# JCNCScreen

Intro

JCNCScreen is a frontent for linuxcnc (http://linuxcnc.org/). Linuxcnc consists of several processes that control a cnc-machine, i.e. a mill.
Of cause, linuxcnc offers frontends too, but none of them suits my needs.

I'm quite outdated, so I need magnifiers to use a default frontend like axis. But in shop I don't like to wear glasses. But that was only one reason to start this exhausting adventure.

The most important reason so far was, that distance to go values beside tip location is not optional, but mandatory – at least for me.

So I started the journey and now I proudly present:
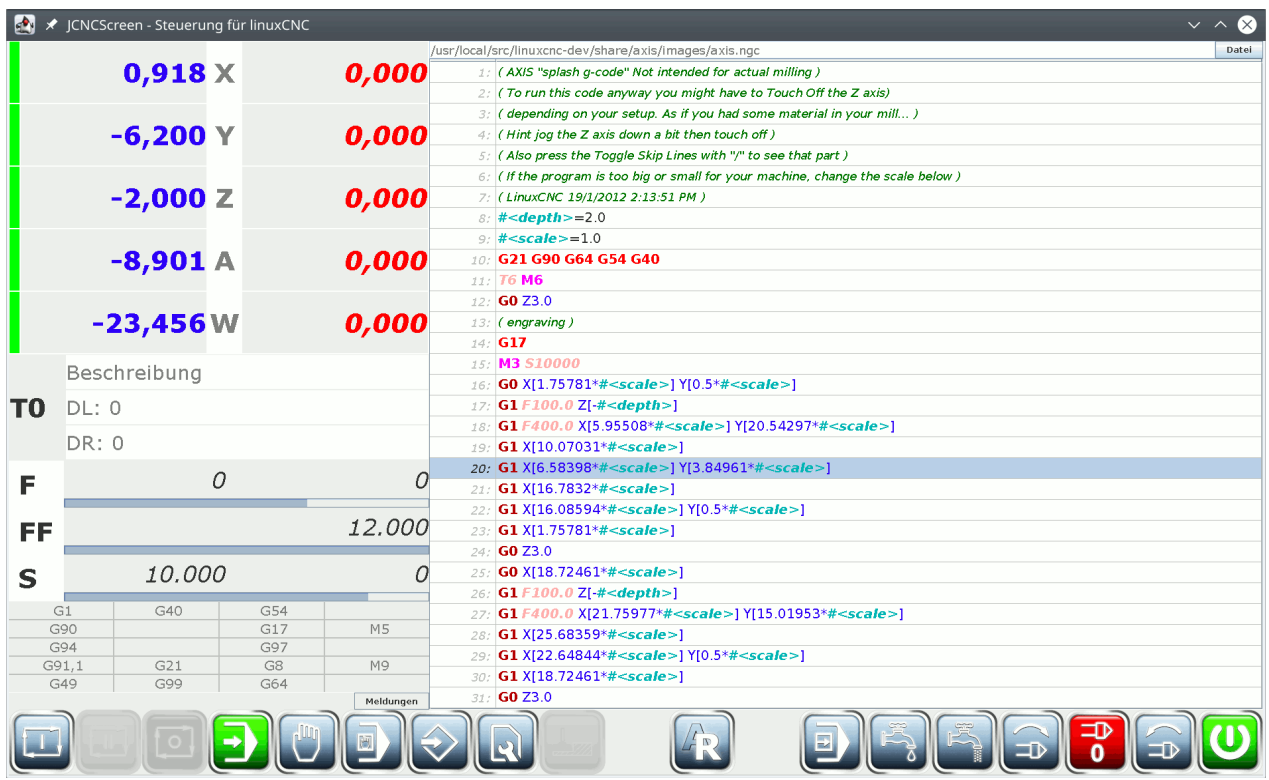
The blue numbers show the tip position and the red numbers are the distance to go for each axis. If relative positions will be changed to absolute values, than the blue numbers turn into green.

On the bottom line you'll find symbol-buttons, that turn application into different mode, or that execute certain actions.
JCNCScreen was designed to be the only active application at the machine. So it uses the whole screen (1920x1200).
The application checks the orientation of the screen and turns itself accordingly into portrait or landsape mode:

JCNCScreen
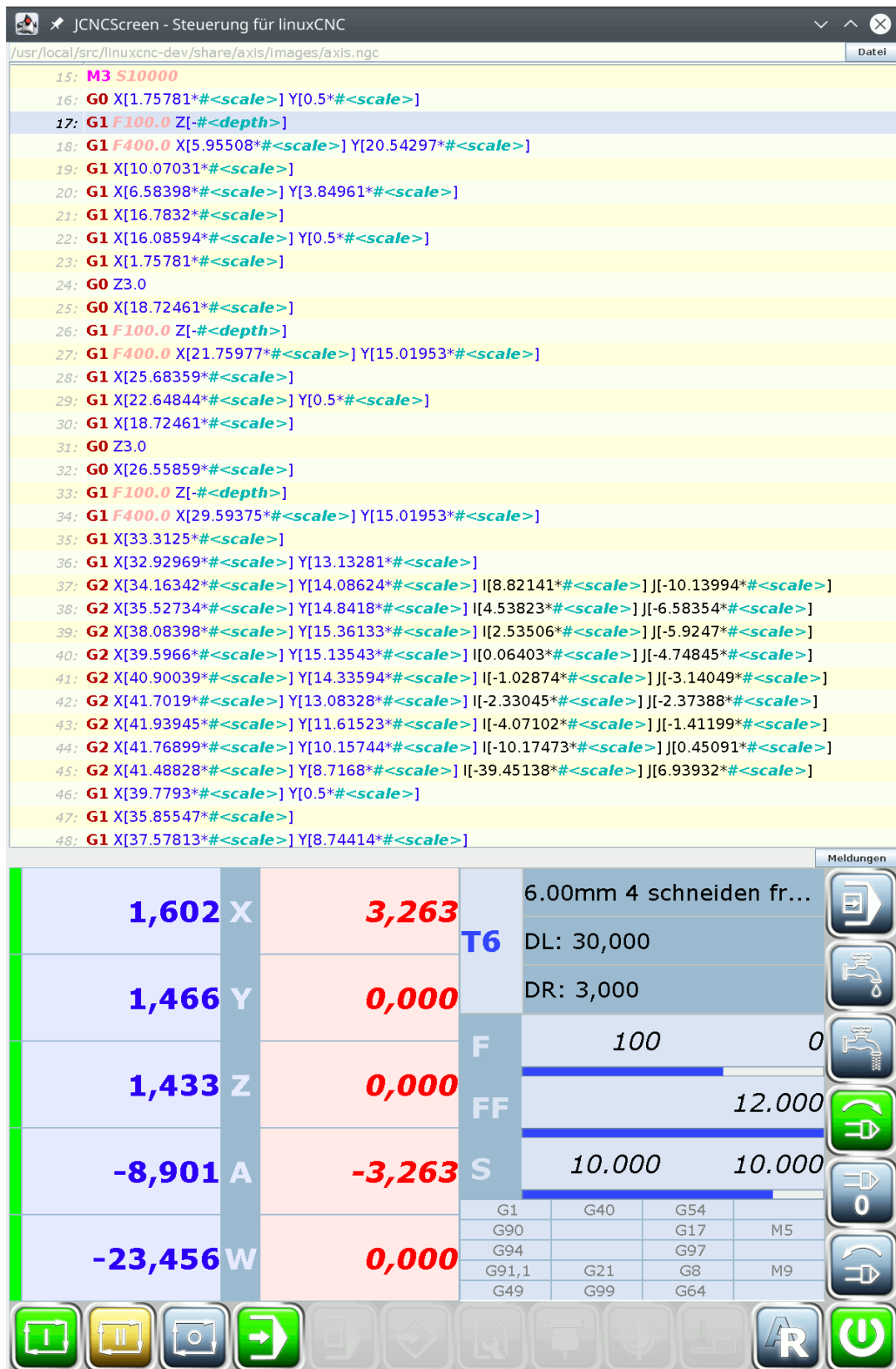
The application can switch into fullscreen mode by commandline parameter (-full). Then the window decorations disappear and you cannot activate another window any more.

JCNCScreen

As you can't argue about taste, application is customizable, or themeable. For demonstration purpose I provided a darkmode theme.

You can activate the darkmode by commandline switch (-theme dark).
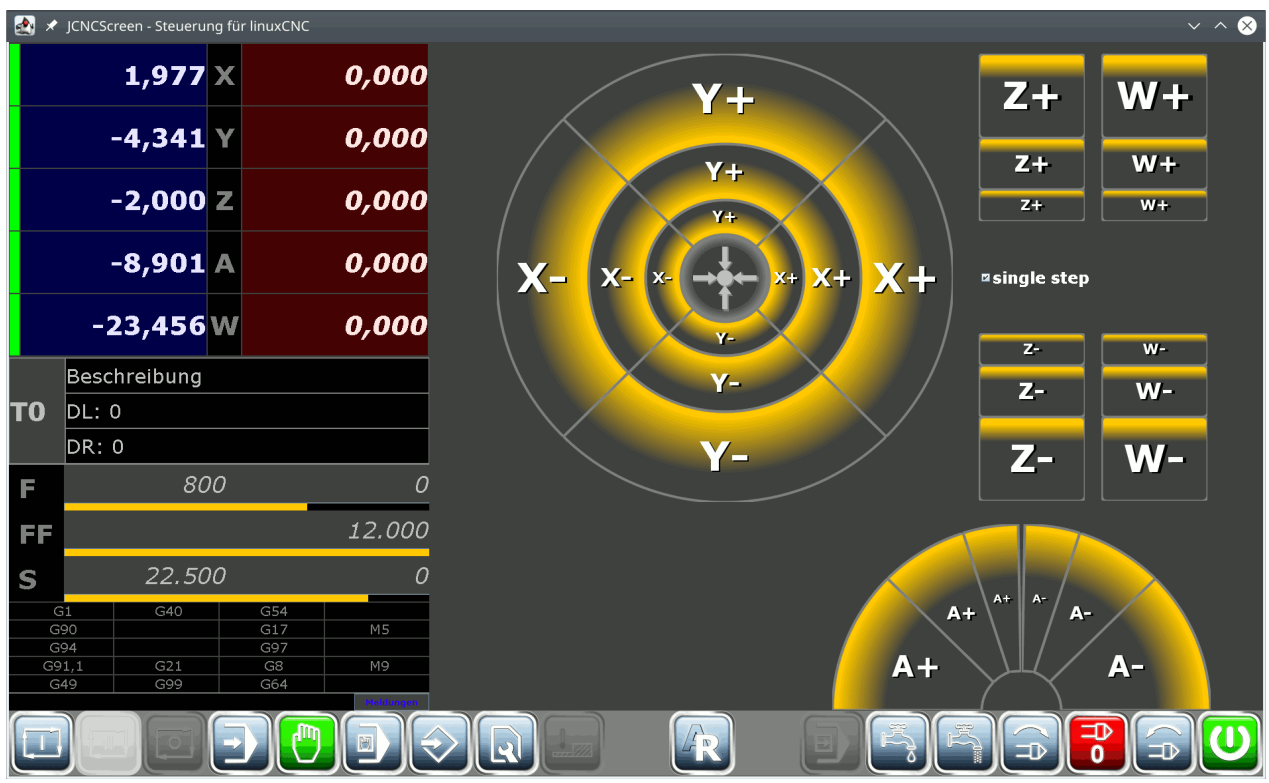

When you start the application from linuxcnc in simulation mode, all symbol buttons at the bottom are disabled.
First you have to turn off the emergeny stop (power button turns into yellow), and then you have to turn on the machine (power button turns into green).

Stil the most symbol buttons keep being disabled, cause the axis need to be homed first.

JCNCScreen

Its the symbole with the 4 arrows pointing towarе the central dot – in the middle of the circle with X- and Y-jog buttons. Control-leds in front of the position fields shine red. When an axis is homed, the led turns into green.

JCNCScreen

Application-modes

	Auto	When this button shines green, you can execute gcode-commands from file
	Manual	In the mode, you can move (jog) single axis by pressing a button. Two modes are available: single step and continuous jog – they can be toggled by a checkbox
	MDI	Manual data input – in this mode, you can enter single commands manually and execute them.
	Edit	Edit gcode-command files. The editing is separated from the file execution (auto mode), so different files can be loaded at each gcode-lister. The root of the gcode-files will be taken from ini-file of linuxcnc. That directory is the root for the filemanager. All other files from the box are invisible to JCNCSreen.
	Settings	Here you can change look of screen elements, find the tool-manager, tooltable for linuxcnc, fixture settings, ...
	Touch	Helps on measuring workpiece fixtures (with electronic sensing device) **This page is not ready yet**

JCNCScreen

Aktions:

	Start	Active in auto and manual mode and serves to start or continue command execution.
	Pause	Is active as soon as commands are executed. You can interrupt the execution, but linuxcnc backend stays active in between execution of a command. So you can continue by pressing start-button.
	Stop	Is active as soon as commands are executed. Opposed to „pause"-button „stop" will abort execution of commands without the chance to continue.
	Flood cooling	Is active at auto- and manual mode
	Mist cooling	Is active at auto- and manual mode
	Spindle CW	Turn spindle on in clockwise direction
	Spindle Stop	Stop turning spindle
	Spindle CCW	Turn spindle on in counter clockwise direction

JCNCScreen

Switches:

	Single-Step	Additional switch for auto-mode. If you're unsure about you gcode file, you can turn single-step on. Linuxcnc will then execute only one motion command and wait for you to press „start"-button again.
	Relative Position	The axis position show the tooltip position relative to the active workpiece origin (G54, G55, whatever has been commanded)
	Absolute Position	The axis position show the tooltip position in absolut / machine offsets (as if G53 would have been activated)

## A closer look:

The auto-mode:

Before you can activate this mode, all axis need to be homed. On the upper right you can start the file-manager to load a gcode-file.

JCNCScreen

By pressing „Start" the execution of the commands from the loaded file begins. File in the list advances automaticalle, so that the current command is always highlighted and visible.

With „single-step" activated only one motion command will be executed. Then linuxcnc backend waits for the next press of the „start" button.

Manual command input (MDI)
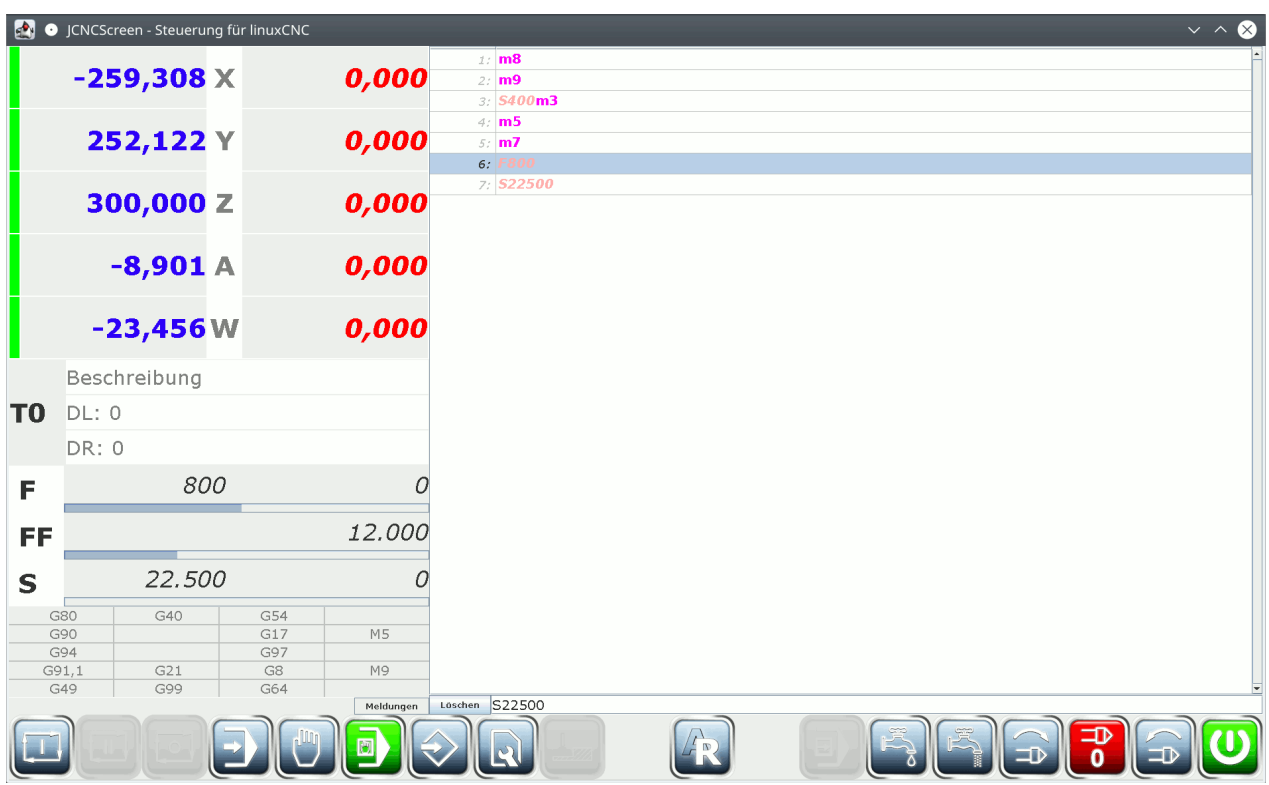
In the edit-line (to the right of the „delete" button) you can enter single commands, or commands that can work together in one line.

Pressing „Start" will execute that commands and save them into the list. The command history will be saved at shutting down the application, so you can use them again on next startup.

Moving the cursor in the list of command will copy the highlighted command into the edit-line. So you can execute it again.

If the list contains too many commands, you can remove entries from the list by pressing „delete"-button.

JCNCScreen

Edit gcode-file:

Usually gcode-files will be created by using a CAM or editor. But – sometimes small changes are required just in time …

For not having to swith to another application, a small editor is integrated.

At the top right a filemanager can be startet to load and/or create a file. As already mentioned, the filemanager is restricted to the directory from the ini-file of linuxcnc.

JCNCScreen
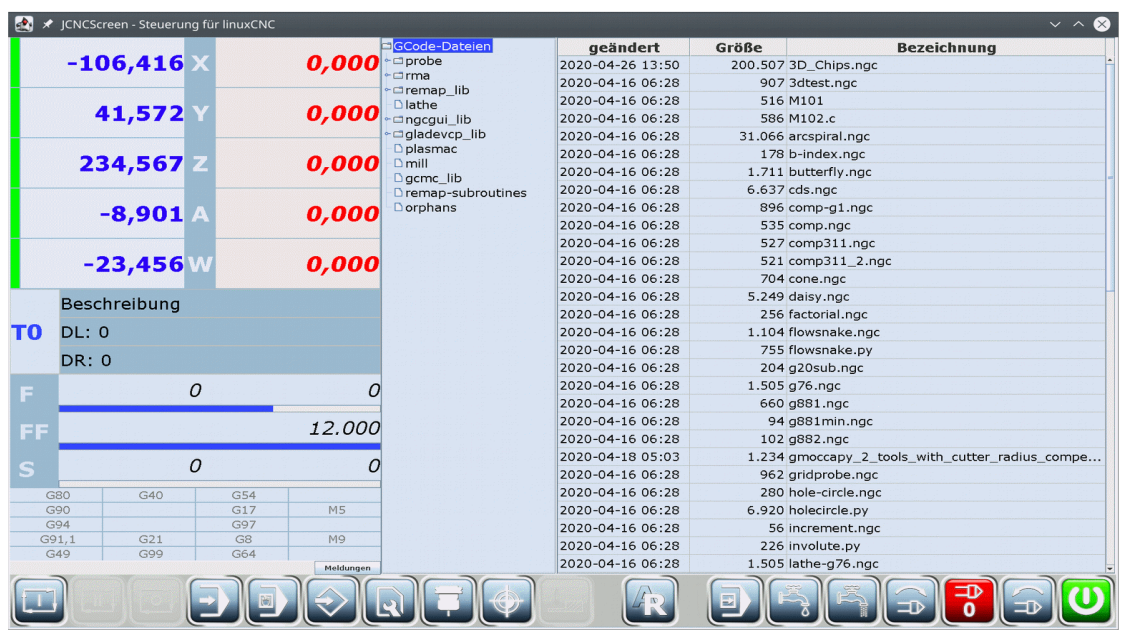
To organize Gcode files, you can create directories from the filemanager. Start the filemanager and when the tree is active (cursor motion moves highlighted entry in tree, not in list) you can press INSERT-key and a dialog pops up, where you can enter the new name.

TAB changes between tree and list. Even it the filelist is empty (on new created directories), pressing INSERT will create a new file. As with directories a dialog pops up, where you can enter the new name.

Any file from list can be loaded into editor by pressing ENTER or with double-click of a mouse.

The editor is a line-editor, which means, moving the cursor copies the highlighted line into the edit-line, where it can be changed.

Hitting „replace" replaces the highlighted line with the content from edit line. Hitting „insert" adds the edit-line to the file.

To ease editing of bigger files, you can search for text. Enter the search pattern into the edit line and press „search".

When you change a file, its filename turns into a brighter color (above the list). Beside the filemanager button there's a button to save a changed file. After loading a file, this button is disabled. It will be enabled on the first change.

JCNCScreen

Tooltable (of linuxcnc):

Displays a list of tools, known to linuxcnc. (tool-table file is taken from ini-file of linuxcnc). You can sort the tooltable by clicking at the column-header.

If you need to change length or diameter of a tool, select the tool from list. That will copy tool-properties to the little tool-editor below the list. Length is „L:" and „D:" stands for diameter.

Pressing „Save" sends the new properties to backend of linuxcnc.

Tool-Manager:

If you have a spindle with toolchanger collets, you might want to manage your tools from outside the machine too. That's the purpose of tool-manager.

Any tool mounted in a collet keeps its properties even after unmounting the tool from spindle.

Tool-manager works the same way as the file-manager:

JCNCScreen
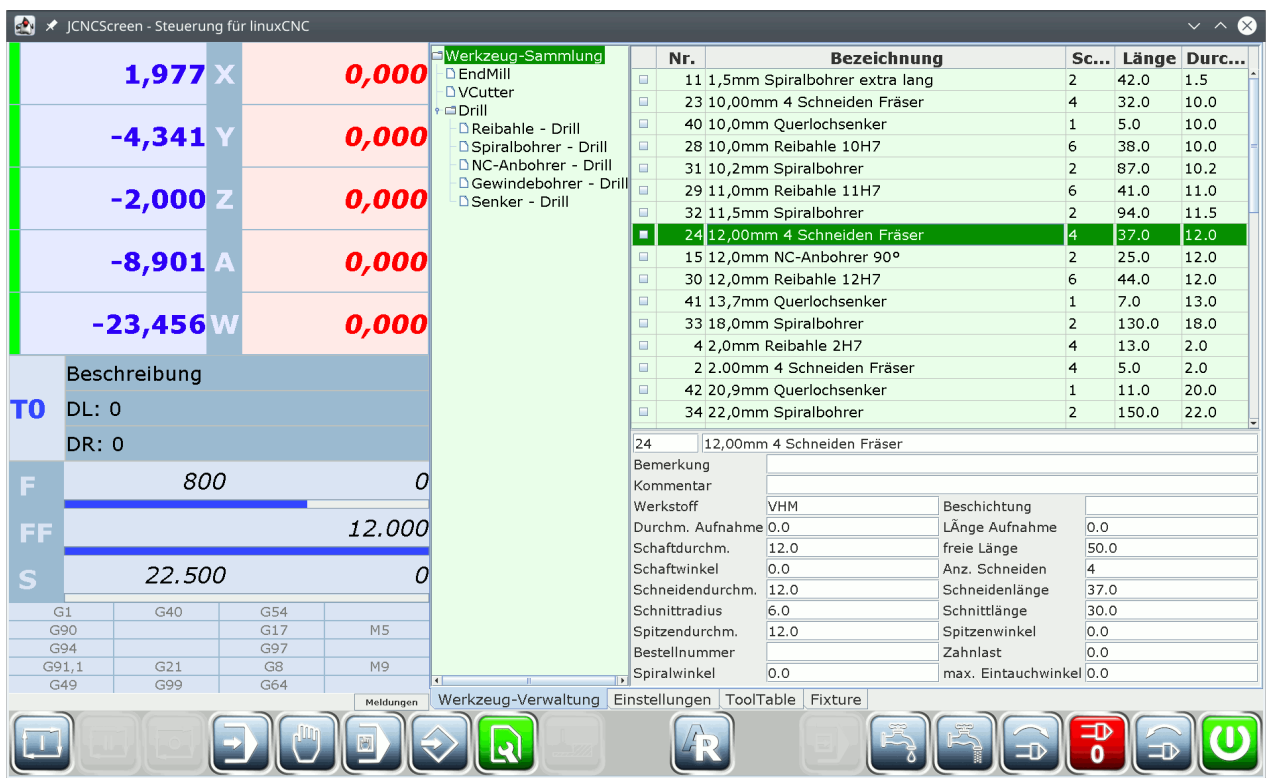
From the tree pressing INSERT will create a new tool category. TAB toggles between tree and list. Pressing INSERT from the list, creates a new tool-entry.

Below the tool list you'll find the tool-editor.

You start the tool editing by pressing ENTER on an existing tool, or by INSERT from the tool list. In tool edit mode, TAB is limited to the entryfields from tool-editor. You need to explicitly end the tool editing.

F10 saves your changes and leaves tool editng, F8 ends tool editing without saving your changes.

The tool-manager uses a database for storage, so you don't have to care about filenames and/or file location.

JCNCScreen

See the known tool propertis supported by the file-manager:
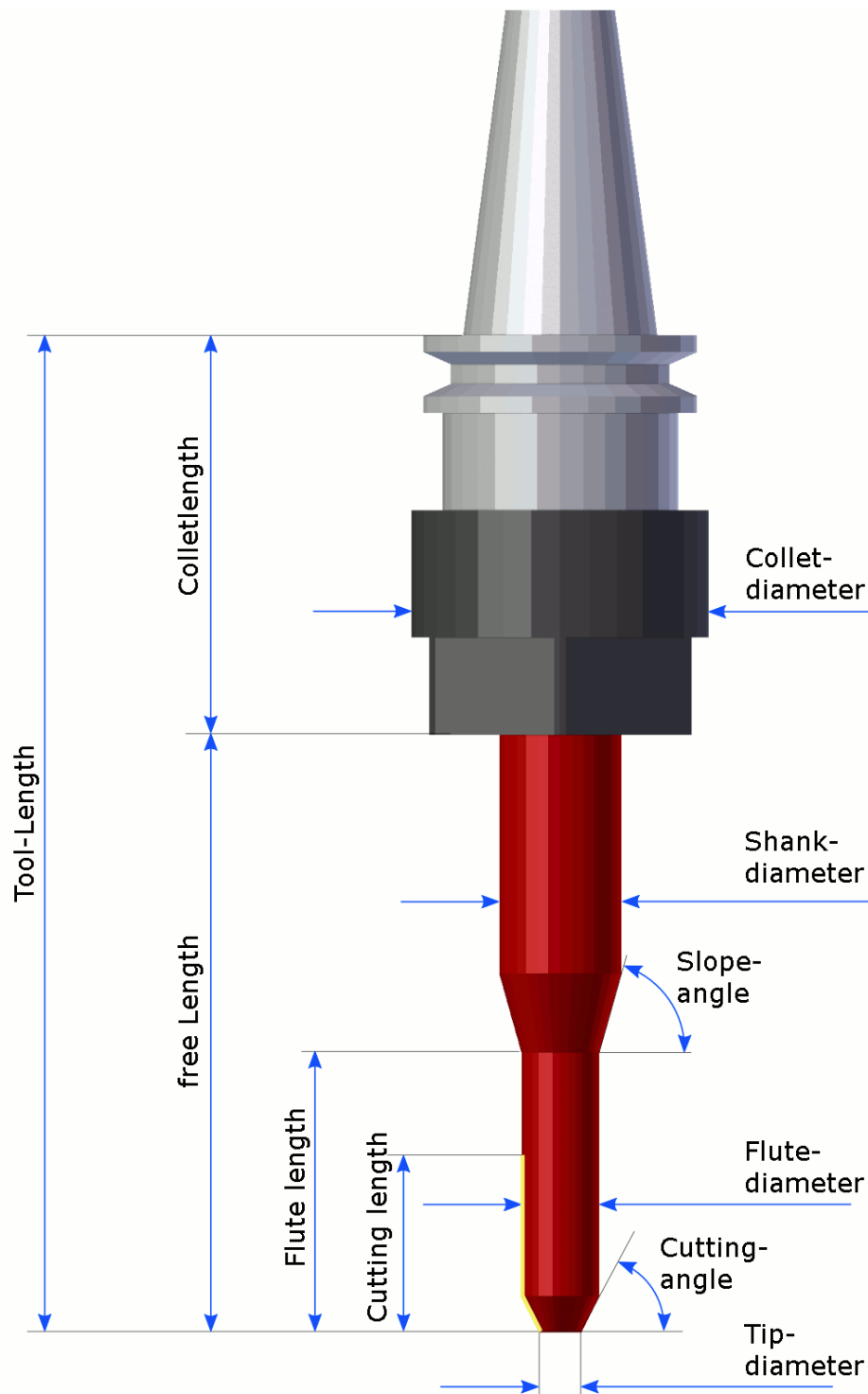
JCNCScreen

The tool-manager does not have a direct connecton to linuxcnc. To be able to share toollists between multiple applications, JCNCScreen uses plugins for the export of tool-data.

One plugin exports the tool-data in the format for linuxcnc, so the file can be used by the linuxcnc backend as tooltable. Another plugin exports the tooldata to CamBam and another plugin creates an XML-file with all supported properties. The latter can be used as a backup of the database too.

Of cause it is not very useful to export all tools at once. Therefor each tool-entry has a checkbox, where you can select a tool. When you move the cursor through tool-list, pressing Spacebar toggles tools selection state (select or deselect it).

When ready with selecting desired tools, press F12 to start the export.

Application looks at the plugin-directory for the installed plugins and offers a dialog to choose the export-plugin. Addtionally the file name and location can be choosen.

JCNCScreen

Fixtures:

linuxcnc supports several coordinate system, or workpiece origins. I.e. you could measure origin of several workpieces and put each into different coordinate system.

G53 means that the machine position should be used (no active workpies origin). G54 is the first workpiece origin, G55 the second, ...

Beside the workpiece origin a common offset can be entered. This will be useful to get rid of differences between measured origin and real milled path.

Offset-values are added to all workpiece origins at the same time.

Each workpiece origin can be transmitted separately to linuxcnc backend.

JCNCScreen

Errorlog:

As errors are human, it would be nice to know, what was the situation or command sequence that leaded to the error?

For that purpose every action gets recorded by JCNCScreen, no matter whether its a user message, an error message or a user command.

To display the messagelog, all application-actions will be disabled. First you have to close messagelog.

Messagelog will be saved at application shutdown, so you have the messages in the list on next startup.