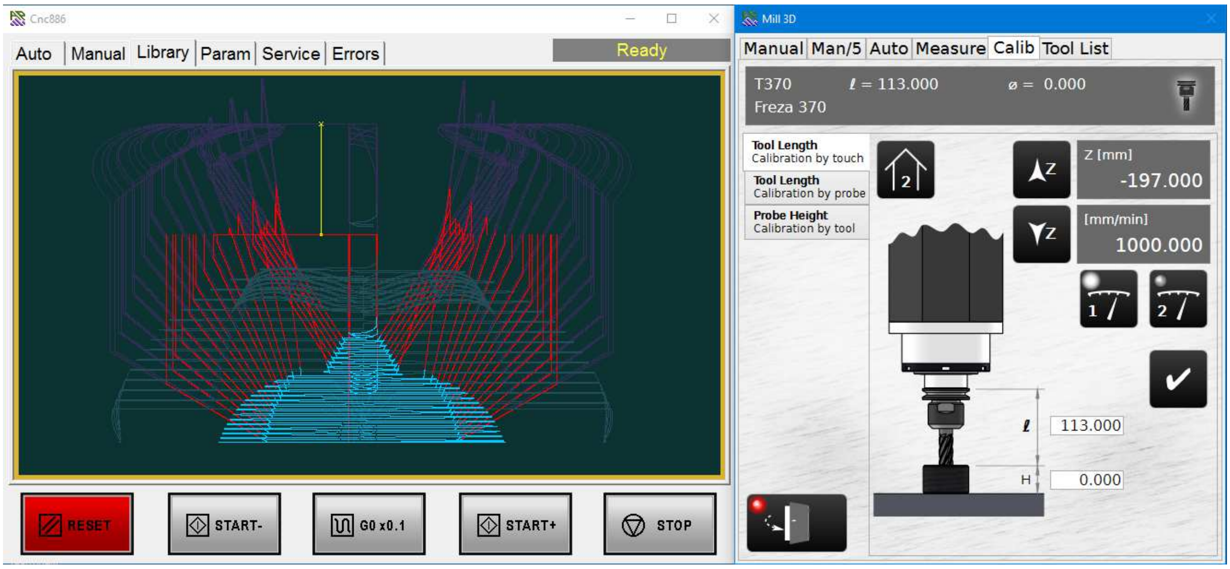
Y [mm] WCS 159.914

Z [mm] WCS 100.000

A [deg] WCS -0.121

C [deg] WCS 0.000

[mm/min] 1000



T370 ℓ = 113.000 ⌀ = 0.000
Freza 370

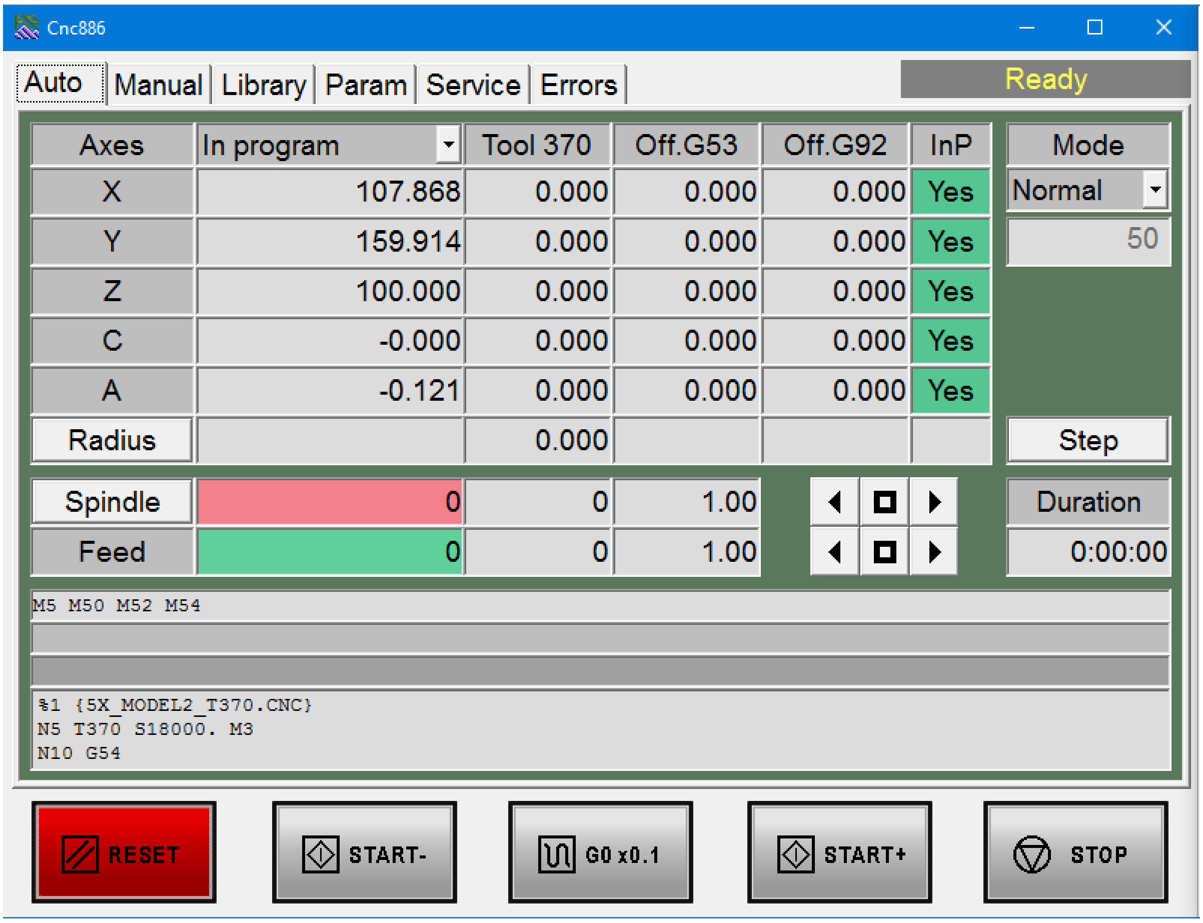
Z [mm] -197.000

[mm/min] 1000.000

ℓ 113.000

H 0.000

The standard user interface is organized into three basic fields.



Axes	In program	Tool 370	Off.G53	Off.G92	InP	Mode
X	107.868	0.000	0.000	0.000	Yes	Normal
Y	159.914	0.000	0.000	0.000	Yes	50
Z	100.000	0.000	0.000	0.000	Yes	
C	-0.000	0.000	0.000	0.000	Yes	
A	-0.121	0.000	0.000	0.000	Yes	
Radius		0.000				Step
Spindle	0	0	1.00			Duration
Feed	0	0	1.00			0:00:00

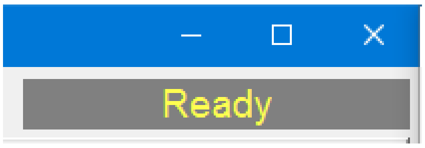
```
M5 M50 M52 M54
```

```
%1 {5X_MODEL2_T370.CNC}
N5 T370 S18000. M3
N10 G54
```

- Status field
- Tabs
- Button bar

The status field is in the upper right corner of the system screen. It is always visible. Tabs fill most of the screen area. Some tabs are further separated into child tabs. The button bar is at the bottom of the screen and is always accessible.

2.1. Status field

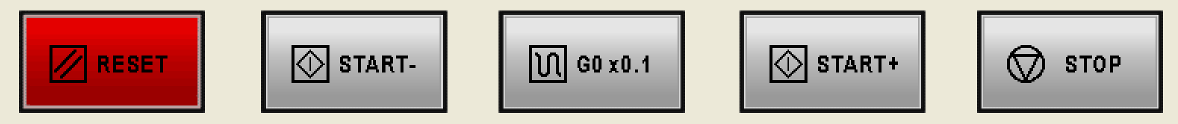


The status field shows the current system status in yellow letters. If you want to perform an operation and it does not happen as you would expect, the problem may be that the system state does not allow it. In this case, pay attention to this window. You will also see **Error** in case of error.

Possible system states are:

Status	Message
Power supply to the power part switched off	Power OFF
Initiation after switch-on the power part	Initialising
The system is in order, some axis is not referred	Not referred
Referring is in process	Referencing
The system is ready, referred	Ready
Program is in process	Running
Program suspended by M0 function	Stopped by M0
Program suspended by STOP button	Stopped
Manual shift of axis is in process	Manual move
Spindle started manually	Manual bin
Loading of the program is in process	Loading: xxxxxxx
Power part powered; movements blocked	Door open / Blocked
Error condition	Error

2.2. Button bar



Along the bottom edge of the screen is a panel with five buttons. The buttons can have different functions depending on which tab is active. The buttons change color depending on their status, they behave the same as the backlit buttons. Thus, they can be pressed and unpressed, as well as luminous or non-luminous.

RESET
This button is used to stop an ongoing operation, whether it is program run, referencing, manual spindle operation, etc. If it is red, it indicates that the machine is at rest. If it is not lit, the machine is in one of the operating modes.

START-
This button is used to start manual movements of the axes backwards. Glows green while moving.

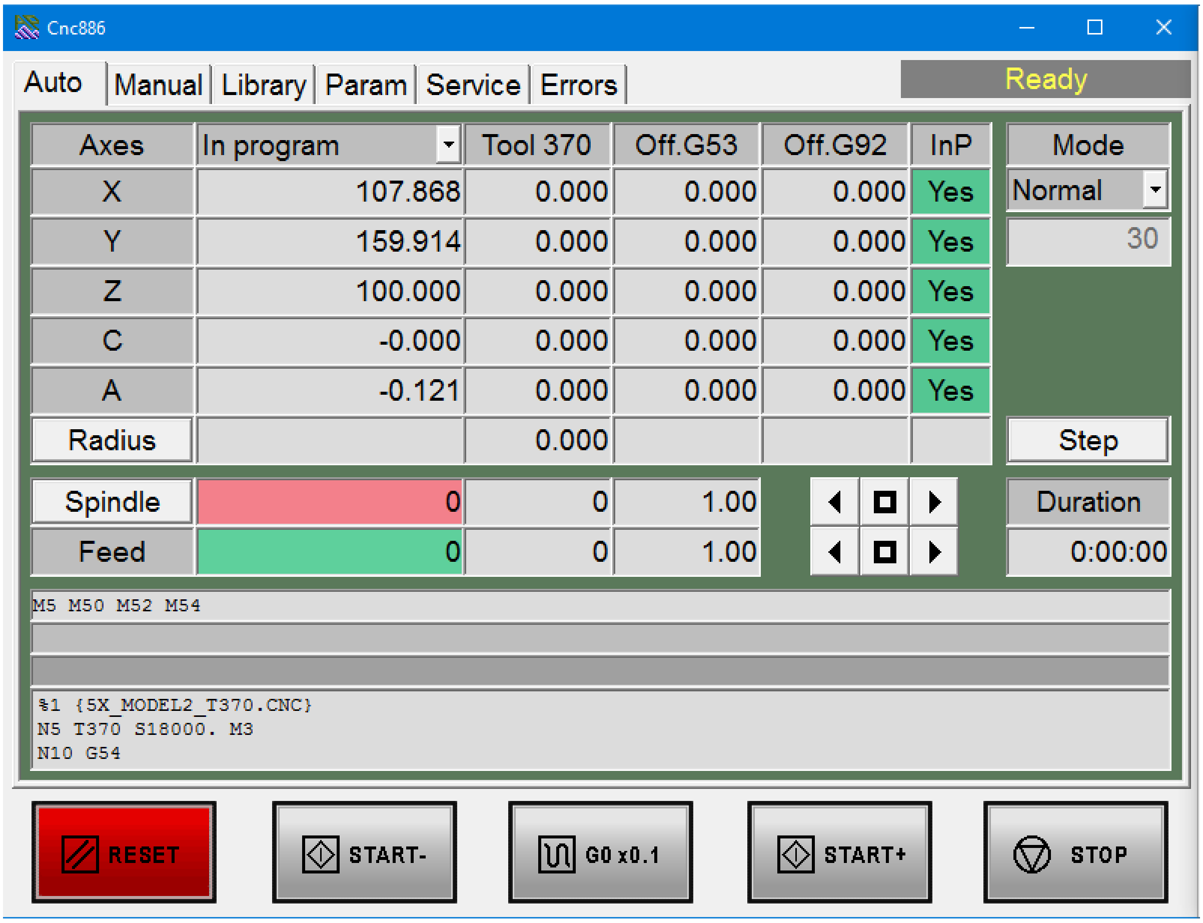
G0 x0.1
This button is used to reduce the rapid positioning. When activated, all rapid positioning moves are slowed down ten times. Pressing the speed override again cancels. When speed reduction is on, the button is highlighted in white. Speed reduction is especially suitable for tuning technology programs. It gives the operator time to react in the event of an imminent collision of the machine with objects in the work area. In older versions, the button was labeled SPEED. In newer versions, this button is called **G0x0.1**, which should better describe its function.

START+
This button is used to start manual forward movements of the axes and to start the program run and to resume program run after pausing the M0 functions or with the STOP

key. It is illuminates green when the program is moving and running automatically, unless it is paused by the M0 function or the **STOP** button.

STOP
This button is used to pause a running program. Illuminates red when the program is paused and waiting for the **START+** button.

2.3. Auto tab



Axes	In program	Tool 370	Off.G53	Off.G92	InP	Mode
X	107.868	0.000	0.000	0.000	Yes	Normal
Y	159.914	0.000	0.000	0.000	Yes	30
Z	100.000	0.000	0.000	0.000	Yes	
C	-0.000	0.000	0.000	0.000	Yes	
A	-0.121	0.000	0.000	0.000	Yes	
Radius		0.000				Step
Spindle	0	0	1.00	◀ ■ ▶		Duration
Feed	0	0	1.00	◀ ■ ▶		0:00:00

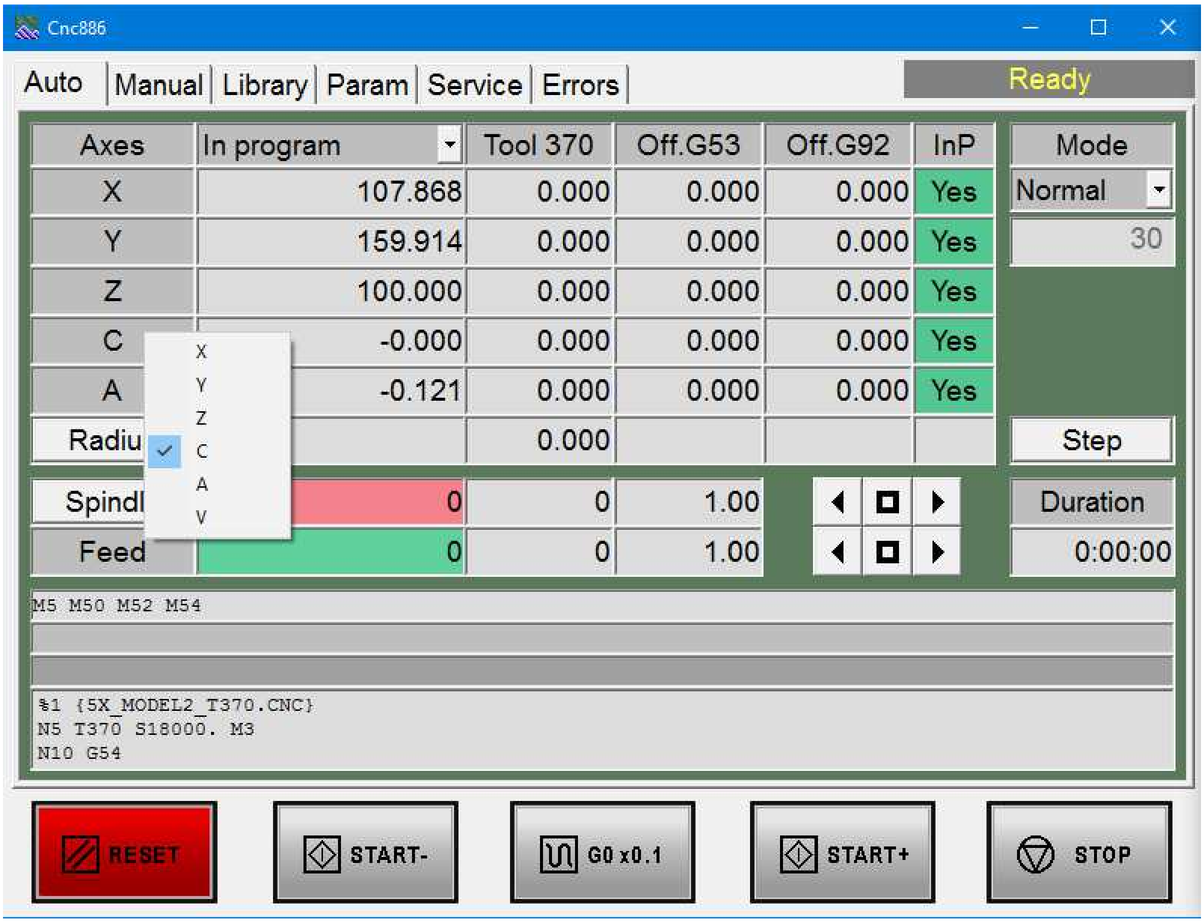
```
M5 M50 M52 M54
```

```
%1 {5X_MODEL2_T370.CNC}
N5 T370 S18000. M3
N10 G54
```

The **Auto** tab is used to monitor and control the operation of the technological program.

Axes
The largest area is occupied by information about the status of individual axes. Depending on the machine type, both the number and the names of the axes change. In the case of our example, it is a system with six axes with the names X, Y, Z, C, A and V. The table then shows the data on the current position (see 2.3.1 Position display), number and value of the tool offset (Tool column), about the offset value of the coordinate system (Off.Gxx column) and about the offset value generated by the G92 function (Off.G92 column). More about the meaning of individual values is given in chapter *3.1 Coordinate systems*. The last column is the status InP (In Position), which signals the correspondence of the required position with the actual position within the tolerance set in the configuration. This value changes while the machine is running. It should always be Yes at rest. If not, the system reports an error.

The standard interface has been designed for up to five displayed axes. If the system has more than five axes, you can click the mouse button in the box displaying the axis name to bring up a menu that offers the names of all the axes in the system. You select which axis you want to display in the given row.



Axes	In program	Tool 370	Off.G53	Off.G92	InP	Mode
X	107.868	0.000	0.000	0.000	Yes	Normal
Y	159.914	0.000	0.000	0.000	Yes	30
Z	100.000	0.000	0.000	0.000	Yes	
C	-0.000	0.000	0.000	0.000	Yes	
A	-0.121	0.000	0.000	0.000	Yes	
Radiu		0.000				Step
Spindl	0	0	1.00			Duration
Feed	0	0	1.00			0:00:00

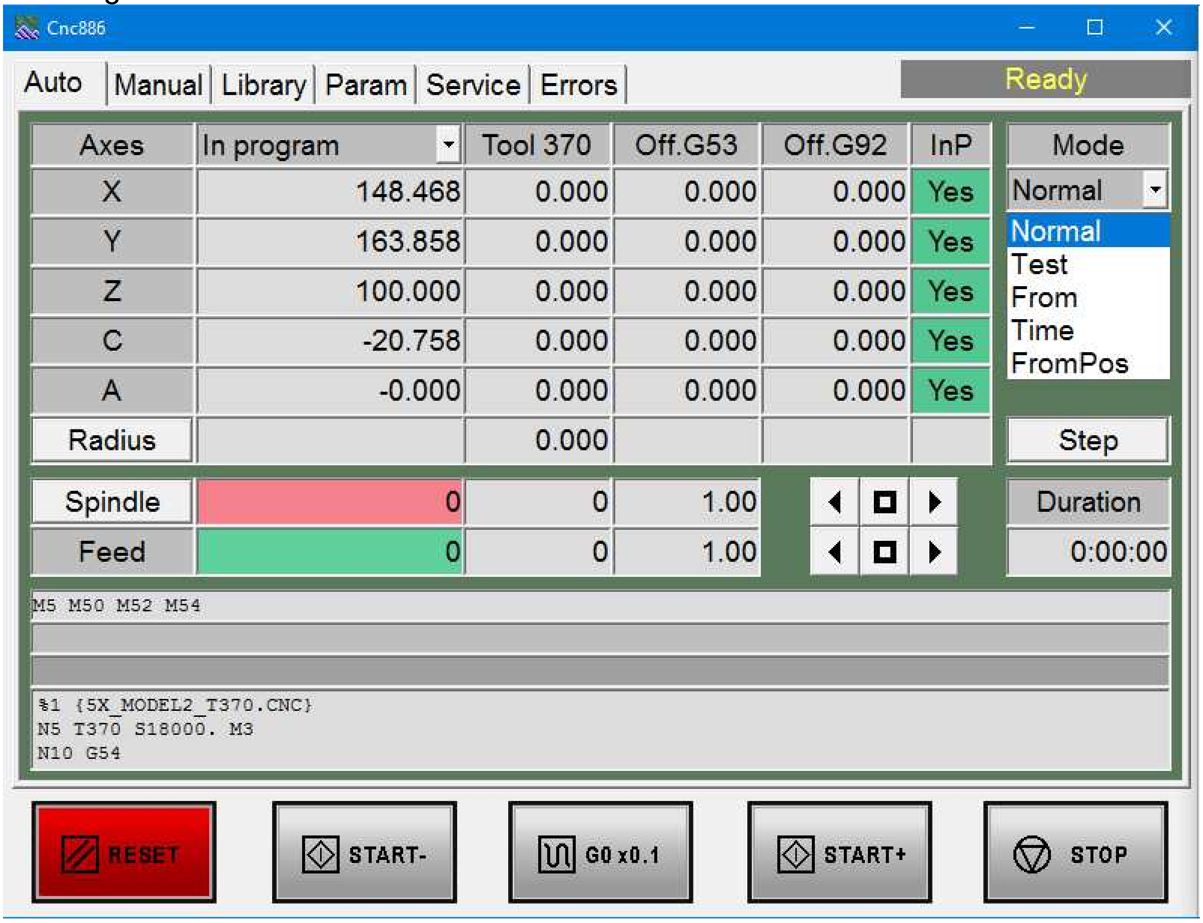
```
M5 M50 M52 M54
```

```
%1 {5X_MODEL2_T370.CNC}
N5 T370 S18000. M3
N10 G54
```

The last row of the table is the **Length** row used to display the radius correction, resp. length correction. Clicking Length toggles between displaying the two corrections. The **Length** inscription changes to **Radius** when the radius compensation is displayed.

Mode

To the right of the axis table is a field titled Mode. Below it is a drop-down menu for selecting a mode.



Axes	In program	Tool 370	Off.G53	Off.G92	InP
X	148.468	0.000	0.000	0.000	Yes
Y	163.858	0.000	0.000	0.000	Yes
Z	100.000	0.000	0.000	0.000	Yes
C	-20.758	0.000	0.000	0.000	Yes
A	-0.000	0.000	0.000	0.000	Yes
Radius		0.000			
Spindle	0	0	1.00		
Feed	0	0	1.00		

Mode: Normal

Normal
Test
From
Time
FromPos

Step

Duration: 0:00:00

```
M5 M50 M52 M54
```

```
%1 {5X_MODEL2_T370.CNC}
N5 T370 S18000. M3
N10 G54
```

Normal mode

If Normal is active, the entire program will be run after pressing **START+**.

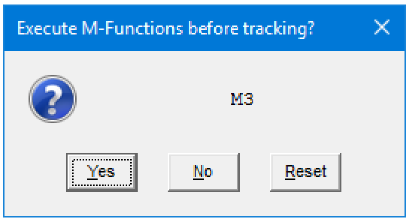
Test mode

If the Test is active, the entire program is checked after pressing **START+**. The system processes the entire program with no axis movement or M functions. However, all corrections, work offsets, etc. are considered, so that if no program error has been detected, it will not occur even during actual operation. The result is also a verification that the limits of axis movement are not exceeded and the size of the resulting workpiece is measured, so that it is possible to determine the size of the required blank. These results are listed in the report in the Errors tab.

From mode

In From line mode, the program is tested up to the line whose number is entered in the field below the mode. On this line, the program test stops, and the following lines are already executed in normal mode, when the axes move, and the M functions are executed. For the correct transition between test and normal mode, the system offers to perform all M functions, active according to the program. Some machines, from the logic of their function, require that the functions be started before approaching the starting point, some again after approaching the starting point. The choice is made by the manufacturer and therefore the menu dialog may appear at a different stage than described below.

The dialog offers a list of all M functions.



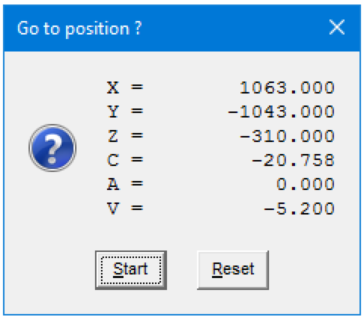
Execute M-Functions before tracking?

M3

There are three options to choose from:

- Yes: Execute the offered function immediately
- No: Functions are not performed.
- Reset: End of the program.

If we answer No, we still have the option to run the function manually by some other method, for example from a tab *2.4.3 Manual Functions tab.*



Go to position ?

```
X =     1063.000
Y =    -1043.000
Z =     -310.000
C =      -20.758
A =        0.000
V =       -5.200
```

After responding to the dialog with the function menu, the system offers another action, which is to approach the starting point. The coordinates are absolute.

Selecting **Reset** will end the program without moving.

After pressing **Start** (or START+), the movement is performed from the current position to the position at the beginning of the line from which we entered the execution of the program. The movement is performed in the order of the axes specified by the manufacturer. During the entire movement, the control system is in the **Stopped** state, in which it is possible to perform several operations, such as starting or stopping M functions. At the end of this, the machine stops in the state corresponding to the STOP button. Waiting for the START+ button to continue. Even at this point, i.e., after the end of the approach, the M functions can be started and stopped. After pressing START+ the program continues in Normal mode

Remarks:

1. The travel path depends on the machine type. For a five-axis milling machine, for example, the procedure is chosen in which the C and A axes are first rotated, then the approach is performed in the X and Y axes and only finally in Z. The approach is performed at the speed set by the manufacturer. If you are not sure of the trajectory along which the approach will take place, it is better to reduce the speed in the Feed line. If one of the cnc886 system extension plugins is running, the approach speed is user-adjustable and equal to the speed of manual movement.
2. During the program run, the exact match of the line number with the set value is tested. Therefore, if the program does not contain a line with the specified number, the program will run in Test mode.

Duration mode

If the Duration mode is active, after pressing START+ a simulated passage of the entire program is performed, including starting and braking ramps, deceleration in dynamically problematic places, etc. The system measures the consumed time. Depending on the performance of the processor, it runs about 30 times to 60 times faster than when the program is actually running. The resulting program run time, displayed in the Duration field, is a very good estimate of the program run time. It does not only count the execution times of M functions. If M functions are not an essential component of the program, even with many-hour programs, the estimation error is in seconds. The measurement result is also recorded in the Errors tab, where the total distance traveled and the average feed rate are also displayed.

Quick mode

Quick mode is enabled on some machines. In this mode, the axis movements are performed at the maximum possible speed along the trajectory given by the active program. M functions do not switch on. This mode is usually allowed for machines with two-dimensional movement.

FromPos mode

This mode is very similar to From Line mode. However, the starting position is not specified by the line number, but by the current machine position. The program starts in simulated mode and checks the position of the simulated passage with the actual position. If the values match (with the manufacturer-defined accuracy), the simulation is stopped and the transition to normal mode is prepared as in the case of From line mode.
This mode is only allowed on some machine types. It is preferably used mainly in biaxial systems.

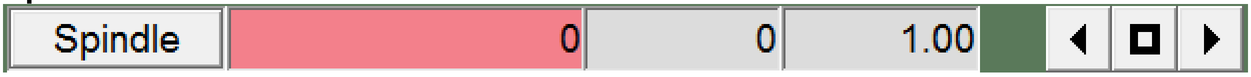
Step button

The **Step** button is used to switch between the normal program run mode and the step mode. When the step mode is selected, the button remains pressed. Each press changes the mode. The mode can be changed while the program is running if the program is paused.

Duration field

To the right of the axis table is a field titled Duration. Here, the execution time in hours, minutes and seconds is displayed during program run. The information remains displayed even after the end of the program and gives information about the execution time of the last running program.

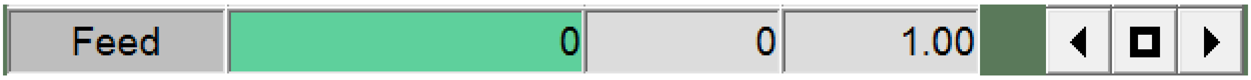
Spindle field



The line denoted by the heading Spindle is used to monitor and adjust the spindle operation. For machines that do not have a spindle, this line is either blank or is used to control another type of device.

The current speed in rpm is displayed in the red field. A positive speed value corresponds to function M03 - spindle forward, a negative value corresponds to function M04 - spindle reverse. The field on the right shows the speed prescribed by the program. The rest of the line is used to correct the speed. The left arrow decreases the speed in 10% (0.1) steps, the right arrow increases the speed in 10% (0.1) steps. The middle button resets the correction to 100% (1). The current value of the correction is displayed in the field to the left of the buttons. The correction can take values from 50% (0.50) to 150% (1.50).
If the machine has more spindles, it is possible to cycle through the display of individual spindles by clicking on the Spindle field. At the same time, the field with the display of the speed correction changes, which is individual for each spindle.

Feed field



Feed 0 0 1.00

The line denoted by the Feed heading is used to monitor and adjust the feed speed. The current feed in mm/min is displayed in the green field. The field on the right shows the feed rate prescribed by the program. The rest of the line is used to correct the feed. The left arrow decreases the feed in 10% (0.1) increments, the right arrow increases the feed in 10% increments. After reducing the speed below 10% (0.1), the buttons change the value by 1% (0.01). The middle button resets the correction to 100% (1). The current value of the feed rate correction is displayed in the field to the left of the buttons. The correction can take values from 0% (0.00) to 150% (1.50). At 0%, the machine halts, but the program is running! Setting non-zero correction resumes motion.

Active M functions

Below the Feed line is a field in which the active M functions are displayed.

Active G functions

The next line shows the active G function while the program is running.

Active ECHO

The next line (blank in the example) shows the text displayed by the ECHO command during the run.

Program lines

The last field of the tab shows the part of the program that is currently being executed. The line currently being executed is displayed first. In Step mode, the finished line is displayed first after stopping, the line ready for the next step is displayed second.

2.3.1. Axes position

The position of the axis is displayed based on the actual value read from the actuator. Therefore, it can fluctuate even when the machine is at rest. The system allows you to choose how to display the position.

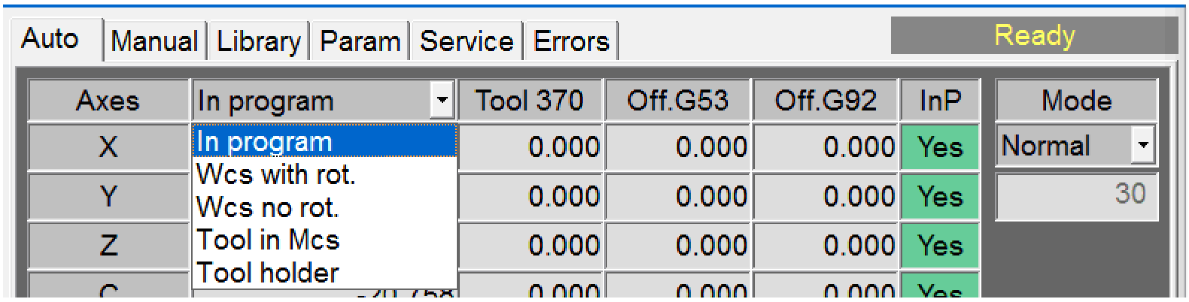
In the following text, we will use the designations "Mcs" for the machine coordinate system and "Wcs" for the workpiece coordinate system.

There are five modes

Mode	Displayed value
In program	The displayed value is the position of the "machining point" of the current tool in the workpiece coordinate system, including the offsets of the G5x and G92 functions. If tool radius offsets are applied, the "machining point" is the point on the tool circumference.
Wcs with rot.	The displayed value is the position of the tip of the current tool in the Wcs product coordinate system. Tool radius compensations are not applied.
Wcs not rot.	The displayed value is the position of the tip of the current tool in the Wcs product coordinate system, but without its rotation. Tool radius offsets are not applied.
Tool in Mcs	The displayed value is the position of the tip of the current tool in the Mcs machine coordinate system. Tool radius compensations are not applied.
Tool holder	The displayed value is the position of the tool holder in the Mcs machine coordinate system. Therefore, no tool corrections or offsets or rotations of the coordinate system are applied.

For more about the meaning of display modes, see. *3.1 Coordinate* systems.

We switch modes by expanding the drop-down list and selecting an option.



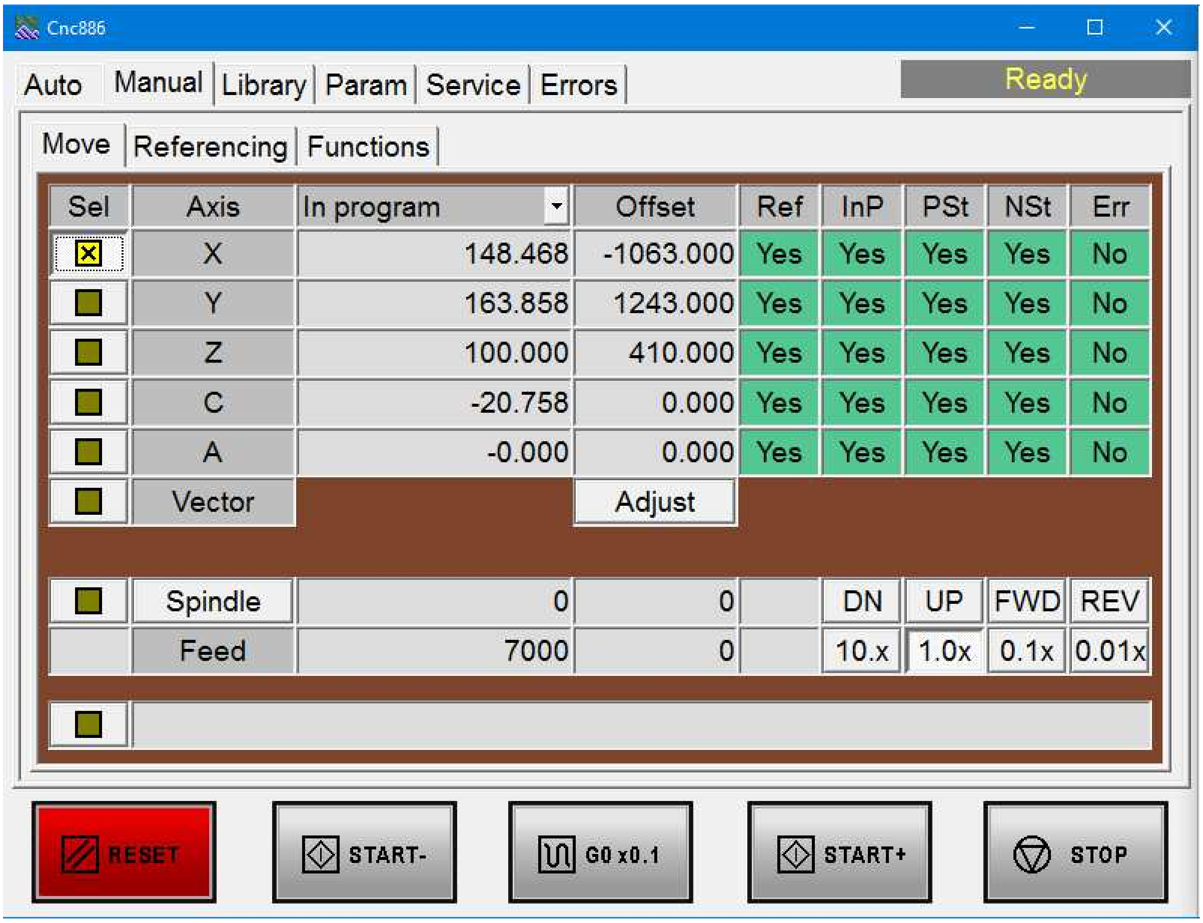
Axes	In program	Tool 370	Off.G53	Off.G92	InP
X		0.000	0.000	0.000	Yes
Y		0.000	0.000	0.000	Yes
Z		0.000	0.000	0.000	Yes

Mode
Normal
30

2.4. Manual tab

The Manual tab combines functions that are performed manually by the operator. In this mode, the system has a limited ability to control the correctness of the operations performed and the operator is responsible for their results.

2.4.1. Manual Move tab



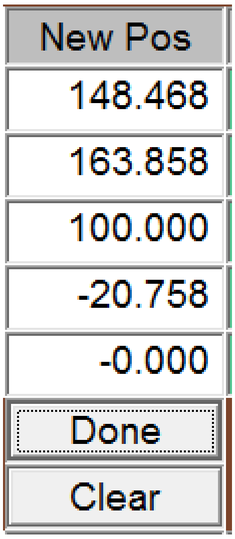
Sel	Axis	In program	Offset	Ref	InP	PSt	NSt	Err
☒	X	148.468	-1063.000	Yes	Yes	Yes	Yes	No
□	Y	163.858	1243.000	Yes	Yes	Yes	Yes	No
□	Z	100.000	410.000	Yes	Yes	Yes	Yes	No
□	C	-20.758	0.000	Yes	Yes	Yes	Yes	No
□	A	-0.000	0.000	Yes	Yes	Yes	Yes	No
□	Vector		Adjust					
□	Spindle	0	0		DN	UP	FWD	REV
	Feed	7000	0		10.x	1.0x	0.1x	0.01x
□								

The Manual-Move tab has a table at the top to display the status of the axes and an associated **Adjust** button (may vary on some systems).
Below the table is a line for manual spindle control and a line with four buttons for selecting the manual feed rate.
The last is the line for entering simple commands.

Axes table
The axes status table contains the following columns:

- **Sel** is a column of buttons that we use to select the axis to move. The inside of the square lights up for the selected axis, as we can see, for example, for the X axis in the picture. When you leave the tab, the selection is automatically reset. It is a certain safety measure that prevents the axes from moving by mistake. Sometimes it can be a little delayed, but it minimizes the likelihood of damage.
- **Axis** shows the axes names

- **The position display** is governed by the same rules as in the Auto tab. An exception is the display in the extreme positions of the axis movement range. If the lower end position is reached, the **Lower Limit** is displayed; if the upper end position is reached, the **Upper Limit** is displayed. Extreme positions are specified by machine constants in the configuration file.
- **Offset** shows the manually set offset of the Wcs coordinate system. To change this setting, press the **Adjust** button. When pressed, the Offset heading changes to **New Pos** and the Adjust button changes to **Done** and **Clear**. Column items become editable. You can enter the position in them (in the sense of the "Wcs with rot"). After pressing the **Done** button, the offset of the coordinate system is adjusted so that the system reports the current current position "Wcs with rot" equal to the value you entered. The **Clear** button resets the offset. The change with the **Adjust** button is only accessible when the machine is at a standstill. The offset value is stored in dedicated R parameters and stored on the computer disk. Therefore, it will be retained after the system is turned off and on. **Remark:** For most Cnc886 applications, there are add-ons that make the Wcs setup process easier. They also support setting the rotation of Wcs versus Mcs. These are plugins MILL3D, Speedy, PTV Control panel and more.



148.468
163.858
100.000
-20.758
-0.000

- **Ref** is an axis status flag indicating whether referencing has taken place
- **InP** is the status flag of the axis, indicating the status In Position as well as in the Auto tab
- **NSt and PSt** are the states of the limit sensors on the axes. In the Yes state, everything is OK, and the position is out of range of these sensors.
- **Err** is an error status flag of the servo amplifier.

Spindle line

The line starts with an indicator square. If the spindle is started manually, the square is lit. To the right of the **Spindle** text is the field for the current spindle speed and next to it for the set spindle speed. The settings are enabled by the **DN** and **UP** buttons. **DN** reduces the speed, **UP** increases the speed with a step determined by the machine configuration. The line ends with the **FWD** button, which starts the spindle movement forward, and the **REV** button, which starts the spindle movement backward. The direction of rotation can also be changed while the spindle is running.

The spindle is stopped with the **RESET** button on the button panel.

If the machine has more than one spindle, it is possible to cycle through the display of individual spindles by clicking on the Spindle field. At the same time, the field with the display of the required speed changes, which is individual for each spindle separately. The UP, DN, FWD, REV and RESET keys refer to the currently selected spindle.

If the machine has no spindle, it is empty or is used to control another device.

Feed line

The four buttons allow you to change the manual feed rate. In machine constants, each axis is assigned a speed for manual feed. This value is multiplied by 10x, 1.0x 0.1x or

0.01x, depending on the selection with the buttons. The selected feed rate and the current feed rate are displayed in the windows.

Field for manual Command Input
For manual approach to a specified position or for machining very simple shapes, the system allows you to enter a one-line command in the syntax of the CNC language. Touching the indicator square of the text field activates the field and the command can be edited. Select the field by pressing the indicator square in the same way as we select the individual axes. If the field is selected, the command can be executed by pressing START +. At that point, the system compiles a three-line program, the first line of which represents the header, the second line contains the command you entered, and the third line contains the M30 function for terminating the program. This program automatically activates and starts.

Vector
For some machines, such as five-axis milling machines, the **Vector** button can be activated in the configuration file. It enables manual movement in the tool axis and thus exits from the workpiece when the program is interrupted, etc. The item is selected in the same way as movement in the individual axis and the movement is started with the **START+** buttons (in the direction of the tool from the product) resp. **START**- (in the direction of the tool into the product). The calculation of the direction is firmly tied to the mechanical construction of the machine and is performed by the system.

Button bar
When you open the **Manual Move** tab, some buttons on the panel have slightly different functions:

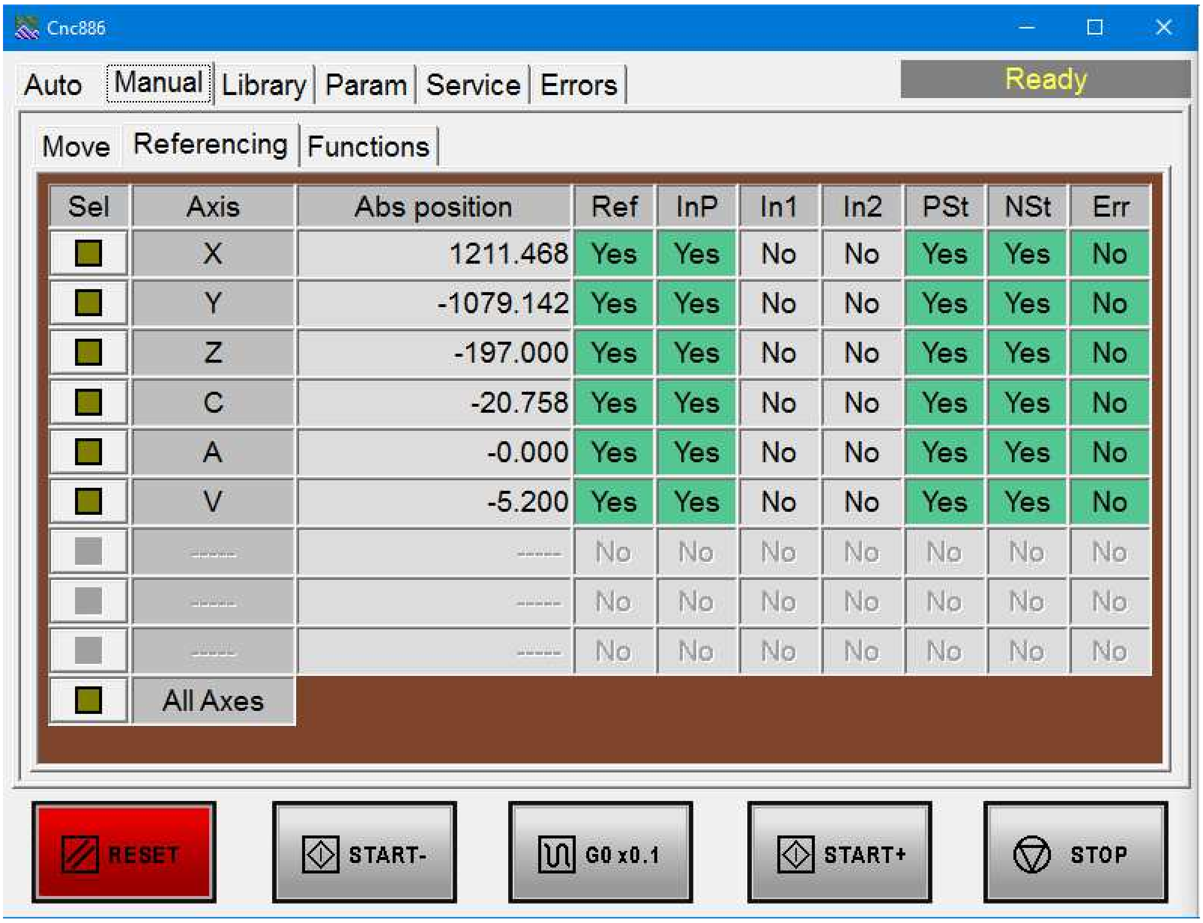
START- If one of the axes is selected (the indicator square is lit next to it), by pressing **START-** we move backwards in the given axis. Release the button to end the movement. The movement can also be performed with the spindle lowered.

START+ If one of the axes is selected (the indicator square is lit next to it), by pressing **START+** we move forward in the given axis. The movement is terminated by releasing the button. The movement can also be performed with the spindle lowered. **START+** is also used to run a one-line command if it is selected. You cannot start program execution on this tab.

2.4.1.1. Limit positions

If the axes are referenced, the system will not allow manual movement beyond the limits set by the machine constants. However, if the axis is not referenced, the system cannot perform any limit position monitoring. In such a state, the operator can then reach the emergency limit switches, which results in an error state.

2.4.2. Manual Referencing tab



Sel	Axis	Abs position	Ref	InP	In1	In2	PSt	NSt	Err
■	X	1211.468	Yes	Yes	No	No	Yes	Yes	No
■	Y	-1079.142	Yes	Yes	No	No	Yes	Yes	No
■	Z	-197.000	Yes	Yes	No	No	Yes	Yes	No
■	C	-20.758	Yes	Yes	No	No	Yes	Yes	No
■	A	-0.000	Yes	Yes	No	No	Yes	Yes	No
■	V	-5.200	Yes	Yes	No	No	Yes	Yes	No
□	------	------	No	No	No	No	No	No	No
□	------	------	No	No	No	No	No	No	No
□	------	------	No	No	No	No	No	No	No
■	All Axes								

The Manual-Referencing tab contains a table for displaying the status of the axes.

Axes table
The axes table contains the following columns:

- **Sel** is a column of buttons that we use to select the axis to reference. The inside of the square lights up for the selected axis, as we can see, for example, for the X axis in the picture. When you leave the bookmark, the selection is automatically reset. It is a certain safety measure that prevents the axes from moving by mistake. Sometimes it can be a little delayed, but it minimizes the likelihood of damage.
- **Axis** shows the axes names
- **Abs position** displays the position of the axis in the sense of "Tool holder". The mode cannot be switched in this tab.
- **Ref** is an axis status flag indicating whether referencing has taken place
- **InP** is the status flag of the axis, indicating the status In Position as well as in the Auto tab
- **In1 and In2** are the states of the sensors used to indicate the reference position. Settings of the servo amplifier and referencing algorithm determine which of the sensors is used and how.

- **NSt and PSt** are the states of the limit sensors on the axes. In the Yes state, everything is OK and the position is out of range of these sensors.
- **Err** is an error status flag of the servo amplifier.

All axes

To make work easier, the **All axes** item is available. Selecting it and pressing **START+** will cause all axes to be referenced in the order specified by the manufacturer.

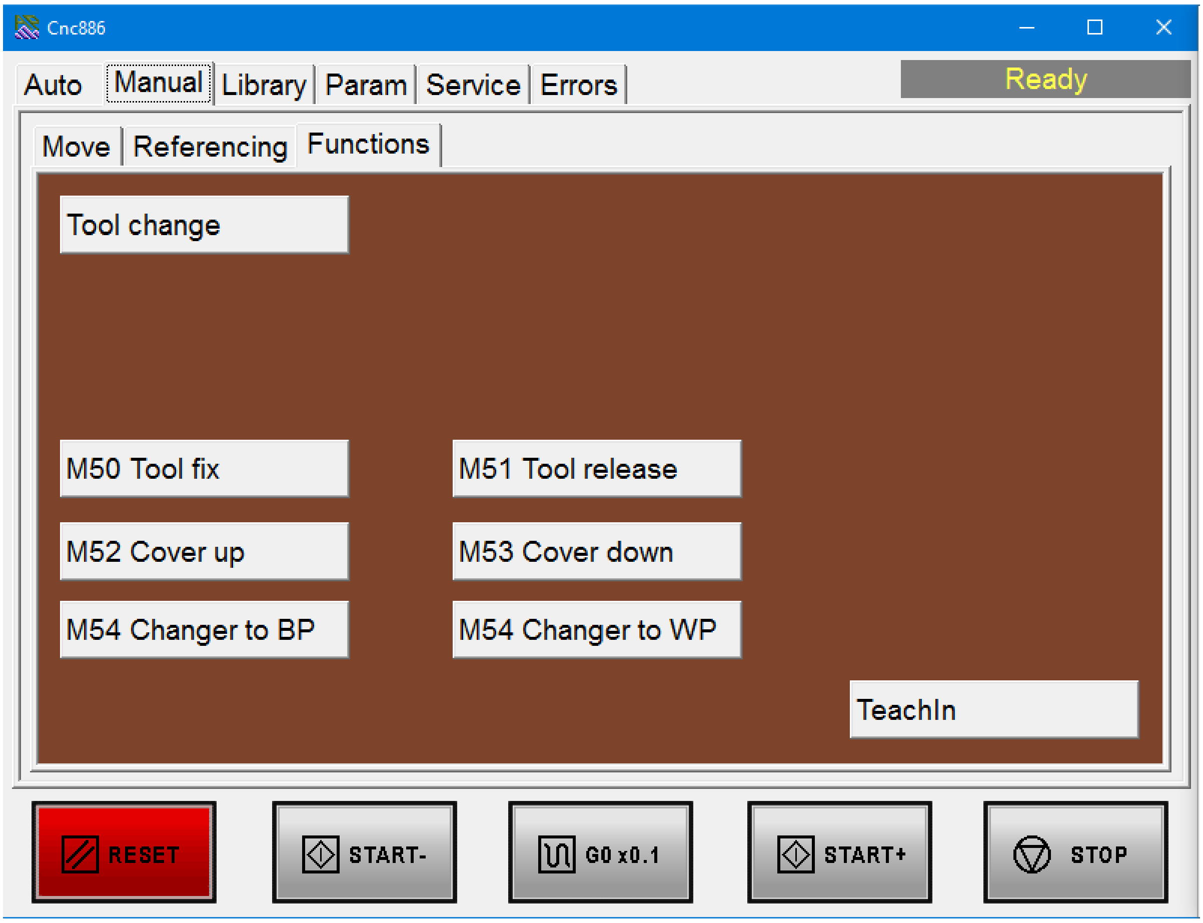
Button bar

When you open the Manual-Referencing tab, some buttons on the panel have slightly different functions:

START+ If one of the axes is selected (the indicator square is lit next to it), press **START+** to start the referencing process.

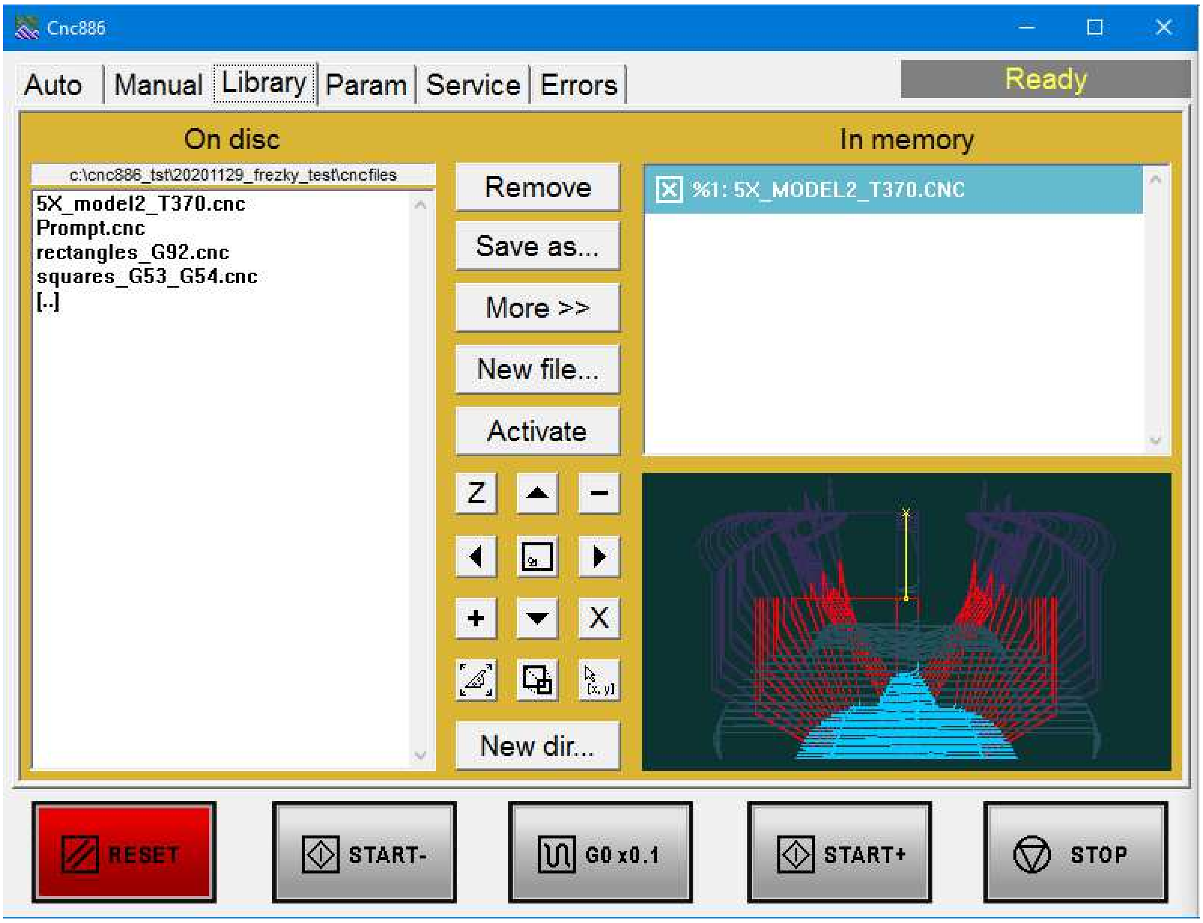
RESET Interrupts the referencing process. When the program is running or the spindle is started manually, it stops working.

2.4.3. Manual Functions tab



The content of this tab depends entirely on the specific machine. There are buttons for various functions. Their number and the system's response to them is variable.

2.5. Libray tab



c:\cnc886_tst\20201129_frezky_test\cncfiles

5X_model2_T370.cnc
Prompt.cnc
rectangles_G92.cnc
squares_G53_G54.cnc
[..]

☒ %1: 5X_MODEL2_T370.CNC

The **Library** tab is used to load programs from a disk or other source into memory, save to disk, delete from memory and disk, select a program to run, etc. The tab contains a window with a list of files with programs on the disk titled **On disc**, a window with a list of programs in memory titled **In memory** and a window with a preview of the program selected to run (see below). Then there are the buttons **Remove**, **Save as…**, **More >>**, **New file...**, **Activate** and **New dir...** Some buttons change their function and name depending on whether the program selection in memory or on the disc is active. All operations can only be performed when the machine is at a standstill.

On disc
This is a list of program files on disc in the directory shown in the list header. The default directory is determined by machine constants. List is filtered to files with .cnc extension. The directory structure can be browsed. Clicking on the name of a file or directory selects it (its name is displayed in white on a gray background) and is ready for action according to the buttons.
You can use the cursor keys on the keyboard to move the cursor inside the window. When browsing the directory structure, move the cursor to the selected directory and enter it with the ENTER key. To move one level up in the directory tree, use the item "**[..]**" located before front of the first subdirectory in the list.

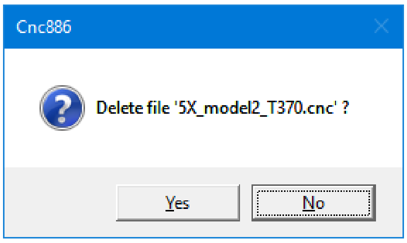
To quickly find a file, just press the first letter of the file name on the keyboard. The cursor automatically moves to the first file with the given initial character.

In memory

This list contains programs loaded into memory from the disc or the serial line. The loaded program shows its number after the % character and optional name. Clicking on a list item selects it for further actions.

Remove/Delete

This button label and function depends on whether program on disc or in memory is selected. For program in memory the label is **Remove** and allows you to remove selected program immediately. For program on disc the label is **Delete** and allows you to delete the file after confirmation. Directory on disc can also be deleted, but only if it is empty.

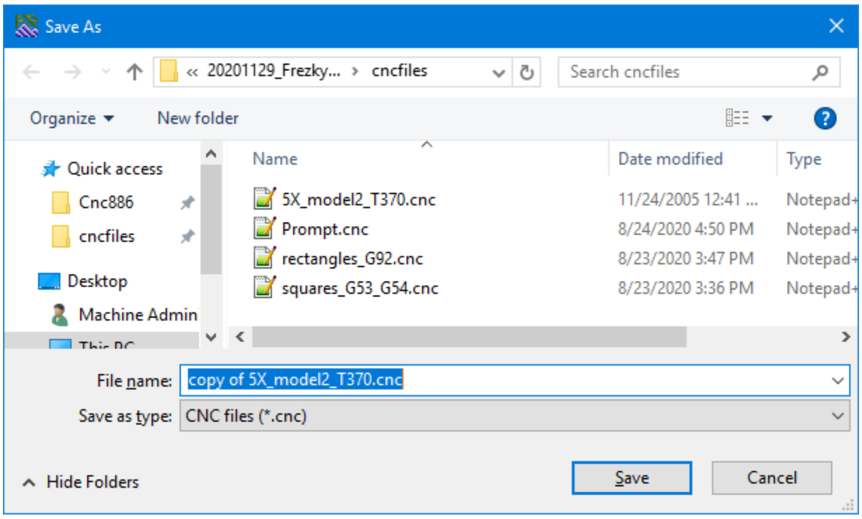


Cnc886

Delete file '5X_model2_T370.cnc' ?

Save as…

The Save as… button allows you to save the selected program to disc under the specified name. Programs loaded in memory are stored as they were loaded; programs located on disc are copied. In both cases, a standard dialog box appears asking you to enter a file name with .cnc extension. If you change the extension, the file will not be visible in the On Disc window! If you enter a file name that already exists, the system will ask if you want to overwrite it. If you give a negative answer, the action will be aborted.



Save As

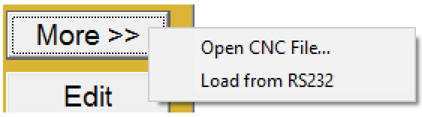
Name	Date modified	Type
5X_model2_T370.cnc	11/24/2005 12:41 ...	Notepad+
Prompt.cnc	8/24/2020 4:50 PM	Notepad+
rectangles_G92.cnc	8/23/2020 3:47 PM	Notepad+
squares_G53_G54.cnc	8/23/2020 3:36 PM	Notepad+

File name: copy of 5X_model2_T370.cnc

Save as type: CNC files (*.cnc)

More >>

The **More >>** button allows you to load the program into memory from another source. When "Load from RS232" is selected, the system immediately switches to the Load: state and expects characters to arrive via the serial line. When selecting "**Open CNC File...**", the file selection dialog appears. You have the entire space of the local computer and files available over the network to choose from. Once selected, the number of characters read is displayed in the status window. If you need to cancel the operation, use the **RESET** button. After successful loading, the program name will be displayed in the **In memory** window. At the same time, the program is set as active, i.e. ready to run. If you want to save the program loaded in this way to disk, use Save as…. The "Load from RS232" option was used mainly in the past and remains in the system for backwards compatibility. It may be removed or replaced by something more useful in the future.



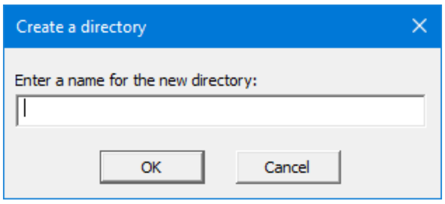
New file…/Edit

This button label and function depends on whether program on disc or in memory is selected. If a directory or in memory program is selected, the label is **New file…** and allows you to create a new file with the CNC program. For program on disc, the label is **Edit**. It allows you to edit the file with program. Press it to open the editor. More about the editor is in a chapter *2.9 Editor*.

Load/Activate

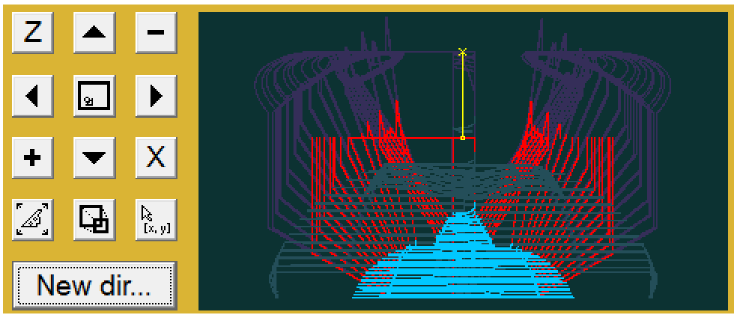
This button label and function depends on whether program on disc or in memory is selected. When program loaded in memory is selected, use **Activate** button to select it to run. Active program is marked with a checked square in the In memory field. If a file from disc is selected, this button has the label **Load**. It loads a file into memory, displays its number and name in the In memory list, and marks it as selected to run. If a program with the same number already exists in memory, the original program will be overwritten by the new one.

New dir…

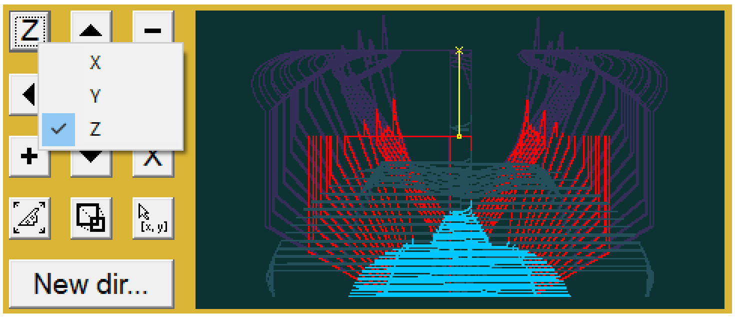


The **New dir...** button is used to create a subdirectory of currently active directory. A dialog box appears asking for the name of the new directory. The next step is intuitive and will not surprise any Windows user. Cases of incorrectly entered names, attempts to create an existing directory, etc. are handled.

Preview window

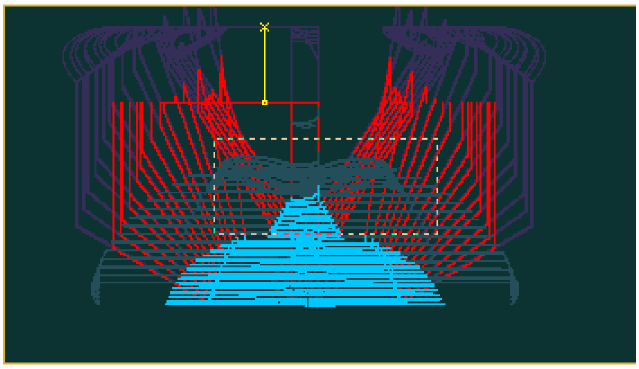


The preview window is used to display the machine path. This function is executed by clicking in the window area. A colored pattern appears on a black background, showing the machine's trajectory in the selected axes. This function is intended primarily for orientation in technological programs. When displayed, the control system runs the entire program as in TEST mode. (However, it does not check that the limit positions are exceeded.) The program run time depends on its length and on the selected value of the interpolation step for the test, which is defined by the configuration file.

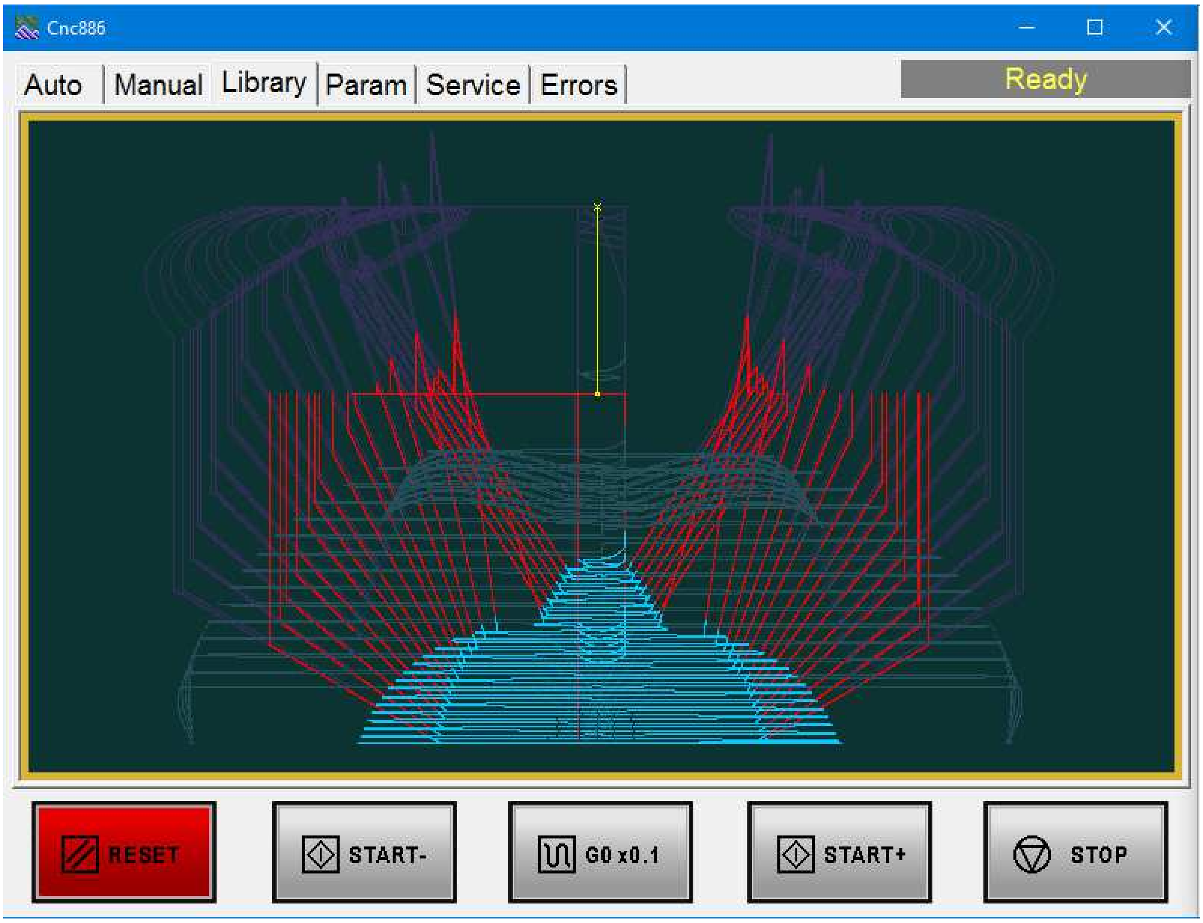


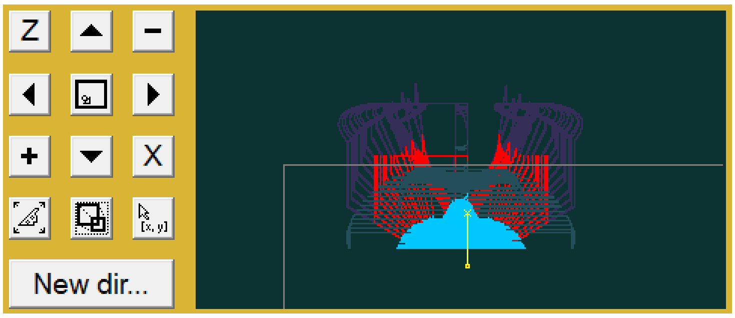
To the left of the window is a set of buttons used for Zoom + and − arrows for display shifts and axis selection (upper left for vertical axis and lower right for horizontal axis)

The button in the middle sets the default display so that the available machine area in the selected axes fits in the window. Axes allows you to select the button in the upper left corner (vertical axis) and the lower right corner (horizontal axis). The button optimizes the display scale so that the entire product fits in the preview window.
During the program run, the current status of the tool position is displayed schematically. You cannot use the display adjustment or axis selection buttons while the program is running.
You can also adjust the zoom with the mouse by selecting the rectangle to display. Left-click, drag the mouse and release. The selected area is displayed in the preview window.



There is also a Enlarge button . It enlarges the preview to fill the application window. Click in the preview area to return to the original view.



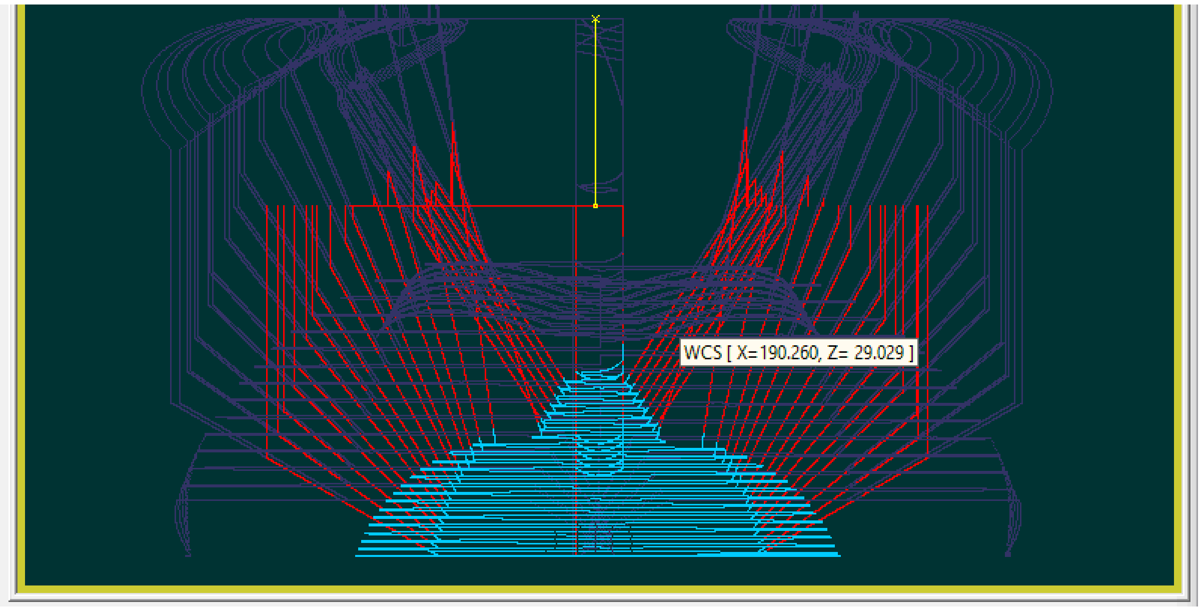


For greater clarity, the trajectories performed at rapid traverse are differentiated in color, and the trajectories of the tool tip and the trajectories of the point determining the limit positions are also differentiated. In the case of the figure, the workpiece is marked in cyan. However, because it is machined by a five-axis system with a tool of a certain length, the paths of the tool holder, and thus also the paths of the machine, are relatively complicated. However, these paths ultimately determine whether the program can be executed without exceeding the limit positions of the individual axes.

A similar situation where the tool tip paths do not correspond to the machine trajectories is the use of radius corrections.

The previous figure shows the inappropriate setting of the origin of the machining coordinate system, which leads to the limit positions being exceeded. In addition to the paths, the preview field also shows the contour of the available machine surface in the X and Z axes. The product itself is inside the rectangle, but the machine trajectories cross the contour.

Last button [x, y] toggles cursor position display. When the button is pressed, moving cursor in preview area displays its position in MCS. Option to display WCS coordinates is provided by plugins. This allows you to estimate placement and distances within the program.



WCS [X=190.260, Z= 29.029]

Drag and drop

The "drag and drop" function is supported in the Cnc886 system. When you drag a CNC code file from another window (such as Explorer) to the "Library" window, it is automatically loaded and activated.

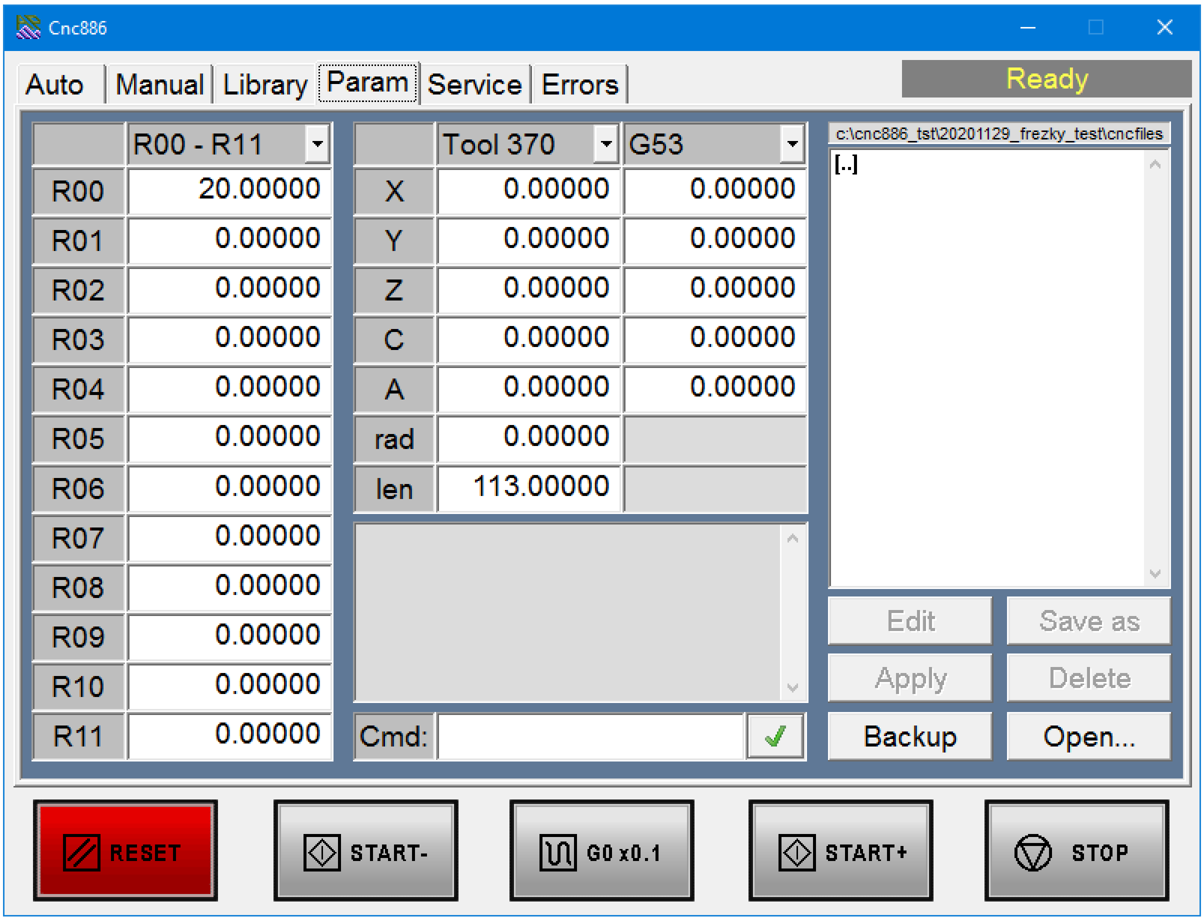
2.5.1. Automatically loaded programs

A program (or programs) can be specified in the system configuration files to be loaded into memory at Cnc886 startup. These are then displayed in the **In memory** list as if they were loaded with the Load button. It may be set in the configuration that programs that are automatically loaded cannot be removed.

This option is particularly advantageous if the automatically loaded program contains service subroutines that address the machine-dependent specialties to which the user programs refer. These can be, for example, subroutines that support tool exchange.

A program with the number %0 plays a special role. It can contain implementations of machine cycles determined by functions G200 to G399. The last activated function is then called by function G79. (see *4.11.9 Group 9: Machine cycles G200 - G399, G79*).

2.6. Param tab



	R00 - R11
R00	20.00000
R01	0.00000
R02	0.00000
R03	0.00000
R04	0.00000
R05	0.00000
R06	0.00000
R07	0.00000
R08	0.00000
R09	0.00000
R10	0.00000
R11	0.00000

	Tool 370	G53
X	0.00000	0.00000
Y	0.00000	0.00000
Z	0.00000	0.00000
C	0.00000	0.00000
A	0.00000	0.00000
rad	0.00000	
len	113.00000	

Cmd: ✓

c:\cnc886_tst\20201129_frezky_test\cncfiles
[..]

The tab is used to set R parameters, tool offsets and values for functions G53 to G59. These parameters are discussed separately in chapter *4.4.1 R Parameters.* After leaving this tab, the system saves the settings of corrections and parameters to the hard drive and reads them again the next time the system is switched on, so that the system is in the same state. It often happens that the settings for different programs differ significantly. It is therefore possible to write the settings of selected parameters to a file and then load it. Parameters that are not mentioned in the file will remain unchanged. It is therefore possible to load several files in succession, thus combining their effect. Settings files have .par extension and have a very simple format. It is plain text file. Each line beginning with a percent sign (%) is a comment and is otherwise ignored. The other lines must contain commands of type

R<parameter number> = < expression> ;

or

TOOL_COR[<tool number>,<cor. name>] = < expression> ;

or

CS_G54[<axis name>] = <expression> ;
CS_G55[<axis name>] = < expression> ;
etc.

or

CLEAR_ALL_RPARAMETERS;

or be empty. The trailing semicolon is required. The expression is entered in the usual way using addition, subtraction, multiplication and division (signs +, -, *, /) including the possibility to use parentheses and selected mathematical functions (more in *4.5 Assignment command*). The elements of expressions are constants or values of parameters.

Special command CLEAR_ALL_RPARAMETERS resets all parameters to zero. This command can only appear in .par file and cannot be entered manually.

Example file:

```
% -------Parameter file example------
%
% Program parameters
%
  R3 = 182.4 ;  R4 = 15 ;
  R5 = (R4 + 3*R3)/25.4 ;
%
% Tool #1
%
  TOOL_COR[  1,X     ] = 12.85 ;
  TOOL_COR[  1,Y     ] = 50.00 ;
  TOOL_COR[  1,Z     ] = 50.00 ;
  TOOL_COR[  1,Length] = 180.00 ;
%
% END
```

The tab area consists of three fields with dropdowns for setting R parameters, tool offsets and offsets of functions G53 to G59. In the right part there are tools for managing files with parameters, i.e. the file selection window and the buttons **Edit**, **Save as…**, **Apply**, **Delete**, **Backup** and **Open**. In the middle at the bottom of the tab is a field labeled **Cmd:** for setting a parameter with a manually entered command and above it is a list with history of entered commands and their results.

R00-R11

Below the drop-down list is a list of twelve values of R parameters. Clicking on a value places the cursor in the given line and the value can be adjusted with the keyboard by entering the expression. The value is saved to the parameter by clicking in another field or by pressing Enter. A record of the change will appear in the history list. If the value was entered incorrectly, the record is marked with the error sign "??". Otherwise, it is marked OK.

Expanding the drop-down menu offers the selection of a different interval of displayed parameters. In this way, parameters in the range from R00 to R301 can be adjusted. Additional parameters can be set by a command in the field **Cmd:**

Tool 0

Below the drop-down list is a list of tool offset values in mm for the given tool. There is also radius compensation in the row **rad** and length correction in the row **len**.

You can choose another tool by expanding the drop-down list. There are a total of 1000 tools (T0 to T999)

G53
Below the drop-down list is a list of coordinate shift values for the given G function. We can choose from the functions G53, G54, G55, G56, G57, G58 and G59.

.par files
The list contains files with .par extension and directories on the disk in the directory shown in the window header. The default directory is determined by machine constants. Clicking on the name of a file or directory will select it, and it is ready for action according to the buttons below the window.

Edit
The Edit button allows you to edit the file on disk. If a file is selected in the window, the editor opens. More about the editor in a separate chapter *2.9 Editor*.

Save as…
The Save as… button allows you to save the selected parameter file to disk under new name. A dialog box appears asking you to enter a file name with a **.par** extension. If you change the extension, the file will not be visible in the window! If you enter a file name that already exists, the system will ask if you want to overwrite it. If you give a negative answer, the action will not take place.

Delete
The Delete button allows you to delete a file or directory from the disk. The system will ask for confirmation before performing the action. The directory can only be deleted if it is empty.

Apply
The Apply button loads the selected parameter file. Result of the operation is displayed in the history list.

Cmd:
Clicking in the white text field labeled Cmd: allows you to enter the command in the same format as in the .par file. This allows access to parameters that are not in the range of R0 to R301. In practice, however, this is rarely needed. The command is executed by pressing ENTER on the keyboard or by clicking green checkmark.

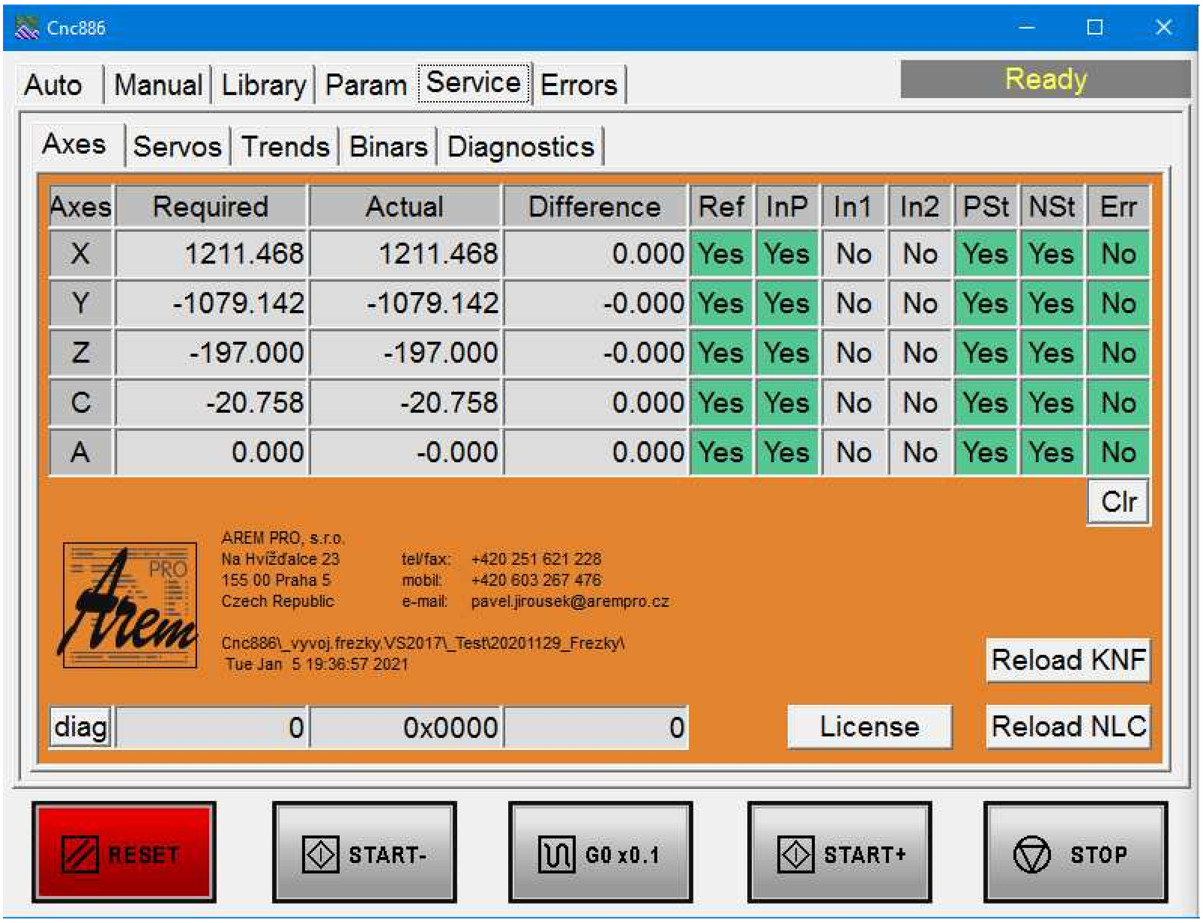
Open…
The Open... button is used to load the parameter file using the Windows system dialog for selecting the file.

Backup
The Backup button is used to create a file containing the current status of all parameters. Note that only non-zero parameters are written, all other parameters are reset to zero with first line being CLEAR_ALL_RPARAMETERS. The file has a text format in a form that can be used with the Apply command. The resulting file can be used as a backup of machine settings, for example, after tuning tool offsets, etc.
After pressing Backup the same dialog will appear as after Save as….

2.7. Service tab

The Service tab is intended for machine diagnostics and for solving special situations. In particular, the Service-Binary sub-tab, which allows the operator to access the individual bits of the outputs, allows manipulations that need to be performed with great care and forethought.

2.7.1. Service Axes tab



Axes	Required	Actual	Difference	Ref	InP	In1	In2	PSt	NSt	Err
X	1211.468	1211.468	0.000	Yes	Yes	No	No	Yes	Yes	No
Y	-1079.142	-1079.142	-0.000	Yes	Yes	No	No	Yes	Yes	No
Z	-197.000	-197.000	-0.000	Yes	Yes	No	No	Yes	Yes	No
C	-20.758	-20.758	0.000	Yes	Yes	No	No	Yes	Yes	No
A	0.000	-0.000	0.000	Yes	Yes	No	No	Yes	Yes	No

AREM PRO, s.r.o.
Na Hvížďalce 23
155 00 Praha 5
Czech Republic

tel/fax: +420 251 621 228
mobil: +420 603 267 476
e-mail: pavel.jirousek@arempro.cz

Cnc886_vyvoj.frezky.VS2017_Test\20201129_Frezky\
Tue Jan 5 19:36:57 2021

diag	0	0x0000	0

The Service-Axes tab contains a table displaying the status of the axes. Right below the table is **Clr** button.

Further down is the **diag** diagnostic line and the three buttons **License**, **Reload KNF** and **Reload NLC**.

Axes table

The axis status display table contains the following columns:

- **Axes** displays the names of axes
- **Required** shows the required axis position in Tool holder coordinates
- **Actual** shows the actual axis position in Tool holder coordinates
- **Difference** shows the difference between the required axis position and the actual axis position. This difference is calculated by the system. However, since the position value is read at a different time than the required value is generated, the

values can differ significantly, especially during rapid movement. Therefore, this information makes more sense at rest.

- **Ref** is an axis status flag indicating whether referencing has taken place
- **InP** is the status flag of the axis, indicating the status In Position as well as in the Auto tab
- **In1 and In2** are the states of the sensors used to indicate the reference position. Settings of the servo amplifier determine which of the sensors is used and how.
- **NSt and PSt** are the states of the end emergency sensors on the axes. In the Yes state, everything is OK and the position is out of range of these sensors.
- **Err** is an error status flag of the servo amplifier.

Clr

The button resets the error status of the amplifiers if the status of any of them is **Yes** in the column **Err**.

Diag

The line is used only for service purposes of the control program. The last box shows the total distance traveled in the program. The length of trajectory of the program is also evaluated in Test mode.

License

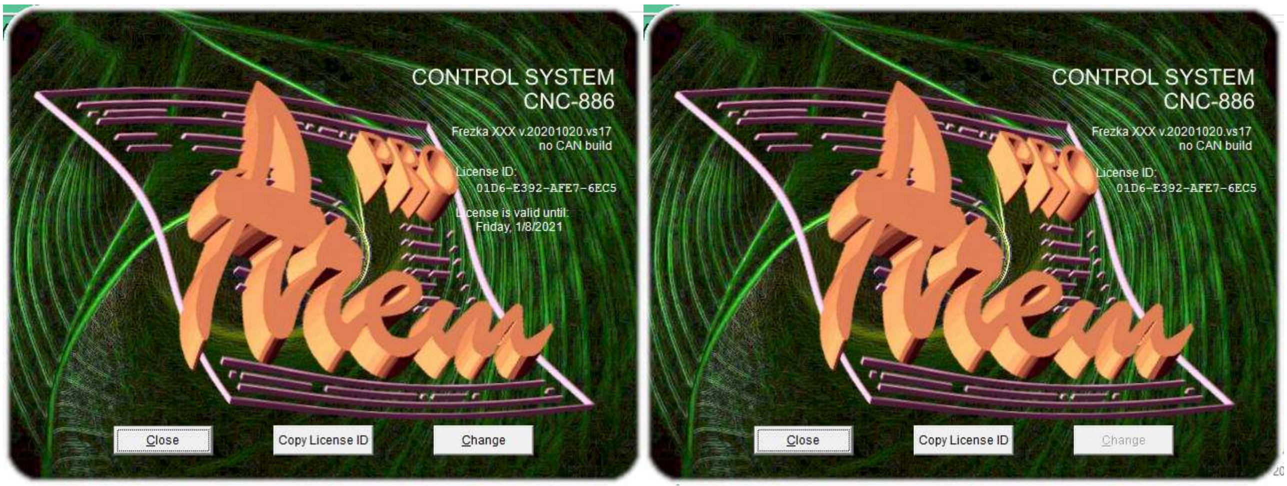
After pressing the button, information about the date until which the license of the control system is valid will appear. It is also used to enter the code that modifies the license.

Reload KNF
Reload NLC

These two buttons allow you to load modified parts of the machine configuration from disk files without having to restart Cnc886. Not all configuration parameters can be changed this way. Therefore, these functions should only be used by a service technician or the operator in consultation with him.

2.7.1.1. Checking and setting up the license

Checking and setting the license only applies to users with whom, for business reasons, an agreement has been concluded to limit the use of the system for a certain period of time.
After pressing the License button, a window will appear with information about the machine type, License ID and license expiration date. At the bottom of the window are the **Close**, **Copy License ID** and **Change.**

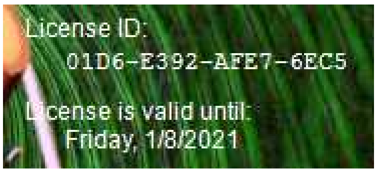


CONTROL SYSTEM
CNC-886
Frezka XXX v.20201020.vs17
no CAN build
License ID:
01D6-E392-AFE7-6EC5
License is valid until:
Friday, 1/8/2021

CONTROL SYSTEM
CNC-886
Frezka XXX v.20201020.vs17
no CAN build
License ID:
01D6-E392-AFE7-6EC5

Depending on the system settings, the license expiration information may be hidden, as shown in the figure on the right. In this case, you can display this information by pressing Ctrl + Shift + L. The Change button is only available when the license validity information is displayed.

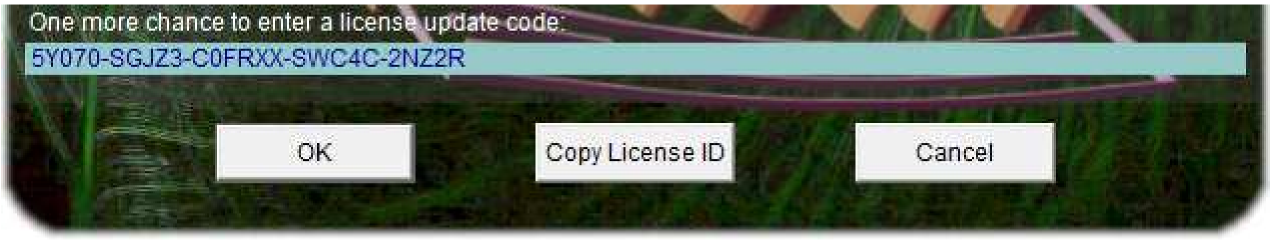
The detail shows the License ID and an example of the validity period. When communicating with the supplier about changing the license validity period, it is necessary to communicate the License ID and the current validity period setting.



License ID:
01D6-E392-AFE7-6EC5
License is valid until:
Friday, 1/8/2021

Because it is easy to make a mistake when writing down the License ID, the **Copy License ID** button is present. ID. By pressing it, the License ID will be transferred to the Clipboard, so with the following Ctrl + V we can transfer the License ID to an e-mail, text file, etc.

If the vendor sends you a license code to change the license expiration date, it will look like the example: 5Y070-SGJZ3-C0FRXX-SWC4C-2NZ2R

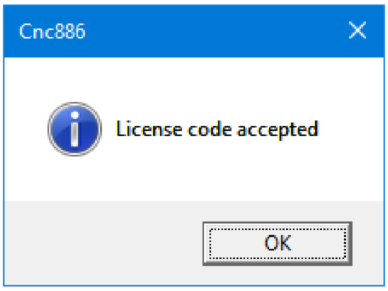


One more chance to enter a license update code:

5Y070-SGJZ3-C0FRXX-SWC4C-2NZ2R

After pressing the Change button, this license code must be entered in the prepared field. Again, you can use Ctrl+C and Ctrl+V

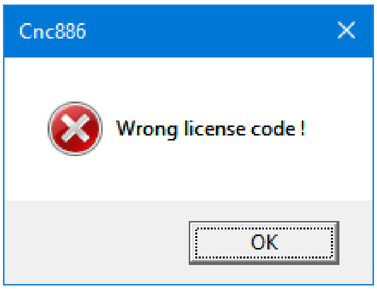
After pressing OK, the system confirms the license change



Cnc886

License code accepted

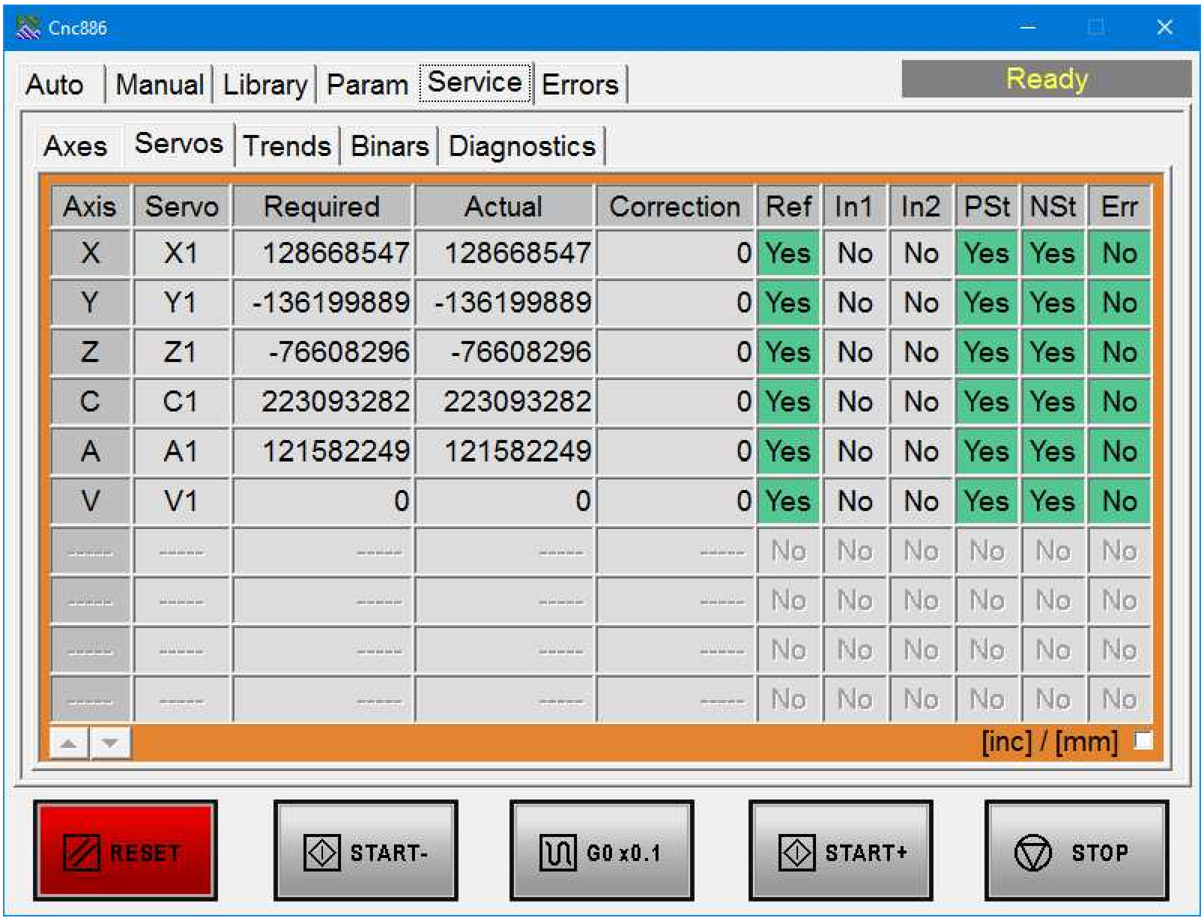
If the attempt is unsuccessful for any reason, it will appear



Cnc886

Wrong license code !

2.7.2. Service Servos tab



Axis	Servo	Required	Actual	Correction	Ref	In1	In2	PSt	NSt	Err
X	X1	128668547	128668547	0	Yes	No	No	Yes	Yes	No
Y	Y1	-136199889	-136199889	0	Yes	No	No	Yes	Yes	No
Z	Z1	-76608296	-76608296	0	Yes	No	No	Yes	Yes	No
C	C1	223093282	223093282	0	Yes	No	No	Yes	Yes	No
A	A1	121582249	121582249	0	Yes	No	No	Yes	Yes	No
V	V1	0	0	0	Yes	No	No	Yes	Yes	No
-----	-----	-----	-----	-----	No	No	No	No	No	No
-----	-----	-----	-----	-----	No	No	No	No	No	No
-----	-----	-----	-----	-----	No	No	No	No	No	No
-----	-----	-----	-----	-----	No	No	No	No	No	No

[inc] / [mm] □

The tab displays information about the position of the servos. The axes mentioned so far were virtual axes, which do not necessarily have to be realized with the same number and the same geometry of the actuators = physical axes. A common case is, for example, driving a portal by the simultaneous interaction of two servos.

For better diagnostics, it is possible to monitor the behavior of not only virtual axes, but also the status of individual servos. Their position can be displayed in integer units, which are sent to the servo system, or converted to physical units. The display is controlled by a check box.

[inc] / [mm] □

The **Axis** column contains the name of the virtual axis to which the servo belongs.

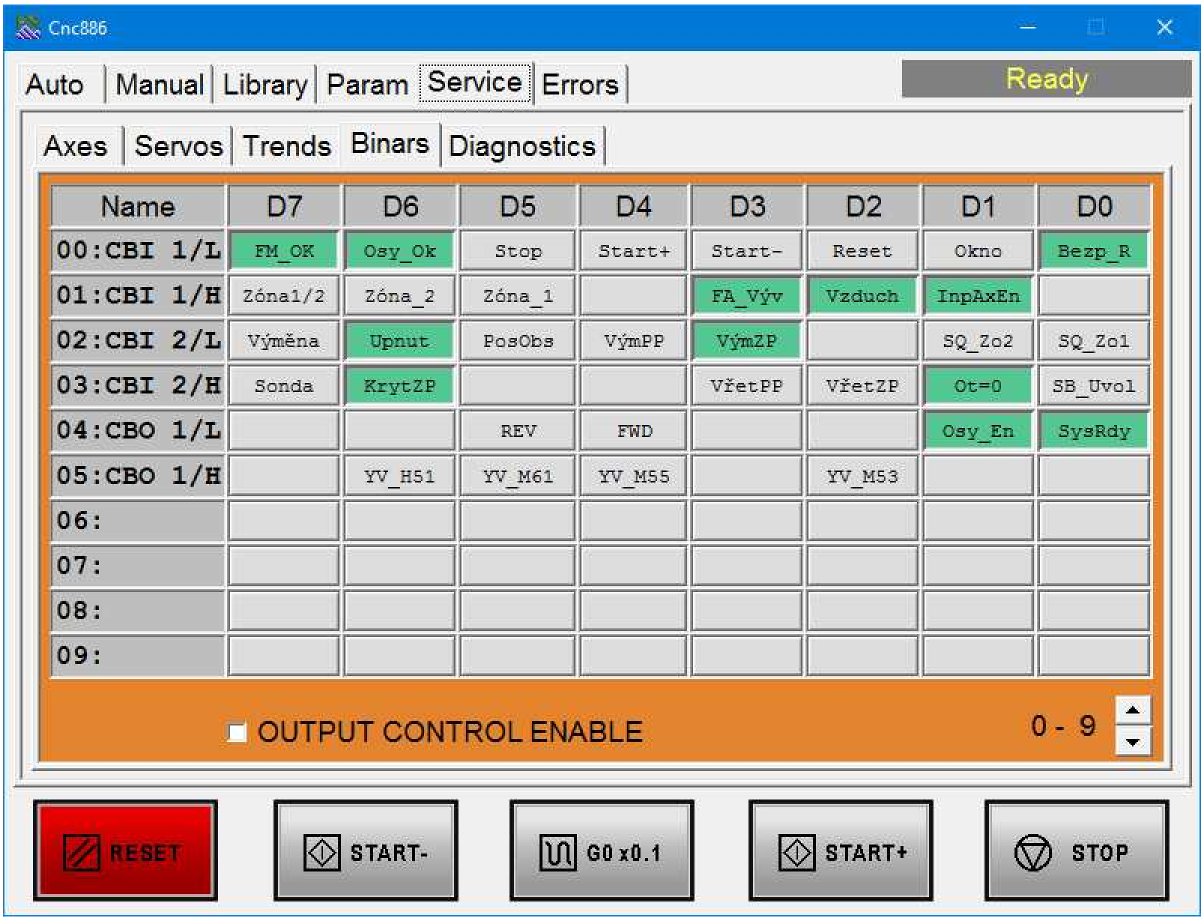
The **Servo** column is a list of servo names.

The **Required** shows the desired position of the servo in selected units.

Actual displays the current position of the servo in selected units.

Correction displays the correction calculated and applied by the control system to the physical axes positioning. This correction is applied to compensate for the non-linearity of the pitch of ball screws, gear racks, etc. to achieve the highest possible system accuracy. When calculating corrections, the system relies on values measured, for example, interferometrically.

2.7.3. Service Binars tab



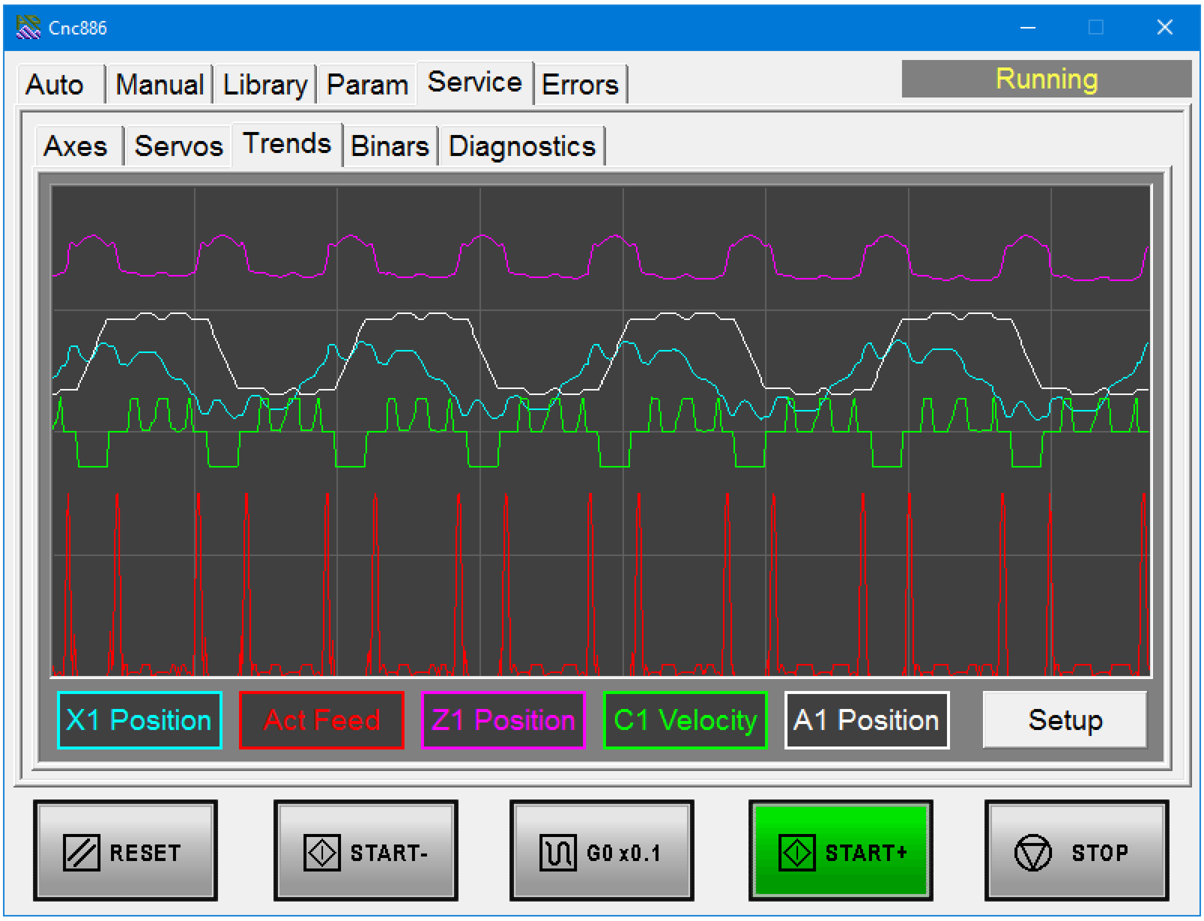
Name	D7	D6	D5	D4	D3	D2	D1	D0
00:CBI 1/L	FM_OK	Osy_Ok	Stop	Start+	Start-	Reset	Okno	Bezp_R
01:CBI 1/H	Zóna1/2	Zóna_2	Zóna_1		FA_Výv	Vzduch	InpAxEn	
02:CBI 2/L	Výměna	Upnut	PosObs	VýmPP	VýmZP		SQ_Zo2	SQ_Zo1
03:CBI 2/H	Sonda	KrytZP			VřetPP	VřetZP	Ot=0	SB_Uvol
04:CBO 1/L			REV	FWD			Osy_En	SysRdy
05:CBO 1/H		YV_H51	YV_M61	YV_M55		YV_M53		
06:								
07:								
08:								
09:								

□ OUTPUT CONTROL ENABLE   0 - 9

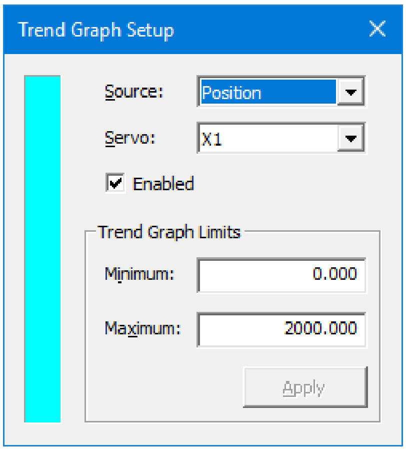
The Service-Binaries tab contains a table of binary inputs and outputs of the system. They are organized by eight bits into bytes. The meaning of the individual bits and thus the labels in their fields are completely dependent on the specific design of the machine.
The field for the individual bits indicates status 0 in gray, status 1 in green.
The output bits of the system can be changed manually by touching (clicking) on the field. Every touch means a change of value. To avoid changes by mistake, you must first activate the **OUTPUT CONTROL ENABLE** button. When you leave the tab, the button is automatically deactivated.
Note that some output cannot be changed because they are forced by Cnc886 to certain value.

2.7.4. Service Trends tab

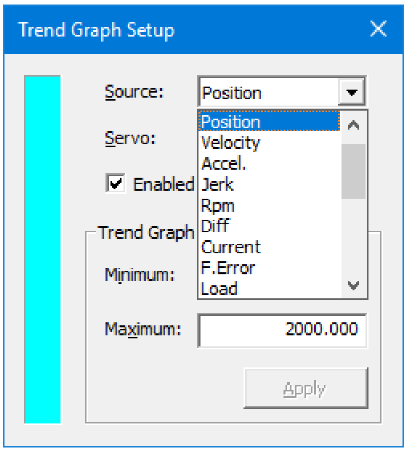
This tab allows you to monitor graphs of various system parameters.



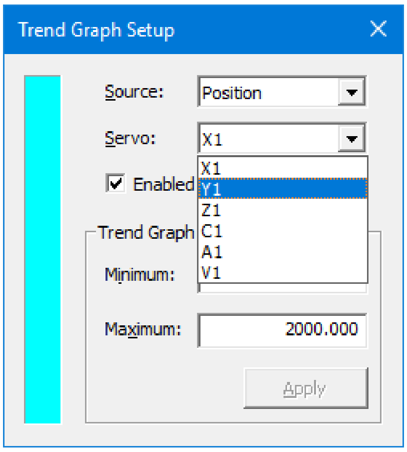
Click on the colored box below the graph to display the configuration dialog for setting the source and scale for the selected graph. Data are available for individual physical axes – servos and depending on the machine from other sources.



Source: Position
Servo: X1
☑ Enabled
Minimum: 0.000
Maximum: 2000.000

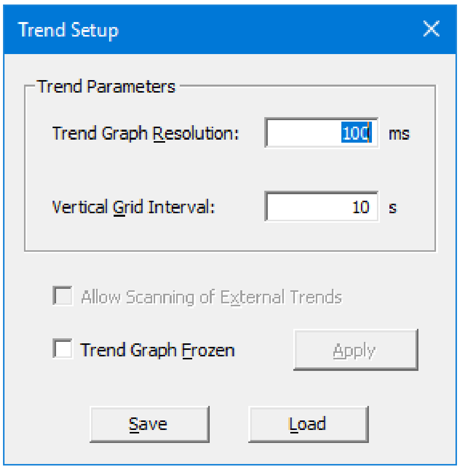


Source: Position
Servo:
☑ Enabled
Minimum:
Maximum: 2000.000



Source: Position
Servo: X1
☑ Enabled
Minimum:
Maximum: 2000.000

It is possible to monitor for example the position, speed, acceleration, jerk, servo speed, motor current, path deviation and total load in percent for each. Additional sources provide number of other diagnostic data.
You can turn the graph off or on again by checking Enable.



Trend Graph Resolution: 100 ms

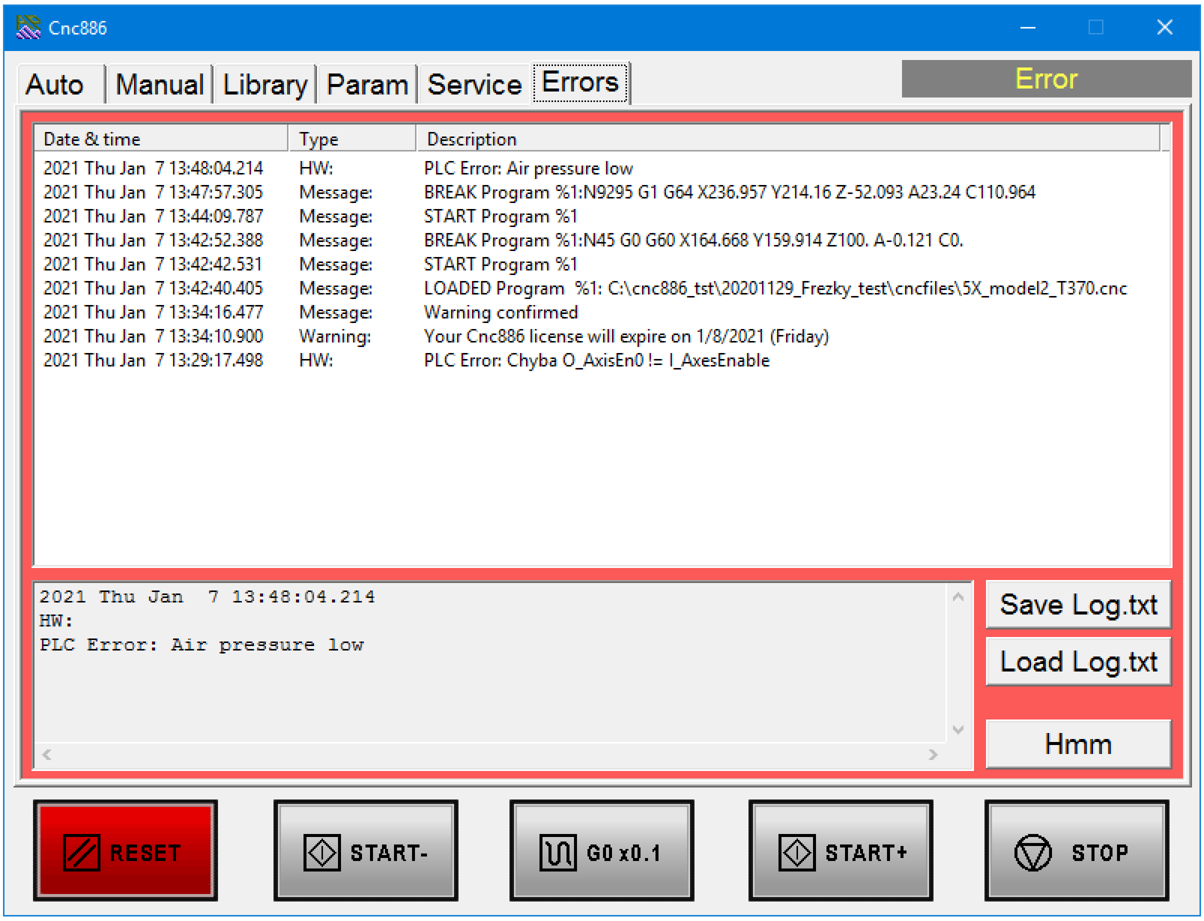
Vertical Grid Interval: 10 s

□ Allow Scanning of External Trends

□ Trend Graph Frozen

The **Setup** button allows you to change the scale on the horizontal axis of the graph and the density of the grid. If something interesting is visible on the graph and we do not want the image to disappear before we look at it properly, we can click on the graph to fix the image. It stops moving. Click again to get back to real time display.
We can also control the freezing and revitalization of the graph with a check mark in the settings dialog.
The settings of the entire Trends window can be saved to a file with the Save button and later restored with the Load button.

2.8. Errors tab



Date & time	Type	Description
2021 Thu Jan 7 13:48:04.214	HW:	PLC Error: Air pressure low
2021 Thu Jan 7 13:47:57.305	Message:	BREAK Program %1:N9295 G1 G64 X236.957 Y214.16 Z-52.093 A23.24 C110.964
2021 Thu Jan 7 13:44:09.787	Message:	START Program %1
2021 Thu Jan 7 13:42:52.388	Message:	BREAK Program %1:N45 G0 G60 X164.668 Y159.914 Z100. A-0.121 C0.
2021 Thu Jan 7 13:42:42.531	Message:	START Program %1
2021 Thu Jan 7 13:42:40.405	Message:	LOADED Program %1: C:\cnc886_tst\20201129_Frezky_test\cncfiles\5X_model2_T370.cnc
2021 Thu Jan 7 13:34:16.477	Message:	Warning confirmed
2021 Thu Jan 7 13:34:10.900	Warning:	Your Cnc886 license will expire on 1/8/2021 (Friday)
2021 Thu Jan 7 13:29:17.498	HW:	PLC Error: Chyba O_AxisEn0 != I_AxesEnable

```
2021 Thu Jan  7 13:48:04.214
HW:
PLC Error: Air pressure low
```

The Errors tab displays a record of error conditions and other events. The window at the top of the tab allows you to view the history of records since the last time the system was turned on. The data are provided with the date and time of the record, the type and specification of the record. In addition to error states, we can find records of loading a file into memory, starting a program, ending a program, or stopping it with an error or the RESET button. For such an event, there is also a record of which line the program was interrupted on. This allows us to resume the program from the line where the program was stopped using the **From mode**.

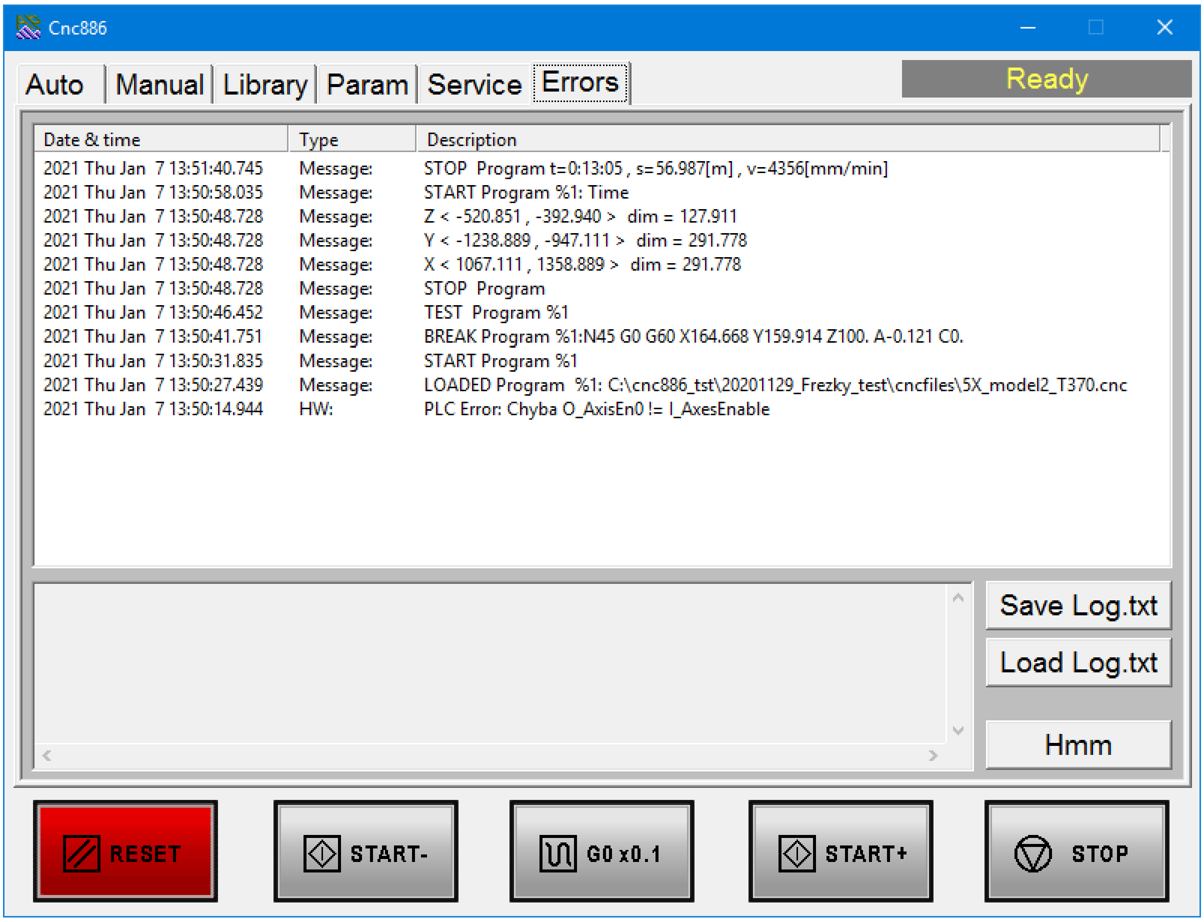
In the lower window, an error message is displayed if an error is active and the system is in the Error state. The area around the field is red.

Confirm the error status with the **Hmm** button. However, if the error persists, the acknowledgment will fail, and the error will reappear. The cause must be eliminated first. This applies to external errors, such as tripped circuit breaker or amplifier errors. Program errors will only reoccur by running the program again.

Errors and other messages displayed in this window are or are not saved to disk for service purposes, depending on the settings in the configuration file. Two buttons are available for operator use. **Save Log.txt** and **Load Log.txt**. As the name suggests, they allow you to save the contents of the message list to disk and reload them. Saving is done in a special file Log.txt, which is not related to the previously described file for service purposes. Saving and reloading can be used in cases where the operator wants to record information such as the last program started, the line when the program was interrupted, etc.

2.8.1. More information in the Errors tab

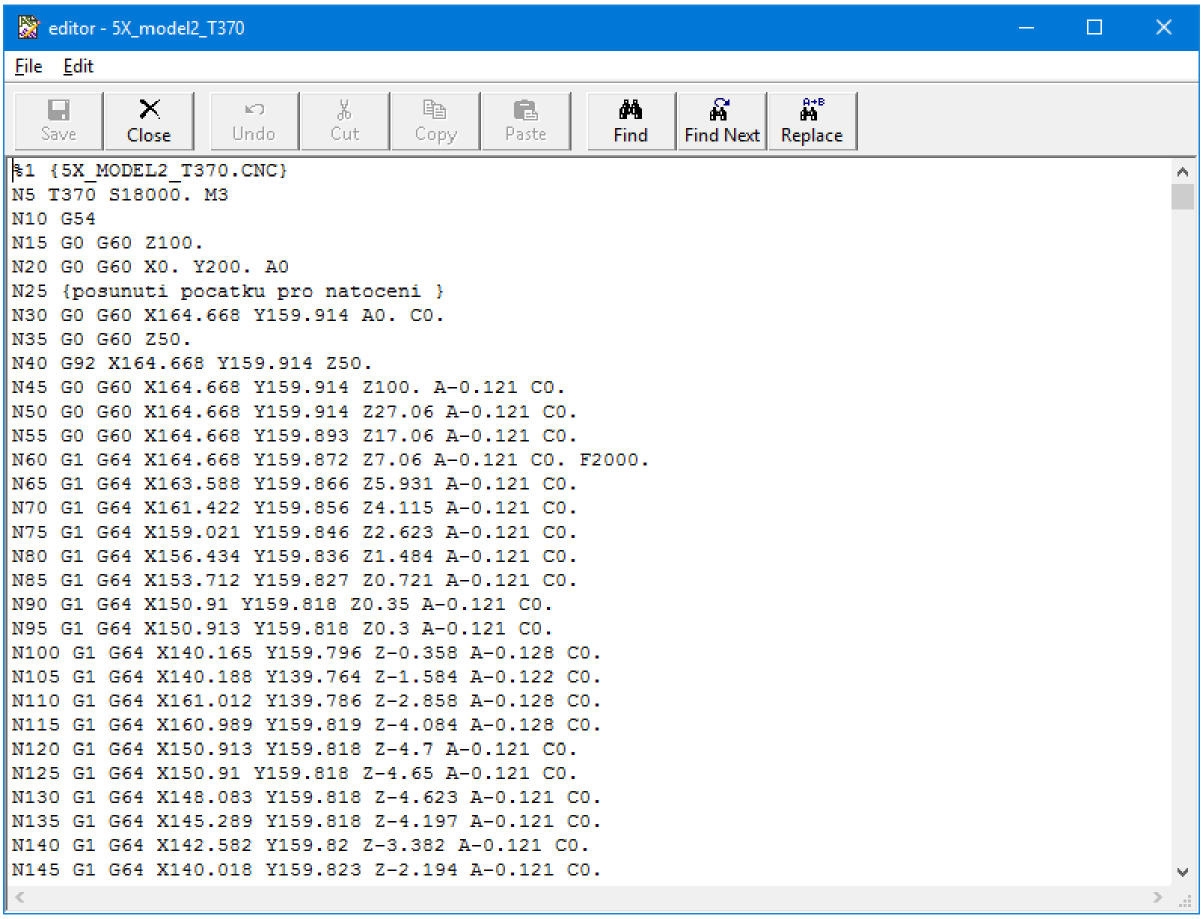
Other useful information is also stored in the records. For example, in the **Test** mode, the minimum and maximum values of the selected axes are monitored, and at the end, the dimension of the semi-product into which the planned product fits is listed. In the picture we see a statement that informs us that the dimensions of the product will be X=291.778 Y=291.778 Z=127.911.



Date & time	Type	Description
2021 Thu Jan 7 13:51:40.745	Message:	STOP Program t=0:13:05 , s=56.987[m] , v=4356[mm/min]
2021 Thu Jan 7 13:50:58.035	Message:	START Program %1: Time
2021 Thu Jan 7 13:50:48.728	Message:	Z < -520.851 , -392.940 > dim = 127.911
2021 Thu Jan 7 13:50:48.728	Message:	Y < -1238.889 , -947.111 > dim = 291.778
2021 Thu Jan 7 13:50:48.728	Message:	X < 1067.111 , 1358.889 > dim = 291.778
2021 Thu Jan 7 13:50:48.728	Message:	STOP Program
2021 Thu Jan 7 13:50:46.452	Message:	TEST Program %1
2021 Thu Jan 7 13:50:41.751	Message:	BREAK Program %1:N45 G0 G60 X164.668 Y159.914 Z100. A-0.121 C0.
2021 Thu Jan 7 13:50:31.835	Message:	START Program %1
2021 Thu Jan 7 13:50:27.439	Message:	LOADED Program %1: C:\cnc886_tst\20201129_Frezky_test\cncfiles\5X_model2_T370.cnc
2021 Thu Jan 7 13:50:14.944	HW:	PLC Error: Chyba O_AxisEn0 != I_AxesEnable

Furthermore, the program was run in Duration mode and the result is duration of 13 minutes 5 seconds, total distance 56.987m at an average speed of 4356 mm / min.

2.9. Editor

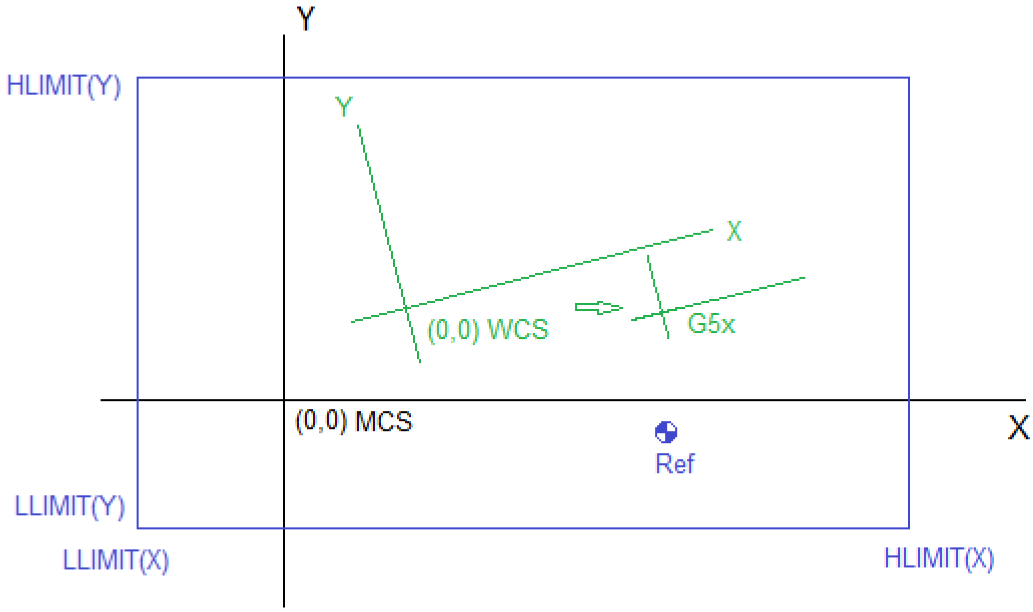


```
%1 {5X_MODEL2_T370.CNC}
N5 T370 S18000. M3
N10 G54
N15 G0 G60 Z100.
N20 G0 G60 X0. Y200. A0
N25 {posunuti pocatku pro natoceni }
N30 G0 G60 X164.668 Y159.914 A0. C0.
N35 G0 G60 Z50.
N40 G92 X164.668 Y159.914 Z50.
N45 G0 G60 X164.668 Y159.914 Z100. A-0.121 C0.
N50 G0 G60 X164.668 Y159.914 Z27.06 A-0.121 C0.
N55 G0 G60 X164.668 Y159.893 Z17.06 A-0.121 C0.
N60 G1 G64 X164.668 Y159.872 Z7.06 A-0.121 C0. F2000.
N65 G1 G64 X163.588 Y159.866 Z5.931 A-0.121 C0.
N70 G1 G64 X161.422 Y159.856 Z4.115 A-0.121 C0.
N75 G1 G64 X159.021 Y159.846 Z2.623 A-0.121 C0.
N80 G1 G64 X156.434 Y159.836 Z1.484 A-0.121 C0.
N85 G1 G64 X153.712 Y159.827 Z0.721 A-0.121 C0.
N90 G1 G64 X150.91 Y159.818 Z0.35 A-0.121 C0.
N95 G1 G64 X150.913 Y159.818 Z0.3 A-0.121 C0.
N100 G1 G64 X140.165 Y159.796 Z-0.358 A-0.128 C0.
N105 G1 G64 X140.188 Y139.764 Z-1.584 A-0.122 C0.
N110 G1 G64 X161.012 Y139.786 Z-2.858 A-0.128 C0.
N115 G1 G64 X160.989 Y159.819 Z-4.084 A-0.128 C0.
N120 G1 G64 X150.913 Y159.818 Z-4.7 A-0.121 C0.
N125 G1 G64 X150.91 Y159.818 Z-4.65 A-0.121 C0.
N130 G1 G64 X148.083 Y159.818 Z-4.623 A-0.121 C0.
N135 G1 G64 X145.289 Y159.818 Z-4.197 A-0.121 C0.
N140 G1 G64 X142.582 Y159.82 Z-3.382 A-0.121 C0.
N145 G1 G64 X140.018 Y159.823 Z-2.194 A-0.121 C0.
```

The image is a sample screen when editing a CNC file.
Working with the editor is intuitive and should not require detailed explanation.
Because the Cnc886 is integrated into the Windows environment, the operator can also use the editors that are part of the operating system to edit programs and other files.
You can also set Cnc886 not to use the built-in editor and use another editor installed on your computer instead.

3. More about control system

3.1. Coordinate systems



The figure illustrates the meaning of the terms used. For illustration, a simple case of a biaxial system with X and Y axes is chosen.

3.1.1. MCS Machine coordinate system

The blue rectangle represents the available area of the machine. The manufacturer shall determine at his discretion the limits within which the X and Y axes can be used. These values are stored in configuration files. In our case, this is

• Lower limit of axis X	LLIMIT(X)
• Upper limit of axis X	HLIMIT(X)
• Lower limit of axis Y	LLIMIT(Y)
• Upper limit of axis Y	HLIMIT(Y)

These values also automatically define the position X = 0 and Y = 0, which defines the origin of the machine coordinate system. We will call this coordinate system MCS (Machine coordinate system). The origin of the MCS does not necessarily lie within the available area of the machine.

3.1.1.1. Servos coordinate system and referencing

Servo amplifiers work in units representing fractions of the motor shaft revolution. Currently, the unit represents 2^{-20} turns, i.e., approximately one millionth. This unit does not necessarily have to be the same as the resolution of the inverter. It can work in coarser resolution. However, the information is transmitted in the above-mentioned units.
Based on the data from the configuration file, the system converts the servo amplifier units into physical units, i.e. millimeters or degrees.
If the actuators are not equipped with an absolute position sensor, the position data is completely random when the system is switched on. The given servo therefore has a

randomly determined position 0, i.e., a randomly set coordinate system. It is therefore necessary to find the relationship between the servos coordinate system and the MCS machine coordinate system. This is achieved in the referencing process. For this purpose, the machine is equipped with sensors whose position in the MCS is known in advance and stored in configuration files. During referencing, the machine approaches the reference sensor and, when found, determines the offset between the MCS coordinate system and the servos coordinate system.

The reference sensors can be placed in the MCS as desired. In our figure, the reference point is marked ⊕ and its position is given by the reference position in the X axis and the reference position in the Y axis.

In the case of an absolute position sensor, a referencing process is not necessary. The offset between the MCS and the drive system is fixed and specified during machine configuration.

3.1.1.2. Physical and virtual axes

From the point of view of the system user, the axes that we refer to as virtual are essential. These are most often the X, Y, Z axes, as well as the rotary axes A, B, C. They express the machining coordinates, the position of the machine, etc. However, the movement in each of these virtual axes can be realized by the interaction of several servomechanisms = physical axes. In some kinematics, the action of servomechanisms can be projected into multiple virtual axes. However, because for service reasons we sometimes need to know the status of these physical axes, physical axes at the service level are mentioned in Cnc886.

In the following text, all axes have the meaning of **virtual axes**.

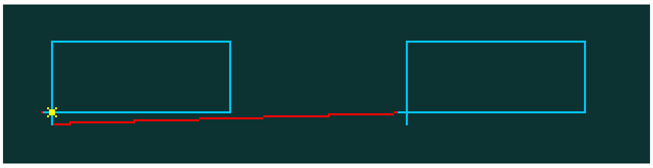
3.1.2. WCS Workpiece coordinate system

The WCS coordinate system is the Workpiece coordinate system. It is offset from the MCS and can be rotated in the XY plane. WCS XY plane tilt is not currently supported by cnc886.

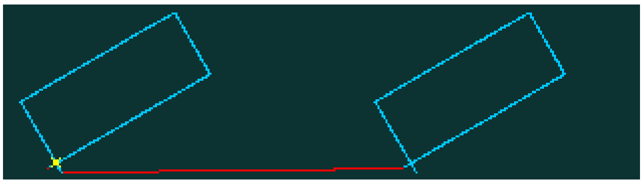
WCS offset can be set in the Manual Motions tab. Usually, however, the Cnc886 is supplemented with an additional interface that allows you to work with multiple WCS coordinate systems, supports their convenient setting, adjustment as well as the definition of the angle of rotation. Within WCS, further coordinate system shifts are possible with G5x functions (see *4.11.4 Group 4:*) and G92 (see *4.11.5 Group 5:*)

3.1.2.1. Shift using functions G53 to G59

Functions G53 to G59 allow you to select an additional offset from the WCS origin. (see picture above). The G5x offset is applied only after the rotation is applied. This means that the offset values are always applied in the MCS direction. The figure shows the machining trajectory of two identical rectangles, where their displacement is realized by activating the G54 function. The G54 offset is in the X axis only.



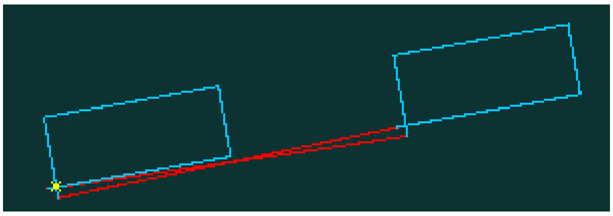
After setting the WCS rotation to 30 °, the trajectory changes as follows:



3.1.2.2. Program shift by G92 function

The last offset that affects the position is the program offset of the G92 function. Its value is displayed in the Auto tab in column. With G92, this offset is set so that the current position of the selected axes is "declared" as a different position. The G92 function is usually used in program cycles.

The G92 function offset is applied before the WCS rotation. Therefore, if we move the rectangles with G92, the trajectory after WCS rotation will look like this:



3.2. 3.2. Kinematic transformations and tool corrections

The MCS and WCS coordinate systems define the positions of the points on the machine or on the product. These points need to be reached at work with a tool or other element that can move in the coordinate systems.

Even in the case of a simple biaxial system X Y, however, the point does not move, but some mechanical element of non-negligible dimensions. In order to be able to talk about "position" at all, it is necessary to define a point on the mechanism to which we will relate. It is reasonable to set this point to represent, for example, the tool clamping point, the pivot point of the five-axis head, etc. The LLIMT and HLIMIT positions (i.e., the MCS system) should also be accessible by this point.

Within Cnc886, we call this mechanism and its reference point the "Tool holder".

The "Tool holder" can be equipped with a number of clamping points for tools. Some are only shifted relative to the "Tool holder". For some, the position of the tool tip may depend on the position of other axes. The way in which the position of the tool tip relative to the "Tool holder" is determined is determined by the kinematic transformation. Cnc886 allows you to define different kinematic transformations for different types of machines and their equipment. But it does not allow you to modify them. They are fixed for the given Cnc886 application.

3.2.1. Tool corrections

Tool corrections are parameters used in kinematic transformation. It is possible to assign corrections of individual virtual axes, parameter length and tool diameter to each tool.

However, since this description may not be sufficient to apply the kinematic transformation, Cnc886 allows the tool to assign additional parameters to it.

One of the parameters is the **Type** of tool. According to it, Cnc886 can internally switch which kinematic transformation and what parameters to use.

Example of defining tool parameters:

```
  TOOL_COR[020,Length] = 78.921858;
  TOOL_COR[020,Radius] = 5.920000;
  TOOL_COR[020,  Type] = 10.000000;
  TOOL_COR[020, Par01] = 1.000000;
```

3.2.2. Radius compensation

Radius compensation is applied when it is necessary to machine using the circumference of the tool, not by its center. However, because the system knows the position of a point on the tool circumference only from the context of the part program, radius compensation is not applicable outside this program. Thus, if the machine is stationary, the program is not running, it is not possible to perform the conversion.

Thus, this correction goes beyond the kinematic transformation and is mentioned separately.

3.3. 3.3. System function logs

The Cnc886 system logs system events into two basic files. ErrorLog.txt and diag.log file

3.3.1. ErrorLog.txt file

The ErrorLog.txt file was originally intended to record error messages. It got its name from that. At present, however, it also contains a record of other events. Thus, the name no longer fits completely.

The file is intended mainly for additional diagnosis of problems. This diagnosis should be made by the machine supplier. The content of the file is not easy to understand to the average user. What is stored in the file depends, to a large extent, on the equipment of the machine and on requirements of the manufacturer.

The system adds records to the file until the file size exceeds 20MByte. It then copies ErrorLog.txt to the ErrorLog.bak file and begins creating a new ErrorLog.txt. This mechanism ensures that files do not grow beyond all limits, and at the same time we have an event history with a sufficient size of 20MByte to 40MByte. Usually, 20MByte is enough to cover a week of machine operation.

In case of troubleshooting, you will likely be asked by the supplier to send these files for analysis. The sooner you back up these two files after a problem occurs, the better your chances of effective analysis.

3.3.2. diag.log file

The diag.log file contains mainly information about the details of Cnc886 startup. It plays an important role in finding a problem with the start of the control system.

4. Programming language description

A CNC program is a sequence of commands for a control system. The program contains the necessary information for movements and other activities of the machine.

The language for Cnc886 is derived from the standard ISO code. It was gradually expanded with new and new elements that seemed useful to us to cover customers' needs. Extensive options have emerged for calculations, program branching, cycles, and subroutine calls.

4.1. Execution of the program by the cnc886 system

When executing a program running cnc886, two threads work together, or if you want processes. The first, let's call it "Interpreter", analyzes language commands and solves all transformations and corrections. The result is a sequence of instructions for the second process, let's call it the "Processor", which performs specific movements and other activities. It only responds to the current state of the machine and operator commands.

The Interpreter continues to analyze the program until one of the following situations stops it:

1. End of program
2. Filled space for instructions or data
3. Occurrence of a function that requires pausing the interpretation until the Processor executes all prepared instructions.

In the case 2. or 3. the work continues as soon as the conditions allow.

Therefore, the interpreter can parse the code many lines ahead. How much is not determined by the number of program lines but by the free space for instructions and their data. Part of this data is, for example, the trajectories of the tool holder in the MCS recorded with the density given by the configuration parameters. These are usually units of thousands of millimeters. With the usual amount of memory for the Interpreter's data, the Interpreter is in front of the Processor by about 1 meter of trajectory.

4.2. Program structure

The program is plain text with instructions organized into lines. In Cnc886, everything in the code is case sensitive. The program file must be saved in UTF-8 or ASCII encoding. ASCII supports only characters from English alphabet. UTF-8 is required for files containing characters with diacritics or special characters like Japanese or Chinese.

The first line is reserved for the program number and optional display name.

The following lines (command lines) can be numbered. The line number is placed at the beginning after the N character.

The program ends on the line that shows the end of program function - M30. It does not have to be the last line.

Program file must end with empty line.

The following program illustrates the usual structure of a program.

```
Example:
%15    { displayed name }
N1     ……    { first instruction line of program }
N16    ……
N156   ……
N945 M30    { program end }
```

4.2.1. First line - program number with program name

The first line must be preceded by a **%** character to start the program. The % character is followed by a program number, which consists of 1 to 8 digits. The program number must be in the range 0 to 99999999

The choice of the number of the main program and subprograms depends on the user and is not defined by the system. A program with the number %0 plays a special role. It can contain implementations of machine cycles determined by functions G200 to G399. The last activated function is then called by function G79. We recommend a suitable division of the range value between the main programs and the subprograms for a better orientation.

The program number can be extended by the program name {comment}.

The number and header are decoded during loading of the program to memory.

```
Example:
%12 {product 124}
```

4.2.2. Comment

A comment can also be considered a special command. Comment is a sequence of characters enclosed between the characters **{** and **}**.

4.2.3. Command lines

Each command line of the CNC program may or may not begin with a line number preceded by the letter N. The line number must be in the range 1 to 99999999. The line contains a sequence of commands. Most commands are calls to a function. The designations of the individual functions are derived from the standard ISO code, where a letter (or more letters) followed by a number designates the functions.

Over time, the basic function set in Cnc886 has been expanded with additional options. However, letter and number naming has been abandoned and functions take the form of keywords such as IF, THEN, GOTO, etc.

Numeric values can be a function numbers or are function parameters. They specify, for example, speed, rpm, coordinate value, etc. In Cnc886, the values are given in the form of an expression. Numeric constants or references to so-called R parameters, MP parameters and CP parameters can appear in the expression.

```
Command line example:

N10  G1  X100  Y5.4  F1000  S500  M3  { comment }
```

N10	{ line number }
G1	{ G-function for linear motion }

```
X100     { naming of axes and target coordinates }
F1000    { feed rate in mm / min.}
S500     { spindle speed rpm for main spindle }
M3       { M function for starting the spindle in the CW direction }
{......} { the comment is in braces "{...}". }
```

4.3. Value and expression

In the text, the term value will mean the number specified by the expression. We express the expression in the usual way. It can contain numbers, references to parameters, operators +, -, *, /, ^, | , & and functions returning a numeric value. Brackets can be used in expressions to a practically unlimited depth.

4.3.1. Arithmetic functions

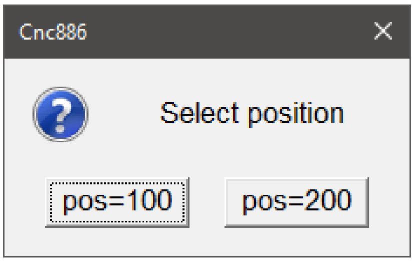
Term Arithmetic function include all functions that return a numeric value. As we will see later, there are still functions that return a logical value.

Function	**Description**
SIN(expression)	sine of the expression value in degrees
COS(expression)	cosine of the expression value in degrees
TAN(expression)	tangent of the value of the expression in degrees
CTG(expression)	cotangent of the expression value in degrees
ASIN(expression)	arc sine of the expression value. The result is in degrees
ACOS(expression)	arc cosine of the expression value. The result is in degrees
ATAN(expression)	arc tangent values of the expression. The result is in degrees
ACTG(expression)	Arc cotangent expression values. The result is in degrees
FLOOR(expression)	Returns the highest integer, less than or equal to the expression.
SQRT(expression)	square root of the value of the expression.
LLIMIT(axis)	Returns the lower limit of the axis specified by its name.
HLIMIT(axis)	Returns the upper limit of the axis specified by its name.
POSITION(axis)	Returns the last specified axis position in the program
ACT_LINE_NUM	Returns the current line number
PROMPT(parameters)	Returns the number of the button selected by the operator
ACT_TOOL	The number of the current tool
CYCLE_NUM	Number of the defined machine cycle

PROMPT function

PROMPT(** <title>,<but1>,<but2>..**)

The PROMPT function displays a dialog box, titled <title> and containing the buttons described <but1>, <but2>, etc. All texts must be enclosed in quotation marks. The operator clicks on a button and the function returns its number. So for but1 it will be value = 1, for but2 value = 2, etc..



```
Example of PROMPT:

N20 R0 = PROMPT("Select position","pos=100","pos=200")
```

4.3.2. Elements of expression

The expression element can be a number, a parameter value, or an arithmetic function. The expression can also be an expression in parentheses.

4.3.3. Arithmetic operations

Arithmetic operations include basic addition, subtraction, multiplication, and division operations. Cnc886 also allows so-called "bitwise" logical operations AND, OR and XOR, i.e. logical product, logical sum and exclusive or. Each of the operands is first converted to an integer value. Then the logical function AND, OR and XOR between the bits at the same position of binary representation is evaluated.

Operation	**Operator**
Addition	**+**
Subtraction	**-**
Multiplication	*****
Division	**/**
Bitwise AND	**&**
Bitwise OR	**|**
Bitwise XOR	**^**

4.4. R parameters, MP parameters and CP parameters

Expressions can include parameter values. The system works with three types of parameters. They differ in their primary purpose and treatment.

4.4.1. R Parameters

The system provides the user with a total of 12000 parameters, called R parameters and called R0 to R11999. They are represented by a 64-bit floating point format.

R parameters are used to store program variables, they can be used to perform simple calculations, test their value for zero, etc. Most of them are parameters having a predetermined function. For example, the values of tool corrections and offsets used for functions G53 to G59 are stored in R parameters. On the one hand, this reveals various programming possibilities, on the other hand, it brings certain dangers if they are handled carelessly.

R parameters differ from other parameter types. When the program is started, the system copies the whole set of parameters (base set) into the memory and works with copies throughout the program run. If it would work with the base set directly, the values of the R parameters may be changed by running the program and restarting the program may perform a different action. However, because sometimes this behavior is desirable, special functions and prefixes are provided that allow you to work with both sets of parameters. These are the SAVED prefix and the LOAD_RPARS function. (see below) It is worth noting that simulation does not affect the base set.

The base set of R parameters is saved to disk at the end of the program (either by RESET, error, or M30 in the normal way). The values of the R parameters modified by program with the prefix SAVED, are stored with new values. It is thus possible to count manufactured pieces, etc.

The LOAD_RPARS (number1, number2) command overwrites parameters with indexes from number1 to number2 from the base set to the copy set.

The R parameter is determined by the CNC code with the character R or Q and the index <expression>. The characters R and Q are interchangeable. However, this index <expression> may depend on another R parameter. This provides the possibility of indirect links. We will explain the determination of the R parameter on examples.

R8 Q8 R(8)	Direct parameter indexing. The eighth parameter is used
RR8 R(R8)	Indirect indexing. The value of parameter R8 is read. This value is truncated to integer, and the result is used as the parameter index. E.g.: R8=11.25, then parameter R11 will be read.
R(R1+3)	The index R of the parameter is given by the sum of the contents of the parameter R1 and the number 3, truncated to integer

The indirect indexing can be arbitrarily nested. However, indirect addressing of higher orders is rarely used.

Some R parameters that have special meaning can be accessed without knowledge of their index. These are tool corrections and coordinate system offsets for functions G53 to G59.
We refer to tool corrections as follows:

```
        TOOL_COR[001,X]
        TOOL_COR[001,Radius]
        TOOL_COR[001,Length]
        TOOL_COR[001,Type]
        TOOL_COR[001,Par01]
```

The first parameter in parentheses indicates the tool number, the second compensation axis, or the tool radius or its length, type or additional parameter.

The coordinate system offset values with functions G53 to G59 are accessed as follows:

```
        CS_G54[Y]
        CS_G55[X]
        ...
```

4.4.2. MP Parameters

MP parameters are intended primarily for storing variables that serve as input parameters for M functions, if they use any. Therefore, it is not recommended to use them as program variables.
It should be noted that a command that changes the value of an MP parameter can be evaluated by the Interpreter long before the function that uses it is executed. This is in a situation where the function is on the line preceding the change. This is because the change is made by the Interpreter process, while the function is performed by the Processor process. Therefore, if we already change the values of MP parameters in the program, it is necessary that the change of the parameter precedes all functions that use it. So preferably at the beginning of the whole program.

The MP parameter file currently contains 2000 items. Unlike R parameters, one set is always used. Therefore, the values that were before the program was run are not restored after execution. We can access the MP parameters as follows:

MP<expression>

```
Example:
  MP100
  MP(400)
  MP(R8+1)
```

4.4.3. CP Parameters

In addition to MP parameters, there are also parameters that are used to configure the system. The Cnc886 commands are not modifiable. They are read from the configuration files and then remain unchanged. However, they are readable. We can read CP parameters as follows:

CP<expression>

```
Example:
  CP100
```

4.5. Assignment command

The command line can include one or more assignment statements. They are performed in the order in which they appear on the line.

The command has the form:
<R parameter> **=** *<expression>*
<MP parameter> **=** < *expression* >

or

SAVED *<R parameter>* **=** < *expression* >

Instead of the **=** character, you can also use **:=**

The command calculates the expression on the right side and stores the result in the R parameter or the MP parameter. As mentioned earlier, CP parameters are read-only.

The following examples show some of the possibilities of using R parameters and assignment commands.

```
Example 1:  N10 R8=6.25 {R8 becomes 6.25}
Example 2:  N10 R8=R6   {R8 takes the values of parameter R6}
Example 3:  N1  R6=4
            N2  R4=10
            N10 R8=RR6 {R8 takes the value 10}


Example 4:  N20 R1=2.5*(R2+R3) R1= R1/R5
Example 5:  N20 R16=2.5*R6 R8=R145+R16
            N30 XR8 YR16
Example 6:  N10 G1 XR2 YRR3 FR3
```

As mentioned earlier, the prefix SAVED causes the value of the expression to be stored in both the copy set and the base set. The prefix applies to only one assignment statement.

```
Example 7:  N10 SAVED R1=R1+1
```

Examples of using arithmetic functions:

```
R0:=45
R2:=R1*SIN(R0)
R3:=R1*COS(R0)
R2 =R1*SIN( 180 + R0 )
R5 =SQRT( R6*R6 + R7*R7 )
```

4.6. GOTO, CALL, IF - Program flow control

The GOTO, CALL, CALL%, BN, B% commands and the IF THEN structure are used to control the program flow.
The line number is used as a parameter in commands. It automatically follows that unnumbered lines cannot be the target of a program jump or subroutine call.

It may happen that we need to use a conditional or unconditional jump in a program generated by some CAD-CAM SW postprocessor. Sometimes it is not possible to use a line number within a postprocessor as a parameter. Therefore, the ACT_LINE_NUM function has been implemented in cnc886, which returns the number of the current line. It is then possible to create cycles or program branches that do not require knowledge of the line number.

```
Example: Jump ten lines forward

N895 GOTO (ACT_LINE_NUM+10)
```

Remark:
Because not every line needs to be numbered, ACT_LINE_NUM is the number of the last line that was numbered.

4.6.1. Unconditional statements

4.6.1.1. Jump in program GOTO

The command has the form:

GOTO *<expression>*

```
Example: (jump to line N10)
N157 G1 X50 Y20 F1000
N158 GOTO 10
N159 G0 X50
```

The same effect can be achieved with older versions using the BN10 command. The BN command has been retained for backward compatibility.
The line number can be specified by an expression. The value of the expression is converted to an integer and used as a line number. The error occurs when the line with the given number does not exist. So it is possible to write:

```
N158 GOTO 1000*R1
```

If there is a GOTO statement on the line, there must be no more commands on it.

4.6.1.2. Subprogram call CALL and CALL%

The command has the form:
CALL *< expression >*
CALL% *< expression >*

There are two variants. The CALL statement makes a call within the program. The command parameter is the line number. The CALL% variant calls another program as a subroutine and the parameter is the program number (identified after the % character on the first line).

In both cases, calling function M17 will cause return from the subroutine.

A **RETURN** command is also available for subroutines called with CALL.

```
Example: (calling subroutine on line 1000)
N157 G1 X50 Y20 F1000
N158 CALL 1000
N159 G0 X50
```

N158	Line number
CALL,CALL%	keyword
10	line number resp. program where the call is to be made

If there is a CALL or CALL% statement on the line, there must be no more statements on it.

4.6.2. Conditional branching of the program

4.6.2.1. Conditional jump

This command is intended primarily for the execution of cycles. BN commands are retained for backward compatibility.

The conditional jump statement has one of the following forms:

GOTO *< expression > Rparameter*
GOTO *< expression > !Rparameter*
BN *< expression > Rparameter*
BN *< expression > !Rparameter*

The conditional jump to the line <expression> is performed according to the state of the specified R parameter. After each test, 1 is automatically subtracted from the parameter. If there is no ! character before the R parameter, a jump is made when the parameter is not zero. If the ! character is specified, a jump is made when the parameter is zero.
Conditional branching is mostly used to implement program cycles. However, the initial filling of the R parameter must not be part of the cycle, otherwise an infinite loop will be created.

```
Example: Jump to the line 25.

N10 R12=8
N25 G0 Z8
N150 G1 Z0 F500
N157 G1 X50 Y20 F1000
N158 BN25 R12
N159 G0 X50
```

4.6.2.2. IF THEN structure

The IF THEN structure represents a conditional statement. If used on a line, there must be no more statements on the line before the word IF.

The command has the form:

IF *<logical expression>* ***THEN*** *<commands>*

A logical expression is formed in the usual way from operands, which are a logical variable or a numeric relation, and using logical operators. Parentheses can be used in expressions.

Logical operators:

!	logical negation. Refers to the operand specified after the operator
&&	logical AND between two operands
||	logical OR between two operands

Operands can be:

BIN<num> Indicates the logical value of a binary variable. For this purpose, the binary inputs and outputs of the system are numbered from zero to a variable limit. The list of binary variables is in the Service Binars tab of the control system and they are numbered so that the number 0 corresponds to the variable at the top right (bit D0). The numbering continues to the left up to number 7, then from the right on the second line, etc.

ODD(expr **)** A Boolean function that returns "true" if the integer part of the expression in parentheses is odd.

Numeric relation

A numeric relation is a comparison of the values of two numeric expressions. Operators are available

==	Equal
!=	Not equal
<	Less
>	Greater
<=	Less or equal
>=	Greater or equal

Evaluation of logical expressions requires additional comment. The interpreter of the CNC program processes the lines in advance before their actual execution. This calculation also addresses all issues related to "look ahead" functions, such as speed planning, path modification by radius correction, etc. When evaluating logical expressions in which there is at least one value of the binary variable BIN, the Interpreter must pause this calculation and wait until all previous statements are executed because they can change the value of a binary variable. Therefore, movement always stops at this point in the program. The recalculation of subsequent movements then starts again.

<commands> after the keyword THEN is a list of commands as we can enter on the command line, except for the other IF THEN structure.

```
Example:
N10 IF (R12==8) THEN GOTO 1200
N25 IF BIN36 THEN CALL 128
N50 IF (R1>10) && (R1<20) THEN M3 S12000 F7000
```

4.7. T - Tool correction selection

The **T** command contains the tool correction number. It is active until a new tool is selected.

T<expression>

A maximum of 1000 tools can be stored in the system.

Following form can be used to access R Parameters containing tool parameters:

TOOL_COR[tool number, correction name]

```
Example:
       TOOL_COR[001,X     ] {tool correction number 1, axis X}
       TOOL_COR[001,Radius] {tool correction number 1, Radius}
       TOOL_COR[001,Length] {tool correction number 1, length}
```

The parameter specified in this way can be used inside expressions as well as the assignment target.

```
Example of tool correction adjustment:
N10 TOOL_COR[001,Radius] = TOOL_COR[001,Radius] - 0,1
```

```
Example of applying tool correction:
N10 G1 X100 Y50 T1     {apply T1}
N20 ..
N130 G1 X450 T56       {apply T56}
N240 ..
N450 G1 Y18 T1         {apply T1}
N490 M30               {end, cancel all corrections}
```

4.8. F - Feed

4.8.1. Feed rate

The tool feed rate (speed) is determined by the F command and is specified in mm / min.

F<expression>

The feed set by the F command applies to all subsequent lines until it is changed by another F command. The criterion of the minimum speed applies at the junction of two movement sections with different feeds. Thus, at the point of transition from one section to another, the lower of the stated feeds will be planned.

4.8.2. Linear axes

The speed depends on the selected interpolation type {e.g., G0 / G1 / G2 etc.} and setting system parameters. The default setting is mm/min.

G1/G2/G3
All axes programmed in the CNC line are interpolated so that the resulting tool speed corresponds to the defined feed rate F in mm/min.

```
Example:
N10  G1  X100  Y50 Z20  F5000
```

G0

Linear movement with speed F defined by machine configuration. Maximum speed is limited by slowest axis.

```
Example:
N10  G0  X100  Y50
```

4.8.3. Rotary axes

At this point, the term "rotary axes" refers to axes that change the tilt of the tool. Examples are the A and C axes of five-axis milling machines.
Where the movement of the rotary axes is accompanied by movements of the linear axes, the feed rate is considered to be the required tool tip speed and is in mm/min.

```
Example:
N10 G1 X100 Y100 C180 F500 {C - rotary axis}
```

Where only the movement of the rotary axes without the movement of the linear axes is indicated on the line, the required speed is considered to be the movement in degrees per minute.

4.9. S - Spindle rotation speed

The spindle speed is programmed in revolutions per minute. The rotation speed of the spindle is limited by the maximum speed set in the machine constant.

```
Example:
N10 G1 X100  Y50  F10 S1500 M3
```

4.10. TI - Dwell time

Dwell times are programmed in seconds in the CNC program.

```
Example:
N10 TI2.5   {time as a constant}
N10 TIR2    {time in R parameter}
```

The dwell time is performed after the execution of the M functions with priority and before the start of the movement indicated on the same line.

4.11. G - functions

The G functions are arranged in functional groups, and only one function from each group can be activated at a time.

Some functions are active before the program starts. Some functions are modal, so they affect all other lines until they are canceled by another function. The others are valid only on the CNC line on which they are listed.

- Group 1 Interpolation G0, G1, G2, G3
- Group 2 Plane selection G17, G18, G19
- Group 3 Tool radius compensation G40, G41, G42
- Group 4 Coordinate system shift G53 - G59
- Group 5 Coordinate system shift by G92 and G93
- Group 6 Speed characteristics G60 G64
- Group 7 Target position G11, G12, G13
- Group 8 Positioning absolute/incremental G90, G91
- Group 9 Machine cycles G200 - G399, G79

4.11.1. Group 1: Interpolation

All interpolation commands consist of a word indicating the type of interpolation and a list of interpolated coordinates with the end point. The end positions of the individual axes are indicated by an expression. So it can be a simple number or a complex arithmetic expression.
The way the individual path sections are connected is affected by the G60 / G64 functions. If G64 is active, the system connects the individual lines of the CNC program without loss of speed or adjusts the speed according to the dynamics of the entire system. In the case of G60, the movement stops at the end of each section.

The meaning of the coordinate values of the individual axes in the program depends on many other factors modifying the coordinate system. The end point can be entered absolutely or incrementally according to the active function G90 / G91. The absolute machine coordinate system can be set with G11. It can be shifted with functions G53 to G59 and function G92. The coordinate system can be affected by the SCALE command or rotated based on a parameter.

4.11.1.1. G0 rapid positioning

All axes reach the programmed position at the same time. The tool moves in a straight line. When moving, acceleration ramps for rapid positioning are applied. The programmed speed F is not effective
The rapid positioning speed is determined by the machine configuration and possibly reduced by the system so that none of the axes exceeds its speed limit.

```
Example:
N10 G0 X100 Y50
```

4.11.1.2. Linear interpolation G1

The tool feed rate is determined by the F command. Linear interpolation is performed in all axes simultaneously. The maximum tool feed rate may be limited by the slowest axis. G1 is active even if no function of this group is specified.

```
Example:
N10 G1 X100 Y500 Z250 C100 A20 F1000
```

4.11.1.3. Circular interpolation G2 and G3

The tool feed rate is determined by the F command. Circular interpolation can only be performed in one plane, which is determined by the plane selection functions {G17 / G18 / G19}. In addition to the end point, the command specifies either the center of the arc or its radius.

G2 Clockwise circular interpolation (CW)
G3 Counterclockwise circular interpolation (CCW)

Other axes listed in the G2/G3 command are interpolated linearly.

Circular interpolation with RC radius

We determine the radius of the circle with the command

RC<expression>

A positive value of the expression creates an arc with an angle of 0 to 180 degrees, a negative value of the radius indicates an arc with an angle greater than 180 degrees.

```
Example:
N10 G1 X100 Y0 F1000
N20 G2 X110 Y0 RC10   {Create arc from current point to end point
with radius 10mm.}
```

Circular interpolation with center determination

Circular interpolation can also be determined by the end point and the center of the circle. The words are used to determine the center of the circle:

I X axis coordinate of the center (first system axis)
J Y axis coordinate of the center (second system axis)
K Z axis coordinate of the center (third system axis)

The center coordinates are always relative to the interpolation starting point.

Due to the numerical stability of the calculations, we strongly recommend not to use arcs with an angle close to 180 degrees. It is best to divide arcs or entire circles into parts with angles up to 90 degrees.

4.11.2. Group 2: Plane selection G17, G18 a G19

The interpolation plane must be set for circular interpolation and for radius compensation. G17 is included by default. All plane selection functions are modal.

- G17 = plane X-Y (resp. first and second system axes)
- G18 = plane X-Z (resp. first and third system axes)
- G19 = plane Y-Z (resp. second and third system axes)

4.11.3. Group 3: Tool radius compensation

The following instructions enable and disable tool path compensation with respect to its radius. The goal is to move the tool so that the points described by commands G1, G2 and G3 are reached by the circumference and not by the center of the tool.

- Cancel tool radius compensation G40
- Tool radius compensation from the left G41
- Tool radius compensation from the right G42

All functions in the group are modal.
The plane of application of tool radius compensation is determined by commands G17/G18/G19.

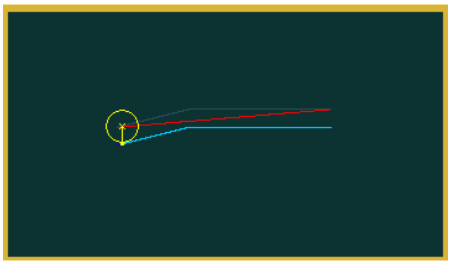
```
Example 1: Applying and canceling tool radius compensation
N10 G1 G41 X100 F500
N20 G1 G40 X150 F500
```

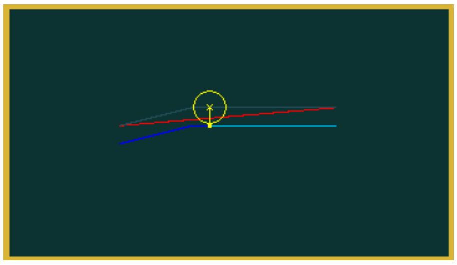
When passing the inner corners, the tool approaches the end of the section so as not to disturb the next section. However, if there is no point in the next section that can continue machining, an error will be declared. When passing through the outer corners, a linear or circular section is inserted by the system, depending on the shape of the path. The selection is made by the system based on the configuration parameter settings.

4.11.3.1. Using G41, G42 and G40

Almost every cut / machining starts outside the contour, continues by approaching the part contour, then cutting the part, and ends by exiting the cut contour.
The start of the machining process (drilling, piercing, etc.) is placed outside the contour because, in the vast majority of cases, it damages its immediate surroundings. The approach to contour follows.
During these operations, a movement must occur somewhere, compensating for the radius of the tool so that the contour of the product touches the tool with the circumference and not with the center. It is obvious that the compensation must already be performed at the moment of approaching the contour.
Functions G41 and G42 work in such a way that the compensation movement takes place during the first movement in which the function is active.

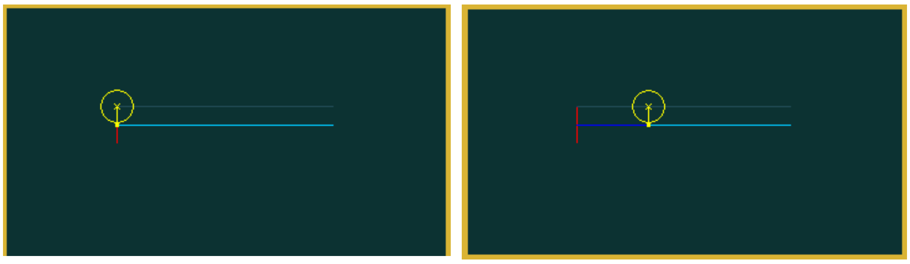
```
N50 G0 X-2 Y0 G40
N60 G1 X0  Y0 G41
N70 G1 X4  Y0
```





The Cnc886 system can also deal with a situation where the G41 / G42 function is listed on a line with motion that would not be without motion compensation. That is, if the end coordinates are the same as the start coordinates.

```
N50 G0 X-2 Y0 G40
N60 G0 X-2 Y0 G41
N70 G1 X0  Y0
N80 G1 X4  Y0
```



Similarly, the G40 function performs a compensating movement on the first movement when G40 is active.

4.11.4. Group 4: Coordinate system shift G53 - G59

The function call for the coordinate system shift is G53 - G59. G53 is current in the preparatory functions. All functions are modal. The shift is canceled by M30.

```
Example of coordinate system shift usage:
N10 G54 G0 X0
N20 B%4711       {part 1}
N30 G55 G0 X0
N40 B%4711       {part 2}
N50 G53 G0 X0
```

4.11.5. Group 5: Coordinate system shift by G92 and G93

The G92 function sets the program coordinate shift during program run so that the actual position becomes the desired position. There is no movement.
The G93 function cancels the offset that was programmed by the G92 function for all axes. Calling G92 without specifying a single axis has the same effect.

```
Example :
N10 G0 X100 Y50  { into position }
N20 G92 X0 Y0    { setting new desired position }
N30 G0 X100 Y50  { into position }
N40 G92 X0 Y0    { setting new desired position }
N50 G0 X100 Y50  { into position }
N60 G93          { cancel shift G92 }
```

It is not possible to combine functions G92 and G93 with functions from group 1 (interpolated moves), as it would not be clear which function should use given coordinates.

4.11.6. Group 6: Speed characteristic G60, G64

The G60 and G64 functions are reserved for changing the speed characteristic.
The G60 and G64 functions are modal.

- G60 use the start and end ramps at the beginning and end of the line, respectively
- G64 passes between two lines without losing speed

G60

At the beginning of the line, the system engages a starting ramp and a braking ramp before the end of the line. A line change is made when the deviation from the end position in all programmed axes is less than the value set by the machine constant.

G64

This function can have several possible variants of behavior depending on the setting of machine constants.

1. The system passes between two lines at the programmed speed F without losing speed. In this case, the operator must predict the behavior in terms of the dynamics of the entire system.
2. The system can evaluate the difference in the angles of the following sections of the trajectory and if the set limit is exceeded, it will automatically include a stop, as with the G60 function.
3. The system can calculate the actual dynamic stress of the whole system based on the machine constants and adjust the speed F between the two rows accordingly so that the system mechanics is exposed to a reasonable stress. This adjustment also applies when passing through circular interpolation.
4. Combination of 2 and 3.

```
Example of G60 and G64 usage:
N10 G1 X50 Y20 F2000
N20 G1 G64 X80 Y40
N30 G1 X100 Y60
N40 G60
N50 G1 X0 Y20
```

The G64 function also works in a linear section and arc connection, two arcs, etc. In the following example, the passage of a rounded corner is programmed without pausing.

```
Example:
N5  G64
N10 G1 X80 Y80 F2000
N20 G2 X100 Y100 RC20
N30 G1 Y150
N40 G60
N50 G1 X0 Y20 F1000
```

Linear interpolation without tangential transitions {angle of two sections> 10 °} can lead to jerky axis movements.

4.11.7. Group 7: Target position G11, G12, G13

The functions complement the movement functions from Group 1 and determine which part of the machine and how to reach the target position. Functions are not modal.

4.11.7.1. Function G11

The target position is in the MCS system and is to be reached by the tool holder. The function is advantageous especially in situations where it is necessary to move to a position that does not move with respect to the reference point, e.g. tool changer, semi-finished product feeders, etc.

4.11.7.2. Function G12

The target position is in the MCS system and is to be reached with the tool tip.
The function is advantageous, for example, when the tool approaches a calibration probe or another fixed location on the machine.

4.11.7.3. Function G13

The function specifies that the position is to be reached in the WCS by tool, but in a specific way. The movement is divided into several phases. Only some axes move in each phase. The method of approach is defined by the manufacturer. (This distributed movement is also used when approaching in **From position** or **From line** mode).
To illustrate, let's give an example of a three-axis milling machine. When approaching the machining point from the tool changer, it is advisable to first move with the XY axes, and finally with the Z axis.
Even more important is the G13 function of a five-axis rotary table milling machine (axes A and C). It is advisable to approach the machining point first by turning the axes A and C, then approach the XY axes (in MCS) and finally traverse the Z axis.
Without the G13 function, it is almost impossible to program such a sequence of movements.

4.11.8. Group 8: Positioning absolute/incremental G90, G91

Functions G90 and G91 are modal.

- G90 Absolute positioning

The coordinates are relative to the zero point of the WCS coordinate system or to the zero-point created by G92.

```
Example: Absolute positioning
N10 G0 G90 X10 Y10
N20 G1 X30 Y30 F1000
N30 X45 Y15
N40 X10 Y10
```

- G91 Incremental positioning

The coordinates are relative to the zero-point formed by the end position of the previous movement.

Auxiliary coordinates {I, J, K} for circular interpolation are not affected by G90 / G91.

```
Example: Incremental positioning
N10 G0 G90 X10 Y10
N20 G1 G91 X20 Y20 F1000
N30 X15 Y-15
N40 X-35 Y-5
```

4.11.9. Group 9: Machine cycles G200 - G399, G79

The system allows you to define special operations, which in many systems are referred to as machine cycles. These are often drilling cycles, hole milling, etc. These cycles can almost always be implemented by a sequence of language commands with input parameters. In Cnc886, such a sequence can be called as a subroutine.

In order to support method used by other control systems, we have introduced the following functions into the system. In addition, we also allow dual way to reference R parameters; with the letter R or Q.

G200 - G399

The function specifies the cycle number that will be called by the G79 function. The function only records its number in the parameter CYCLE_NUM available to the called function file.

G79

The function is identical to calling CALL% 0. Therefore, the program with the number %0 must be present in the memory for the actual execution of the cycle. It is assumed that program %0 contains subroutines for all defined machine cycles and that this program is automatically loaded at Cnc886 start-up.

Example of cycle determination:

```
%1 { Program volajici G200 a G79 }
N100 G00 Z+100 G40
N100 G00 X0 Y0
N110 G200 { cycle definition }
     Q200=2   { set-up clearance         }
     Q291=-15 { depth                    }
     Q206=250 { feed rate for plugging   }
     Q202=5   { plugging depth           }
     Q210=0   { dwell time at top        }
     Q203=20  { surface coordinate       }
     Q204=100 { 2nd set-up clearance     }
     Q211=0.1 { dwell time at depth      }
N120 G0 X+30 Y+10 M3 S12000 G79
N130 G1 X+80 Y+50 G79 F10000
N140 G0 Z+100
N999 M30
```

In the example shown, cycle G200 is activated, which is then called on lines N120 and N130.

The implementation in program% 0 can then look like this.

```
%0 { called program }
IF ( CYCLE_NUM == 200 ) THEN GOTO 200000
ERROR "Unknown cycle"
M17

N200000 ECHO "Cycle 200"
        G0 Z(Q203+Q200)
        G1 Z(Q203+Q291) F(Q206)
        TI(Q211)
        G0 Z(Q204)
        TI(Q210)
        ECHO ""
        M17
```

4.12. ECHO, WARNING, ERROR

4.12.1. Function ECHO

The function is used to write texts to a special line in the Auto tab of the control system. It allows you to display the program phase, operating instructions when pausing M0 functions and so on.
The keyword ECHO is followed by a string of characters in quotation marks.

```
Example:
N10 ECHO "Roughing"
```

4.12.2. Function WARNING

The WARNING function displays the text that follows it. It appears in a special window that overlaps cnc886. At the same time, the text is written to the record in the Errors tab.

```
Example:
N10 WARNING "10000 pieces produced. Perform maintenance!"
```

4.12.3. Function ERROR

The ERROR function raises a program error and displays the text that follows it. At the same time, the text is written to the record in the Errors tab.

```
Example:
N10 ERROR "Tool changer position occupied!"
```

4.13. SCALE - scale change

The command has the form:

SCALE = *<expression>*

The value of the expression in the SCALE statement specifies the number by which the required coordinates of the CNC plan are to be multiplied and also the radius of the RC circle. The SCALE command behaves "intelligently" by multiplying the values of only those axes that make sense. Thus, for example, the rotary axes carrying the tool are not multiplied.
The scaling is applied to all other executed lines of the program until the SCALE value changes.
Theoretically, negative expression values can be used, but they will not work well if circular interpolation or radius correction occurs in the CNC program.
At the end of the program, SCALE is automatically reset to 1, i.e., without scaling.

4.14. M - functions

M-functions are predefined for controlling various parts of the machine. Most of them are designed to control those subsystems that are not related to the trajectory of the machine. They control spindles, chucks, cooling and more. Some M functions are de facto defined by the standard. Others are defined by the manufacturer of the specific machine. They are therefore machine dependent.

The command to call the M function has the form:

[$<expr>]M<expr> [<params list>] [MODE=<expr>]

[$<expr>] represents an optional function prefix. For example, it is used to distinguish the spindle number if more than one is installed on the machine. The meaning of the prefix is given by the implementation of the M function and depends on the specific machine.

M<expr> is a command to call an M function whose number is specified by an expression. Therefore, the number of the called M function does not have to be fixed. It can be read from R parameter and/or calculated.

[<params list>] The function can be modified with up to ten parameters. The parameter list is a comma-separated sequence of expressions. The whole list is enclosed in parentheses

([<expr1>], [<expr2>], ... [<exprN>])

The meaning of individual parameters depends on the specific function and their position in the list. It is allowed to omit the parameter. Then the expression is not in place of the parameter. The function must then have defined how it will behave in such a case. For example, it will use some default value.

```
Params list example:
      M150(1,,R8,(CP200+3)) {Parameter 2 is omitted}
```

[MODE=<expr>] Another modifier of the M-function is the MODE command. This command can again affect its behavior in a way specified by the machine tool builder.

4.14.1. Execution of M-functions

In the Cnc886 system, each M function is performed by a separate thread (process). Therefore, all M functions are run at the same time and work in parallel. Their order on the line is not important.

The function is considered executed if its thread terminates and confirms completion. When we talk about performing a function, it means starting it and waiting for it to complete.

If the M functions are listed on a line together with the command for moving the axes, the operations are performed in the following order:

1. M functions designated “before move”
2. Axes movement
3. M functions designated “after move”

Most M functions are designated "before move". Designation of M functions is defined by the machine manufacturer.

The M functions in the CNC line always run when the axes are at rest. This behavior can be overridden by the **NOWAIT** prefix at the beginning of the line. With this modifier, all M functions are performed simultaneously with the movement indicated on the line. The transition to the next line occurs only after the completion of functions and movements. The **NOWAIT** prefix is valid only for the given line.

If a line with an M function appears with a sequence of lines with movements, the movement is stopped before the M function is started.

To illustrate, here are some examples of calling M functions:

```
Examples of M functions calls:

Example 1.
  N10 M3 S10000

Example 2.
  N10 $2M3 $2S10000

Example 3.
  N10 M191 MODE=1

Example 4.
  N10 M150(1,0,0) MODE=1

Example 5.
  N10 NOWAIT M60 M62 M70
```

4.14.2. Standard M-functions

M0 The CNC program stops as soon as the CNC line has been processed. The program can continue after pressing START+.

M3 Turn the spindle clockwise. The function is considered complete when the required speed is reached.

M4 Turn the spindle counterclockwise. The function is considered complete when the required speed is reached.

M5 Spindle stop. The function is considered complete when zero speed is reached. For positioned spindles, it automatically stops in the zero position.

```
Example: {Spindle}
N10 M3 S500      {start spindle}
N20 G1 X... Y...
N30 M5           {stop spindle}
N50 C45 F300     {positioning the spindle as a C-axis}
```

M17 End of subroutine. This function causes the subroutine to return to the calling CNC program.

M30 End of program. This function interrupts the running of the CNC program. The system automatically switches to the READY state.

4.14.3. Machine dependent M–functions

The list of applicable M functions depends on the specific machine and is given in the operating instructions.

4.15. Overview of G-functions

Flag meaning:

- ! active when the program is started
- & the function is effective for only one line
- § modal

Group		!	&	§	Description
1	G0		●		linear interpolation at a rapid positioning speed
	G1	●	●		linear interpolation
	G2		●		circular interpolation CW
	G3		●		circular interpolation CCW
2	G17	●		●	X-Y plane selection
	G18			●	X-Z plane selection
	G19			●	Y-Z plane selection
3	G40	●		●	cancel tool radius compensation
	G41			●	tool radius compensation from the left
	G42			●	tool radius compensation from the right
4	G53	●		●	shift the coordinate system
	G54			●	shift the coordinate system
	G55			●	shift the coordinate system
	G56			●	shift the coordinate system
	G57			●	shift the coordinate system
	G58			●	shift the coordinate system
	G59			●	shift the coordinate system
5	G92		●		setting the desired position
	G93	●	●		cancel G92
6	G60	●		●	passage with braking ramp engagement
	G64			●	passage without loss of speed
7	G11		●		the target is a tool holder in MCS
	G12		●		the target is tool tip in MCS
	G13		●		sequential tool movement in WCS
8	G90	●		●	absolute positioning
	G91			●	incremental positioning
9	G79		●		machine cycle call
	G200 - G399			●	Define a machine cycle to call with G79 function

Only one G function from a group can be selected at a time. All modal functions remain effective until a new function is programmed for the same group.

4.16. Overview of standard M - functions

M	Definition
0	programmed stop
3	rotating the spindle clockwise
4	rotating the spindle counterclockwise
5	spindle stop
17	end of subroutine
30	end of program / reset

5. 5. Basic actions

This part of the user's guide is intended to give you a quick guide when you need to perform a specific operation and, if necessary, to guide you to one of the previous chapters of the text.

5.1. System states

Recapitulation of system states:

Status	Message
Power supply to the power part switched off	Power OFF
Initiation after switch-on the power part	Initialising
The system is in order, some axis is not referred	Not referred
Referring is in process	Referencing
The system is ready, referred	Ready
Program is in process	Running
Program suspended by M0 function	Stopped by M0
Program suspended by STOP button	Stopped
Manual shift of axis is in process	Manual move
Spindle started manually	Manual bin
Loading of the program is in process	Loading: xxxxxxx
Power part powered; movements blocked	Door open / Blocked
Error condition	Error

see also *2.1 Status*

5.2. Switching on the power supply of the power section

When started, the Cnc886 program reports the status **Power OFF** in the status window. In this state, it expects you to press the hardware START/Motors On button on the electrical cabinet or control panel. When pressed, the system enters the **Initialising** state and then the **Not referred** state. If something is wrong, it may also be in the **Error** state.
The **Power OFF** status is also reported after pressing Central Stop. Even in this case, it is necessary to press START to restore the power supply to the power section. However, note that Central Stop is a lock button. It is therefore necessary to release it in advance.

5.3. Entering the protected zone

Machines have different ways and degrees of protection against human entry into the dangerous areas where the machine or tools move. These subsystems usually work with the control system. Through the control system, the operator requests entry into the protected zone if necessary. The system puts the machine in a state that allows entry. This is often the case when the drives are switched off. However, such a shutdown practically always leads to a servo fault condition without prior preparation.

The Cnc886 system is ready to work with security subsystems. Usually, the following procedure is performed:

1. The operator requests access with designated button.
2. The system puts the drives to a standstill so that loss of power does not cause an error. The power supply (or servo safety signal) is disconnected either by the system or by breaking the zone, which is not a fault in this state.
3. Machine signals that it is in safe-to-enter condition. The operator can enter.
4. After leaving the zone, the operator restores the safety lock of the zone to restore power to servos and allow further operation.

The state when the system is secured is signaled as "Door open", or other text suitable for the machine configuration.

5.4. Axes referencing

Referencing starts in the Manual-Referencing tab (see *2.4.2 Manual Referencing tab*). Go to the Manual tab, then the Referencing tab. Select the axis to reference, or All axes to perform referencing of the whole machine. Press the START+ button on the screen to start the process of referencing. You can interrupt it by pressing RESET.
Referencing can only be started from the Not refered and Ready states.

5.5. Manual control of axes and spindles

Manual axis movements are started in the Manual Move tab. (See *2.4.1 Manual Move tab*) Go to the Manual tab, then to the Move tab. In the **Sel** column, select the axis. Press the START+ key to move the axis in the positive direction. The movement continues as long as you hold down the button. When released, the axis stops. The opposite direction of movement is started by pressing the START- button.
The traversing speed can be selected in multiples of the value specified by the machine constant for each axis. The multiples are **10x**, **1x**, **0.1x** and **0.01x**. Selected with on-screen buttons.
If the axis has been referenced, the system will not let you go outside the limits set by the machine constants in the configuration file. If the axis is not referenced, the check is not performed, and the operator must take extra care. Approaching the emergency limit sensor can bring unnecessary complications.
Manual feed can only be started in the Not refered, Ready, and Manual bin states.

The spindles are started by pressing the **FWD** and **REV** buttons (forward or backward direction). The system enters the Manual bin state. The spindle rotates and rotates even after the press is completed. The speed is adjusted with the **Up** and **Dn** buttons. To stop the spindle, press RESET.
The spindle can also be started in Stopped and Stopped M0 mode. To stop the spindle without switching off the running program, press STOP.

5.6. Manual control of machine specific functions

Other machine functions are started in the Manual-Functions tab. (see *2.4.3 Manual Functions tab*) Go to the Manual tab, then to the Functions tab. A tab with a set of buttons appears on the screen. The individual buttons call up the individual functions. Their number and significance is determined by the specific machine design.

5.7. Loading technological programs and their management

Loading of technological programs and their management is performed in the Library tab. (see *2.5 Libray tab*)
Loading and managing can be done in the Power OFF, Not referred, and Ready states.

Load a program from disk.
First, select the program to load in the **On disc** window. Then press the **Load** button. After loading and displaying, the system returns to its original state. If the load was successful, the program is ready to run.
You can also load the program by dragging it from the Explorer window to the Library tab.

Select a program to run.
If you want to run a program other than the last loaded one, select it in the **In memory** window and press **Activate**.

Delete a program
First select the program to be deleted in the **On disc** or **In memory** window, then press Erase / Delete.

Save the program to disk
First select the program to save in the **On disc** or **In memory** window, then press **Save as…**.

Create a new directory
First click in the On Disk window and then press **New Dir…**

Load from another source
is done by pressing the **More >>** button. The system will offer loading from available local and network drives. It also offers reading via the "Load from RS232" serial line. This option has been retained for compatibility with older versions of the system. The system enters the Loading: state. Now upload the program from the computer where you created it. After loading and displaying, the system returns to its original state. Loading can be interrupted by pressing RESET. If the download was successful, the program is ready to run.

5.8. Modifying corrections and parameters

The setting of corrections and parameters and the management of files with corrections and parameters is performed in the Param tab. (see *2.6 Param tab*)
Settings and management can be performed in the Power OFF, Not referred, and Ready states.

5.9. Acknowledging error message

If an error condition occurs in the system, the running program is terminated, and Error is displayed in the status window. Go to the Errors tab. At the bottom of the screen, you will find a window describing the error that occurred. In case of an error that requires your intervention (for example, a tripped circuit breaker, an amplifier error, etc.), eliminate the cause. Then press the **Hmm** button to confirm the error. If your intervention was not successful, the error will reappear. If successful, acknowledging all errors will put the system into a standstill state.

5.10. Clearing the servo errors

If the cause of the error condition was an error of a servo amplifier, you must reset it before acknowledging the error. The **Clr** button in the Axis Service tab is used for this purpose. (see *2.7 Service tab*). Enter this tab. In the Err column, the servos with error are marked in red. After pressing the Clr button, the amplifier reset process starts. It can take approx. 5 seconds to perform. If successful, it would change the red mark in the Err column to gray. Then you can acknowledge the errors in the Errors tab.

5.11. Running technological program

In order to run the technological program, it must first be loaded into memory from the serial line or from disk and activated. (see *2.5 Libray tab* and *Loading technological programs and their management*)

If a program is activated, it is possible to start it with the START+ button. However, the system must be in the Ready state. The START+ button allows you to start the program from all tabs except Manual Move and Manual Referencing.

5.11.1. Program test

The program can also be run in a mode in which no machine action is performed, but only interpreted and tested for syntactic correctness and correctness of the coordinate system settings. During the test, all required corrections as well as the axis limit position test are active. If the test is run without error, a program error cannot occur during normal operation with the same setting. At the same time, after the test, you can find the product dimensions in the Errors tab, (see *2.8.1 More information in the Errors tab*). We set the program test by selecting the Test mode in the Automatic tab.

5.11.2. Measuring the duration of the program

By setting the Duration mode in the Auto tab (see *2.3 Auto tab*) and running the program, we get a very good estimate of the program execution time. A list of information after the end of the run can be found in the Errors tab, (see *2.8.1 More information in the Errors tab*)

5.11.3. Starting from the line resp. from position

The program can also be started from the specified program line. The program runs from the beginning like in Test mode, in which no machine actions are performed. After finding the set line, the system offers to execute M functions that are active according to the program and then offers to approach the starting position. This is the position the system would reach at the end of the previous line. After the start, the program is in the Stopped state, i.e., as after pressing STOP. You can continue by pressing START+. Then the program continues normally.

Set starting line by selecting the **From** mode and fill the field for the line number in the Auto tab. (see *2.3 Auto tab*)

We often need to run the program from the selected line after its interruption by an error, disturbance of the work zone by careless operator or after pressing RESET for various reasons. We then need to know on which line the program stopped. The system automatically prepares the field for setting the line number with the correct value in the event of a program interruption. Information about the line can also be found in the event log in the Errors tab and therefore also in the ErrorLog.txt file

As with the line, some machines can be started from the current machine position. The program then finds the first position in the program close to the current position.

5.11.4. Execute the program line by line

Step mode allows you to execute the program line by line. In this mode, after executing a line, the program pauses and the STOP button on the button panel lights up. The system is in the Stopped state. The next line is executed after pressing START+.

Step mode is on when the **Step** button in the Auto tab is pressed. This button also turns the mode on and off. (see *2.3 Auto tab*) Step mode can be entered at any time, but it is convenient to enter it from Ready or Stopped mode.

5.11.5. Program suspension

To suspend the program, press the STOP button on the button panel. The system is in the Stopped state. The program resumes running after pressing START+. The STOP button allows you to suspend the program from all tabs.

Suspending causes the axis movements to stop. The spindle and M functions remain running.

5.11.6. Adjusting the feed rate

The feed rate of the tool during program run is given by the request from the program multiplied by a coefficient, which can be set in the Auto tab. (see *2.3 Auto tab*) You can set the value from 0% (0) to 150% (1.5). It can be changed before start as well as while the program is running.

5.11.7. Spindle speed adjustment

The spindle speed during program run is given by the request from the program multiplied by a coefficient, which can be set in the Auto tab. (see *2.3 Auto tab*). You can set the value from 50% (0.5) to 150% (1.5). It can be changed before starting as well as while the program is running.

5.12. Resetting the technological program

The technological program is reset (stopped without possibility to resume) by pressing the RESET button on the button panel. The RESET button allows you to stop the program from all tabs. All movements are stopped, the spindle is switched off and all functions are performed, like at the end of the program. The program cannot be resumed. It can only be restarted from the beginning, from a position or from a line.

5.13. Work in service mode

Working in service mode is enabled by the Service Binars tab. Go to the Service tab, then to the Binars tab. A description of working with the tab is in chapter *2.7.3 Service Binars tab*.

Remember that when manipulating binary outputs, the system cannot check the meaningfulness and security of your interventions!

5.14. License check and settings

In the Service Axis tab, click the **License** button to display a dialog that shows how long the license is valid. At the same time, you are offered to enter another dialog, which allows you to enter code that modifies the license. (see *2.7.1.1 Checking and setting up the license*)

5.15. Switching off the system

Shutting down the system is done in several steps. You must first end the Cnc886 operation. Thus, to complete or interrupt the execution of the technological program, editing, etc. Then it is possible to quit the Cnc886 program by clicking on the cross in the upper right corner of the window. It is then possible to shut down the Windows operating system.

Finally, it is possible to switch off the power with the main switch.