

# MyCNC Screen Configuration

## MyCNC GUI elements

Label

Display

BDisplay

RadioDisplay

KDisplay

KSpinBox

KSpinBox2

LED Display

SVG Display

Myitems Widget

Button

A PushButton screen item can be used on myCNC screen. A typical button definition is shown below

```
<gitem where="main"
position="160;0" width="80" height="80"
image="button-refresh" action="file-refresh" type="button" />
```

Attributes description

- **where** - parent widget name the button placed to
- **position** - X and Y position of the button inside a parent widget
- **width** - button width in pixels
- **height** - button height in pixels
- **type** - type definition of the item (should be type="button")
- **image** - a image (icon) file in SVG format for the button
- **action** - action for the button (which procedure will be executed if event triggered)
- **event** - event type the button is sensitive to. Event can be

- **pressed** - action is executed if the button pressed
- **released** - action is executed if the button released
- **both** - there are separate actions (";" semicolon separated) for **pressed** and **released** events
- **skinbase** - besides the Image file for each button there is a common **skin** SVG file for all the buttons. For selected buttons skin file can be redefined with **skinbase** attribute. This SVG file will be used as a bottom layer for the button image.

## XButton

XButton is a Push Button with built in Light indicator. Light is mostly used to show a current state of CNC controller input, output pin, CNC global variable register value etc.

A XButton XML definition example is shown below

```
<gitem where="x-mill"
  xattr="3;3;16;16;led;red;round"
  address="outputs" number="#OUTPUT_SPINDLE".
  position="720;10" width="70" height="70"
  image="M/button-m03" action="plc-run:M03/#5524"
  type="xbutton" />
```

### Attributes description

- **where** - parent widget name the button placed to
- **position** - X and Y position of the button inside a parent widget
- **width** - button width in pixels
- **height** - button height in pixels
- **type** - type definition of the item (should be type="button")
- **image** - a image (icon) file in SVG format for the button
- **action** - action for the button (which procedure will be executed if event triggered)
- **event** - event type the button is sensitive to. Event can be
  - **pressed** - action is executed if the button pressed
  - **released** - action is executed if the button released
  - **both** - there are separate actions (";" semicolon separated) for **pressed** and **released** events
- **skinbase** - besides the Image file for each button there is a common **skin** SVG file for all the buttons. For selected buttons skin file can be redefined with **skinbase** attribute. This SVG file will be used as a bottom layer for the button image.
- **xattr** - defines light position, size and attributes - semicolon separated
  - X,Y position of the led/light inside the button
  - width and height of the led
  - type of light (actually this field is always "led", reserved for future options)
  - shape of the standard led light. Standard shape can be "round" and "rect"
- **address** - defines CNC controller hardware the light attached to
  - "inputs" - the light attached to input pin
  - "outputs" - the light attached to output pin
  - "number" - defines pin number the light attached to. A number can be assigned directly, for example

number="0" for pin #0  
 number="15" for ipn #15  
 or through pin definition file used in Hardware PLC - "pins.h". In this case sign "#" and the pin name defined in "pins.h" should be instead of pin number. For example  
 number="#OUTPUT\_SPINDLE"  
 and "pins.h should contain this name definition, for example

```
#define OUTPUT_SPINDLE 7
```

## GLView

## NCView

2D Visualization widget.

- Basic setup.

```
<gitem where="olicut" name="ncview" type="ncview"
  position="100;100" width="880" height="440" />
```

- Advanced setup.

```
<gitem where="olicut" name="ncview" type="ncview"
  position="100;100" width="880" height="440"
  singlepath="yes"
  bgColor="#00E0E0"
  ColorT0="gray:1"
  ColorT1="#D000D0:5"
  ColorT2="#D05000:5"
  ColorT3="red:2"
  ColorT4="green:2"
  ColorT5="#0000D0:2" />
```

## Logview

## NCList

## CentringView

## Rotation2View

## MyCNC Actions

All button-like screen components (button, xbutton, bdisplay, kspinbox2) runs **Handler** procedure when "Pressed" or "Released" event activated by mouse click or touch screen tap. The Handler

defined in “action” attribute of xXML configuration item.

List of actions is shown below

Action Name	Description
advanced-config	
basic-config	
toggle-button-coordinates	
toggle-button-jog-enable	
toggle-button-soft-limits	
toggle-button-mist	
toggle-button-flood	
toggle-button-spindle	
toggle-button-ccw-spindle-spindle	
toggle-item:	
laser-marker-	

### Player actions

Action Name	Description
player-play	
player-play-confirm	
player-play-edge	
player-play-edge-confirm	
player-jump1	
player-pause	
player-stop	
player-skip-forward	
player-skip-forward-10	
player-skip-backward	
player-skip-backward-10	
player-skip-part-f	
player-skip-part-b	
player-nc-reset	
player-nc-reset-confirm	
player-back-to-path	
player-back-to-path-confirm	
player-play-back	
player-nc-tie	
player-play-step	
player-play-step-backward	
mdi-play	
mdi-open	

### Jog actions

Action Name	Description
jog-0-plus, jog-1-plus, jog-2-plus, jog-3-plus, jog-4-plus, jog-5-plus, jog-6-plus, jog-7-plus jog-8-plus	Jog Positive direction in Axis X, Y, Z, A, B, C, U, V, W
jog-0-minus, jog-1-minus, jog-2-minus, jog-3-minus, jog-4-minus, jog-5-minus, jog-6-minus, jog-7-minus, jog-8-minus	Jog Negative direction in Axis X,Y,Z,A,B,C,U,V,W
jog-0-plus-1-plus	Simultaneous two-axes Jog X+ Y+
jog-0-plus-1-minus	Simultaneous two-axes Jog X+ Y-
jog-0-minus-1-plus	Simultaneous two-axes Jog X- Y+
jog-0-minus-1-minus	Simultaneous two-axes Jog X- Y-
jog-overspeed-inc jog-overspeed-dec	Increment/Decrement Jog Overspeed value (%)
jog-overspeed-set:	Set given Jog Overspeed value (%)

### Motion Settings Actions

Action Name	Description
step-float-inc step-float-dec	Increase/decrease Jog Step Size
step-size-inc step-size-dec	Increase/decrease Jog Step Size
motion-linear-acceleration-inc motion-linear-acceleration-dec	Increase/decrease Motion Linear Acceleration value
motion-linear-jog-speed-inc motion-linear-jog-speed-dec motion-linear-jogspeed-inc motion-linear-jogspeed-dec	Increase/decrease Motion Linear Jog Speed value

Action Name	Description
motion-jog-speed-x-inc, motion-jog-speed-x-dec, motion-jog-speed-y-inc, motion-jog-speed-y-dec, motion-jog-speed-z-inc, motion-jog-speed-z-dec, motion-jog-speed-a-inc, motion-jog-speed-a-dec, motion-jog-speed-b-inc, motion-jog-speed-b-dec, motion-jog-speed-c-inc, motion-jog-speed-c-dec, motion-jog-speed-u-inc, motion-jog-speed-u-dec, motion-jog-speed-v-inc, motion-jog-speed-v-dec, motion-jog-speed-w-inc, motion-jog-speed-w-dec, motion-jog-speed-xy-inc, motion-jog-speed-xy-dec, motion-jog-speed-xyz-inc, motion-jog-speed-xyz-dec	Increase/decrease Motion <b>Jog</b> Speed value for given Axis
motion-rapid-speed-x-inc, motion-rapid-speed-x-dec, motion-rapid-speed-y-inc, motion-rapid-speed-y-dec, motion-rapid-speed-z-inc, motion-rapid-speed-z-dec, motion-rapid-speed-a-inc, motion-rapid-speed-a-dec, motion-rapid-speed-b-inc, motion-rapid-speed-b-dec, motion-rapid-speed-c-inc, motion-rapid-speed-c-dec, motion-rapid-speed-u-inc, motion-rapid-speed-u-dec, motion-rapid-speed-v-inc, motion-rapid-speed-v-dec, motion-rapid-speed-w-inc, motion-rapid-speed-w-dec, motion-rapid-speed-xy-inc, motion-rapid-speed-xy-dec, motion-rapid-speed-xyz-inc, motion-rapid-speed-xyz-dec	Increase/decrease Motion <b>Rapid</b> Speed value for given Axis

Action Name	Description
motion-feed-speed-x-inc, motion-feed-speed-x-dec motion-feed-speed-y-inc motion-feed-speed-y-dec motion-feed-speed-z-inc motion-feed-speed-z-dec motion-feed-speed-a-inc motion-feed-speed-a-dec motion-feed-speed-b-inc motion-feed-speed-b-dec motion-rapid-speed-c-inc motion-rapid-speed-c-dec motion-rapid-speed-u-inc motion-rapid-speed-u-dec motion-rapid-speed-v-inc motion-rapid-speed-v-dec motion-rapid-speed-w-inc motion-rapid-speed-w-dec motion-rapid-speed-xy-inc motion-rapid-speed-xy-dec motion-rapid-speed-xyz-inc motion-rapid-speed-xyz-dec	Increase/decrease Motion <b>Rapid</b> Speed value for given Axis
motion-overspeed-inc motion-overspeed-dec motion-overspeed	Increase/decrease/set Motion <b>Overspeed</b> value (%)
spindle-speed-inc spindle-speed-dec spindle-speed	Increase/decrease/set Current Spindle Speed (S) for current operation. New "S" value in running g-code will overwrite Current Spindle Speed
spindle-speed-restore	<b>Restore Default</b> Spindle Speed (S)
spindle-overspeed-inc spindle-overspeed-dec spindle-overspeed	Increase/decrease/set Spindle Speed (S)

### Built-in Editor Actions

Action Name	Description
editor-arrow-up	
editor-arrow-down	
editor-ncline-edit	
editor-ncline-insert	
editor-ncline-remove	
editor-font-inc	
editor-font-dec	
editor-numbers-toggle	
editor-selection-begin	
editor-selection-end	
editor-selection-remove	
editor-selection-insert	
editor-new	
editor-save-as	

Action Name	Description
editor-save	

File Manipulation Actions

Action Name	Description
save	
open	
load-file:	
load-macro:	
file-close	
file-close	
dxf-import	
file-close	
image-import	

Visualization Actions

Action Name	Description
zoom-in	
zoom-out	
fit-to-view	
zoom-in	
zoom-in	
3dview-xy	3D Visualization: show XY view
3dview-xz	3D Visualization: show XZ view
3dview-yz	3D Visualization: show YZ view
3dview-iso	3D Visualization: show ISO view
3dview-custom:	3D Visualization: show Custom view, defined by Alfa,Beta,Gamma angles for rotation matrix

Widget Manipulations Actions

Action Name	Description
show-inputbox	
show-widget-centring	
show-widget-rotation	
show-widget-sawcutting	
show-widget-diagnose	
show-widget-config	
show-widget-edit	
show-widget-lib	
show-widget-tools	
show-widget-lof	
show-widget-report	
show-widget-support	
show-widget-user	



Action Name	Description
show-widget-work	
mywidget-show:	
mywidget-hide:	
mywidget-toggle:	
pendant-widget-open	
measure-surface	
toggle-virtual-keyboard	
mywidget:	

## Application Actions

Action Name	Description
close-application	Close myCNC Control software
cnc-config-save	Save myCNC configuration files to disk

## Hardware Manipulation actions

Action Name	Description
servo-pid-on	
servo-pid-off	
hw-pwm-inc	
hw-pwm-dec	
hw-dac-inc	
hw-dac-dec	
hw-direct-	
thc-arc-voltage-ref-inc	
thc-arc-voltage-ref-dec	
thc-jog-speed-dec	
thc-jog-speed-inc	
thc-jog-pos	
thc-jog-neg	
dev-thc-jog-pos	
dev-thc-jog-neg	
toggle-widget-centring	
toggle-widget-rotation	
toggle-widget-sawcutting	
toggle-widget-user	
toggle-button-flood	
toggle-button-coolant	
toggle-button-spindle	
toggle-button-ccw-spindle	
toggle-button-mist	
toggle-button-cv-mode	

## PLC Actions

Action Name	Description
soft-plc-run:	
soft-plc-stop:	
*soft-plc-stop:	
plc-run:	
direct-plc:	
multidev-plc-run:	
switch-cnc-gvariable	

## Job Actions

Action Name	Description
direct-run:	
direct-run-confirm:	
radio-confirm	
mode-cutting-ignore	
mode-cutting-accept	
mdi-save-list	
mode-show-ruler	
mode-show-dimension	
mode-show-workarea	

## OS/System Actions

Action Name	Description
system-cmd	Run system command
application-close	Close MyCNC application
application-close-confirm	Run confirmation dialog to Close MyCNC application
application-minimize	Minimize MyCNC Application

## CNC Variables manipulation Actions

Action Name	Description
direct-set-cnc-var	
cnc-variable:	
cnc-gvariable-dec, cnc-gvariable-inc	
cnc-variable-dec, cnc-variable-inc, cnc-variable-change, cnc-variable-toggle, cnc-variable-set, cnc-variable-clear, cnc-variable-vset, cnc-variable-switch	
cnc-vm-variable-dec, cnc-vm-variable-inc	
device-variable-dec, device-variable-inc	
item-switch:	
run-numpad:	
item:	
load-item:	
cnc-nvariable-	
cnc-nvariable:	

fake	Empty Handler Action. Nothing happens if run this action
myitem-value-inc, myitem-value-dec	Increment/Decrement XML Item value by name

HMI Actions

Action Name	Description
search-nc-position	Open dialog to define line number for <b>Run From Here</b> command
file-refresh	
row-n-column	Open <b>Row And Column Multiplication/Nesting</b> Dialog
dialog-rotate	Open Dialog for NC-program rotation
mypopup-show:	Show custom defined Popup widget
mypopup-toggle:	Toggle custom defined Popup widget
mode-jog-unlimited	Switch to Unlimited Jog Mode
show-simple-edit	Show simple text editor window (Ver #1)
show-simple2-edit	Show simple text editor window (Ver #2)
rotate-nc-last	Rotage NC program to previously defined angle
cnc-cutchart-save	Save current cutting data to Cutchart
cnc-cutchart-load	Load cutting data for selected Cutting Mode
run-from-here	<b>Run From Here</b> command - start job file from selected line

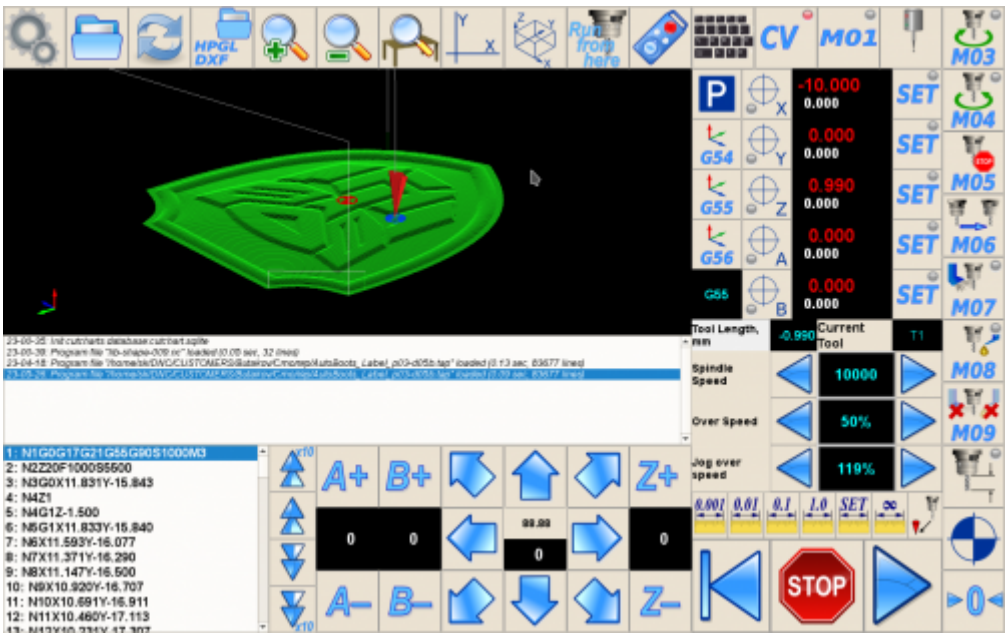
MyCNC screen configuration examples

[Add Rotation Dialog on Main Screen](#)

Add "Servo On" - "Servo Off" buttons

We need to add buttons for Servo On/Servo Off to **1280M5** profile

Original screen is shown below-



Put our new buttons instead of **CV** button

1. Find **cnc-screen.xml** configuration file in **1280M5** profile folder
2. Edit the file (in Midnight commander press F4 on the **cnc-screen.xml** file)
3. Find in the file section with **CV** button definition (F7, search "CV")

Here it is-

```
<gitem where="magic" position="960;0" width="80" height="80"
image="CV/CV" action="cnc-gvariable-toggle-5710"
xattr="60;4;16;16;led;red;round" name="display-gvariable-5710"
type="xbutton" />
```

4. Put under this button small popup window which will be a container for two buttons - **Servo ON** & **Servo Off**. Add **quick-popup-layout** section with popup window definition -

```
<quick-popup-layout>
  <current>popup-servo</current>
  <layout stretch="0" name="popup-servo" wa="80;160;right"
orientation="vertical" skin="skin/metal-01">
    <widget stretch="1" spacing="0" name="toolbar-servo"
orientation="vertical">myitems</widget>
  </layout>
</quick-popup-layout>
```

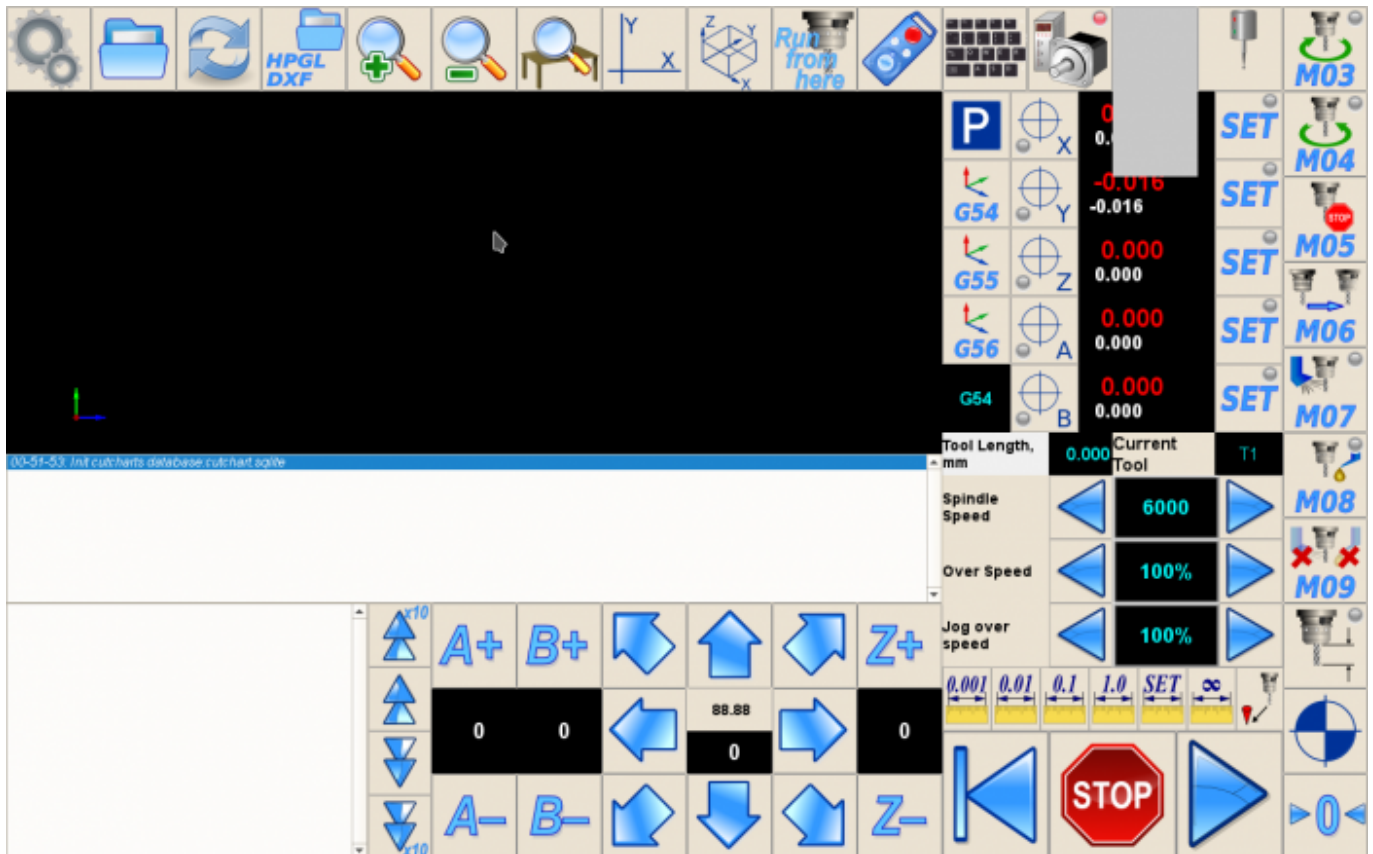
- Layout for Servo buttons named "popup-servo", size of layout is **80x160**, layout placed to the right side of the button that activates the popup (**wa** means "Window Attributes").
- Layout contains a Window named "toolbar-servo".

5. Fix **CV** button to show/hide **popup-servo** -

~~image="CV/CV" action="cnc-gvariable-toggle-5710"~~ image="motor/servo-driver-wide"  
action="mypopup-toggle:popup-servo"

```
<gitem where="magic" position="960;0" width="80" height="80"
image="motor/servo-driver-wide" action="mypopup-toggle:popup-servo"
xattr="60;4;16;16;led;red;round" name="display-gvariable-5710"
type="xbutton" />
```

6. Save **cnc-screen.xml**, restart the software and press new button wit **Servo driver**. Small popup grey colour window will be shown on the right side of the button.



7. Add two buttons definition in the popup window

```
<gitem where="toolbar-servo" image="motor/motor-start" action="plc-run:M62/3"
height="80" event="pressed-delay-1000" type="button"/>
<gitem where="toolbar-servo" image="motor/motor-stop" action="plc-run:M63/3"
height="80" type="button"/>
```

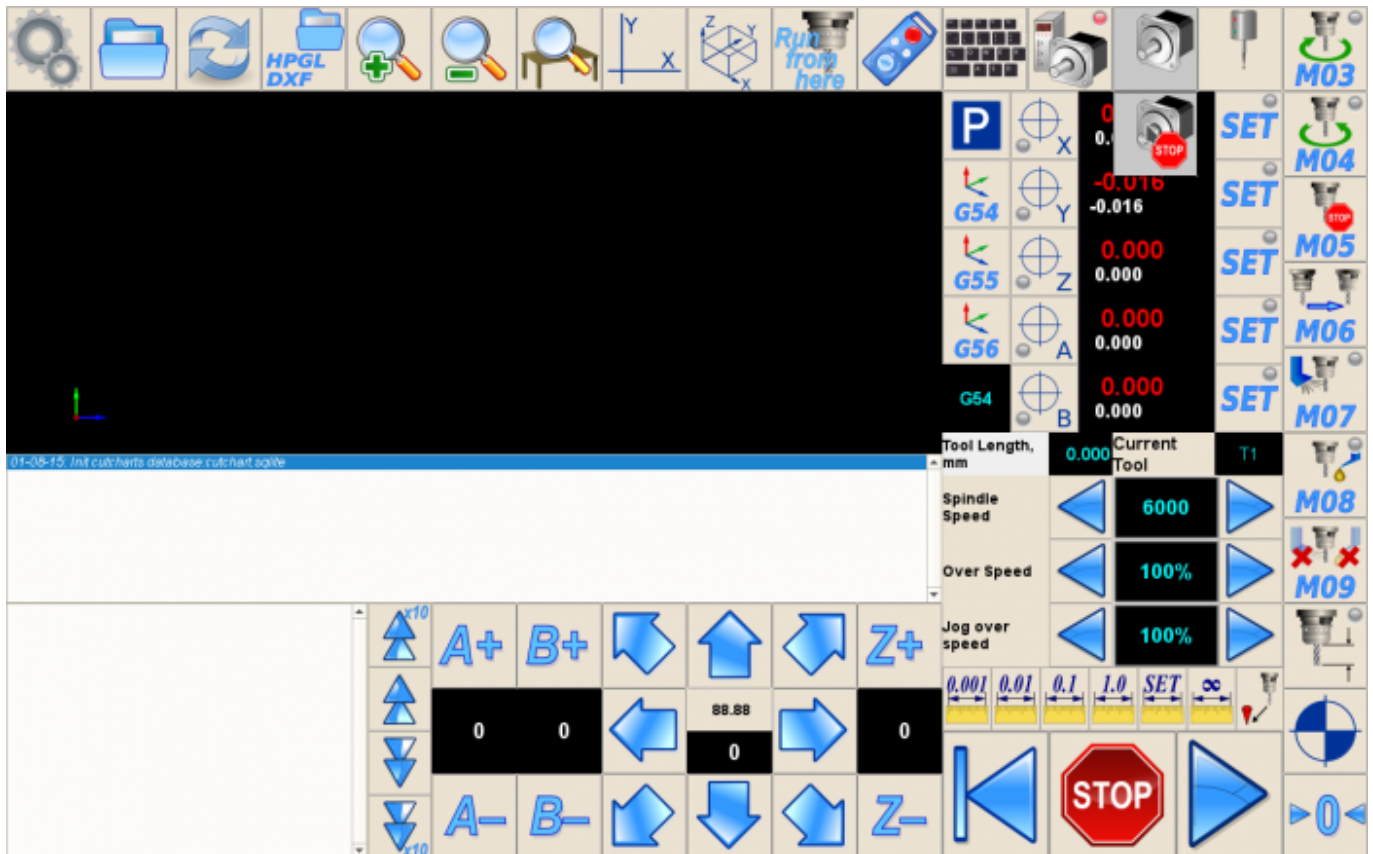
to prevent accidental servo driver activation add 1 sec delay for start button -

```
event="pressed-delay-1000"
```

action function will be started only if the button press and hold for 1 sec (1000 msec). On button will ON relay #3 (M62.plc procedure) Off button will Off relay #3 (M63.plc procedure)

For more complicated configurations special PLC procedure can be written to handle Servo ON/OFF (for example turn On servo power, wait some time and then check servo Fault/Alarm inputs before turn On **Servo On**)

Result should be like this -



## button & xbutton types

Difference between **button** and **xbutton** push-button types easy to see if compare "Servo" button that shows/hide popup window and "Servo-On"/"Servo-Off" buttons. Main "Servo" button contains **LED** display. XButton is simple Push button with added LED display on the top. Led parameters are given in attribute **xattr**.

```
xattr="60;4;16;16;led;red;round" name="display-gvariable-5710"
type="xbutton" />
```

In xattr programmed start x, y positions of Led inside the button, width and height of Led, defines **led** type, led colour (red, green, blue, yellow) and led shape (round, square).

Custom defines skin from SVG file is possible too and will be described in other examples.

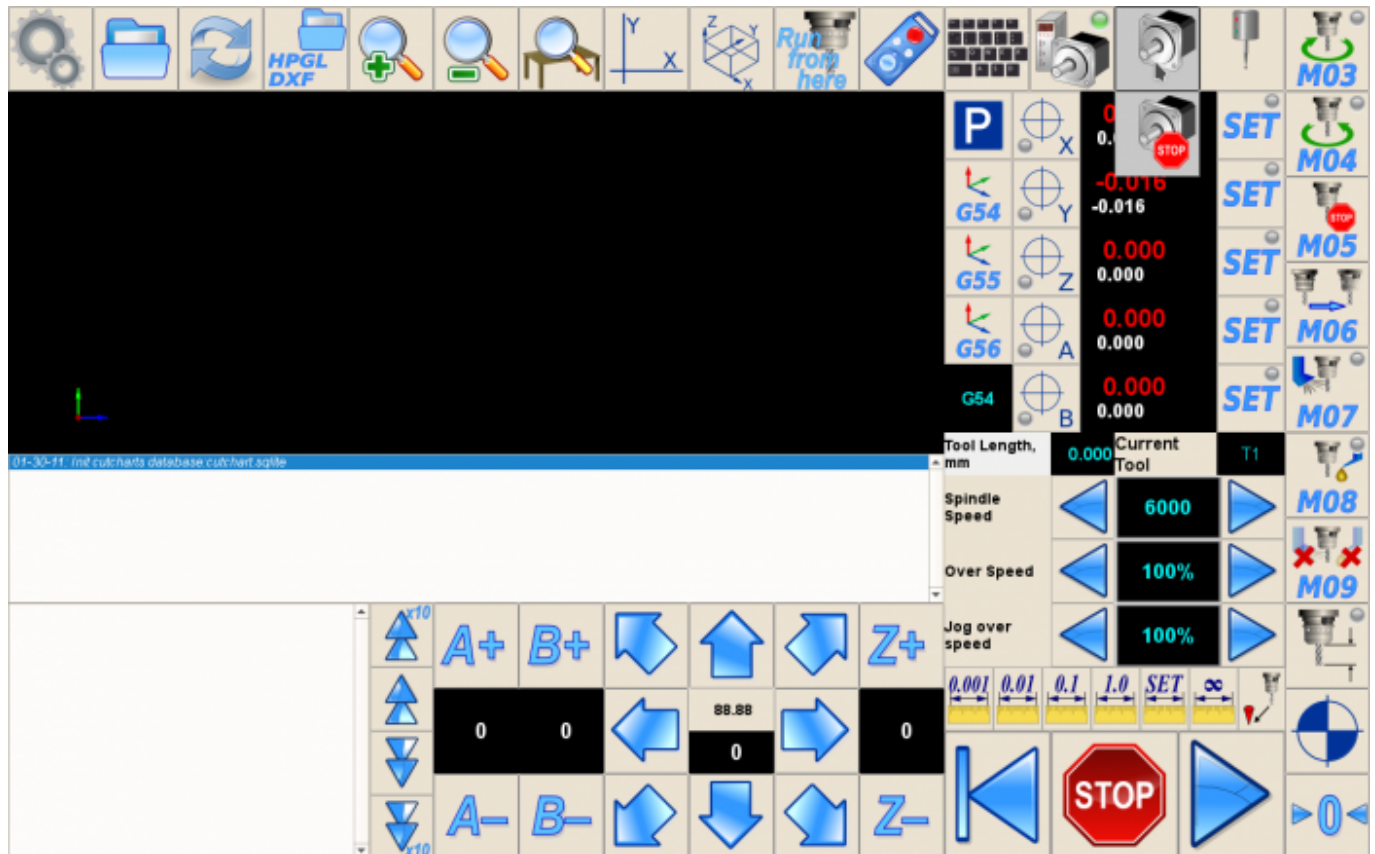
Attribute **name="display-gvariable-5710"** setup LED display to show current state of CNC Global variable #5710 (0 - Led is OFF, 1 or more - LED is ON)

Attributes **address="outputs" number="3"** setup LED display to show current state of output pin #3

To complete this example change led to show output pin #3 to show current servo state and resize the led -

```
<gitem where="magic" position="960;0" width="80" height="80"
image="motor/servo-driver-wide" action="mypopup-toggle:popup-servo"
xattr="56;4;20;20;led;green;round" address="outputs" number="3"
```

```
type="xbutton" />
```

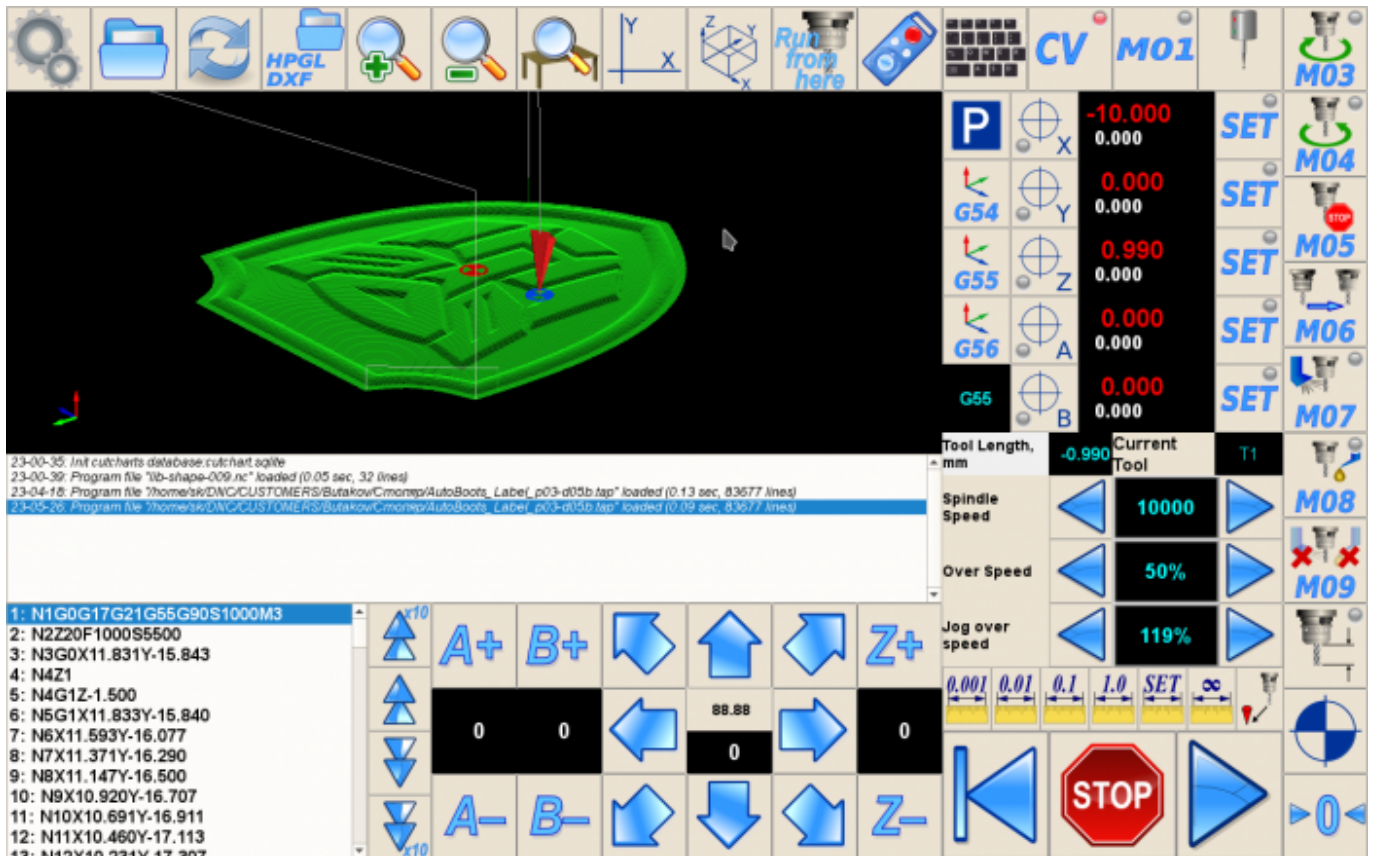


#### Add "Servo On" - "Servo Off" buttons (Version 2)

Just another way to add 2 buttons widget containers with 2 buttons. The same **1280M5** profile

Original screen is shown below-





Put our new buttons instead of **CV** button

1. Find **cnc-screen.xml** configuration file in **1280M5** profile folder
2. Edit the file (in Midnight commander press F4 on the **cnc-screen.xml** file)
3. Find in the file section with **CV** button definition (F7, search "CV")

Here it is-

```
<gitem where="magic" position="960;0" width="80" height="80"
image="CV/CV" action="cnc-gvariable-toggle-5710"
xattr="60;4;16;16;led;red;round" name="display-gvariable-5710"
type="xbutton" />
```

4. Put under this button small popup window which will be a container for two buttons - **Servo ON** & **Servo Off**. Add **mywidget** item - custom small widget with given attributes -

```
<gitem where="magic" position="1120;80" height="160" width="80"
type="myitems" name="toolbar-servo"
bgColor="#d0d0d0" hidden="1" />
```

widget name: **toolbar-servo**

widget size: 80×160 exactly to fit 2 80×80 buttons in

built-in led to show **output#0** state

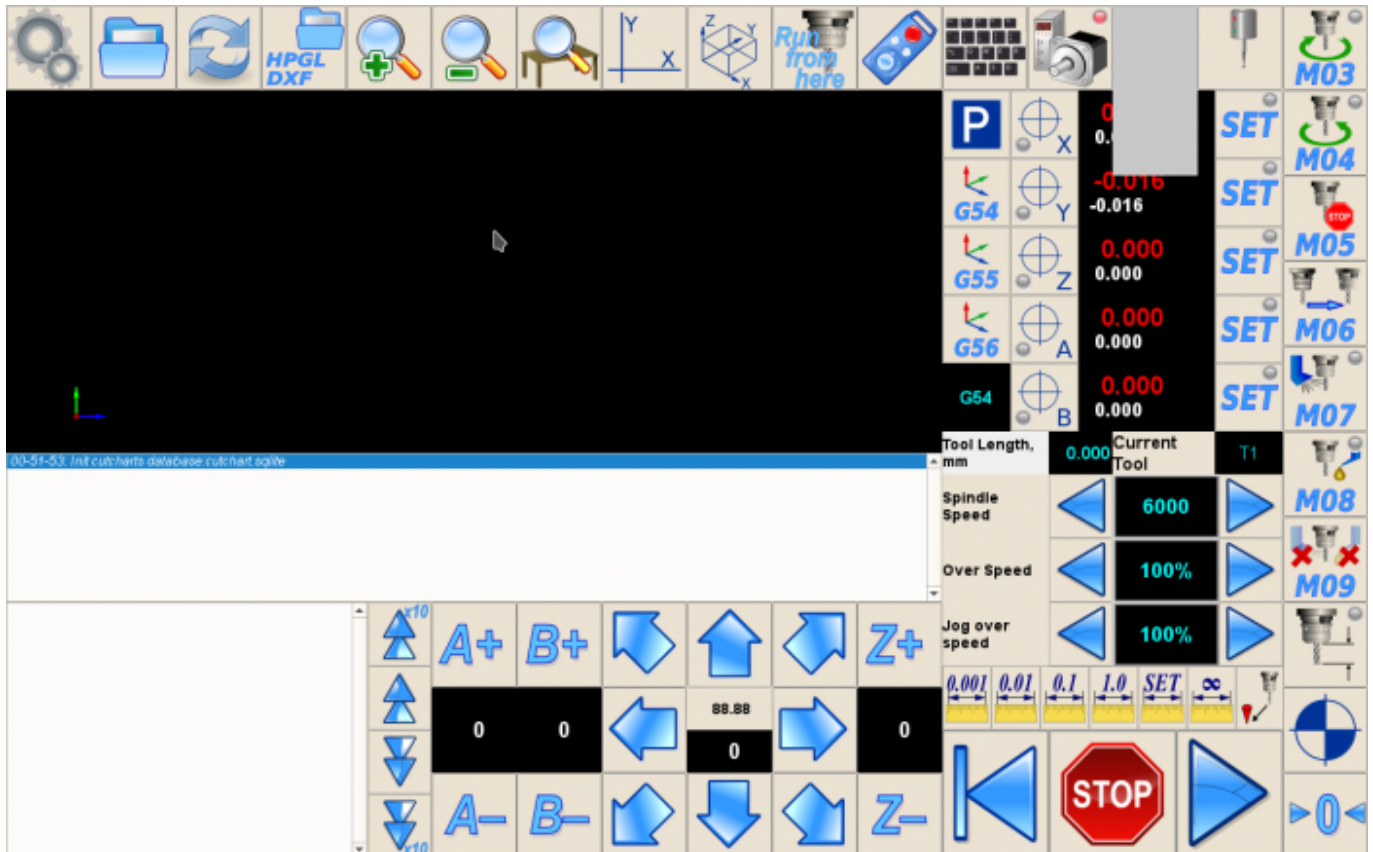
5. Fix **CV** button to show/hide **popup-servo** -



image="CV/CV" action="cnc-gvariable-toggle-5710" image="motor/servo-driver" action="mywidget-toggle:toolbar-servo"

```
<gitem where="magic" width="80" height="80" position="1120;0"
image="motor/servo-driver" action="mywidget-toggle:toolbar-servo"
xattr="56;4;20;20;led:green;round" address="outputs" number="0"
type="xbutton" ></gitem>
```

6. Save **cnc-screen.xml**, restart the software and press the new button with **Servo driver**. Small popup grey colour window will be shown on the right side of the button.



7. Add two buttons definition in the popup window. Definition is very similar to previous example but need to add button position attribute inside widget and button width/height attributes.

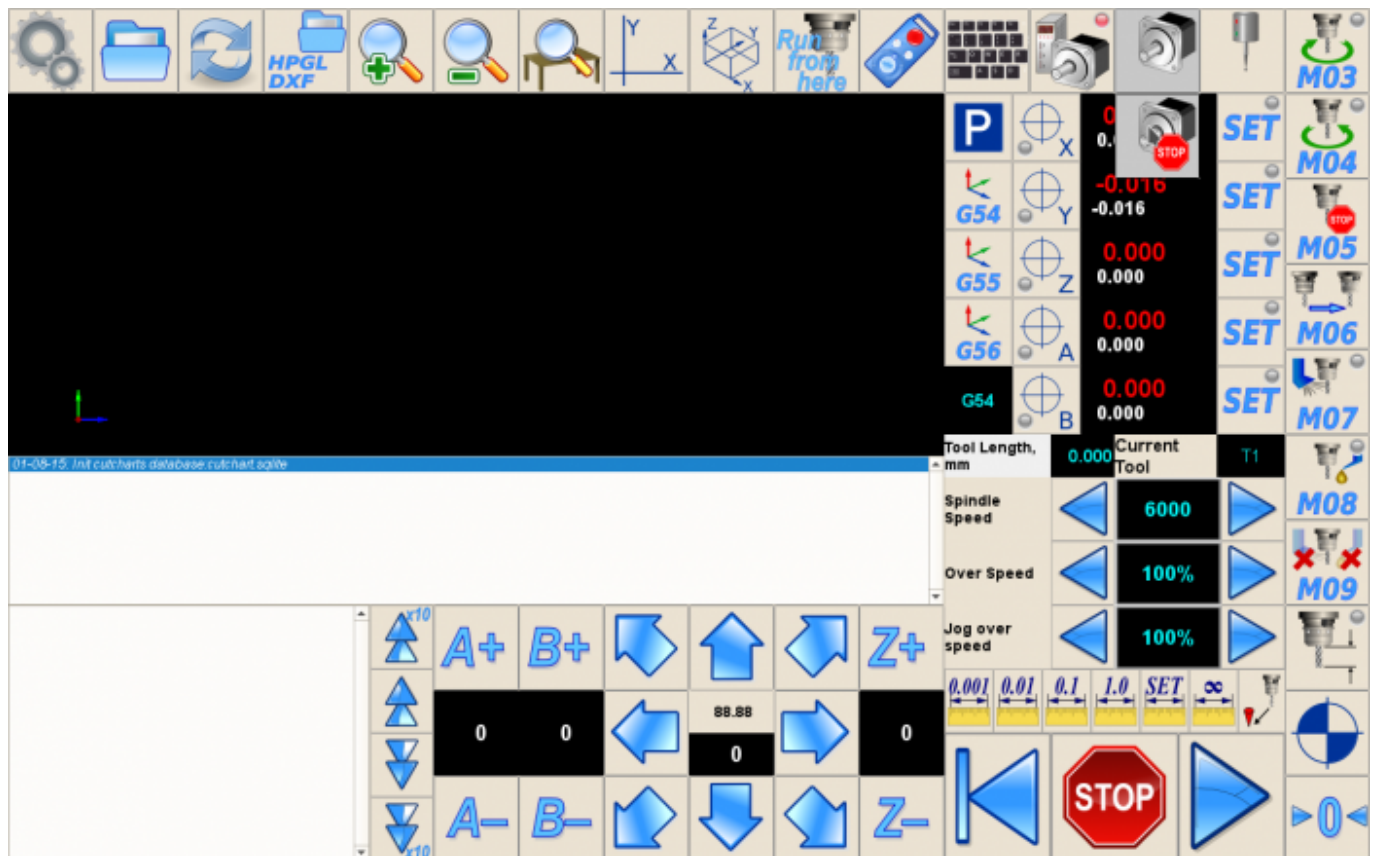
```
<gitem where="toolbar-servo" position="0;0" width="80" height="80"
image="motor/motor-start" action="servo-pid-on" event="pressed-delay-1000"
type="button" />
<gitem where="toolbar-servo" position="0;80" width="80" height="80"
image="motor/motor-stop" action="servo-pid-off" type="button" />
```

to prevent accidental servo driver activation add 1 sec delay for start button -

```
event="pressed-delay-1000"
```

action function will be started only if the button press and hold for 1 sec (1000 msec). On button will run Servo-Pid-ON procedure Off button will run Servo-Pid-Off procedure

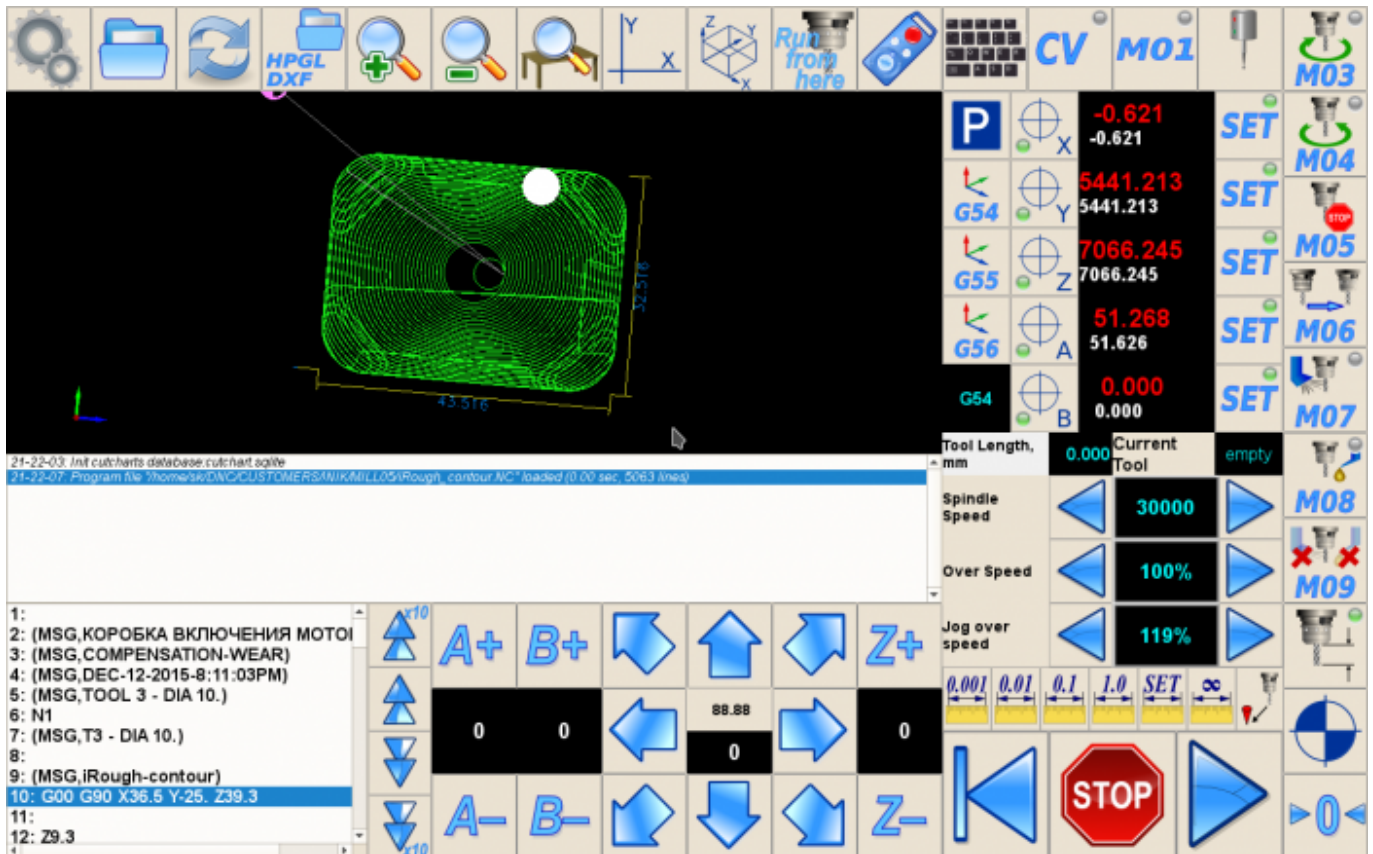
Result should be like this -



**Add Window with some LEDs display**

We need to add to the main screen a window with 8 led displays and some text labels to show the current state of inputs & outputs. We take **1280M5** profile as a base

Original screen is -



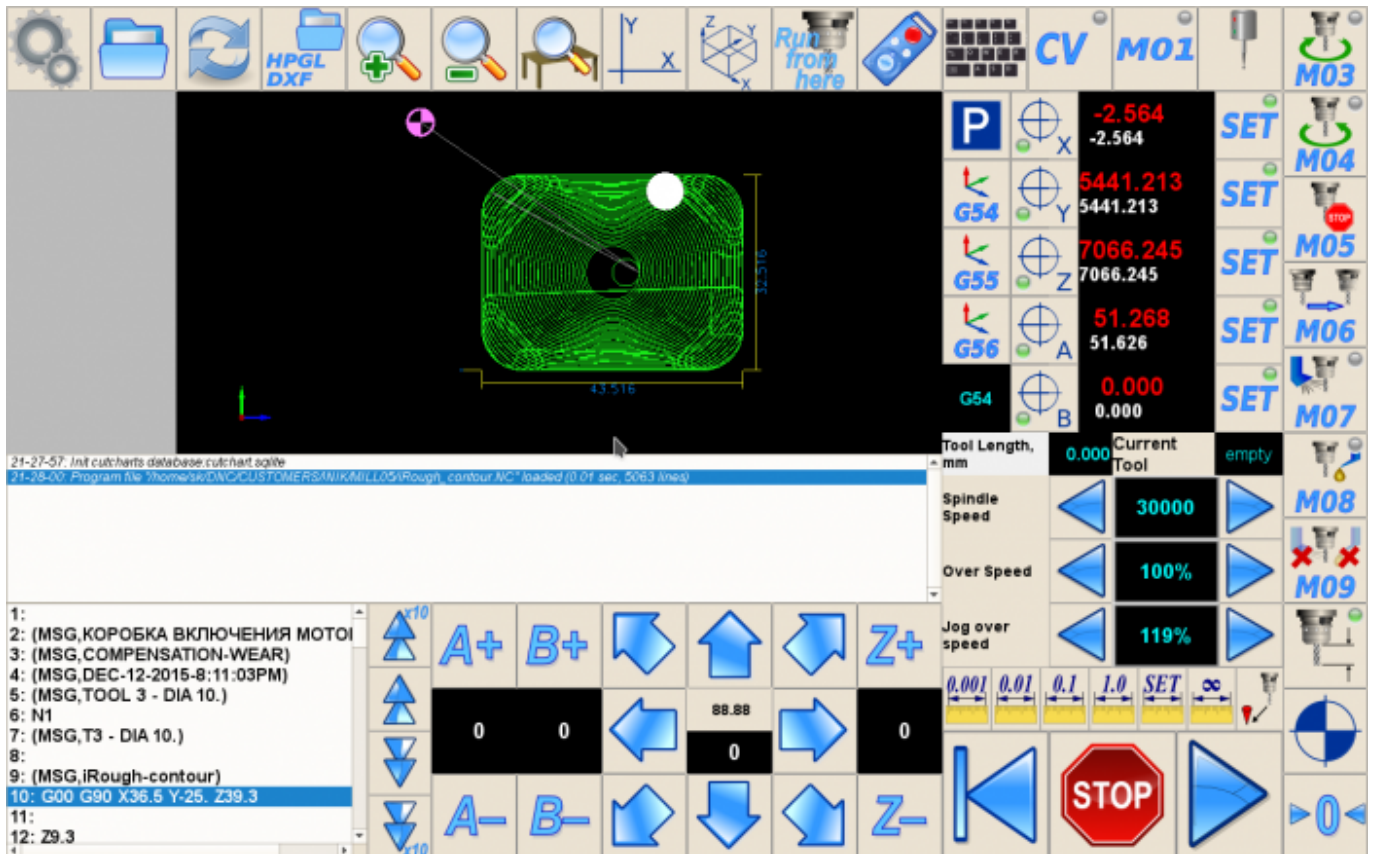
Make the visualisation widget a little narrow and put the new window to the left side from visualization. Find "glview" section in cnc-screen.xml and fix from:

```
<gitem where="magic" position="0;80" width="880" height="340"
bgColor="0xff353535" name="glview" type="glview" />
```

to:

```
<gitem where="magic" position="160;80" width="820" height="340"
bgColor="#ff353535" name="glview" type="glview" />
<gitem where="magic" position="0;80" width="160" height="340"
bgColor="#ffbbbbbb" name="led-panel" type="myitems" />
```

Result is -



Add 1st line label and leds from left and right sides:

```
<gitem inversion="no" where="led-panel" position="5;5" width="20"
displayWidth="20" height="20"
address="outputs" number="11" shape="round" color="green" type="led" />
<gitem inversion="no" where="led-panel" position="25;5" width="130"
labelWidth="110"
displayWidth="20" height="20" labelAlignment="hcenter" labelFontSize="14"
address="outputs" number="14" shape="round" color="green" type="led">
  <message>Side pillar</message>
  <message_ru>Боковой упор</message_ru>
</gitem>
```

**where="led-panel"** - LEDs will be added to window named "led-panel" which was added before

**address="outputs"** - Leds will show state of **Outputs**

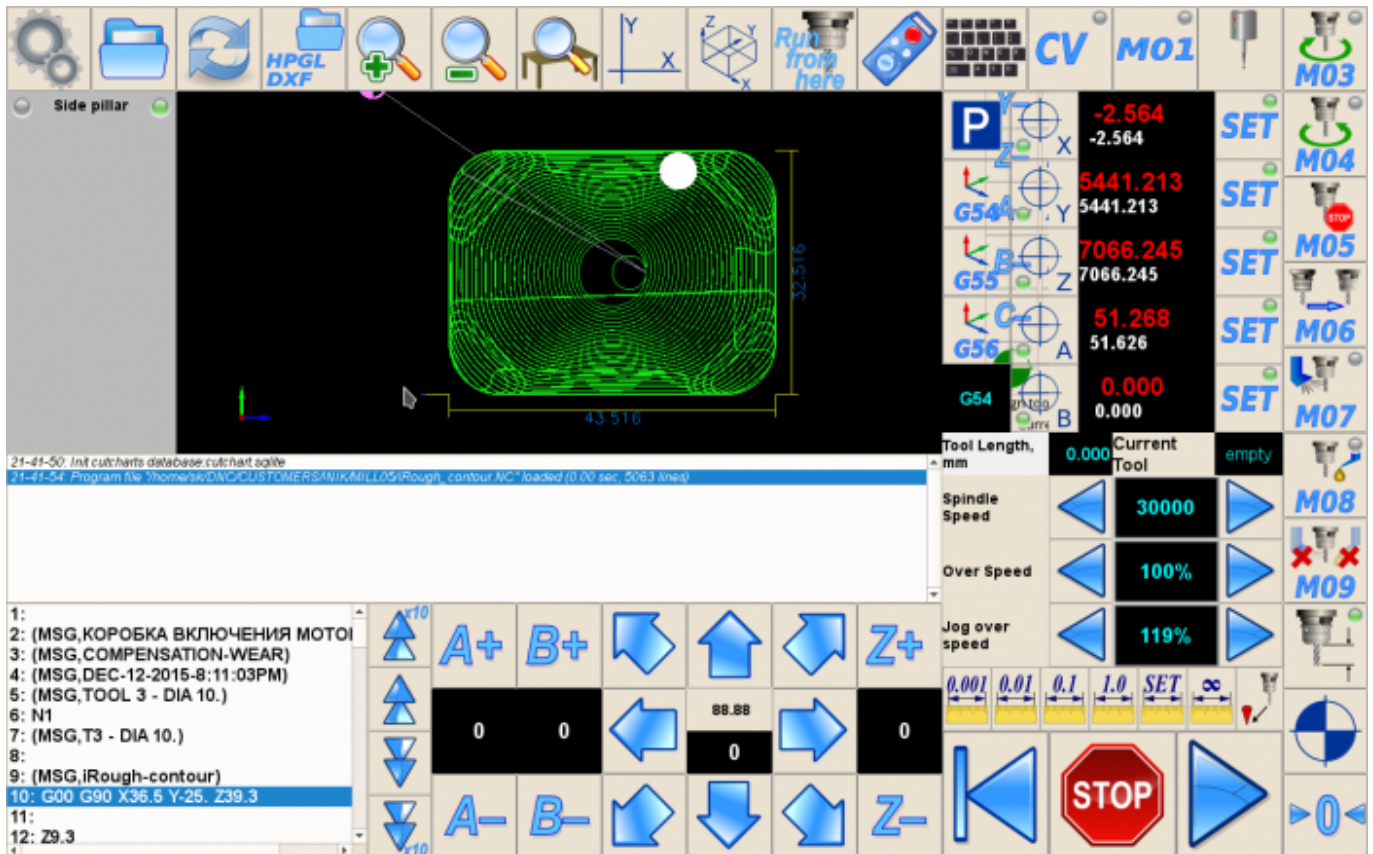
**<gitem inversion="no" where="led-panel" position="5;5" width="20" displayWidth="20" height="20" address="outputs" number="11" shape="round" color="green" type="led" />** - first (left) led is single, no text label assigned

**<message\_ru>Боковой упор</message\_ru>** - second (right) led associated with text label

**position="5;5"** - led is placed to position inside parent widget

Result is -





Add some more LEDs to the widget. Complete code for the widget is -

```
<gitem where="magic" position="0;80" width="160" height="340"
bgColor="#ffbbbbbb"
name="led-panel" type="myitems" />

<gitem inversion="no" where="led-panel" position="5;5" width="20"
displayWidth="20"
height="20" address="outputs" number="11" shape="round" color="green"
type="led" />

<gitem inversion="no" where="led-panel" position="25;5" width="130"
labelWidth="110"
displayWidth="20" height="20" address="outputs" number="14"
labelAlignment="hcenter"
labelFontSize="14" shape="round" color="green" type="led">
<message>Side pillar</message>
<message_ru>Боковой упор</message_ru>
</gitem>

<gitem inversion="no" where="led-panel" position="5;25" width="20"
displayWidth="20"
height="20" address="outputs" number="22" shape="round" color="green"
type="led" />

<gitem inversion="no" where="led-panel" position="25;25" width="130"
labelWidth="110"
displayWidth="20" height="20" address="outputs" number="23"
```

```
labelAlignment="hcenter"
  labelFontSize="14" shape="round" color="green" type="led">
  <message>Rear pillar</message>
  <message_ru>Задний упор</message_ru>
</gitem>

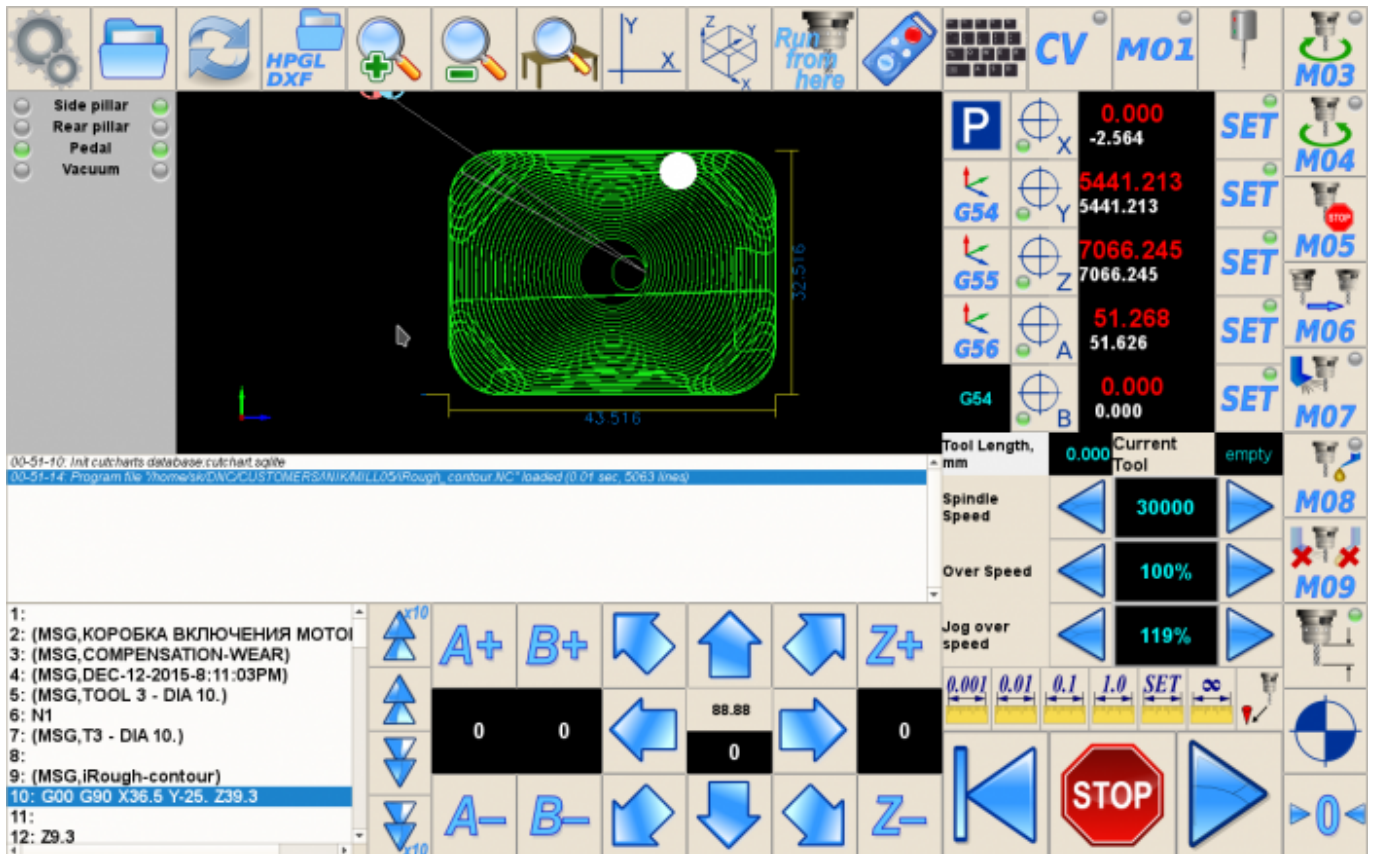
<gitem inversion="no" where="led-panel" position="5;45" width="20"
displayWidth="20"
  height="20" address="inputs" number="13" shape="round" color="green"
type="led" />

<gitem inversion="no" where="led-panel" position="25;45" width="130"
labelWidth="110"
  displayWidth="20" height="20" address="inputs" number="14"
labelAlignment="hcenter"
  labelFontSize="14" shape="round" color="green" type="led">
  <message>Pedal</message>
  <message_ru>Педаль</message_ru>
</gitem>

<gitem inversion="no" where="led-panel" position="5;65" width="20"
displayWidth="20"
  height="20" address="outputs" number="10" shape="round" color="green"
type="led" />

<gitem inversion="no" where="led-panel" position="25;65" width="130"
labelWidth="110"
displayWidth="20" height="20" address="outputs" number="15"
labelAlignment="hcenter"
labelFontSize="14" shape="round" color="green" type="led">
  <message>Vacuum</message>
  <message_ru>Присоски</message_ru>
</gitem>
```

Result is -



### Add Button with led display, that toggles Global Variable value and shows the current state

For described example machine runs the same g-code on left and right sides of the machine table. To increase a performance operator remove ready parts and install a new blank on one side of the table while the machine cuts new parts on another side.

Global variable #500 contains Current active side machine works (or going to work) with. Variable value-

0 - means machine works (or will start to work) with the left side,

1 - means machine works on the right side.

We add a button to toggle current side. We add on-button indication and add LED displays to show current working side.

Button definition is shown below -

```

<gitem where="led-panel" type="xbutton" position="50;90" width="60"
height="60" image="left-right"
action="direct-run:G90 G10 L186 P500 Q1" />
</code>
  
```

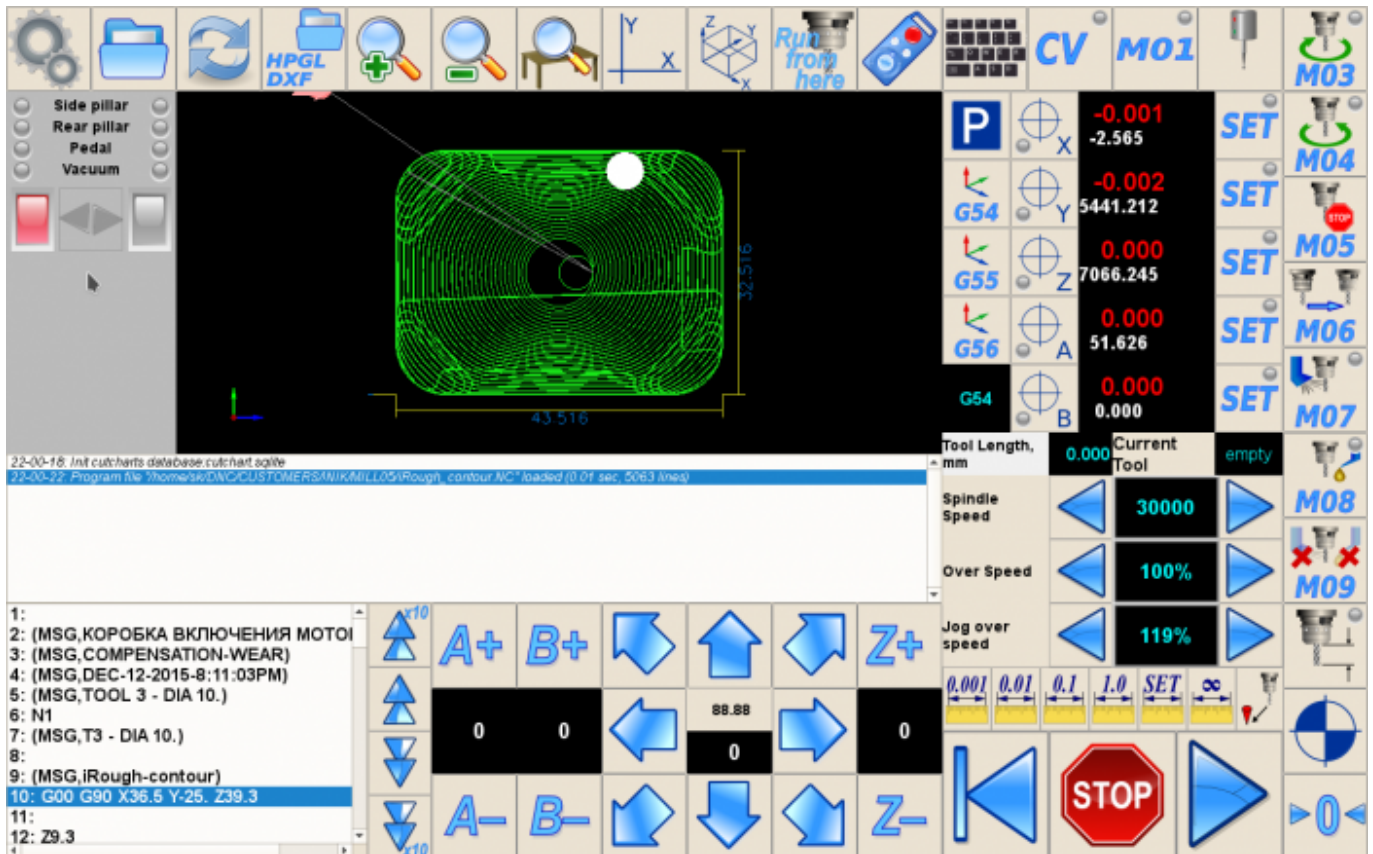
Led displays to the left and right sides from button -

```

<code xml>
<gitem inversion="no" where="led-panel" position="5;90" width="40"
height="60" name="display-cnc-gvariable-500"
shape="square" color="red" type="led" />
  
```

```
<gitem inversion="yes" where="led-panel" position="115;90" width="40"
height="60" name="display-cnc-gvariable-500"
shape="square" color="red" type="led" />
```

Result is -



To add "Current Side" indication on the button -

```
change type <del>"button"</del> "xbutton
add xattr="xattr="0;0;60;60;led;green;round"
add name="display-cnc-gvariable-500" to show current state of Variable #500
add image files array - images="--left;--right"
```

images to show Variable state have the same size as the button itself, so image **-left** or **-right** will be drawn on the top of the button image.

Images for button base, "the left state" and "the right state" are



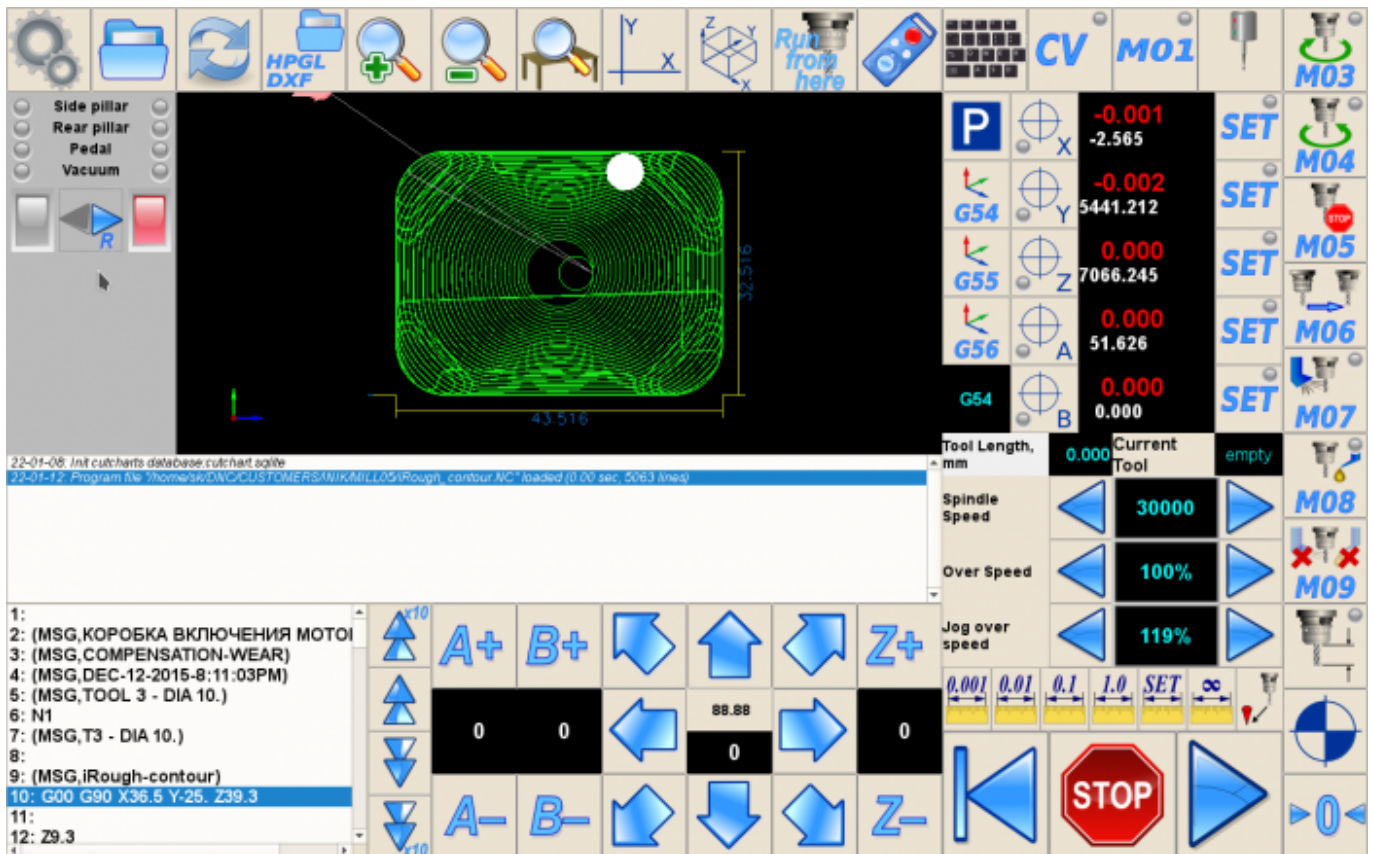
code for button definition is -

```
<gitem where="led-panel" position="50;90" width="60" displayWidth="60"
height="60" image="left-right"
```



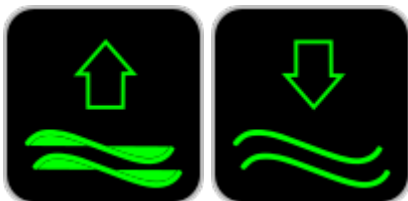
```
type="xbutton" action="direct-run:G90 G10 L186 P500 Q1"
images="--left-right;left--right" name="display-cnc-gvariable-500"
xattr="0;0;60;60;led:green;round"/>
```

Result screen is -

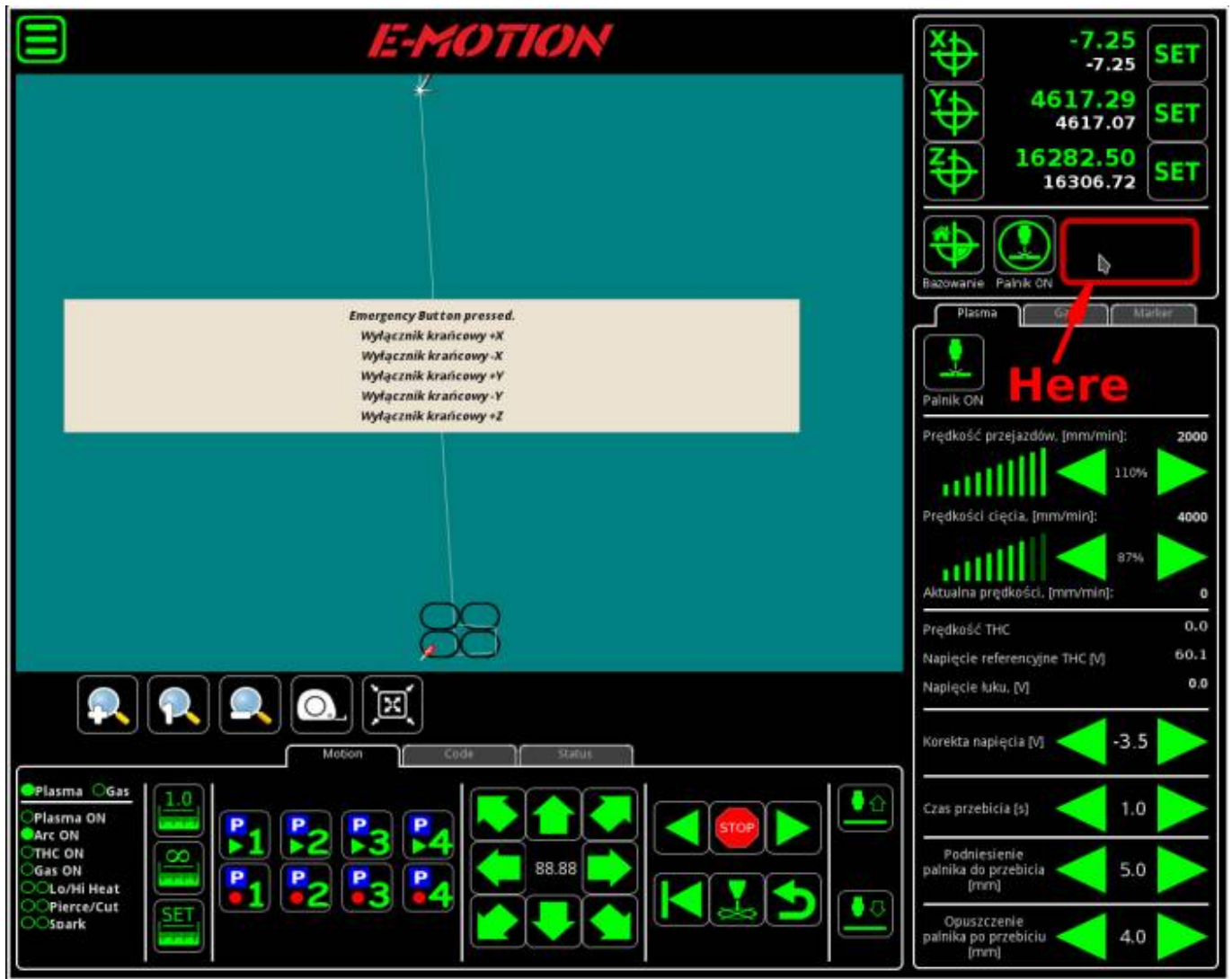


**Add Buttons with LED display that runs PLC procedures and shows Output pin state**

We need to add 2 buttons on GUI main screen to fill up and drain water from plasma cutting machine water table.



Need to add the button to the right from **Palnik ON** button -



Goto cnc-screen.xml in profile folder, find “Palnik ON” line. Here it is: definition of **Palnik ON** button and a text label under it -

```
<gitem where="ecocut-coordinates" position="80;205" width="60" height="60"
image="Ecocut/2a;Ecocut/1a" action="cnc-mode-cutting-ignore" type="toggle-
switch" />

<gitem where="ecocut-coordinates" position="80;265" width="60" height="20"
type="label"
fgColor="white" labelFgColor="white" labelFontFamily="sans-serif"
labelFontSize="12"
fontSize="12" fontStyle="bold" labelAlignment="vcenter,hcenter">
  <message>Test Plasma</message>
  <message_ru>Поджиг</message_ru>
  <message_pl>Palnik ON</message_pl>
</gitem>
```

**where=“ecocut-coordinates”** - widget where button & label are placed

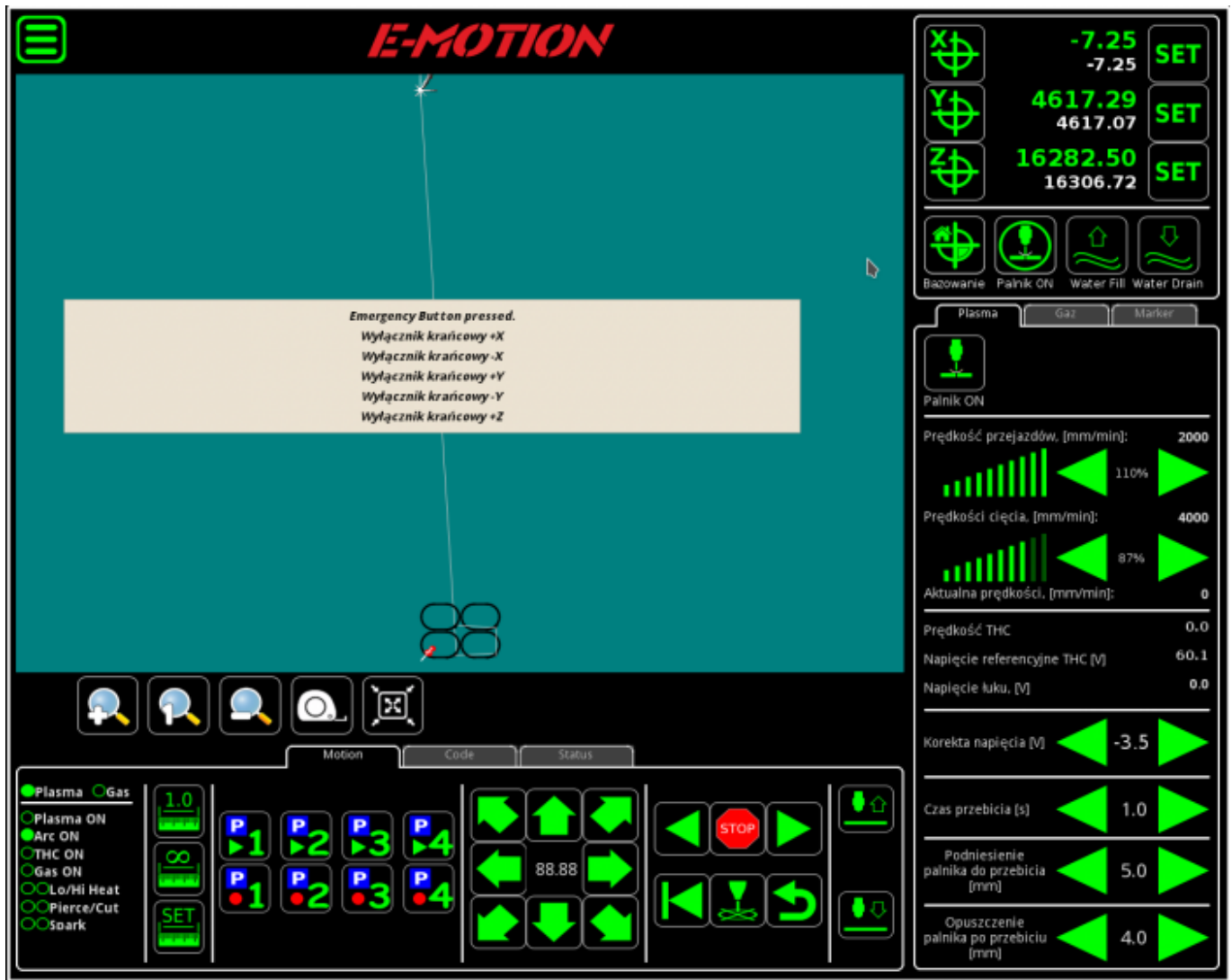
**position=“80;205” width=“60” height=“60”** - position, width and height of the button

We need to add 2 buttons just after it -

```
<gitem where="ecocut-coordinates" position="150;205" width="60" height="60"
image="Ecocut/water-up"
action="direct-plc:M240/1" type="button" />
<gitem where="ecocut-coordinates" position="145;265" width="70" height="20"
type="label" fgColor="white"
labelFgColor="white" labelFontFamily="sans-serif" labelFontSize="12"
fontSize="12"
fontStyle="bold" labelAlignment="vcenter;hcenter" >
  <message>Water Fill</message>
  <message_pl>Water Fill</message_pl>
</gitem>

<gitem where="ecocut-coordinates" position="220;205" width="60" height="60"
image="Ecocut/water-down"
action="direct-plc:M240/0" type="button" />
<gitem where="ecocut-coordinates" position="215;265" width="70" height="20"
type="label" fgColor="white"
labelFgColor="white" labelFontFamily="sans-serif" labelFontSize="12"
fontSize="12" fontStyle="bold"
labelAlignment="vcenter;hcenter" >
  <message>Water Drain</message>
  <message_pl>Water Drain</message_pl>
</gitem>
```

Result is -



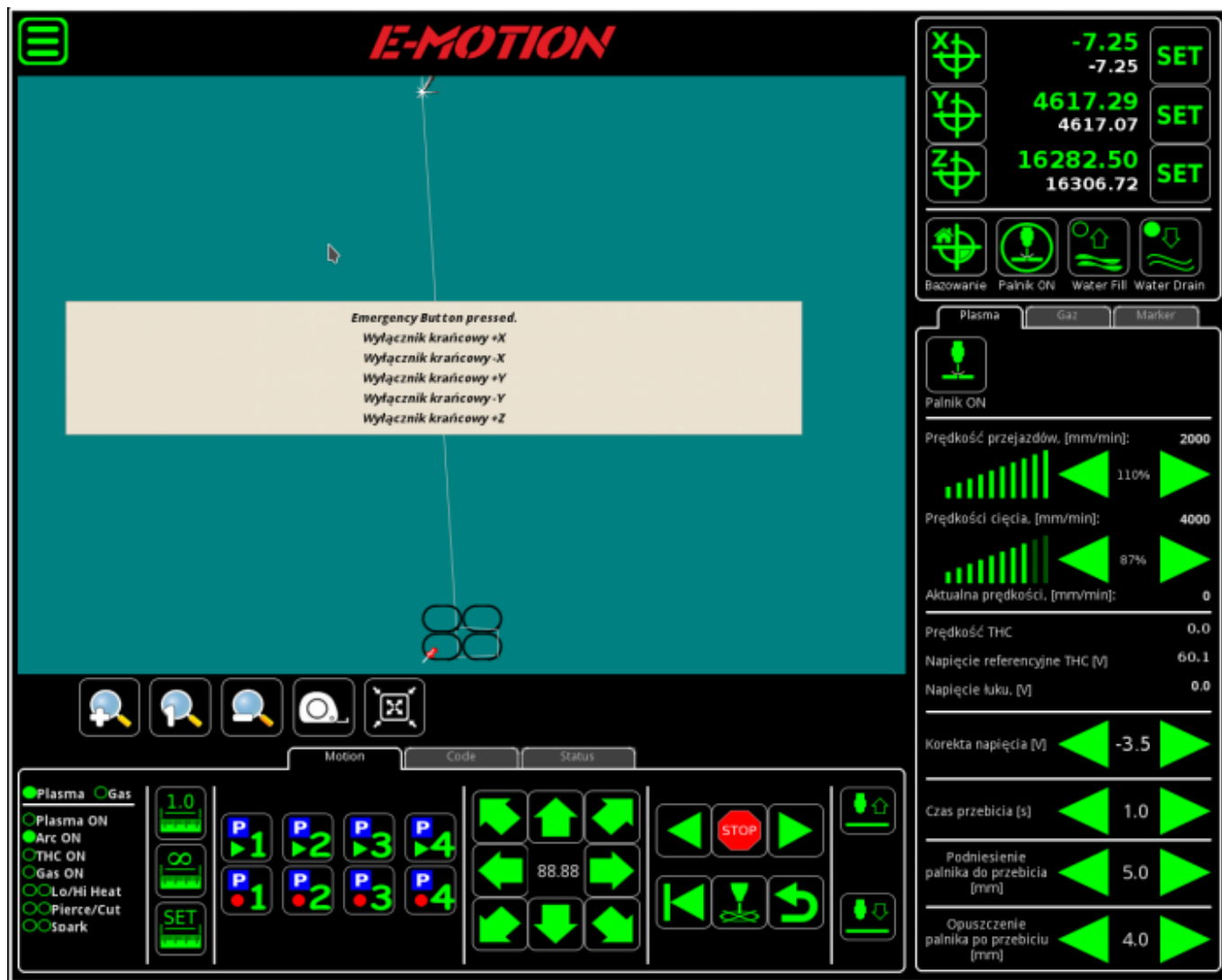
Out pin #13 controls Water Fill,  
Out pin #12 controls Water Drain

Add LEDs display to the buttons -

1. Change type from "button" to "xbutton" 2. Add **xattr** attribute to buttons definition with LEDs position, width, height, colour and shape 3. Add to button definition attribute address="outputs" and number="12" (or number="13") to show output pin state

```
<gitem where="ecocut-coordinates" position="150;205" width="60" height="60"
image="Ecocut/water-up"
action="direct-plc:M240/1"
type="xbutton" xattr="4;4;16;16;led;green;round" address="outputs"
number="13"/>
<gitem where="ecocut-coordinates" position="220;205" width="60" height="60"
image="Ecocut/water-down"
action="direct-plc:M240/0"
type="xbutton" xattr="4;4;16;16;led;green;round" address="outputs"
number="12" />
```

Result is -

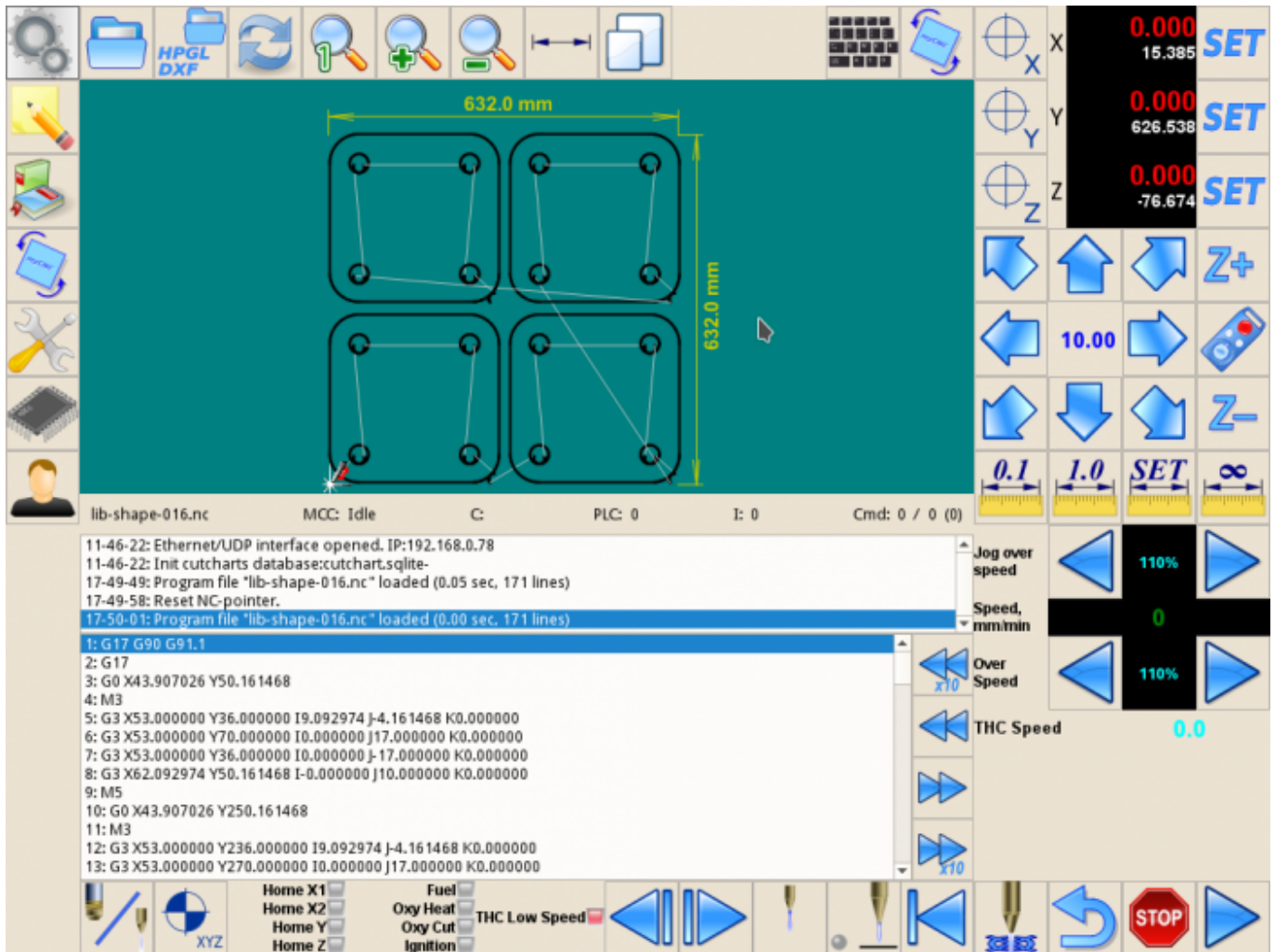


To control Water Fill/Drain PLC procedure M240 with parameter "1" (to Fill) or "0" (to Drain) is used. PLC procedure M240 is described [Here](#)

### Add display to show current PLC status for Gas Cutting

We will fix **1024G** profile. The main screen of the profile is shown below.





Put PLC process display to the top of the empty area next to "THC Speed" display.

To add the display -

- find and open to edit "cnc-screen.xml" file in profile folder
- find reference "THC Speed" display -

```
<gitem fgColor="cyan" where="w-operate" labelFontStyle="bold"
format="%5.1f" K="1." height="30"
type="display" deviation="0.01" name="display-gvariable-5493"
bgColor="transparent"
displayWidth="140" labelWidth="100" fontStyle="bold" fontSize="20"
orientation="horizontal">
```

```
    <message>THC Speed</message>
    <message_ru>Z Слежение</message_ru>
```

```
</gitem></code>
```

\* Add new display (easy way to copy-paste "THC Speed" display and edit it.

\* Remove "name", "K" attributes

\* Add "address" attribute to organize process display -

```
<code>address="plc-proc"
```

- Change type -

```
type="radio-display"
```

- Edit "format" attribute -

```
format="No Cutting=0;Ignition=50;PreHeat=51;Cutting=60"
```

- Translate text inside format attribute if need local language messages -

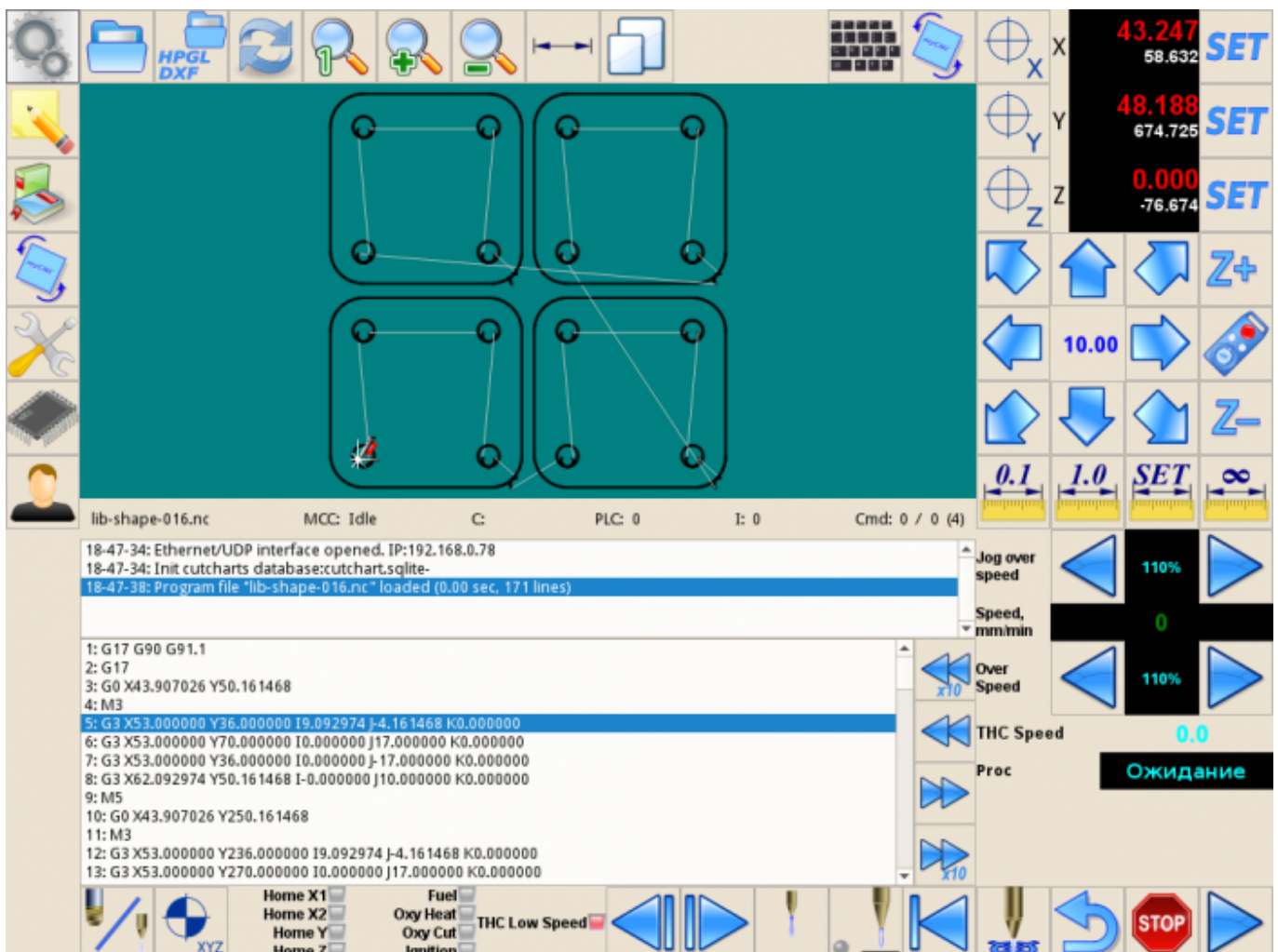
```
format="Ожидание=0;Поджиг=50;Подогрев=51;Резка=60"
```

- Edit bgColor, fgColor, fontSize attributes to get nice looking display

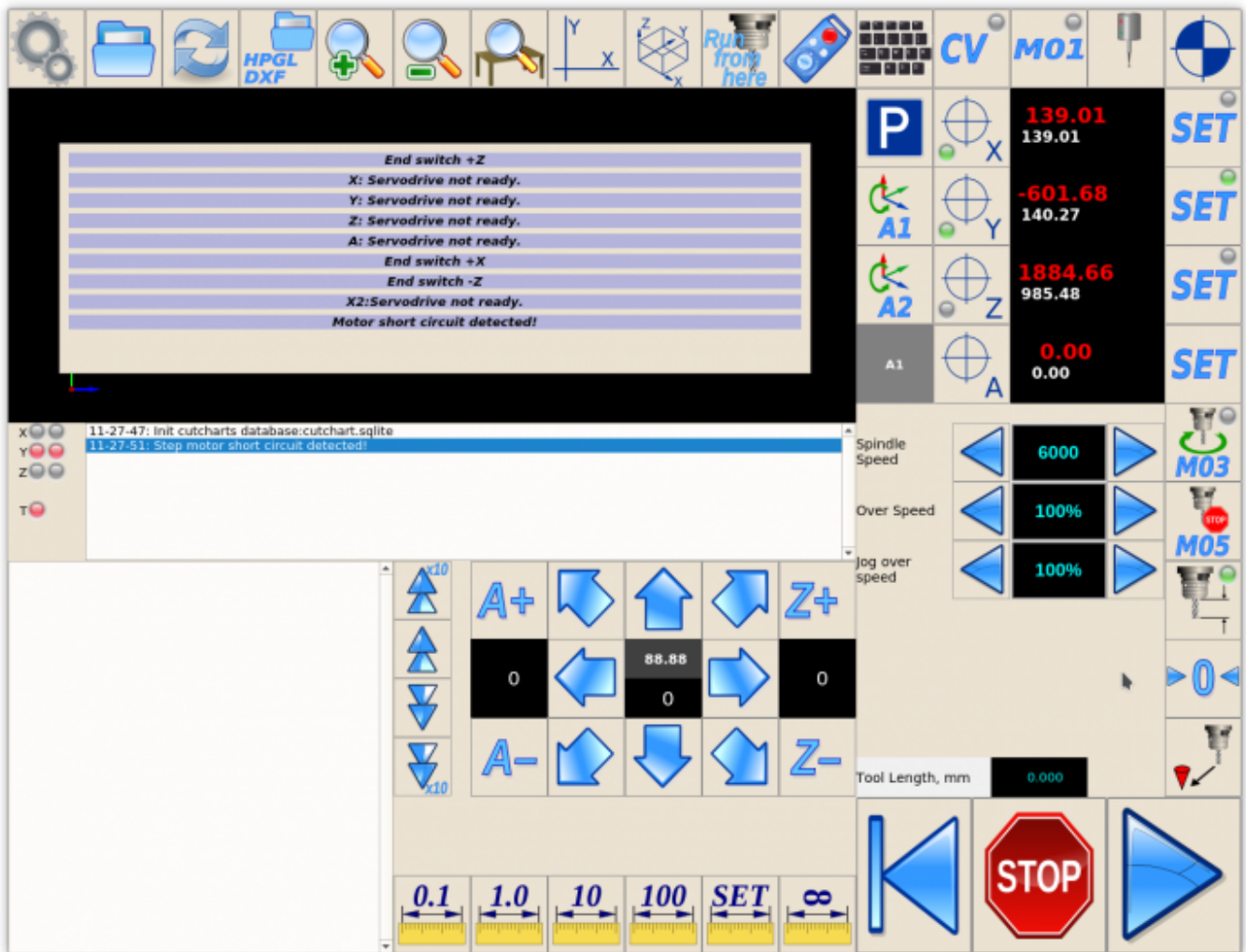
A complete code -

```
<gitem fgColor="cyan" where="w-operate" labelFontStyle="bold"
type="radio-display" address="plc-proc"
format="Ожидание=0;Поджиг=50;Подогрев=51;Резка=60"
height="30" displayWidth="140" labelWidth="100" bgColor="black"
fontStyle="bold" fontSize="16" orientation="horizontal">
  <message>Proc</message>
  <message_ru>Процесс</message_ru>
</gitem>
```

Screen result is -



## Add "Move A to 0" button to run macro with confirmation



For 3 axes machine we have a button **>0<** to moves machine to work 0 position. For machines with rotational axes it is convenient to have similar button to move rotational axis **A** to work 0.

We add this button to the main screen on the left of **>0<** button. Here is **Move To 0** definition -

```

<gitem type="button"
where="magic"
image="parking/move-to-0"
action="direct-run-confirm:M333"
position="1200;640" width="80" height="80">
  <message-confirm>Move to Work (0,0,Tool Lift) position (macro M333). Are
you sure?</message-confirm>
  <message-confirm_ru>Переместить в положение (0,0, Tool Lift) (макро
M333)?</message-confirm_ru>
</gitem>

```

Add on the left **Move To A** definition -

```

<gitem type="button"
where="magic"

```



```

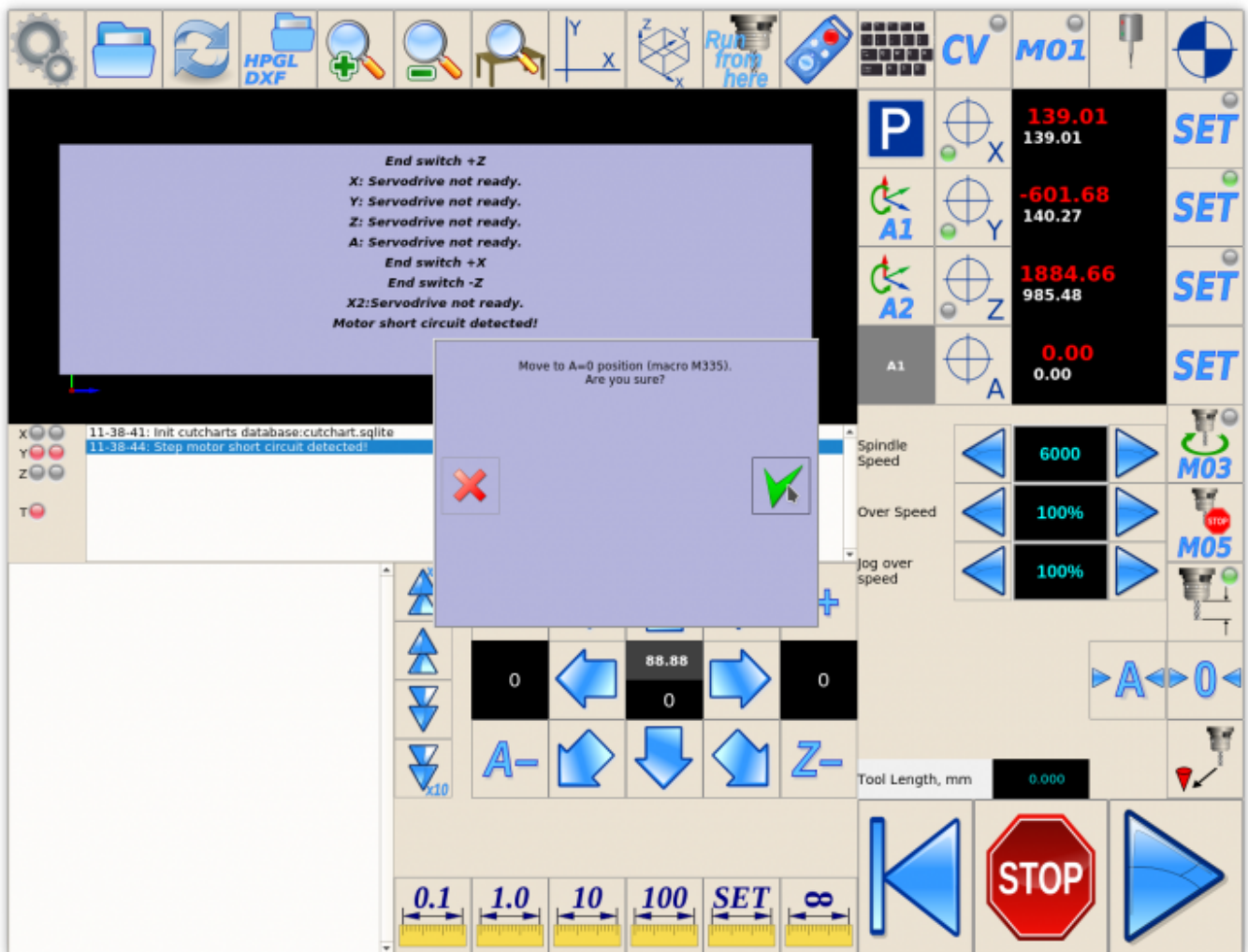
image="parking/move-to-A"
action="direct-run-confirm:M335"
position="1120;640" width="80" height="80">
  <message-confirm>Move to A=0 position (macro M335). Are you
sure?</message-confirm>
  <message-confirm_ru>Вернуть в A=0 (макро M335)?</message-confirm_ru>
</gitem>

```

- **type="button"** - Button definition
- **where="magic"** - Name of Parent Widget
- **image="parking/move-to-A"** - Button skin image filename (SVG format, folder relative to /myCNC/art/buttons-no-theme/)
- **action="direct-run-confirm:M335"** - What to do if button pressed -
  - **direct-run-confirm:** - Open Message box to confirm, then
  - run G-code sequence (";" is used as separator of g-code blocks)
- **position="1120;640" width="80" height="80"** - button position on Parent Widget, width and height of button
- **<message-confirm>Move to A=0 position (macro M335). Are you sure?</message-confirm>** - message to show in Message Box
- **<message-confirm\_ru>Вернуть в A=0 (макро M335)?</message-confirm\_ru>** - message to show for Russian language



Screen with **Move To A** button



Message Box when **Move To A** button pressed

If button OK pressed on Message Box, g-code line **M335** (which is macro actually) will be executed to move A axis to zero.

[M335 macro to move A to zero is described here](#)

### Add Centring Widget on main GUI screen

myCNC software has Centring widget to work with probing sensor and find internal and external centers, edges, corners on parts.

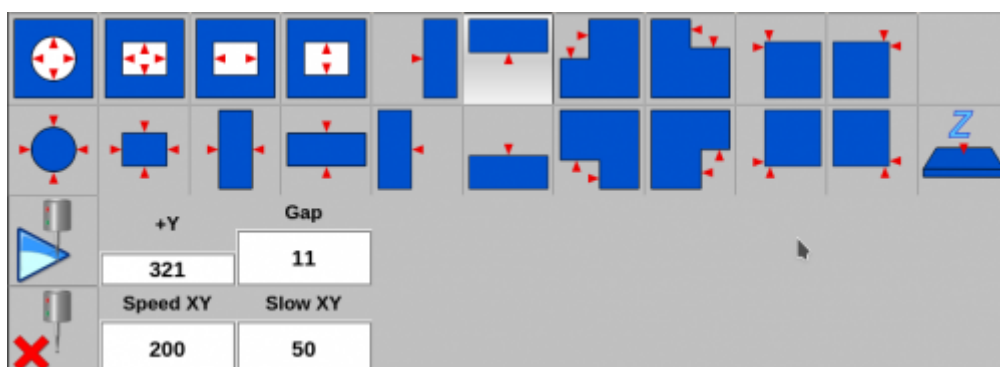


Fig. Centring Widget

There is just 1 line to configure Centring Widget in cnc-screen.xml configuration file.

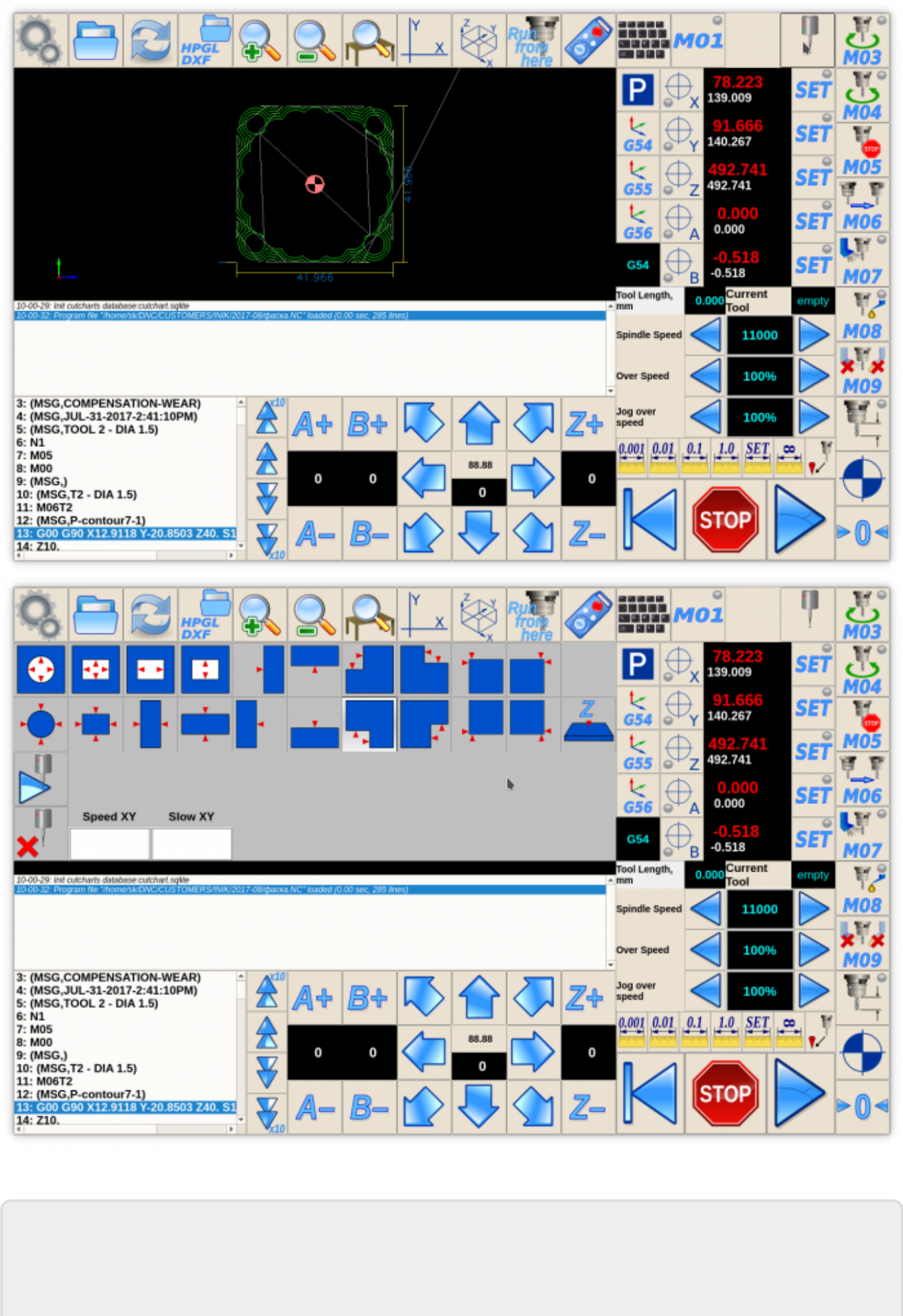
```
<gitem type="centringview" where="magic"
bsize="80"
position="0;80" width="880" height="320"
hidden="1"
bgColor="#c0c0c0">centringview</gitem>
```

- **type="centringview"** - Centring Widget definition
- **where="magic"** - name of Parent Widget
- **bsize="80"** - size of buttons
- **position="0;80" width="880" height="320"** - Position of Centring Widget on the Parent Widget, width and height of Centring Widget
- **hidden="1"** - Centring Widget is hidden on the start of GUI
- **bgColor="#c0c0c0"** - Backgroundcolor of Centring Widget

Centring Widget takes a lot of space on the GUI screen, so it's convenient to have it hidden while normal work and show the widget only when work with Probe sensor. To show/hide Centring Widget usually used on-screen button.

```
<gitem type="button"
image="probing/centring-unit" action="toggle-widget-centring"
where="magic"
position="1120;0" width="80" height="80" />
```

- **type="button"** - Button definition
- **image="probing/centring-unit"** - image for the button skin (in SVG format, relative to images folder "/myCNC/art/buttons-no-theme/
- **action="toggle-widget-centring"** - Action definition for pressing the button - Show/Hide Centring Widget
- **where="magic"** - name of Parent Widget
- **position="1120;0" width="80" height="80"** - Button position on the Parent Widget, width and height on the button in pixels



From:

<http://docs.pv-automation.com/> - **myCNC Online Documentation**

Permanent link:

[http://docs.pv-automation.com/mycnc/mycnc\\_screen\\_configuration?rev=1538404966](http://docs.pv-automation.com/mycnc/mycnc_screen_configuration?rev=1538404966)

Last update: **2018/10/01 10:42**

