



## LMN-StepperController v 4.0

Document revision: 1.2

26.10.04

### Überblick

Für die Auswahl unterschiedlicher Lasereffekte in einer Laserbank kommen unter anderem Strahlschalter zum Einsatz. Der LMN-StepperController (Version 4.0) verwaltet bis zu 32 Schrittmotoren, die über Treiberplatinen angesteuert werden. Durch einen Timerinterrupt können alle Motoren zeitgleich angesteuert werden. In der momentanen Firmwareversion kann bei voller Anzahl von Motoren ein maximaler Schritttakt von 2 kHz erzielt werden. Die derzeit verwendeten Stepper begrenzen die Taktrate jedoch und werden mit einer Schrittdauer von 3 ms betrieben. 24 Schritte (entsprechend  $43,2^\circ$ ) von der Nullposition bis in den Strahl werden demnach in 72 ms ausgeführt. Durch Feintuning, beispielsweise die Verminderung der nötigen Schrittzahl, kann diese Dauer weiter vermindert werden. Da der Wert jedoch bei einer Firmwareversion eine feste Konstante darstellt, kann diese Zeitdauer beim Timing von Lasershows berücksichtigt werden. Die Ansteuerung durch den Steuerrechner erfolgt mittels RS232 über ein ungekreuztes 9-poliges Kabel (Comparameter: 38400, 8N1).

---

LMN-Lasertechnik ist ein Hobbyprojekt ohne kommerzielle Interessen. Alle im vorliegenden Text wiedergegebenen Schaltungen und Verfahren werden ohne Rücksicht auf die Patentlage mitgeteilt. Sie sind daher nur für Amateur- und Lehrzwecke und dürfen nicht gewerblich genutzt werden. Die Verwendung der enthaltenen Informationen erfolgt auf eigene Gefahr! Obwohl sorgfältig und nach bestem Wissen vorgegangen wurde, kann keine Haftung für Schäden jeder Art übernommen werden. Es wird ausdrücklich auf die von Laserstrahlung und dem Umgang mit elektrischer Spannung ausgehenden Gefahren hingewiesen. Konstruktive Kritik, Mitteilung von Fehlern und sonstige Hinweise werden natürlich gern angenommen.

---

## Featureliste

- Mikrocontrollersteuerung
- Betrieb von bis zu 32 Motoren
- bei 32 Motoren maximale Schrittrate von 2 kHz
- Interruptsteuerung für gleichzeitige Ansteuerung aller Motoren
- schnelle und störungssichere Kommunikation mit Steuerrechner über RS232
- automatische Nullung der Spiegelarme bei Systemreset
- Haltestromschaltung für stabile Armpositionen
- Watchdogschaltung gegen Prozessorfehlfunktion
- 100 % kompatibel zu bestehenden Treiberplatinen
- Schutzbeschaltung des Leistungsteils der Treiberplatinen
- in-system-programmierbar für einfache Firmwareupdates

## Befehlsreferenz

Die Kommandos von Steuersystem zum StepperController sind genau ein Byte lang. Die implementierten Kommandos sind in der folgenden Tabelle verzeichnet. Sonstige Binärkombinationen sind verboten. Nach dem Übersenden eines Kommandobytes darf der PC keine weiteren Zeichen senden, bevor der StepperController den Empfang des Kommandos nicht quittiert hat. Es kommen zwei mögliche Quittierungen in Frage:

- ACK (ASCII „A“), das Kommando wurde verstanden und wird ausgeführt
- NACK (ASCII „Z“), das Kommando konnte nicht interpretiert werden

Am Steuerrechner sollte das Warten auf die Quittierung durch ein Timeout abgesichert werden.

<b><i>Binär</i></b>	<b><i>Hex</i></b>	<b><i>ASCII</i></b>	<b><i>Kommando</i></b>
0011.0000	30	0	Identifikation senden
0011.0001	31	1	Softwarereset auslösen
0011.0010	32	2	Leistungsteil offline schalten
0011.0011	33	3	Leistungsteil online schalten
0011.0100	34	4	Schaltzustände berichten
010n.nnnn	40-5F	@,A..Z,[,\,],^,-	Stepper n aus dem Strahl
011n.nnnn	60-7F	',a..z,{, ,},~,DEL	Stepper n in den Strahl
sonst.	-	-	reserviert

*Tabelle 1: Kommandos an den StepperController*

Die Option „Leistungsteil offline schalten“ deaktiviert den 12 V Zweig der Treiberplatinen komplett und verringert damit die Leistungsaufnahme der Schaltung. Es ist jedoch zu beachten, dass die haltestromlosen Motoren anfällig für die Bewegung der Arme sind und durch die Federkraft der Mikroschalter die Nullposition verlassen können. Die Verwendung dieser Option wird daher

ausdrücklich nur bei abgeschaltetem Laser beziehungsweise bei geschlossenem Shutter (man hat doch einen Shutter!!) empfohlen. Der Leistungsteil wird durch einen Systemreset wieder online gebracht. Damit ist sichergestellt, dass alle Motoren wieder eine definierte Position einnehmen.

Eine Ausgabe der Motorpositionen (Sollwerte) kann mit dem Kommando „Schaltzustände berichten“ erreicht werden. Es wird ein ASCII-String zurückgeliefert, der in vier Achterblöcken mit „0“ Motoren in Nullposition und mit „x“ Motoren im Strahl kennzeichnet. Dies ist im wesentlichen eine Testfunktion und wird bei der Ansteuerung durch Showsoftware keine Beachtung finden müssen.

Mit dem Steuersystem (PC) wird der StepperController mit einem 1:1 belegten ungekreuzten 9 poligen Kabel verbunden. Der auf der Platine befindliche 10-polige Pfostensteckverbinder ist in der Pinbelegung auf den Anschluss eines Schneid-Klemm-Steckers (Männchen) ausgelegt (unter Berücksichtigung der Pinvertauschung). Die Parameter der seriellen Übertragung lauten: 38400, 8N1.

## Hardwareänderungen

Die folgenden Veränderungen wurden an der Platine vorgenommen und müssen gegebenenfalls bei einem Redesign der Schaltung berücksichtigt werden:

Ist die Schaltung am aktiven Rechner angesteckt, so konnte kein Fehlverhalten (selbständiges Umschalten etc.) festgestellt werden. Dagegen kommt es bei abgestecktem seriellen Kabel bzw. abgeschaltetem Rechner gelegentlich zu Fehlern (< 1 Fehler/Stunde). Daher wurden 10k Pull-down in RXT und TXD nach dem MAX232 eingebaut. Im Langzeittest (24 h) wurde kein Fehler festgestellt.

## Treiberplatinen

Die eigentliche Schaltung zum Treiben der Stepper. Diese Schaltung wird mit der Controllerplatine über ein 40-poliges Flachbandkabel verbunden. Es können bis zu acht Platinen angeschlossen werden. Jede Platine kann dabei vier Motoren treiben. Durch die Jumperblöcke (siehe Schaltplan) wird jeder Platine in 4er-Schritten Adressraum zugeordnet (Decoder auf der Controllerplatine). Innerhalb dieser Adressbereiche wird mit A0 und A1 der einzelne Motor selektiert (Decoder auf der Treiberplatine). Der Controller schreibt also jeweils in das entsprechende Latch die Bitfolgen, um die Schritte eines Motors auszulösen. Die unteren 4 Bit dienen dabei der Bewegung. Die höherwertigen sind am Treiber (ULN2803) mit Widerständen versehen und können für reduzierten Haltestrom verwendet werden. Diese Funktionalität ist allerdings in der derzeitigen Firmware nicht implementiert. Die Idee dabei ist, dass nach Beendigung der Bewegung, die Bits für den Haltestrom gesetzt werden und somit ein verminderter Haltestrom fließt. Ein Haltestrom ist überhaupt notwendig, da die Strahlschalter mit einem Endanschlagsschalter (Mikrotaster) ausgestattet sind. Ohne Stromfluss würde der Stößel des Tasters den Arm wieder wegstoßen. Cleverer wäre daher eine berührungslose Positionserkennung, bei der dann auf einen Haltestrom verzichtet werden könnte.

Belegung der Stecker Treiberplatine - Schrittmotor:

1. SM0 (Kabel am Motor: braun)
2. SM1 (Kabel am Motor: gelb)
3. SM2 (Kabel am Motor: schwarz)
4. SM3 (Kabel am Motor: orange)
5. IN
6. VCC
7. +12V
8. +12V
9. GND

Auf den Treiberplatinen wird der Jumper J6 für die Schalter an den Schrittmotoren auf Pull-Up gestellt. Wenn der Schalter betätigt wird – Nullposition des Armes ausserhalb des Strahls – liegt LOW am Eingang, sonst HIGH.