



LinuxCNC

Spindel synchronisieren + orientieren
mit Schritt und Richtungsignalen

Meine verwendeten Komponenten.

Servodrive

Estun EDB-10A + Estun Servomotor 1KW 1000 U/min



5I25 Superport FPGA based PCI
Anything I/O card



Theorie - Konfiguration der Spindel

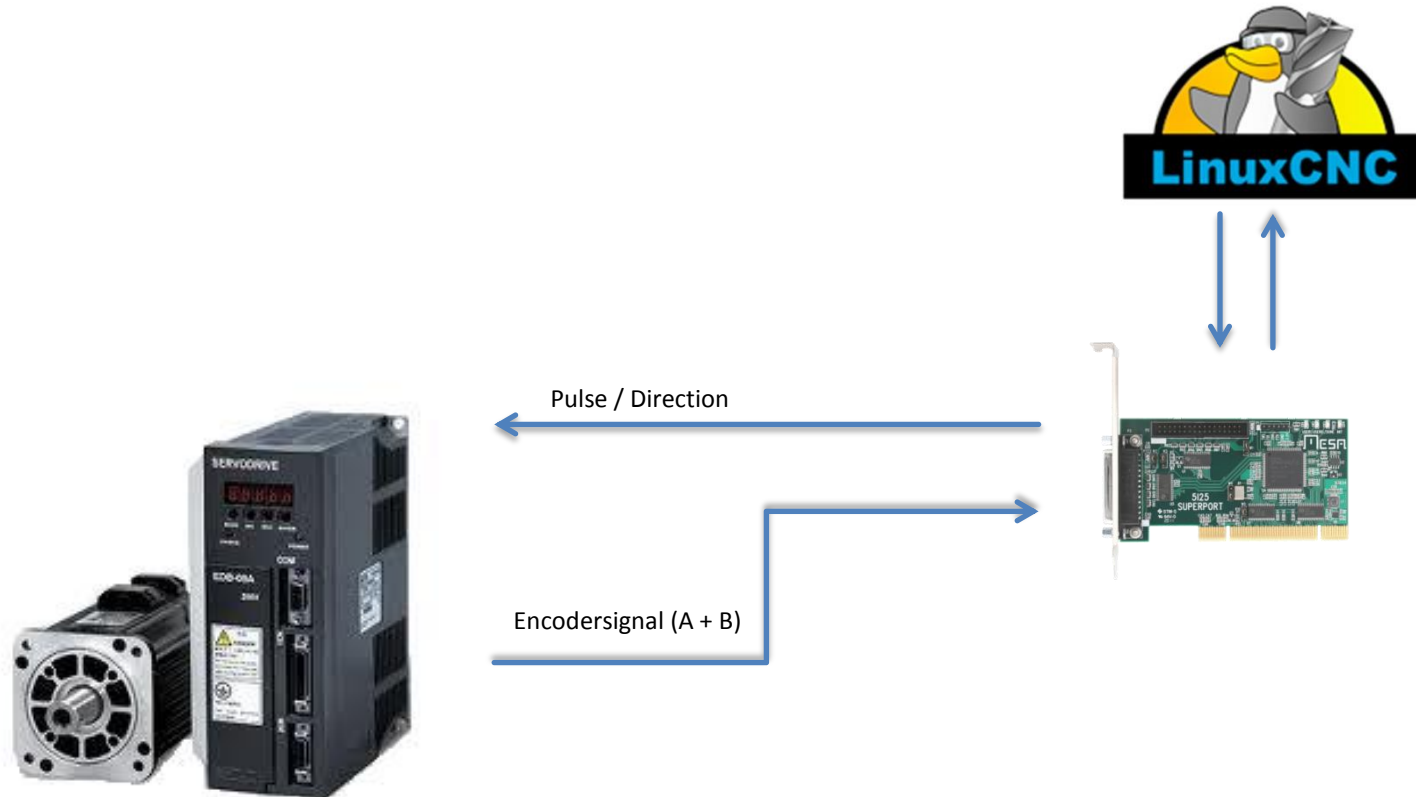
- Drehzahlvorgabe (per Schritt und Richtung)
- Drehzahlrampen (Beschleunigung und Bremsen der Spindel einstellbar)
- Drehzahlvergleich (Achsbewegung beginnt erst wenn die eingestellte Drehzahl erreicht ist)
- Synchronisation (damit diversen G-Befehle (Gewinden) funktionieren)
- Positionierung (über M19 die Spindelposition einstellen)

Aber wie



Theorie - Closed-loop Stepper Konfiguration

LinuxCNC gibt Schritt- Richtungssignale zur Endstufe und der Encoder am Motor meldet die aktuelle Position zurück.



Konfiguration 5i25

Ich nutze Mesaflash um mein Bitfile in die Karte zu laden.

Anschließend habe ich auf den jeweiligen Pins die Funktionen.

```
sudo mesaflash --device 5i25 --readhmid  
sudo mesaflash --device 5i25 --write test.bit  
sudo mesaflash --device 5i25 --reload
```

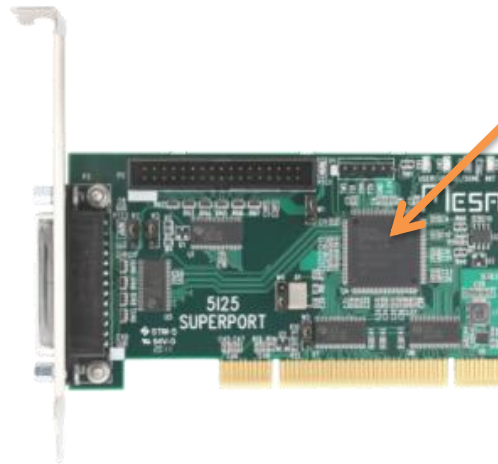
PIN 2 = STEP

PIN 3 = DIR

PIN 11 = A (ENCODER)

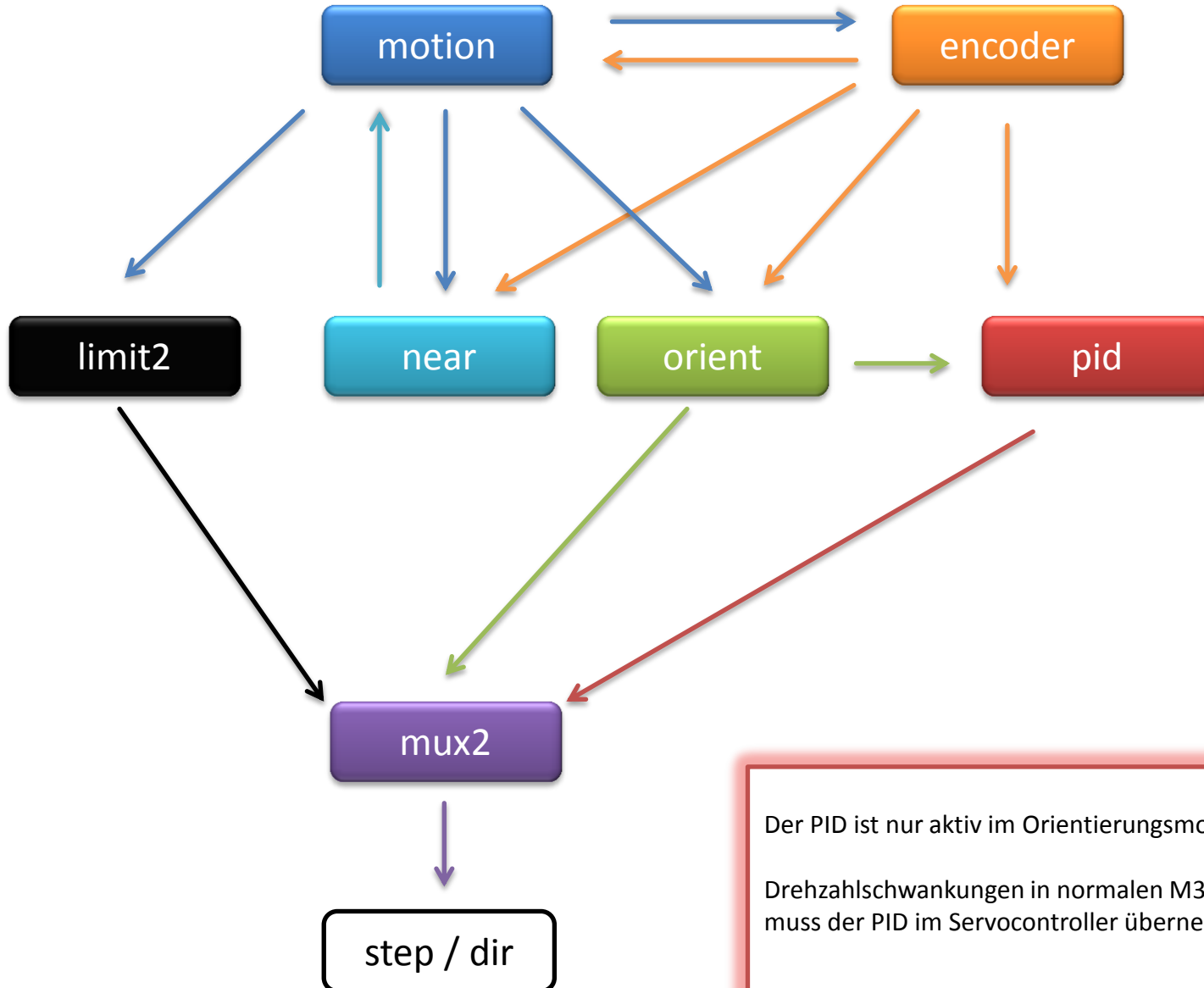
PIN 12 = B (ENCODER)

PIN 13 = C (ENCODER)



Flash bitfile

Übersicht der HAL Komponenten



Der PID ist nur aktiv im Orientierungsmodus.

Drehzahlschwankungen in normalen M3/M4 Betrieb muss der PID im Servocontroller übernehmen.

HAL - Konfiguration der Spindel

Als erstes erstelle ich in der INI eine Sektor [SPINDLE_9] und einige Variablen die dann von der HAL gelesen werden. Damit kann man später die Spindel einfacher parametrieren.

```
*****
```

```
# Spindle
```

```
*****
```

```
[SPINDLE_9]
```

```
DIRSETUP  = 1000      (Impulslänge in Nanosekunden)
DIRHOLD    = 1000      (Impulslänge in Nanosekunden)
STEPLEN    = 1000      (Impulslänge in Nanosekunden)
STEPSPACE  = 1000      (Impulslänge in Nanosekunden)
STEPGEN_MAXVEL  = 1500  (Maximale Schritterzeugung bzw. meine maximale Drehzahl in U/min)
STEPGEN_MAXACCEL = 2000 (Beschleunigung der Schritterzeugung)
OUTPUT_SCALE    = 166.6666 (Impulse pro Sekunde)
```

Berechnung:

```
10000 / 60 = 166.6666 (Encoder Impulse pro Umdrehung durch 60 = Impulse pro Sekunde)
```

```
ENCODER_SCALE = 10000 (Mein Encoder liefert 10000 Impulse pro Umdrehung. Volle Auflösung hier eintragen)
ACCELERATION  = 2000  (mit welcher Beschleunigung und Bremswirkung soll gefahren werden)
MAX_ERROR     = 0.2   (akzeptierter Fehler zwischen Sollwert und Istwert angegeben in U/Sekunde)
OFF_DELAY     = 1.5   (Abschaltverzögerung in Sekunden für den Stepgen)
```

```
P = 300
```

```
I = 0
```

```
D = 0
```

```
FF0 = 0
```

```
FF1 = 0
```

```
FF2 = 0
```

```
BIAS = 0
```

```
DEADBAND = 0
```

```
MAX_OUTPUT = 500 (Die maximale Geschwindigkeit beim orientieren)
```

HAL - Konfiguration der Spindel

SPINDLE

setp	hm2_5i25.0.encoder.00.counter-mode	0	
setp	hm2_5i25.0.encoder.00.filter	1	
setp	hm2_5i25.0.encoder.00.index-invert	0	
setp	hm2_5i25.0.encoder.00.index-mask	0	
setp	hm2_5i25.0.encoder.00.index-mask-invert	0	
setp	hm2_5i50.0.encoder.00.scale	[SPINDLE_9]ENCODER_SCALE	wert aus INI übernehmen
setp	hm2_5i25.0.stepgen.00.dirsetup	[SPINDLE_9]DIRSETUP	wert aus INI übernehmen
setp	hm2_5i25.0.stepgen.00.dirhold	[SPINDLE_9]DIRHOLD	wert aus INI übernehmen
setp	hm2_5i25.0.stepgen.00.steplen	[SPINDLE_9]STEPLEN	wert aus INI übernehmen
setp	hm2_5i25.0.stepgen.00.stepspace	[SPINDLE_9]STEPSPACE	wert aus INI übernehmen
setp	hm2_5i25.0.stepgen.00.position-scale	[SPINDLE_9]OUTPUT_SCALE	wert aus INI übernehmen
setp	hm2_5i25.0.stepgen.00.step_type	0	Geschwindigkeitsmodus
setp	hm2_5i25.0.stepgen.00.control-type	1	
setp	hm2_5i25.0.stepgen.00.maxaccel	[SPINDLE_9]STEPGEN_MAXACCEL	wert aus INI übernehmen
setp	hm2_5i25.0.stepgen.00.maxvel	[SPINDLE_9]STEPGEN_MAXVEL	wert aus INI übernehmen

loadrt limit2	names=spindle-ramp	einmal limit2 mit Namen weiter verwenden
loadrt near	names=spindle-at-speed,spindle-at-pos	zweimal near mit Namen weiter verwenden
loadrt timedelay	names=spindle-active-delay	einmal limit2 mit Namen weiter verwenden
loadrt orient	names=spindle-orient	einmal limit2 mit Namen weiter verwenden
loadrt pid	names=spindle-pid	einmal limit2 mit Namen weiter verwenden
loadrt mux2	names=spindle-pwm-switch	einmal limit2 mit Namen weiter verwenden
loadrt or2	count=0	einmal or2 numerisch weiter verwenden
loadrt not	count=0	einmal not numerisch weiter verwenden
loadrt and2	count=0	einmal and1 numerisch weiter verwenden
loadrt offset	count=0	einmal offset numerisch weiter verwenden

addf spindle-ramp	servo-thread	Komponente in den servo-thread laden damit diese berechnet werden.
addf spindle-at-speed	servo-thread	Komponente in den servo-thread laden damit diese berechnet werden.
addf spindle-at-pos	servo-thread	Komponente in den servo-thread laden damit diese berechnet werden.
addf spindle-active-delay	servo-thread	Komponente in den servo-thread laden damit diese berechnet werden.
addf spindle-orient	servo-thread	Komponente in den servo-thread laden damit diese berechnet werden.
addf spindle-pid.do-pid-calcs	servo-thread	Komponente in den servo-thread laden damit diese berechnet werden.
addf spindle-pwm-switch	servo-thread	Komponente in den servo-thread laden damit diese berechnet werden.
addf or2.0	servo-thread	Komponente in den servo-thread laden damit diese berechnet werden.
addf not.0	servo-thread	Komponente in den servo-thread laden damit diese berechnet werden.
addf and2.0	servo-thread	Komponente in den servo-thread laden damit diese berechnet werden.
addf offset.0.update-output	servo-thread	Komponente in den servo-thread laden damit diese berechnet werden.

HAL - Konfiguration der Spindel

setp spindle-pid.Pgain	[SPINDLE_9]P	wert aus INI übernehmen
setp spindle-pid.Igain	[SPINDLE_9]I	wert aus INI übernehmen
setp spindle-pid.Dgain	[SPINDLE_9]D	wert aus INI übernehmen
setp spindle-pid.bias	[SPINDLE_9]BIAS	wert aus INI übernehmen
setp spindle-pid.FF0	[SPINDLE_9]FF0	wert aus INI übernehmen
setp spindle-pid.FF1	[SPINDLE_9]FF1	wert aus INI übernehmen
setp spindle-pid.FF2	[SPINDLE_9]FF2	wert aus INI übernehmen
setp spindle-pid.deadband	[SPINDLE_9]DEADBAND	wert aus INI übernehmen
setp spindle-pid.maxoutput	[SPINDLE_9]MAX_OUTPUT	wert aus INI übernehmen
setp spindle-pid.error-previous-target true		
setp spindle-ramp.maxv	[SPINDLE_9]ACCELERATION	wert aus INI übernehmen
setp spindle-at-speed.difference	[SPINDLE_9]MAX_ERROR	wert aus INI übernehmen
setp spindle-at-pos.difference	0.01	
setp spindle-at-pos.in1	0	
setp spindle-active-delay.on-delay	0	
setp spindle-active-delay.off-delay	[SPINDLE_9]OFF_DELAY	wert aus INI übernehmen

HAL - Konfiguration der Spindel

orient mit motion verknuepfen

```
net orient-angle    motion.spindle-orient-angle => spindle-orient.angle
net orient-mode     motion.spindle-orient-mode  => spindle-orient.mode
net orient-enable   motion.spindle-orient      => and2.0.in1
```

Position vom Encoder in den pid / orient und motion schieben

```
net spindle-pos      => spindle-pid.feedback    => spindle-orient.position
net spindle-pos      <= motion.spindle-revs    <= hm2_5i25.0.encoder.00.position
```

Encodergeschwindigkeit U/sek in den near und motion schieben

```
net spindle-fb-rps   motion.spindle-speed-in   <= hm2_5i25.0.encoder.00.velocity => spindle-at-speed.in2
```

Positionsvorgabe vom orient in den pid schieben

```
setp offset.0.offset 1
net spindle.orient-cmd spindle-orient.command  => offset.0.in
net spindle-orient-cmd1 offset.0.out           => spindle-pid.command
```

Drehzahlvorgabe U/min aus motion in das limit2 schieben

```
net spindle-speed-rpm motion.spindle-speed-out => spindle-ramp.in
```

Drehzahlvorgabe U/sek aus motion in das near schieben

```
net spindle-speed-rps motion.spindle-speed-out-rps => spindle-at-speed.in1
```

Wenn Solldrehzahl gleich Istdrehzahl dann aus near das bit in motion schieben

```
net spindle-at-speed spindle-at-speed.out => motion.spindle-at-speed
```

Wenn Sollpositon gleich Istpositon dann aus near das bit in motion schieben

```
net spindle-pos-err spindle-at-pos.in2 <= spindle-pid.error
net spindle-at-pos spindle-at-pos.out =>
net spindle-on0 motion.spindle-on => spindle-active-delay.in
net spindle-on1 spindle-active-delay.out => or2.0.in0 => not.0.in
net spindle-on2 and2.0.in0 <= not.0.out
```

Signal ist true wenn in Position zum Beispiel interessant beim WZW

orient und pid aktivieren

```
net orient-active and2.0.out => or2.0.in1 => spindle-pwm-switch.sel
net orient-active => spindle-pid.enable => spindle-orient.enable
net spindle-enable or2.0.out => hm2_5i25.0.stepgen.00.enable
```

motion gibt Signal an Encoder das beim naechsten Z Signal auf 0 gestellt werden soll

```
net spindle-sync motion.spindle-index-enable => hm2_5i25.0.encoder.00.index-enable
sets spindle-sync 1
```

Schrittsignal entweder vom motion oder orient

```
net pwm-switch-in0 spindle-pwm-switch.in0 <= spindle-ramp.out
net pwm-switch-in1 spindle-pwm-switch.in1 <= spindle-pid.output
net pwm-switch-out spindle-pwm-switch.out => hm2_5i25.0.stepgen.00.velocity-cmd
```



LinuxCNC

Spindel Konfiguration

<http://talla83.de/linuxcnc/config.htm>

<http://linuxcnc.org/docs/2.7/html/man/man9/hostmot2.9.html>

<http://linuxcnc.org/docs/2.7/html/man/man9/near.9.html>

<http://linuxcnc.org/docs/2.7/html/man/man9/limit2.9.html>

<http://linuxcnc.org/docs/2.7/html/man/man9/timedelay.9.html>

<http://linuxcnc.org/docs/2.7/html/man/man9/orient.9.html>

<http://linuxcnc.org/docs/2.7/html/man/man9/pid.9.html>

<http://linuxcnc.org/docs/2.7/html/man/man9/offset.9.html>

<http://linuxcnc.org/docs/2.7/html/man/man9/or2.9.html>

<http://linuxcnc.org/docs/2.7/html/man/man9/and2.9.html>

<http://linuxcnc.org/docs/2.7/html/man/man9/mux2.9.html>

<http://linuxcnc.org/docs/2.7/html/man/man9/not.9.html>

<http://linuxcnc.org/docs/2.7/html/man/man9/motion.9.html>