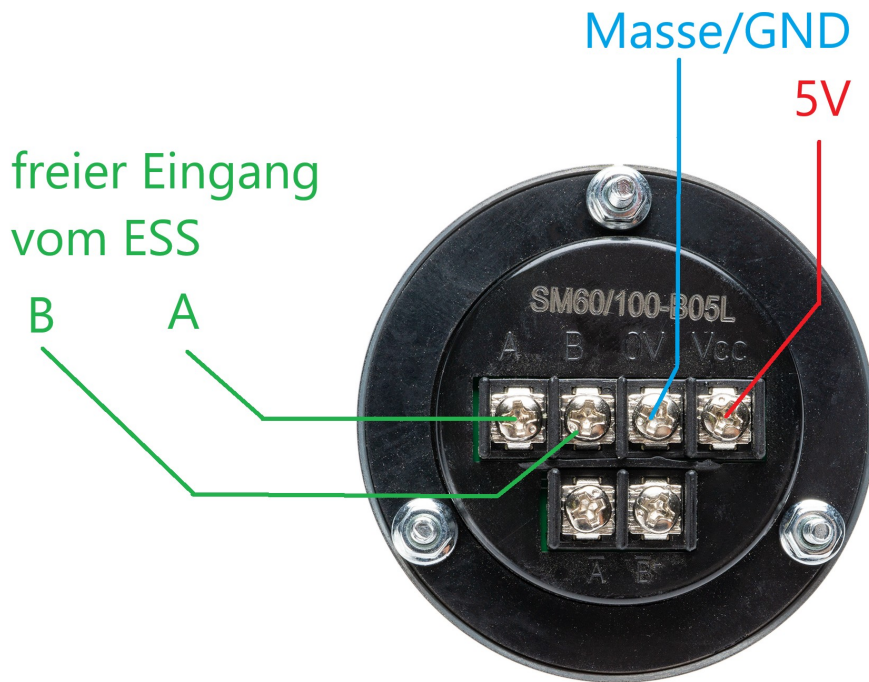


2 Encoder für eine Drehbank X Z für ESS Smoothstepper

Bedienungsanleitung

Alle Rechte an dieser Betriebsanweisung verbleiben bei cnc-technics. Texte, Angaben und Abbildungen dieser Betriebsanweisung dürfen nicht vervielfältigt, verbreitet oder zu Zwecken des Wettbewerbs unbefugt verwertet oder anderen mitgeteilt werden.

Die Zwei Encoder werden an 4 (2x AB-Signale) freien Eingängen vom ESS angeschlossen.
Wenn dies ein 5V Encoder ist können die direkt am ESS angeschlossen werden



Zuordnung in Mach4:

Die 2 Encoder mit den AB Pins müssen im ESS Plugin unter Pins Config zugeordnet werden

Ethernet SmoothStepper (ESS) Configuration v319, Mach4 build 4.2.0.6105 - Instance 0

Info General Motors Spindle Laser Laser... Analog Pins Config Input Signals Output Signals Homing Probing Backlash HC

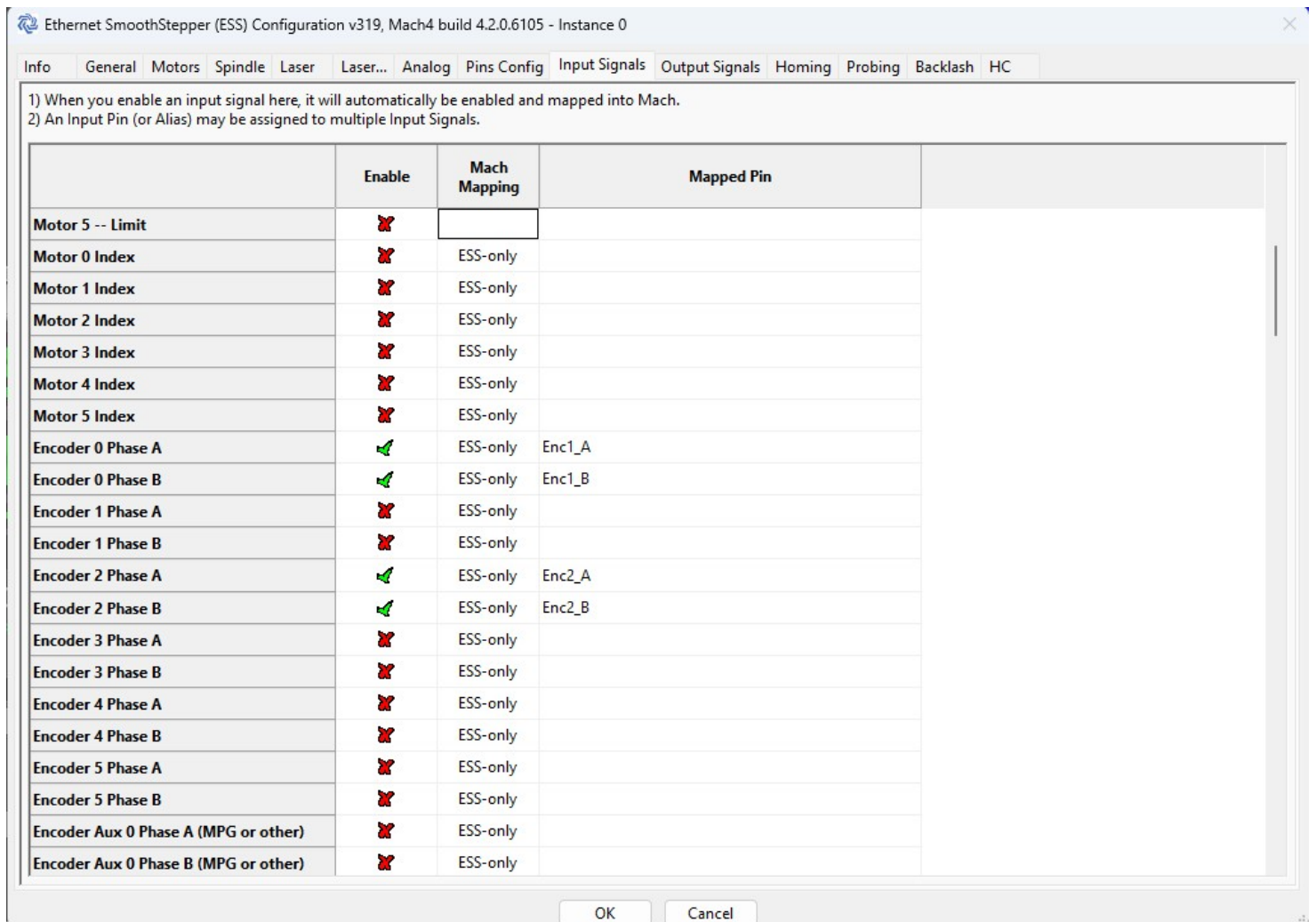
1) Set the pins Active High (Red Arrow Up) or Active Low (Green Arrow Down).
2) Give the Pins you are using an Alias: {P#-#} DESCRIPTION (This is the {Port#-Pin#} for the pin and a description of it).
This makes it MUCH EASIER to identify which pins do what in the Input and Output signal tabs.
3) Assign Noise Filtering, in us, for each input pin, if needed (see the 'Info' tab for more details).
4) A Feed Hold or Stop (Stop, EStop, Disabled or Limit) event can set the Output state to 'Force ON', 'Force OFF', or 'No Change'.
'No Change' means that the output is controlled by Mach4. With 'Force ON' and 'Force OFF' the ESS will force that desired state.
5) Pins are always enabled, only Signals can be enabled or disabled. Connect pins as needed on the Input and Output signal tabs.

Port 2 Pins 2-9 Direction Port 3 Pins 2-9 Direction
☒ Inputs ☐ Outputs ☒ Inputs ☐ Outputs

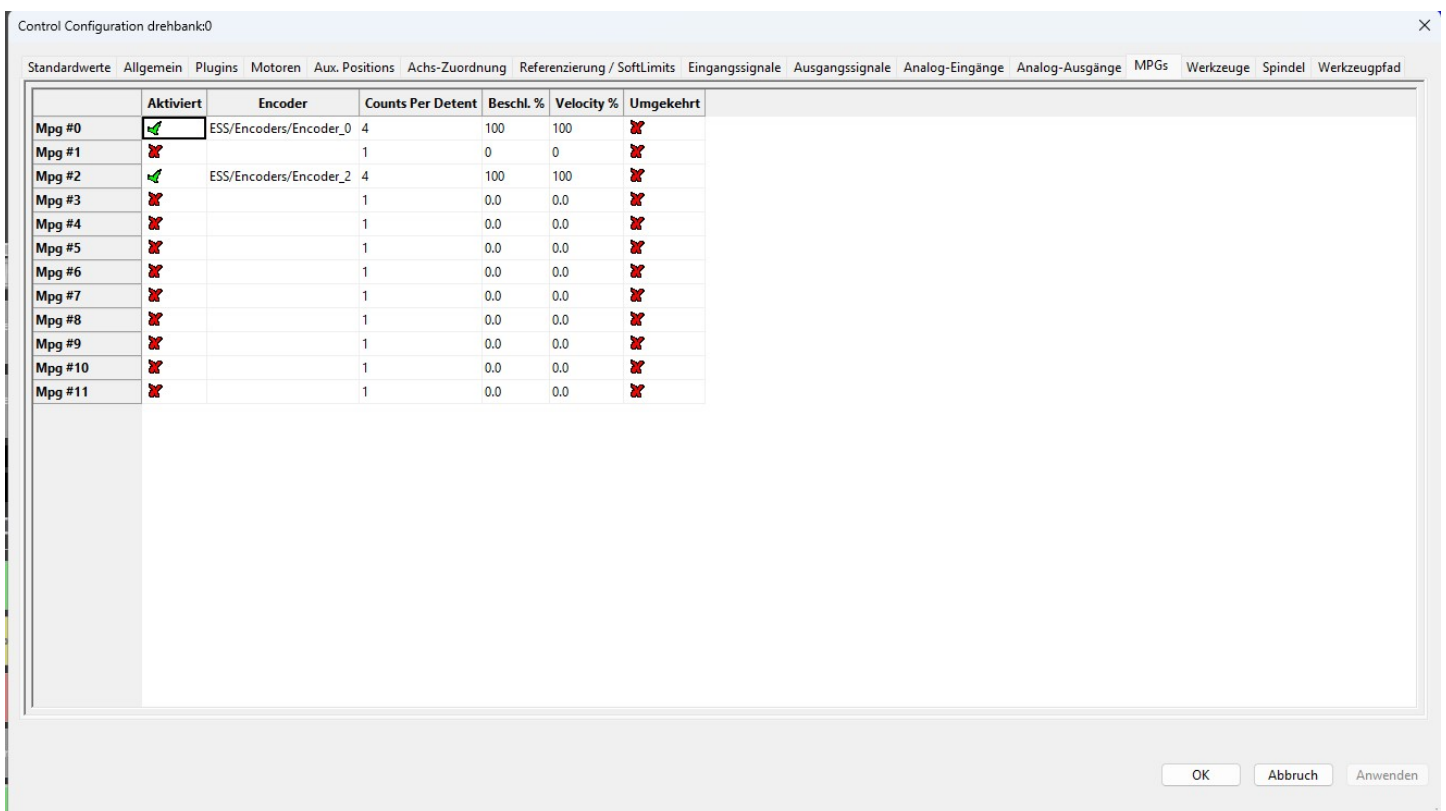
	DIR	Active High/Low	Alias or Name	Noise Filtering	Stop State	Feed Hold State
Port1-Pin1	Out	↑	{P1-1}	----	No Change	No Change
Port1-Pin2	Out	↑	stepX	----	No Change	No Change
Port1-Pin3	Out	↑	dirX	----	No Change	No Change
Port1-Pin4	Out	↑	stepY	----	No Change	No Change
Port1-Pin5	Out	↑	dirY	----	No Change	No Change
Port1-Pin6	Out	↑	stepZ	----	No Change	No Change
Port1-Pin7	Out	↑	dirZ	----	No Change	No Change
Port1-Pin8	Out	↑	{P1-8}	----	No Change	No Change
Port1-Pin9	Out	↑	{P1-9}	----	No Change	No Change
Port1-Pin10	In	↑	Enc1_A	0.00	----	----
Port1-Pin11	In	↑	Enc1_B	0.00	----	----
Port1-Pin12	In	↑	Enc2_B	0.00	----	----
Port1-Pin13	In	↑	Enc2_A	0.00	----	----
Port1-Pin14	Out	↑	{P1-14}	----	No Change	No Change
Port1-Pin15	In	↑	{P1-15}	0.00	----	----
Port1-Pin16	Out	↑	{P1-16}	----	No Change	No Change
Port1-Pin17	Out	↑	ChargePump	----	No Change	No Change

OK Cancel

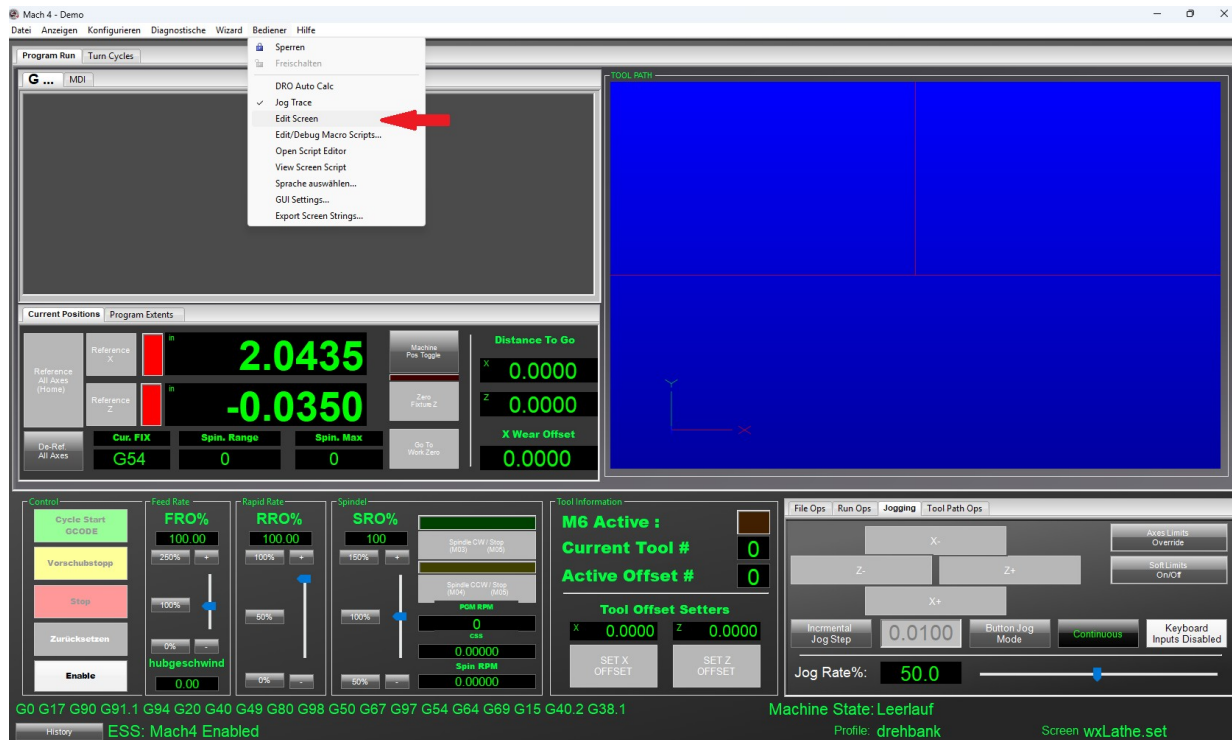
In Input Signals werden sie Mach4 den Encoder Signalen zugeordnet



Und in Mach4 direkt unter MPGs aktiviert

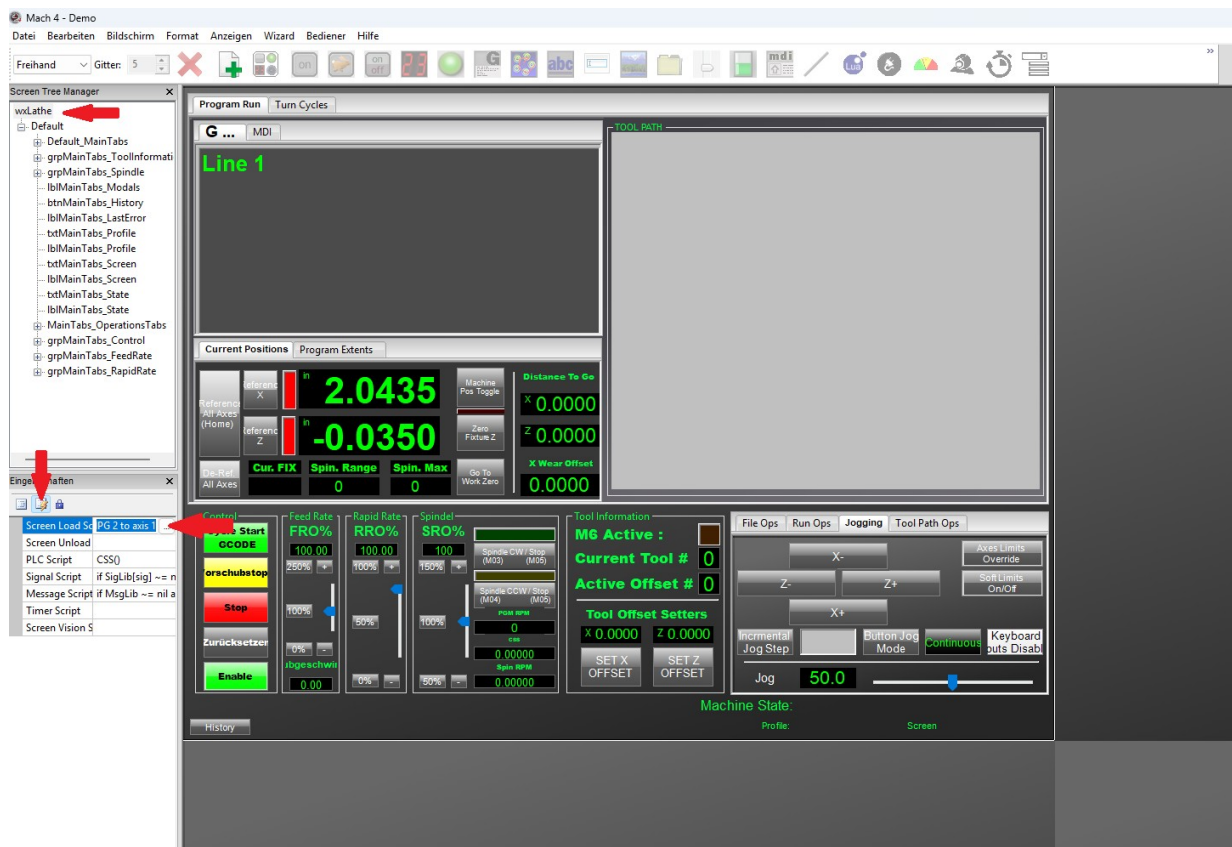


Unter EDIT Screen muss noch eine Anpassung im LUA gemacht werden

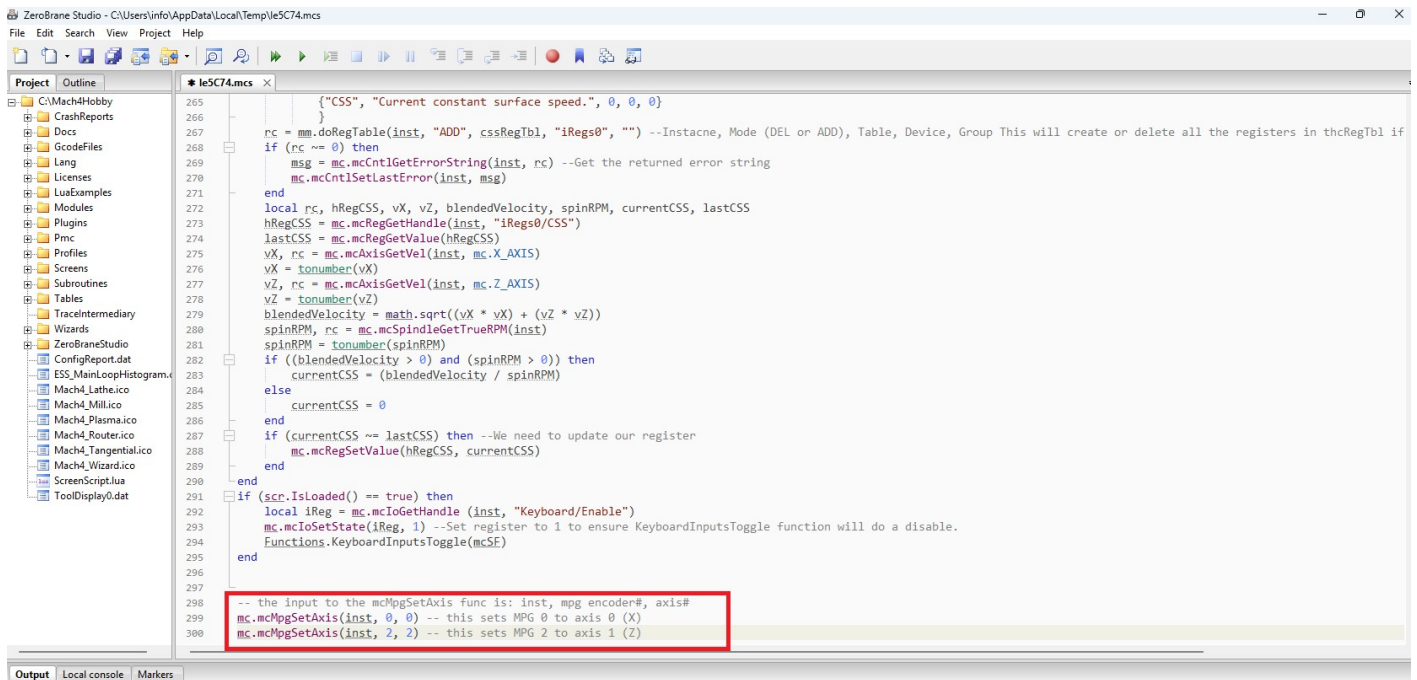


um in das LUA Script zu kommen muss man

- erst wxLathe anklicken
- dann auf das Ereignissfeld
- und dann auf das Screen Load Script



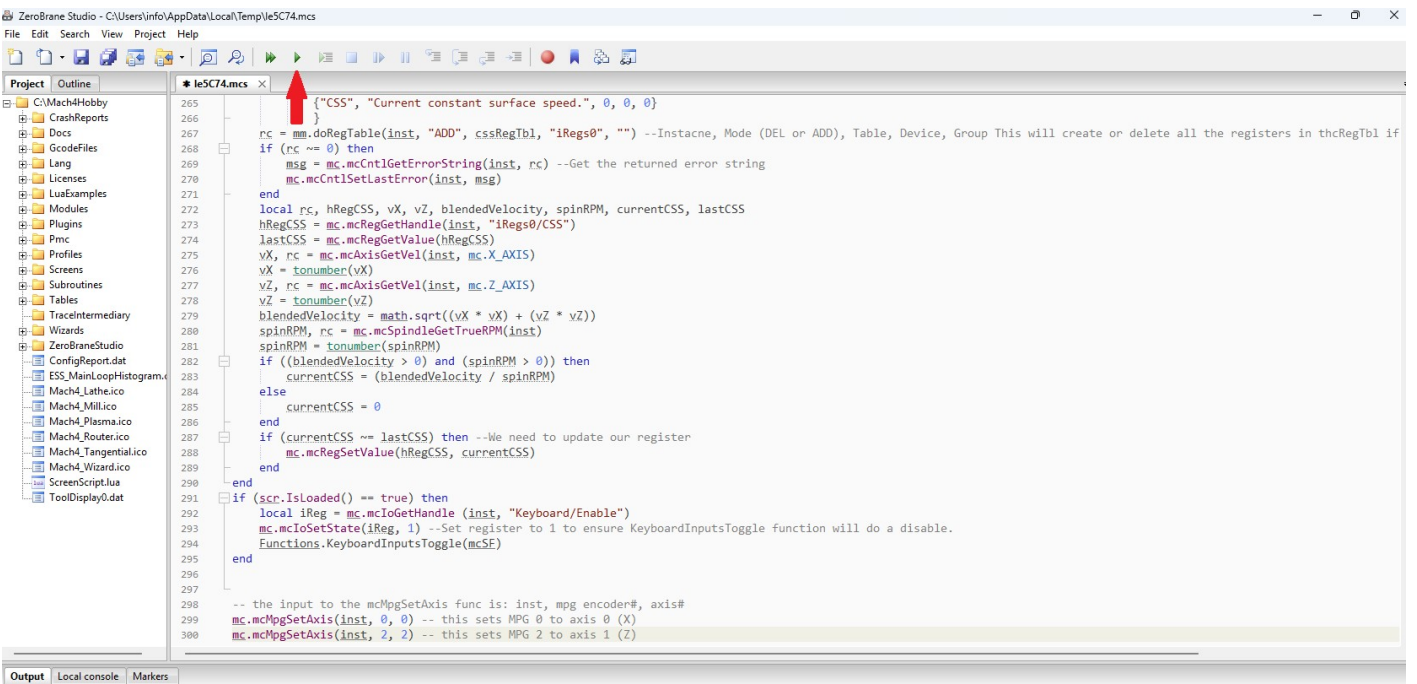
Ganz unten müssen 2 Zeilen eingefügt werden



```
265 { "CSS", "Current constant surface speed.", 0, 0, 0 }
266 }
267 rc = mm.doRegTable(inst, "ADD", cssRegTbl, "iRegs0", "") --Instacne, Mode (DEL or ADD), Table, Device, Group This will create or delete all the registers in thcRegTbl if
268 if (rc ~= 0) then
269     msg = mc.McCtlGetErrorString(inst, rc) --Get the returned error string
270     mc.McCtlSetLastError(inst, msg)
271 end
272 local rc, hRegCSS, vX, vZ, blendedVelocity, spinRPM, currentCSS, lastCSS
273 hRegCSS = mc.McRegGetHandle(inst, "iRegs0/CSS")
274 lastCSS = mc.McRegGetValue(hRegCSS)
275 vX, rc = mc.McAxisGetVel(inst, mc.X_AXIS)
276 vX = tonumber(vX)
277 vZ, rc = mc.McAxisGetVel(inst, mc.Z_AXIS)
278 vZ = tonumber(vZ)
279 blendedVelocity = math.sqrt((vX * vX) + (vZ * vZ))
280 spinRPM, rc = mc.McSpindleGetTrueRPM(inst)
281 spinRPM = tonumber(spinRPM)
282 if ((blendedVelocity > 0) and (spinRPM > 0)) then
283     currentCSS = (blendedVelocity / spinRPM)
284 else
285     currentCSS = 0
286 end
287 if (currentCSS ~= lastCSS) then --We need to update our register
288     mc.McRegSetValue(hRegCSS, currentCSS)
289 end
290 end
291 if (scr.IsLoaded() == true) then
292     local iReg = mc.McIoGetHandle(inst, "Keyboard/Enable")
293     mc.McIoSetState(iReg, 1) --Set register to 1 to ensure KeyboardInputsToggle function will do a disable.
294     Functions.KeyboardInputsToggle(mcSF)
295 end
296
297
298 -- the input to the mcMpgSetAxis func is: inst, mpg encoder#, axis#
299 mc.MpgSetAxis(inst, 0, 0) -- this sets MPG 0 to axis 0 (X)
300 mc.MpgSetAxis(inst, 2, 2) -- this sets MPG 2 to axis 1 (Z)
```

-- the input to the mcMpgSetAxis func is: inst, mpg encoder#, axis#
mc.MpgSetAxis(inst, 0, 0) -- this sets MPG 0 to axis 0 (X)
mc.MpgSetAxis(inst, 2, 2) -- this sets MPG 2 to axis 1 (Z)

danach auch die PLAY Taste zum aktivieren



```
265 { "CSS", "Current constant surface speed.", 0, 0, 0 }
266 }
267 rc = mm.doRegTable(inst, "ADD", cssRegTbl, "iRegs0", "") --Instacne, Mode (DEL or ADD), Table, Device, Group This will create or delete all the registers in thcRegTbl if
268 if (rc ~= 0) then
269     msg = mc.McCtlGetErrorString(inst, rc) --Get the returned error string
270     mc.McCtlSetLastError(inst, msg)
271 end
272 local rc, hRegCSS, vX, vZ, blendedVelocity, spinRPM, currentCSS, lastCSS
273 hRegCSS = mc.McRegGetHandle(inst, "iRegs0/CSS")
274 lastCSS = mc.McRegGetValue(hRegCSS)
275 vX, rc = mc.McAxisGetVel(inst, mc.X_AXIS)
276 vX = tonumber(vX)
277 vZ, rc = mc.McAxisGetVel(inst, mc.Z_AXIS)
278 vZ = tonumber(vZ)
279 blendedVelocity = math.sqrt((vX * vX) + (vZ * vZ))
280 spinRPM, rc = mc.McSpindleGetTrueRPM(inst)
281 spinRPM = tonumber(spinRPM)
282 if ((blendedVelocity > 0) and (spinRPM > 0)) then
283     currentCSS = (blendedVelocity / spinRPM)
284 else
285     currentCSS = 0
286 end
287 if (currentCSS ~= lastCSS) then --We need to update our register
288     mc.McRegSetValue(hRegCSS, currentCSS)
289 end
290 end
291 if (scr.IsLoaded() == true) then
292     local iReg = mc.McIoGetHandle(inst, "Keyboard/Enable")
293     mc.McIoSetState(iReg, 1) --Set register to 1 to ensure KeyboardInputsToggle function will do a disable.
294     Functions.KeyboardInputsToggle(mcSF)
295 end
296
297
298 -- the input to the mcMpgSetAxis func is: inst, mpg encoder#, axis#
299 mc.MpgSetAxis(inst, 0, 0) -- this sets MPG 0 to axis 0 (X)
300 mc.MpgSetAxis(inst, 2, 2) -- this sets MPG 2 to axis 1 (Z)
```

Danach zurück zu Mach4 muss man den Edit Screen beenden

