LinuxCNC

Spindel Konfiguration

Meine verwendeten Komponenten.

Lenze Vector 8200
0,37 KW
IO Standart Modul

VEM Drehstrommotor
0,3 KW
2750 U/min

PC

mit Parallelport

Theorie - Konfiguration der Spindel

Ich konfiguriere in der HAL eine Spindel die das PWM Signal an den Regler gibt.

Max Drehzahl = 2750 U/min somit wären 10V = 2750 U/min.

Links- und Rechtslauf sollen über Relais gesteuert werden.

Ein Initiator auf der Motorwelle soll die Drehzahl in LinuxCNC geben.

Aber wie

HAL/INI - 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_9]
PWM_FREQ        = 1000          (Die PWM Frequenz soll hier mal 1000 Hz sein.
ENCODER_SCALE   = 60            (Mein Initiator liefert 1 mpulse pro Umdrehung . Gerechnet habe ich  1 Impuls  x  60 Sekunden =  60~  Impulse/Minute)
OUTPUT_SCALE    = 2750          (bei  +10V die maximale Drehzahl)
```

In der HAL erstelle ich mit den Komponenten pwmgen , encoder und scale die Ausgabe und das Feedback.

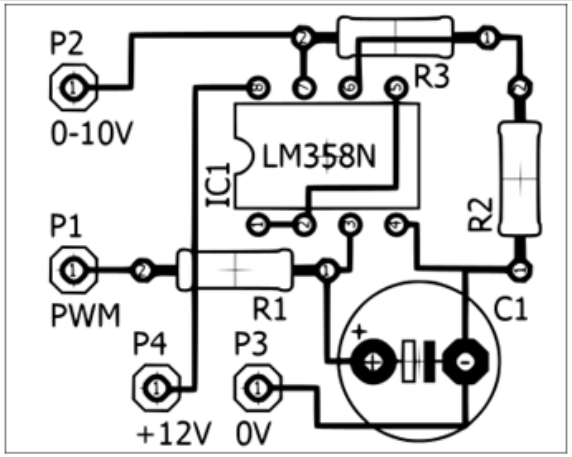
```
#############################################################################
# Spindel mit PWM über Parallelport

loadrt pwmgen output_type=1
addf pwmgen.update servo-thread
addf pwmgen.make-pulses base-thread

setp pwmgen.0.pwm-freq  [SPINDLE_9]PWM_FREQ                 (hier wird z.B. der Wert aus der INI übernommen)
setp pwmgen.0.scale     [SPINDLE_9]OUTPUT_SCALE             (hier wird z.B. der Wert aus der INI übernommen)
setp pwmgen.0.offset 0
setp pwmgen.0.dither-pwm true

loadrt encoder names=initiator                              (Encoder laden und mit Namen weiter verwenden)
addf encoder.update-counters base-thread
addf encoder.capture-position servo-thread

setp initiator.counter-mode true                            (Countermode aktivieren – nur A wird gezählt)

loadrt scale count=1
addf scale.0 servo-thread
setp scale.0.gain [SPINDLE_9]ENCODER_SCALE                  (hier wird z.B. der Wert aus der INI übernommen)

net test initiator.velocity scale.0.in

net spindel-feedback        motion.spindle-speed-in <= scale.0.out

net sollwert                motion.spindle-speed-out        pwmgen.0.value      (Sollwert vom Planer an das PWM geben)
net spindle-an              pwmgen.0.enable <= motion.spindle-on => parport.0.pin-17-out    (PIN17 soll ein Relais schalten damit der FU los legt )
net spindle-dr              motion.spindle-forward => parport.0.pin-14-out      (PIN14 soll ein Relais schalten zum Drehrichtungswechsel)
net spindle-pwm             pwmgen.0.pwm => parport.0.pin-16-out                (PIN16 soll das PWM Signal  rauskommen)
net spindle-index           initiator.phase-A <= parport.0.pin-10-in            (PIN17 mit dem encoder.phase-A verbinden)

#############################################################################
```

PWM zu 0-10V Konfiguration der Spindel

Um ein 0- 10V Signal zu erhalten verwende ich folgende Schlatung.

Pin	Funktion	Wert	Einheit
P1	PMW	5	V
P2	ANALOG	10	V
P3	SP-N	0	V
P4	SP-P	12	V

Bauteil	Wert	Beschreibung
R1	4K7	Widerstand
R2	10K	Widerstand
R3	10K	Widerstand
LM358	-	Operationsverstärker
C1	4,7µF	Kondensator

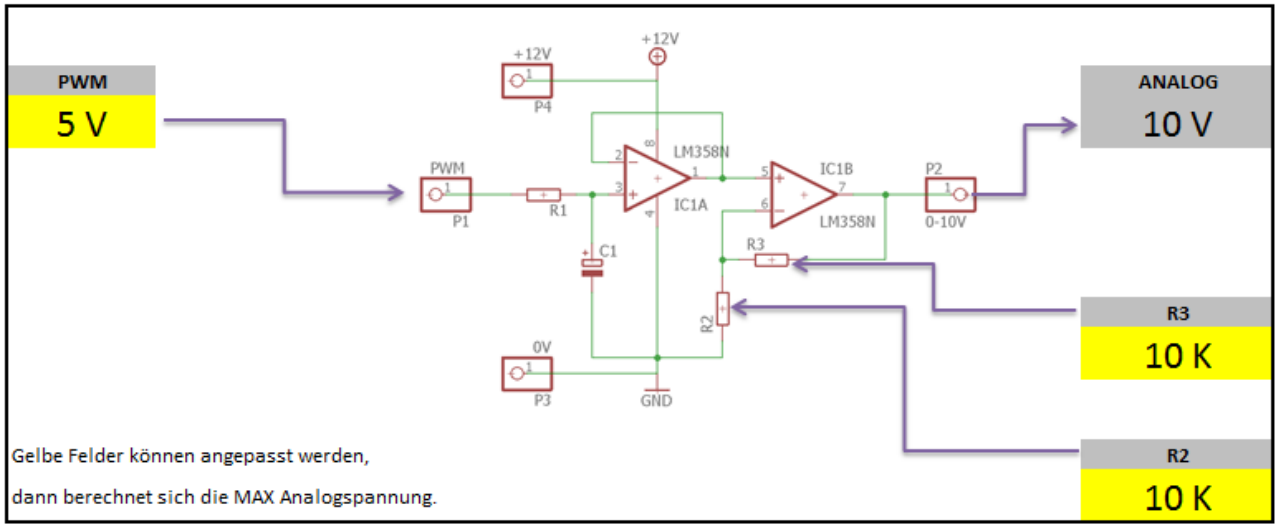


Beschreibung der Schaltung.....
Den ersten Teil des Operationsverstärkers nutze ich als Spannungsfolger.
Der zweite Teil verstärkt dann das PWM Signal um einen Faktor der sich durch
R2 und R3 einstellen lässt.

Die Berechnung der Verstärkungsfaktors geht wie folgt.

1 + R3 /R2

Ein Beispiel dazu: 1 + (10K/10K) = 2
Somit würde eine Spannung von 5V mit 2 Multipliziert werden.



LinuxCNC

Spindel Konfiguration

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

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

http://linuxcnc.org/docs/2.7/html/man/man9/pwmgen.9.html

http://linuxcnc.org/docs/2.7/html/man/man9/encoder.9.html

http://linuxcnc.org/docs/2.7/html/man/man9/scale.9.html