LinuxCNC Axis GUI erweitern

Mit pyVCP können Erweiterungen für das Axis GUI geschrieben werden, die in einem Panel rechts neben der üblichen Axis-Software geladen werden.

Dazu wird dreierlei benötigt:

- 1. Die Definition der GUI-Erweiterung als pyVCP-Datei im XML-Format. Hier werden die Komponenten des GUI wie Buttons, LEDs, Skalen usw. festgelegt.
- Eine POSTGUI-HAL-Datei welche die Verknüpfungen zwischen dem pyVCP-Panel und dem HAL (Hardware Abstraction Layer) von LinuxCNC festlegt. Diese entspricht auf Software-Ebene der Verschaltung von Komponenten, besteht also hauptsächlich aus net-Befehlen.
- 3. Ein Editieren der <mark>INI-Datei</mark> für die Maschine. Dort sind 2 Dinge nötig:
 - Im Display-Abschnitt muss das pyVCP-Panel geladen werden
 - Im HAL-Abschnitt muss die HALUI-Komponente von aussen zugänglich gemacht werden.

Ein B	eispiel
-------	---------

👗 ax	is.ngc - AXIS 2.7.12 on ISEL	↑ _ □ ×
<u>F</u> ile <u>Machine</u> <u>V</u> iew <u>H</u> elp		
😣 💽 🗁 🤌 🍡 🛁 🚺	(🛄 💠 😑 [Z] [S [X] [Y] 과 🍅	Jog Speed
Manual Control [F3] MDI [F5]	Preview DRO	
Axis: • X • Y • Z	X : 0.000	
- + Continuous -	Y: 0.000	118
Home All Touch Off	Z: 0.471	
Tool Touch Off	Vel: 0.000 DTG: 0.000	Jog Z Axis
Override Limits	2	Up
Spindle: 🔊 Stop 🕐		Down
- +	The second and a second s	
Coolant: Dist	S CAR CAR	
	734 5 - 4 V C	
	je standard and a standard and a standard a s	
Feed Override: 100 %		
Rapid Override: 100 %		
log Speed: 290 mm/min		
Max Velocity: 3000 mm/min		
1. (AXIS "splash g-code" Not intende	d for actual milling)	X
2: (To run this code anyway you migh	t have to Touch Off the Z axis)	-
3: (depending on your setup. As if y 4: (Hint jog the Z axis down a bit t	ou had some material in your mill) hen touch off)	
5: (Also press the Toggle Skip Lines	with "/" to see that part)	N
7: (LinuxCNC 19/1/2012 2:13:51 PM)	(ct for your machine, change the scale below)	
8: # <depth>=2.0 9: #<scale>=1.0</scale></depth>		7
ON No tool	Position: Relative Actual	-

Die Axis-GUI erhält 3 neue Elemente im rechten Panel: einen Slider zum Einstellen der Geschwindigkeit und 2 Buttons zum Hoch- und Runterfahren der z-Achse.

Die pyVCP-Datei

Die Datei "mypanel.xml" enthält dies:

```
<pyvcp>
   <labelframe text="Jog Speed">
       <scale>
           <font>("Helvetica",16)</font>
           <width>"50"</width>
            <halpin>"jogspeed"</halpin>
           <resolution>1</resolution>
           <orient>VERTICAL</orient>
           <min >500</min >
           <max >0</max >
           <param_pin>1</param_pin>
       </scale>
   </labelframe>
    <labelframe text="Jog Z Axis">
        <vbox>
            <button>
                <halpin>"jog-z-plus"</halpin>
                <text>" Up "</text>
                <font>('Fixed',16)</font>
            </button>
            <button>
                <halpin>"jog-z-minus"</halpin>
                <text>" Down"</text>
                <font>('Fixed',16)</font>
            </button>
        </vbox>
    </labelframe>
</pyvcp>
```

Diese Datei beginnt immer mit dem Tag **<pyvcp>** und endet mit **</pyvcp>**.

Die Elemente sind auf zwei Frames verteilt. Die obere enthält ein **Scale**-Element, mit dem die Geschwindigkeit eingestellt werden kann.

Unten gibt es zwei Buttons für Up und Down.

Die Verbindung nach aussen erfolgt durch die Zeilen, die mit dem Tag <halpin> beginnen:

```
<halpin>"jogspeed"</halpin>
<halpin>"jog-z-plus"</halpin>
<halpin>"jog-z-minus"</halpin>
```

Diese sind sozusagen die "Ausgänge" des pyVCP-Blocks.

<u>Test</u>

Damit das Panel geladen wird, muss in der <mark>INI-Datei</mark> im Abschnitt [DISPLAY] ein Befehl eingefügt werden:

[DISPLAY]
DISPLAY = axis
.....
PYVCP = mypanel.xml

🔲 Select Item to Probe 🛧 🗕 🗆 🗙			×
<u>P</u> ins	<u>S</u> ignals	Para <u>m</u> ete	rs
parport.0.pin	-17-out		
parport.0.rea	id.time		
parport.0.res	et.time		
parport.0.wri	te.time		
parport.read-all.time			
parport.write-all.time			
pyvcp.jog-z-minus			
pyvcp.jog-z-plus			
pyvcp.jogspe	eed-f		
pyvcp.jogspe	eed-i		
pyvcp.scale.0.param_pin			
servo-thread.time			
stepgen.0.counts			
stepgen.0.dir			
stepgen.0.enable			
stepgen.0.position-cmd			
stepgen.0.position-fb			
stepgen.0.st	ер		
stepgen.1.counts			
stepgen.1.dir			
stepgen.1.er	able .		-
<u>C</u> lose			

Im Menü Machine – HAL meter kann man unter "Pins" die 3 Ausgangssignale sehen. Alle beginnen mit pyvcp.

Wichtig:

Das Signal der Scale, also jogspeed, gibt es in zwei Varianten, mit den Endungen i und f. Das eine stellt einen Integer-Typ bereit, das andere einen Floating point. Das letztere muss später für die "Verdrahtung" benutzt werden.

Die Verbindung zu HAL

Damit diese funktioniert, muss in der INI-Datei die Option HALUI=halui aktiviert sein:

[HAL] . . .

HALUI = halui

Nun kann man mit dem HAL-meter sehen, welche "Pins" HALUI zum "Verdrahten" zur Verfügung stellt.

Für unser Beispiel wichtig sind die folgenden Eingangs-Pins:

halui.jog-speed	
halui.jog.2.plus	(positive z-Richtung)
halui.jog.2.minus	(negative z-Richtung)

Diese müssen nun in der Datei post-gui.hal mit den Signalen des pyVCP-Panels verbunden werden:



Das geschieht in der Datei custom_postgui.hal:

```
net zs halui.jog-speed <= pyvcp.jogspeed-f
net remote-jog-z-plus halui.jog.2.plus <= pyvcp.jog-z-plus
net remote-jog-z-minus halui.jog.2.minus <= pyvcp.jog-z-minus</pre>
```

Die Syntax ist dabei :

Signalname Eingang <= Ausgang

Achtung: für jogspeed muss das Floatingpoint-Signal (mit der Endung "-f") bei pyvcp verwendet werden!

Damit die Datei custom_postgui.hal geladen wird, muss in der INI-Datei der entsprechende Eintrag vorhanden sein:

[HAL]

```
HALFILE = ISEL.hal
HALFILE = custom.hal
POSTGUI_HALFILE = custom_postgui.hal
HALUI = halui
```

Achsen antasten: X,Y,Z null setzen

Hierfür sind keine HALUI-Pins verfügbar, es müssen MDI – Kommandos benutzt werden.

Um diese Kommandos zur Verfügung zu stellen, muss die <mark>INI-Datei</mark> editiert werden. Es wird ein neuer Abschnitt [HALUI] angelegt (wenn noch kein solcher existiert). Darin werden folgende Befehle definiert:

```
[HALUI]
MDI_COMMAND = G10 L20 P0 X0
MDI_COMMAND = G10 L20 P0 Y0
MDI_COMMAND = G10 L20 P0 Z0
```

Dies bewirkt, dass neue HALUI-Pins zur Verfügung gestellt werden. Diese sind einfach durchnummeriert:

```
halui.mdi-command-00
halui.mdi-command-01
```

Um diese zu benutzen erweitern wir unser Panel mypanel.xml und fügen drei neue Buttons ein:

```
<button>
<halpin>"zero_x"</halpin>
<text>"X=0!"</text>
<font>('Fixed',14)</font>
```

</button>

:

und entsprechend für Y und Z

Dann müssen noch die Verbindungen zwischen Panel und HALUI in der custom-postgui.hal-Datei festgelegt werden:

net x0 pyvcp.zero_x => halui.mdi-command-00
net y0 pyvcp.zero_y => halui.mdi-command-01
net z0 pyvcp.zero_z => halui.mdi-command-02

Nun stehen die Buttons zum Nullsetzen der X,Y,Z-Koordinaten zur Verfügung.