

Display de Tempo de Usinagem no emc2 (by ALEBE)

- Primeiro, acrescente estas linhas no seu arquivo .hal

Display de Tempo de Usinagem

```
loadrt pwmgen output_type=0 # This creates a 1Hz signal
loadrt encoder num_chan=1 # This will count the 1Hz signals
loadrt edge count=1 # and this looks for a change in program.is-running to reset the timer

addf pwmgen.make-pulses base-thread
addf encoder.update-counters base-thread # If you don't have a base-thread then put it in the servo-
thread
addf pwmgen.update servo-thread
addf encoder.capture-position servo-thread
addf edge.0 servo-thread

setp pwmgen.0.pwm-freq 1
setp pwmgen.0.scale 1
setp pwmgen.0.offset 0
setp pwmgen.0.value 0.5
setp encoder.0.counter-mode 1
setp edge.0.in-edge false

net run-timer halui.program.is-running => pwmgen.0.enable edge.0.in
net OneHzPulse pwmgen.0.pwm => encoder.0.phase-A
net timer-reset edge.0.out => encoder.0.reset
```

- Verifique se existe um arquivo chamado custompanel.xml. Abra-o com o gedit (botão dir. e “abrir com gedit”).
- Acrescente entre as linhas <pyvcp> </pyvcp>. O conteúdo deverá ficar assim:

```
<pyvcp>
  <hbox>
    <labelframe text="Tempo de Usinagem">
      <number>
        <halpin>"timer"</halpin>
        <format>"4.0f"</format>
        <bg>"white"</bg>
      </number>
    </labelframe>
  </hbox>
</pyvcp>
```

Obs.: Se já houver alguma configuração de botão ou outro indicador de painel neste arquivo, sua posição no painel segue a hierarquia de posição dentro do arquivo.

- No arquivo custom-postgui.hal acrescente esta linha:

```
net timer-display encoder.0.position => pyvcp.timer
```

- E no arquivo .ini acrescente ou edite estas linhas nas seções:

```
[DISPLAY]
PYVCP = custompanel.xml
```

e

```
[HAL]
POSTGUI_HALFILE = custom_postgui.hal
HALUI = halui
```

Fonte:

<http://wiki.linuxcnc.org/cgi-bin/emcinfo.pl?SimpleCycleTimer>