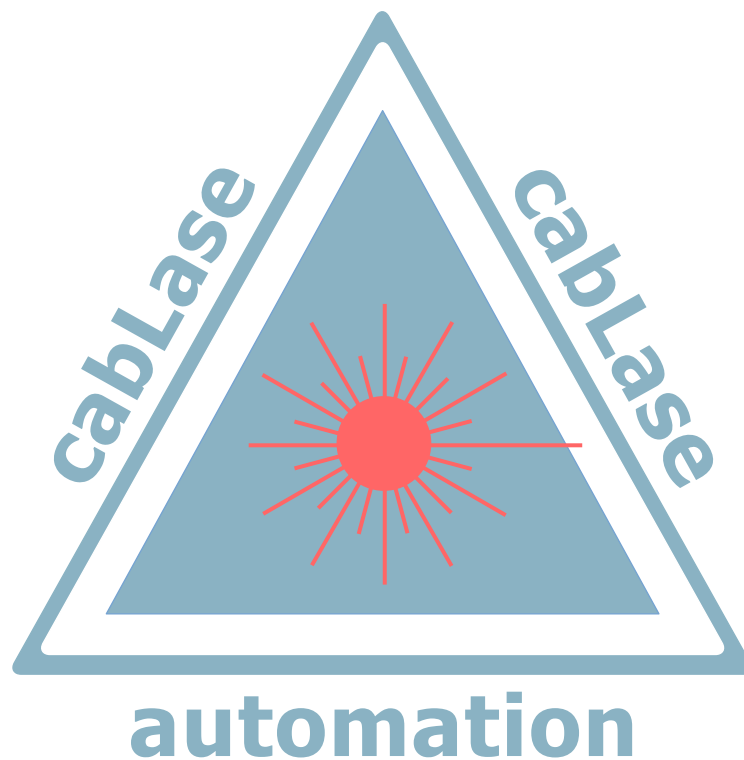


SPS Programmieranleitung



cab FL⁺
Remote API Schnittstelle

Art.-Nr.: 9009855

1	Vorwort.....	5
2	Einleitung.....	7
2.1	Hinweise.....	7
2.2	Bestimmungsgemäßer Gebrauch	7
3	Einleitung.....	8
3.1	Allgemein	8
3.2	Funktionalität der Remote API Schnittstelle	8
3.3	API Kommunikationsübersicht	9
3.4	Verbindung über die TCP/IP Schnittstelle	10
3.5	Hinweise zu cab SPS Demoprogrammen.....	11
3.5.1	Hard- und Software	11
3.5.2	Informationen zu Programmbeispielen	11
3.5.3	Einrichten der Netzwerkverbindung	13
4	Programmbeispiele.....	14
4.1	Beispiel 1.....	14
4.1.1	Beschreibung	14
4.1.2	Layout	14
4.1.3	SPS Programmbeispiel	14
4.1.4	Flowchart.....	15
4.2	Beispiel 2.....	16
4.2.1	Beschreibung	16
4.2.2	Layout	16
4.2.3	SPS Programmbeispiel	16
4.2.4	Flowchart.....	17
4.3	Beispiel 3.....	18
4.3.1	Beschreibung	18
4.3.2	Layout	18
4.3.3	SPS PProgrammbeispiel.....	18
4.3.4	Flowchart.....	19
4.4	Beispiel 4.....	20
4.4.1	Beschreibung	20
4.4.2	Layout	20
4.4.3	SPS Programmbeispiel	20
4.4.4	Flowchart.....	21
5	Einrichten einer TCP/IP Verbindung über cabLase Editor 5.....	22
5.1	Auswahl des Netzwerkadapters.....	22
5.2	Manuelles Verbinden des Lasers mit cabLase Editor 5	24
5.3	Automatisches Verbinden des Lasers mit cabLase Editor 5.....	26
5.4	Ändern der IP-Adresse.....	27
6	Vorbereiten des Stand-Alone Betriebs über cabLase Editor 5.....	28
6.1	Allgemein	28
6.2	Speichern von *.wlj Dateien im Laserspeicher.....	28
6.3	Verwalten von *.dat Dateien.....	29
6.4	Speichern von Laser-Font-Dateien	30

7	Auswahl API Befehle.....	31
7.1	API Befehlssatz.....	31
7.1	Abort.....	31
7.2	ClearJobList	32
7.3	ConnectNetworkShare	32
7.4	Echo	32
7.5	EnableObject.....	33
7.6	ExecuteJobContinuous	33
7.7	ExecuteJobOnce.....	34
7.8	GetAllIOWords	35
7.9	GetFlashJobFileList	36
7.10	GetFontFileList.....	36
7.11	GetKFactor.....	37
7.12	GetNetworkJobFileList.....	37
7.13	GetObjectCenter	37
7.14	GetUSBJobFileList.....	38
7.15	HardwareReset	38
7.16	LoadFlashJob.....	39
7.17	LoadNetworkJob	39
7.18	LoadUSBJob	40
7.19	MakeJobActive.....	41
7.20	RemoveJob	41
7.21	RemoveObject	41
7.22	ReleaseHostControl	42
7.23	ResetObject	42
7.24	ResetUserTransform	42
7.25	SetExternalStartMode	43
7.26	SetObjectString	43
7.27	TakeHostControl.....	44
7.28	TransformObject.....	45
8	API Befehle komplett	46
8.1	Steuerung.....	46
8.2	Objekte	48
8.3	Beschriftungsauftrag	49
8.4	Administration	50
9	Remote API Host Response Codes	51
10	Remote API Systemfehlermeldungen	56
11	Remote API Objekttypen	57
12	Integrationsschnittstellen	58
12.1	Schnittstelle Extern I/O CON2.....	58
12.2	Schnittstelle Remote CON3	59
12.3	Schnittstelle Interlock CON4	61
13	Service	62
13.1	Referenzdokumente.....	62
13.2	Revisionierung der Beispielprogramme	62
13.3	Kontakt	62

Ausgabe: 161121

Art.-Nr. Programmieranleitung: 9009855

Lieferumfang

Der Lieferumfang ist in den Begleitpapieren der Warensendung beschrieben.

Die Lieferung ist auf Vollständigkeit und einwandfreien Zustand zu prüfen. Bei Unstimmigkeiten ist der Lieferant umgehend zu informieren.

Allgemein

Diese Programmieranleitung enthält weder Anweisungen über die grundsätzliche Bedienung eines Computers, bzw. einer speicherprogrammierbaren Steuerung, noch Anweisungen zu den Grundfunktionen der Betriebssysteme Windows® oder Mac OS®.

Informationen über die Bedienung Ihres Computers oder Betriebssystems ist in den entsprechenden Handbüchern des Computers nachzuschlagen.

Herausgeber

cab Produkttechnik GmbH & Co KG
Wilhelm-Schickard-Str. 14
D-76131 Karlsruhe
Telefon: +49 (0)721/66 26 0
Telefax: +49 (0)721/6626 249
<http://www.cab.de>
e-mail: laser@cab.de

Urheberrecht

Diese Dokumentation sowie Übersetzungen hiervon sind Eigentum der cab Produkttechnik GmbH & Co KG.

Das Reproduzieren, Verarbeiten, Vervielfältigen oder Verbreiten im Ganzen oder in Teilen zu anderen Zwecken als der Verfolgung der ursprünglichen bestimmungsgemäßen Verwendung erfordert die vorherige schriftliche Genehmigung von cab.

Jede mögliche Software, die Teil dieses Produkts darstellt, wird unter Lizenz zur Verfügung gestellt und darf nur in Übereinstimmung mit den Lizenzbedingungen verwendet oder kopiert werden.

Warenzeichen

Microsoft® ist ein eingetragenes Warenzeichen der Microsoft Corporation.

Windows XP®, Vista®, Windows 7®, Windows 8® und Windows 10® sind eingetragene Warenzeichen der Microsoft Corporation.

TrueType™ ist ein eingetragenes Warenzeichen von Apple Computer, Inc.

Alle anderen aufgeführten Firmen- und Produktnamen und deren Warenzeichen sind geschütztes Eigentum der jeweiligen Inhaber.

Redaktion

Bei Fragen oder Anregungen bitte an cab Produkttechnik GmbH & Co KG Adresse Deutschland wenden.

Aktualität

Durch die ständige Weiterentwicklung der Geräte können Abweichungen zwischen der Dokumentation und der Funktionalität am Gerät auftreten.

Die aktuelle Ausgabe ist bei cab Produkttechnik GmbH & Co KG Adresse Deutschland erhältlich.

Geschäftsbedingungen

Lieferungen und Leistungen erfolgen zu den Allgemeinen Verkaufsbedingungen von cab.

cab Produkttechnik GmbH & Co KG
Wilhelm-Schickard-Str. 14
D-76131 Karlsruhe
Phone +49 721 6626-0
Fax +49 721 6626-249
www.cab.de
info@cab.de

Frankreich

cab technologies s.a.r.l.
B.P. 50020
Z.A. Nord du Val de Moder
F-67350 Niedermodern
Phone +33 388 722 501
Fax +33 388 722 502
www.cab-technologies.fr
info@cab-technologies.fr

Taiwan

cab Technology Co, Ltd.
9F-8, No. 700,
Jhongjheng Road
Junghe 23552, Taipei, Taiwan R.O.C.
Phone +886 2 8227 3966
Fax +886 2 8227 3566
www.cabasia.net
cabasia@cab.de

Weitere Vertretungen auf Anfrage

USA

cab Technology Inc.

90 Progress Avenue Unit #2
Tyngsboro MA, 01879
Phone +1 978 649 0293
Fax +1 978 649 0294
www.cabtechn.com
info@cabtechn.com

China

cab (Shanghai) Trading Co., Ltd.
Room 11C60,
2299, Yan An Road West,
Shanghai 200336, China
Phone +86 21 6236 3161
Fax +86 21 6236 3162
www.cabasia.net
cabasia@cab.de

2.1 Hinweise

Wichtige Informationen und Hinweise sind in dieser Dokumentation folgendermaßen gekennzeichnet:



Gefahr!

Macht auf eine außerordentliche große, unmittelbar bevorstehende Gefahr für Gesundheit oder Leben aufmerksam.



Warnung!

Weist auf eine gefährliche Situation hin, die zu Körperverletzungen oder zu Schäden an Sachgütern führen kann.



Achtung!

Macht auf mögliche Sachbeschädigung oder einen Qualitätsverlust aufmerksam.



Hinweis!

Ratschläge zur Erleichterung des Arbeitsablaufs oder Hinweis auf wichtige Arbeitsschritte.



Umwelt!

Tipps zum Umweltschutz.



Handlungsanweisung



Verweis auf Kapitel, Position, Bildnummer oder Dokument.



Option (Zubehör, Peripherie, Sonderausstattung).

Zeit Darstellung in Display / Monitor.

2.2 Bestimmungsgemäßer Gebrauch

- Die Softwarebeispiele sind nach dem Stand der Technik und den anerkannten sicherheitstechnischen Regeln programmiert. Dennoch können bei der Verwendung Gefahren für Leib und Leben des Benutzers oder Dritter bzw. Beeinträchtigungen des Gerätes / der Maschine und anderer Sachwerte entstehen.
- Die Software ist ausschließlich zur Verwendung mit geeigneten, vom Hersteller zugelassenen Systemen bestimmt.
- Die Sicherung von Daten, Anwenderinformationen, Systemeinstellungen und maschinenabhängigen Konfigurationsdaten ist Aufgabe des Anwenders. Für deren Verlust oder daraus resultierendem Schaden haftet der Hersteller nicht; das Risiko trägt allein der Anwender.
- Eine andere als die bestimmte oder darüber hinausgehende Benutzung ist nicht bestimmungsgemäß. Für aus missbräuchlicher Verwendung resultierende Schäden haftet der Hersteller nicht; das Risiko trägt allein der Anwender.
- Zur bestimmungsgemäßen Verwendung gehört das Beachten der Montage- und Programmieranleitung, einschließlich der vom Hersteller gegebenen Wartungsempfehlungen und -Vorschriften.
- Diese Programmieranleitung enthält weder Anweisungen über die grundsätzliche Bedienung eines Computers, noch Anweisungen über Grundfunktionen von spezifizierten Betriebs- und Steuerungssystemen.

3.1 Allgemein

Die Steuerung LEC-2 der cab Beschriftungslaser FL+ wurde als leistungsstarke Stand-Alone Lösung entwickelt und verfügt über die Möglichkeit, über Remote API Befehle von extern gesteuert und überwacht zu werden. Die Steuerung quittiert empfangene Befehle mit sogenannten Response Codes, einer von der Steuerung generierte Antwort.

Die Kommunikationsplattform ermöglicht, die cab Beschriftungslaser in moderne Industrie 4.0 Netzwerke und Maschinensteuerungen zu integrieren.

Eine vielfältige Auswahl von Remote API Befehlen erlaubt, Beschriftungsaufträge in den Auftragspeicher der Lasersteuerung zu laden, umzubenennen, auszuführen oder zu verwalten.

Die Kommunikation ist in die Bereiche Host und Client unterteilt. Unter dem Host versteht man die Steuerung des Beschriftungslasers. Clients sind alle übergeordneten Steuerungen auf der Prozessleitebene, welche mit der Lasersteuerung über Remote API kommunizieren.

Grundsätzlich stehen für die Kommunikation zwei Datenschnittstellen zur Verfügung. Die Remote API kann über eine TCP/IP Socket Verbindung, oder über eine serielle Schnittstelle RS 232 eingerichtet werden.

Ziel dieser Programmieranleitung ist es, den SPS Programmierer bei der Integration der cab Beschriftungslaser in automatisierte, verkettete Prozesse zu unterstützen.

Anhand von vier aus der Praxis gewählten Beispielen wird aufgezeigt, in welchen Schritten eine Integration erfolgen kann. Die Darstellung der Automationsprozesse wird mit Hilfe von Flowcharts transparent visualisiert und in die SPS Programmiersprache von Siemens umgesetzt.

Alle Beispiele sind mit den entsprechenden Layouts und SPS Ablaufprogrammen hinterlegt.

Die in dieser Anleitung behandelten Remote API Befehle sind im SPS Projekt als entsprechende Funktionsbausteine hinterlegt und können in beliebiger Reihenfolge kombiniert und verkettet werden.

Die Beispiele beinhalten die digitalen Signale der Schnittstellen CON2 und CON3 zur Synchronisierung der Prozesse und zeigen damit auf, wie Beschriftungen gesteuert und überwacht werden können.



Hinweis!

Falls nicht vorhanden, können alle zu dieser Anleitung zugehörigen Beispielprogramme und Dateien [hier](#) heruntergeladen werden.



Hinweis!

Die Remote API Schnittstelle steht sowohl als TCP/IP Socketverbindung, als auch über serielle Schnittstelle RS232 zur Verfügung.

3.2 Funktionalität der Remote API Schnittstelle

Die LEC Remote API Schnittstelle verwendet ein nachrichtenbasierendes Kommunikationsprotokoll zwischen dem „Client“, einer Steuerung auf der Prozessleitebene und dem „Host“, der Lasersteuerung. Das Protokoll ist bidirektional.

Auf jeden Befehl erhält der „Client“ eine vom „Host“ generierte Antwort, welche ihn über den Status des Befehls informiert. Das Warten auf diese Antwort ist zwingend erforderlich, um eine fehlerfreie Synchronisation in der Kommunikation zu erhalten. Die Antworten geben Aufschluss darüber, ob der Befehl erfolgreich oder mit einem Fehler abgearbeitet wurde.

Alle Befehle müssen mit der Anweisung „Carriage Return“ und „Line Feed“ abgeschlossen werden. Antworten werden automatisch vom Host mit „Carriage Return“ und „Line Feed“ abgeschlossen.

Die LEC Remote API Schnittstelle bietet eine Kommunikationsplattform zur Steuerung und Überwachung von Beschriftungsaufträgen und der Lasersteuerung.

Neben der Abfrage und der Verwaltung von Systemparametern können Beschriftungsaufträge lokal von der Steuerung oder aus dem Netzwerk geladen werden. Die einzelnen Beschriftungsobjekte können von extern editiert oder modifiziert werden.



Hinweis!

Das Senden von Befehlen ohne Abwarten der entsprechenden Antwort vom Host führt zu Problemen in der Kommunikation und kann zu Störungen führen.



Hinweis!

Alle Befehle müssen mit der Anweisung „Carriage Return“ und „Line Feed“ abgeschlossen werden (0X0D & 0X0A oder <CR>&<LF>). Alle Antworten werden mit der Anweisung „Carriage Return“ und „Line Feed“ abgeschlossen (0X0D & 0X0A oder <CR>&<LF>).

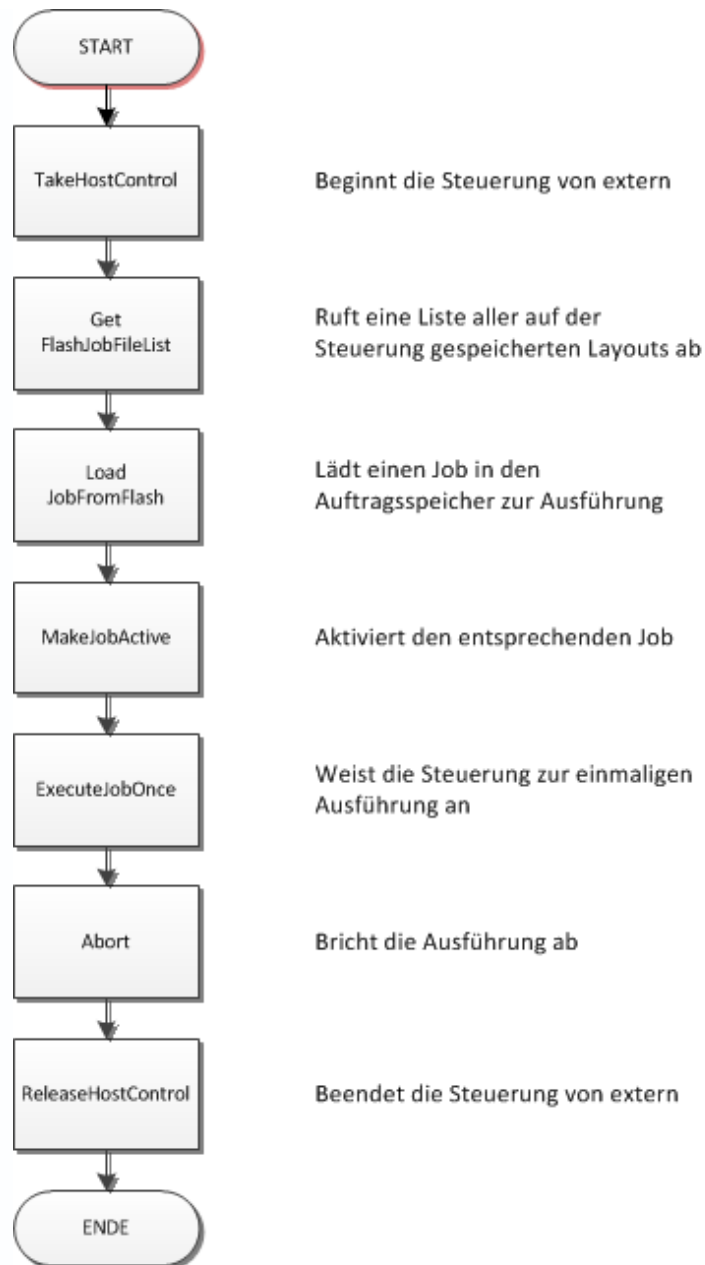


Hinweis!

Für die Übertragung von Befehlen ist der 7-Bit Zeichensatz ASCII 0-127 oder der 8-Bit Zeichensatz ASCII0-255 zu verwenden.

3.3 API Kommunikationsübersicht

Die untenstehende Grafik zeigt eine beispielhafte Kommunikation zwischen Client und Host für die Markierung eines auf der Steuerung gespeicherten Layouts:



Über Get- und Set- Befehle ist es möglich, den Status und die Eigenschaften des Hosts zu überprüfen und einzustellen. Um Eigenschaften zu verändern, muss der Client über TakeHostControl mit dem Host verbunden sein.

**Hinweis!**

Starten Sie die Verbindung zum Host durch Verwenden des Befehls TakeHostControl.

**Hinweis!**

Beenden Sie die Verbindung zum Host durch Verwenden des Befehls ReleaseHostControl.

3.4 Verbindung über die TCP/IP Schnittstelle

TCP/IP ist ein Netzwerkprotokoll, welches erlaubt, über lokale Netzwerke (LAN – Local Area Network) oder das Fernnetz/ Internet (WAN – Wide Area Network) zu kommunizieren. Damit unterschiedliche Programme und Computer mit der Lasersteuerung und dem dort installierten Remoteserver in Verbindung treten können, muss dieser unter einer spezifischen und bekannten IP- und Portadresse erreichbar sein.

Über die Port- und IP Adresse wird sichergestellt, dass die Befehle vom Sender den richtigen Empfänger erreichen.

Die Lasersteuerung verwendet zur Kommunikation über die Remote API Schnittstelle den Port 12500. Die Steuerung kann mit einer statischen oder einer dynamisch über DHCP wechselnden IP-Adresse konfiguriert werden. Bei der Steuerung über Remote API empfehlen wir die Verwendung einer statischen IP-Adresse. Durch diese Vorgehensweise ist die Verbindungseinstellung zwischen Client und Host statisch hinterlegt und kann bspw. beim Ausschalten der Geräte erneut ohne Probleme aufgebaut werden.

Die Nutzung der Remote API Schnittstelle kann von mehreren Computern oder Systemen im Netzwerk verwendet werden. Es kann nur ein Computer oder System gleichzeitig die Steuerung des Lasers übernehmen.

Clients können in beliebigen Netzwerken unter den Betriebssystemen MS Windows oder Linux installiert werden.

**Hinweis!**

Die im Auslieferungszustand eingestellte IP-Adresse entspricht 192.168.1.11.

**Hinweis!**

Die Lasersteuerung LEC nutzt zur Übertragung der Befehle den Port 12500.

**Hinweis!**

Bei Verwendung der Remote API ist es vorteilhaft, eine statische IP-Adresse für die Lasersteuerung zu verwenden.

**Hinweis!**

Nur ein Computer oder System kann gleichzeitig die Steuerung des Lasers übernehmen.

**Hinweis!**

Einstellungen der IP-Adresse oder sonstigen Verbindungseigenschaften sind erst beim nächsten Neustart aktiv.

3.5 Hinweise zu cab SPS Demoprogrammen

3.5.1 Hard- und Software

Die hier aufgeführten Programmbeispiele wurden in einem Projekt mit folgender Hard- und Softwareplattform erstellt:

Hersteller: Siemens

Hardware: CPU 1511-1PN

Programmiersoftware: Tia Portal V13 SPS1 Update 6

Das Projekt wurde unter dem Dateinamen „FL+ Remote API Library_YYYYMMDD“ gespeichert, wobei die Indexierung YYYYMMDD Aufschluss über das Revisionsdatum geben und zur Überprüfung der Aktualität dienen.

Da das Projekt insgesamt vier Programmbeispiele beinhaltet und diese selektiv zur Auswahl bereit stehen sollen, ist jeweils im Netzwerk 4 die Eingabebedingung „Start Sample“ vorhanden. Diese muss aktiviert werden, um Zugriff auf den Laser zu ermöglichen.



Hinweis!

Nur ein Computer oder System kann gleichzeitig die Steuerung des Lasers übernehmen. Bei Verwendung von Siemens CPUs, welche anderen Baureihen angehören, kann die Konvertierung des Programms über das TIA Portal erfolgen.

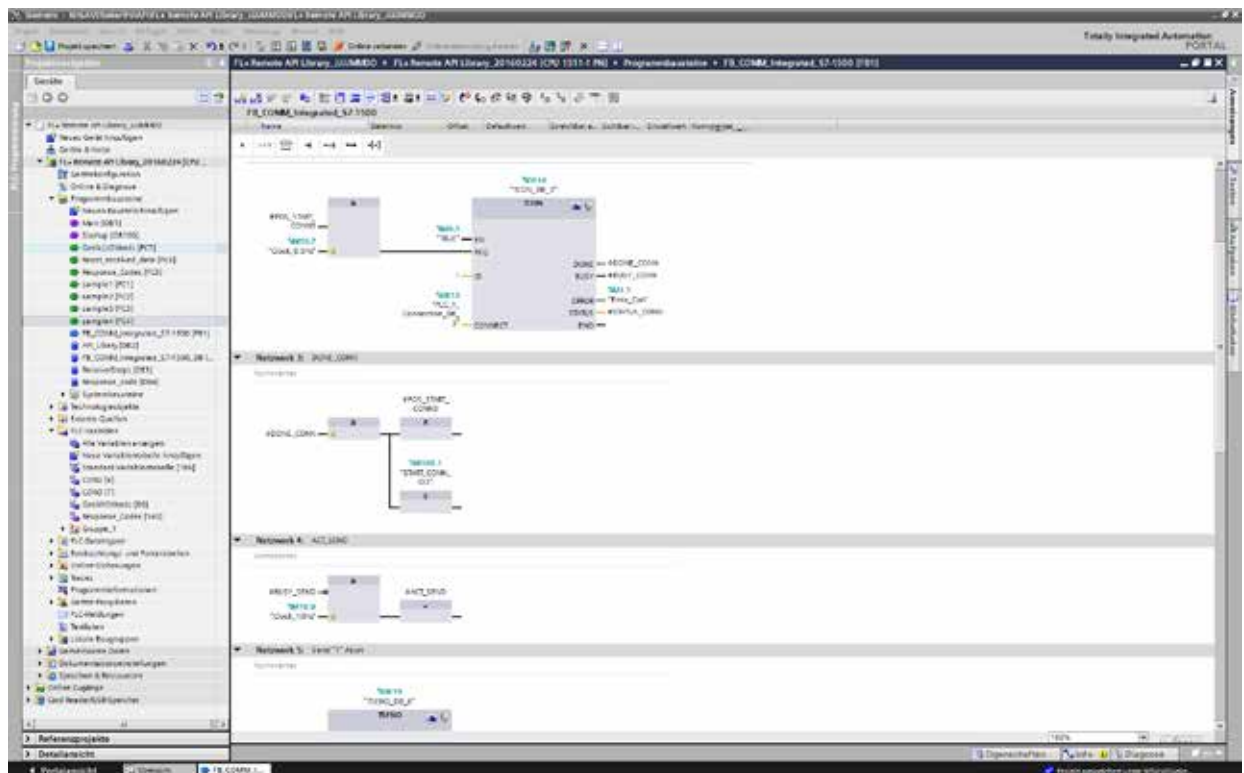


Hinweis!

Wurde im Programmbeispiel die Eingabebedingung „Start Sample“ aktiviert, so ist zeitgleich von der Laser-Software cabLase kein Zugriff auf den Laser möglich.

3.5.2 Informationen zu Programmbeispielen

Das untenstehende Bild zeigt die Ordnerstruktur im Projektordner:



Der Operationsbaustein Main OB1 definiert den grundsätzlichen Programmablauf. Mit Startup OB100 wird das Verhalten des SPS Programms beim Aus- und Einschalten der CPU festgelegt.

Die in der Programmieranleitung behandelten Beispiele 1 bis 4 befinden sich in den folgenden Funktionsbausteinen:

Sample1[FC1]

Sample2[FC2]

Sample3[FC3]

Sample4[FC4]

Neben der eigentlichen Remote API Befehlsstruktur enthalten die Programmbeispiele auch Details zum Einschalten und Überwachen des Lasers auf der Grundlage der digitalen Schnittstellen des Lasers CON2, CON3 und CON4.

Das **Netzwerk 1** im Beispiel behandelt das Einschalten des Lasers. Der Laser kann dann eingeschaltet werden, wenn der Sicherheitskreis CON4 geschlossen, auf der Schnittstelle CON3 das „Power“ Signal anliegt und keine Fehler auf den digitalen Schnittstellen CON2 und CON3 ausgegeben werden.

Das **Netzwerk 5** im Beispiel behandelt die Auswertung des Lasersicherheitskreises auf der Schnittstelle CON4. Ein Öffnen des Lasershutters und damit Austritt von Laserstrahlung ist nur dann möglich, wenn die Sicherheitskreise geschlossen sind.

In jedem Beispiel befindet sich im **Netzwerk 4** die Eingabebedingung „Start Sample“. Diese muss aktiviert werden, um den Zugriff auf den Laser zu ermöglichen und das Beispiel anzuwenden. Wird diese zurückgenommen, so ist es bspw. möglich, an Stelle der SPS mit dem Laser über die Software cabLase eine Verbindung aufzubauen.

Response_Codes [FC5] Ist verantwortlich für die Auswertung der von der vom Laser empfangenen API Response Codes.

Im Funktionsbaustein **FB1** ist der Kommunikationsbaustein TCON zum Aufbau der Verbindung zwischen SPS und Laser festgelegt. Darüber hinaus wird hier definiert, wie Daten zum Laser geschickt und vom Laser empfangen werden.

Im Datenbaustein **DB[2]** befindet sich eine Liste von Funktionsbausteinen aller in dieser Programmieranleitung behandelten Remote API Befehle. Für eine kundenorientierte Programmierung können diese für eine beliebige Verkettung kopiert werden.

Die von der SPS empfangenen Daten werden in den Datenbaustein **DB[3]** in „Receive_Data“ geschrieben. Soll der Empfangsbereich des Datenbausteins **DB[3]** gelöscht werden, muss der Merker 5.4 „Reset_Receive_Data“ temporär auf „high“ gesetzt werden.

Im **DB[3]** ist unter Feld 8 „Response Code String“ ein Feld vorbereitet, welches potentielle Fehlermeldungen als Beschreibungstext ausgibt. Diese Ausgabe kann bspw. auch zur Visualisierung in einer Bedienoberfläche verwendet werden.

Die Wandlung der numerischen API Response Rückmeldungen erfolgt im **DB[4]**. In dieser Bibliothek werden den Zahlen-codes Klartextmeldungen zugeordnet, welche bei Bedarf, bspw. der Wahl einer Fremdsprache, angepasst werden können.

Im Baustein **PLCVariablen/CON2** ist ein Register enthalten, welches die digitalen Ein- und Ausgänge der Schnittstelle CON2 dem Verdrahtungsplan zuordnet.

Im Baustein **PLCVariablen/CON3** ist ein Register enthalten, welches die digitalen Ein- und Ausgänge der Schnittstelle CON3 dem Verdrahtungsplan zuordnet.



Hinweis!

In Beispiel 3 ist die Synchronisation und Überwachung des Laserprozesses nicht über die Verdrahtung der Schnittstelle CON2 realisiert, sondern über die Remote API. Die Hardwareverdrahtung kann entfallen. Das Einschalten und Überwachen des Lasers wird über CON3 realisiert.



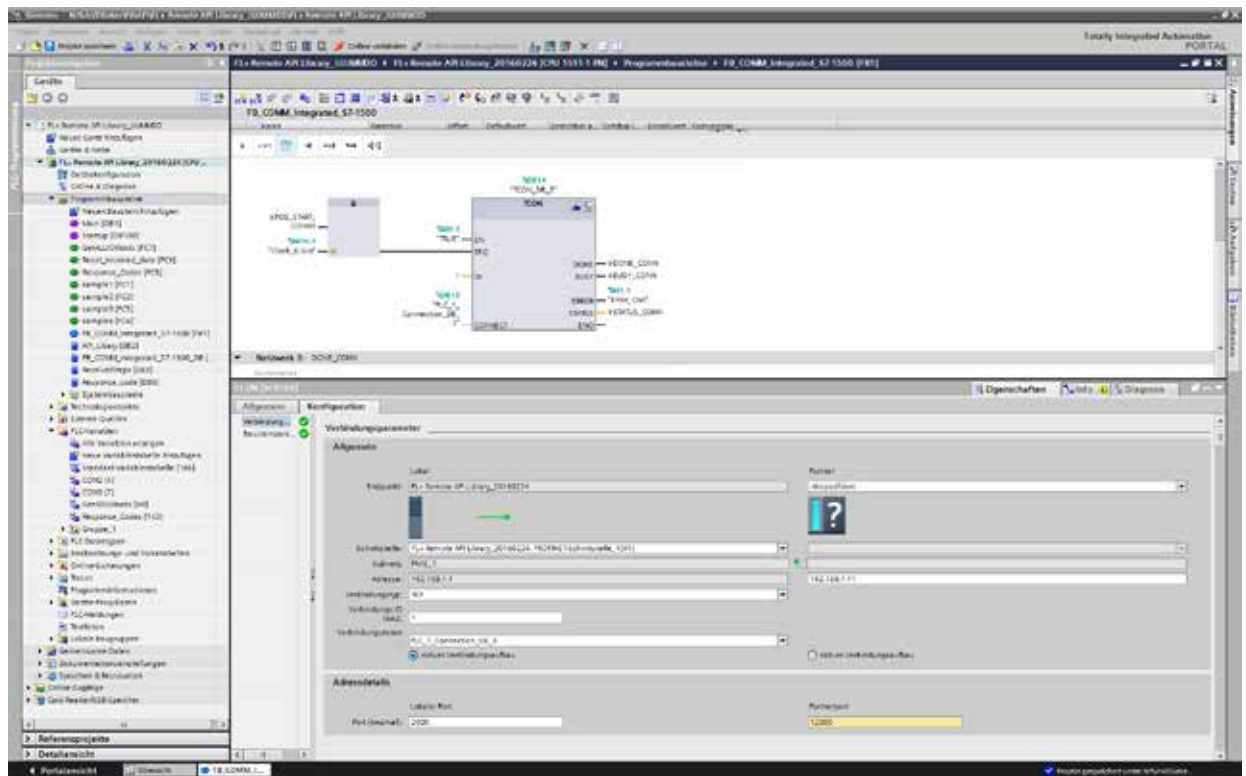
Hinweis!

Wird für den Laser eine abweichende IP Adresse vergeben, muss diese im Baustein TCON angepasst werden, um eine Verbindung einzurichten.

Wird der von cab zur Verfügung gestellte Kommunikationsbaustein TCON nicht verwendet, so ist zu beachten, dass die empfangenen Daten den Datenbaustein DB[3] richtig referenzieren.

3.5.3 Einrichten der Netzwerkverbindung

Im Funktionsbaustein **FB1** ist der Kommunikationsbaustein TCON zum Aufbau der Verbindung zwischen SPS und Laser festgelegt. Darüber hinaus wird hier definiert, wie Daten zum Laser geschickt und vom Laser empfangen werden.



Im Funktionsbaustein werden folgende Angaben gemacht:

- Beschreibung der lokalen und Partnersteuerung
- Lokale und Partner IP-Adresse
- Definition des Speicherbereichs, in welchem Verbindungsdaten abgelegt werden
- Lokale Portadresse
- Portadresse des Partners



Hinweis!

Die im Auslieferungszustand eingestellte IP-Adresse entspricht 192.168.1.11.

4.1 Beispiel 1

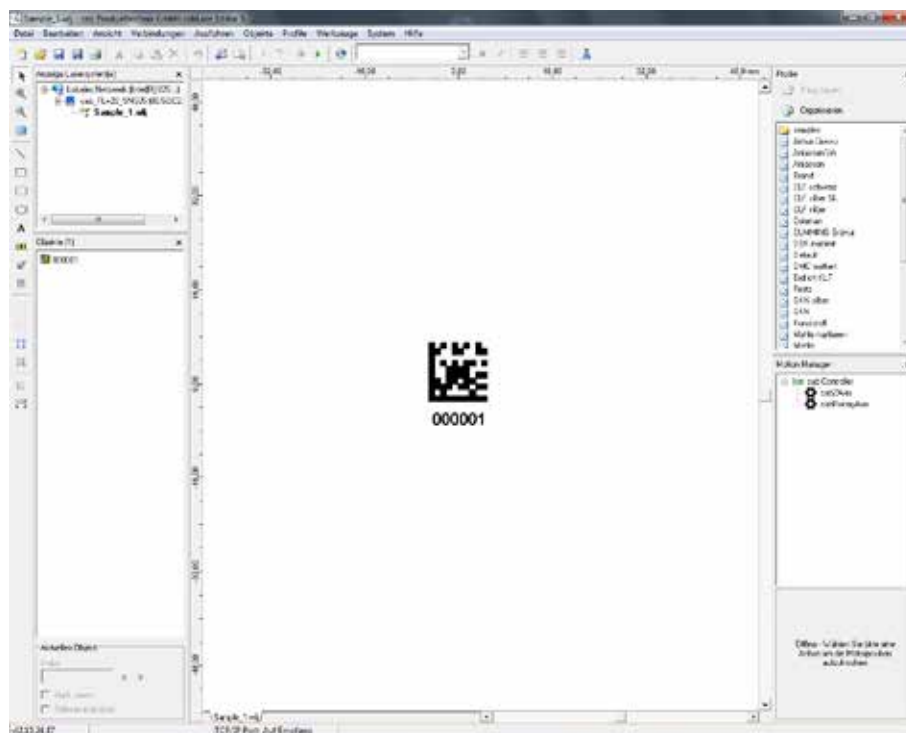
4.1.1 Beschreibung

Bei der Beschriftungsaufgabe handelt es sich um die Kennzeichnung mit einem DatamatrixCode. Beschriftet wird eine fortlaufende 6-stellige Seriennummer. Bei Produktionsbeginn soll die Startnummer übergeben und nachfolgend jeweils um 1 inkrementiert werden. Die Inkrementierung übernimmt die SPS.

Das Layout befindet sich im Flashspeicher der Lasersteuerung unter dem Dateinamen Sample1.dat. Beim Starten des Programms wird das Layout automatisch geladen, editiert, aktiviert und mit den digitalen Eingängen auf der Schnittstelle CON2 synchronisiert.

Einmalig bei Beginn des Prozesses wird der Laser automatisch über die Schnittstelle CON3 eingeschaltet und später mit den Schnittstellen CON3 und CON2 überwacht.

4.1.2 Layout



Hinweis!

Bei korrektem Programmablauf zählt die Seriennummer bei der Beschriftung hoch.

4.1.3 SPS Programmbeispiel

Das Beispielprogramm befindet sich im Projekt FL+ Remote API Library_YYYYMMDD in der Funktion Sample1[FC1].



Hinweis!

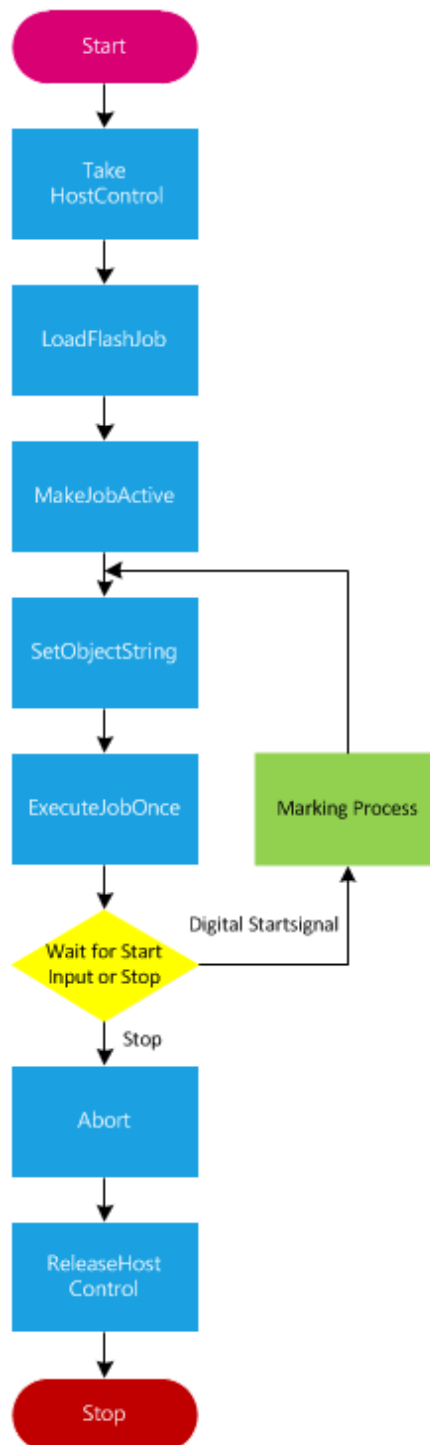
Beachten Sie, dass sich in Netzwerk4 die Eingabebedingung „Start Sample“ befindet. Diese muss aktiviert werden, um den Zugriff auf den Laser freizugeben und das Programmbeispiel anzuwenden.



Hinweis!

Nach Freigabe von „Start Sample“ ist ein Zugriff auf den Laser durch die Software cabLase 5 nicht möglich.

4.1.4 Flowchart



Hinweis!
Blaue Elemente entsprechen den Remote API Befehlen.

4.2 Beispiel 2

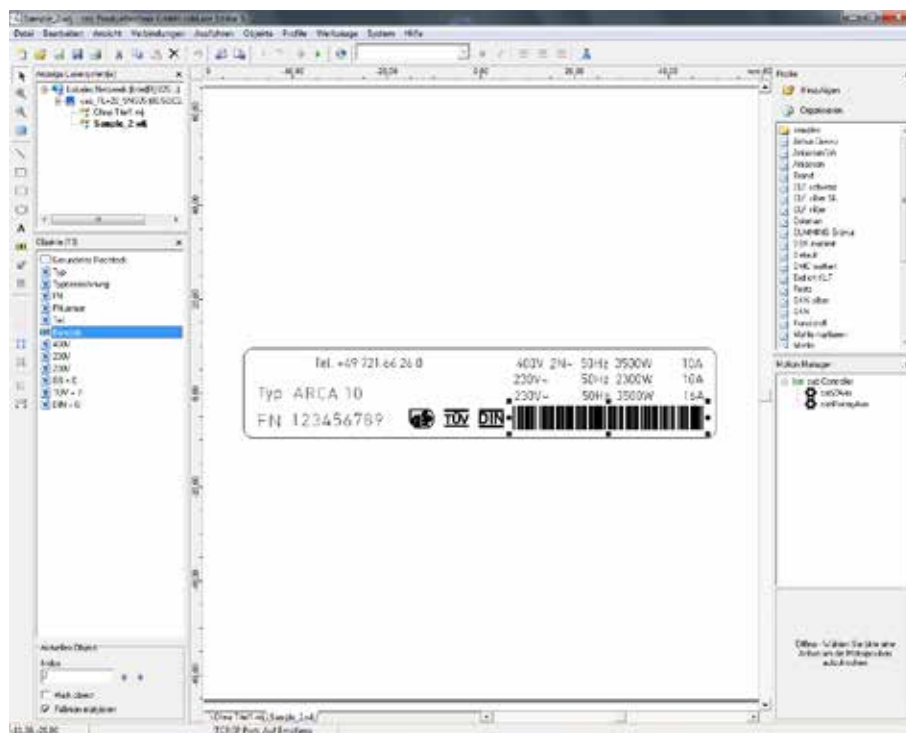
4.2.1 Beschreibung

Bei der Beschriftungsaufgabe handelt es sich um die Kennzeichnung eines Typenschilds mit Leistungsdaten und den Prüfzertifikaten GS, TÜV und DIN. Abhängig von der Produktionssteuerung sollen die unterschiedlichen Prüflogos ein- und ausgeschaltet werden.

Das Layout befindet sich im Flashspeicher der Lasersteuerung unter Sample2.dat. Beim Starten des Programms wird das Layout automatisch geladen, editiert, aktiviert und mit den digitalen Eingängen auf der Schnittstelle CON2 synchronisiert. Am Ende des Prozesses werden die Beschriftungsaufträge aus dem Auftragspeicher gelöscht.

Einmalig bei Beginn des Prozesses wird der Laser automatisch über die Schnittstelle CON3 eingeschaltet und später mit den Schnittstellen CON3 und CON2 überwacht.

4.2.2 Layout



Hinweis!

Bei korrektem Programmablauf werden die Prüfsymbole selektiv zu- und abgeschaltet.

4.2.3 SPS Programmbeispiel

Das Beispielpogramm befindet sich im Projekt FL+ Remote API Library_YYYYMMDD in der Funktion Sample2[FC2].



Hinweis!

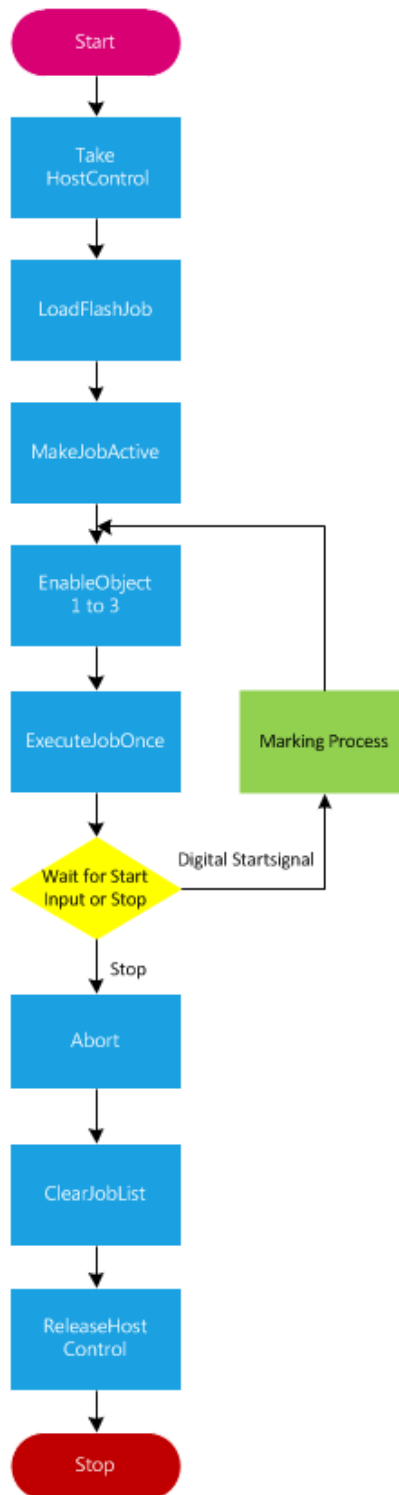
Beachten Sie, dass sich in Netzwerk4 die Eingabebedingung „Start Sample“ befindet. Diese muss aktiviert werden, um den Zugriff auf den Laser freizugeben und das Programmbeispiel anzuwenden.



Hinweis!

Nach Freigabe von „Start Sample“ ist ein Zugriff auf den Laser durch die Software cabLase 5 nicht möglich.

4.2.4 Flowchart



Hinweis!
Blaue Elemente entsprechen den Remote API Befehlen.

4.3 Beispiel 3

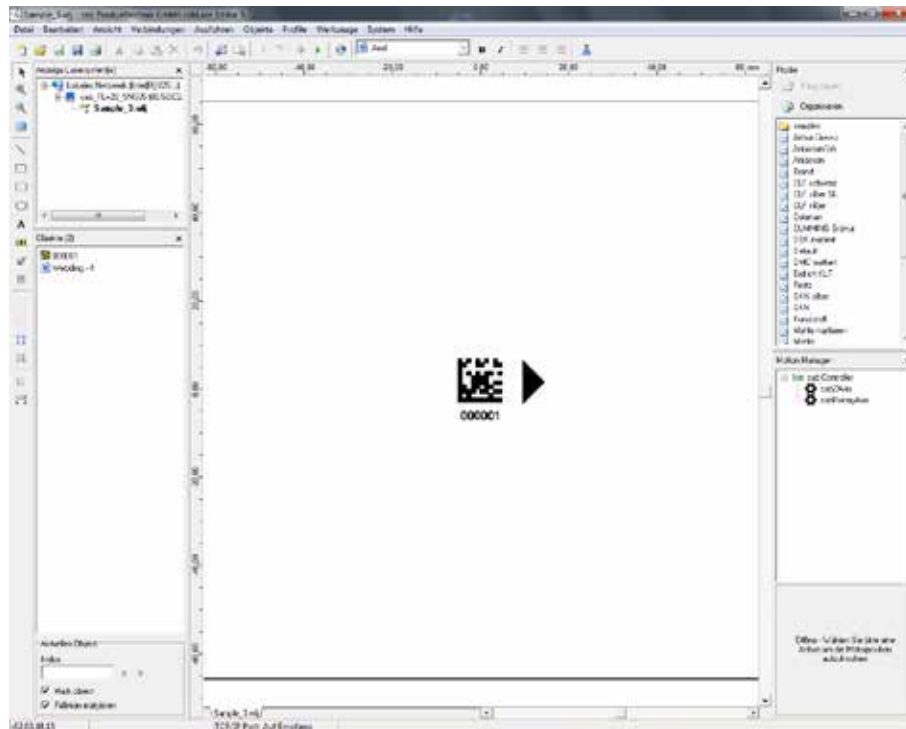
4.3.1 Beschreibung

Bei der Beschriftungsaufgabe handelt es sich um die Kennzeichnung eines Bauteils, welches in unterschiedlichen Varianten bezüglich der Positionierung der Kennzeichnung auf dem Bauteil vorliegt. Abhängig von der Modellvariante soll die Kennzeichnung des Pfeils auf unterschiedliche Koordinaten in X- und Y-Richtung geschoben und gedreht werden.

Das Layout befindet sich im Flashspeicher der Lasersteuerung unter Sample3.dat. Es wird beim Programmstart automatisch geladen, editiert, aktiviert und über die Remote API über den Befehl GetAllIOWord synchronisiert.

Einmalig bei Beginn des Prozesses wird der Laser automatisch über die Schnittstelle CON3 eingeschaltet und später mit den Schnittstellen CON3 und CON2 überwacht.

4.3.2 Layout



Hinweis!

Bei korrektem Programmablauf wandert der Pfeil im Gegenuhrzeigersinn um den Code.

4.3.3 SPS PRogrammbeispiel

Das Beispielprogramm befindet sich im Projekt FL+ Remote API Library_YYYYMMDD in der Funktion Sample3[FC3].



Hinweis!

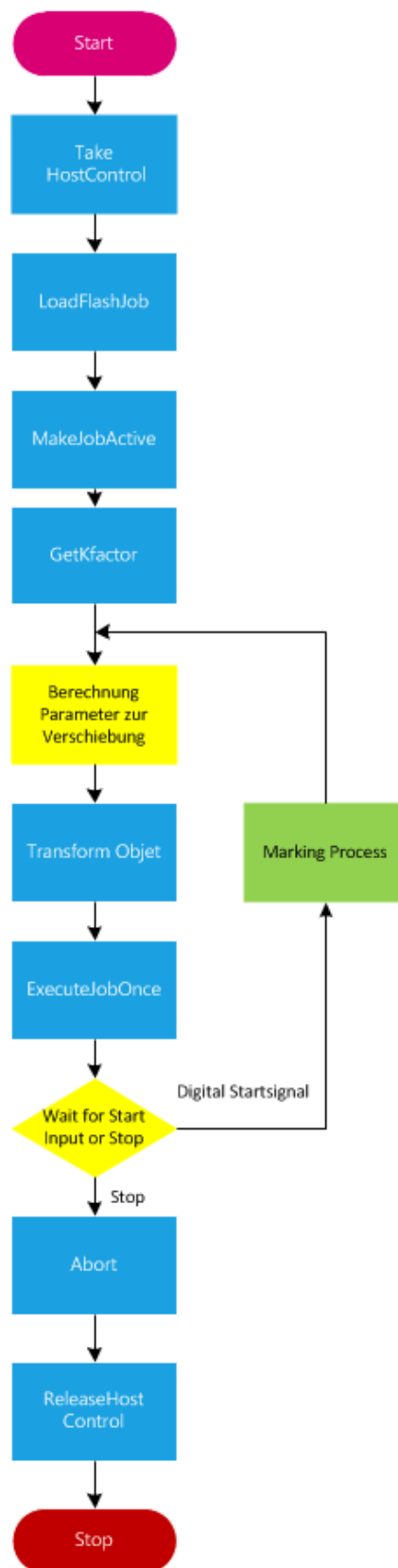
Beachten Sie, dass sich in Netzwerk4 die Eingabebedingung „Start Sample“ befindet. Diese muss aktiviert werden, um den Zugriff auf den Laser freizugeben und das Programmbeispiel anzuwenden.



Hinweis!

Nach Freigabe von „Start Sample“ ist ein Zugriff auf den Laser durch die Software cabLase 5 nicht möglich.

4.3.4 Flowchart



Hinweis!
Blaue Elemente entsprechen den Remote API Befehlen.

4.4 Beispiel 4

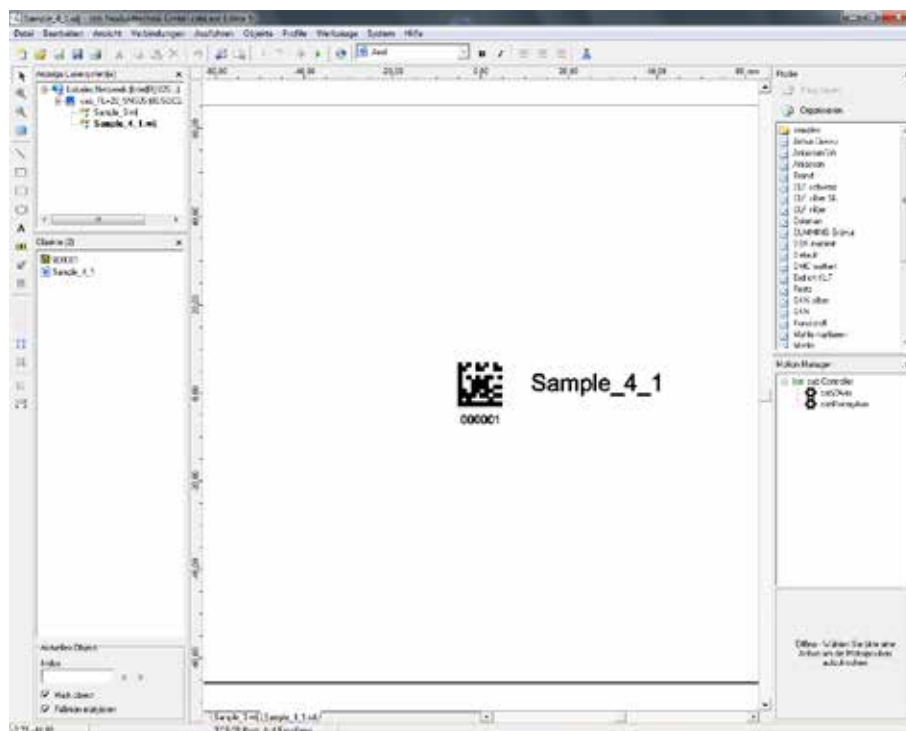
4.4.1 Beschreibung

Bei der Beschriftungsaufgabe handelt es sich um die Kennzeichnung von unterschiedlichen Bauteilen in einer Produktionslinie. Abhängig von der Bauteilvariante sollen die unterschiedlichen Beschriftungslayouts aus dem Firmennetzwerk geladen werden.

Die unterschiedlichen Layouts befinden sich im Netzwerk \\Datenserver\Layouts unter Sample4_1.dat, Sample4_2.dat und Sample4_3.dat. Abhängig von der Auswahl in der Produktion werden die Layouts beim Programmstart automatisch geladen, aktiviert und mit den digitalen Eingängen auf der Schnittstelle CON2 synchronisiert.

Einmalig bei Beginn des Prozesses wird der Laser automatisch über die Schnittstelle CON3 eingeschaltet und später mit den Schnittstellen CON3 und CON2 überwacht.

4.4.2 Layout



Hinweis!

Bei korrektem Programmablauf wechselt der Text neben dem Datamatrixcode von Sample_4_1 auf Sample_4_2 und Sample_4_3.

4.4.3 SPS Programmbeispiel

Das Beispielpogramm befindet sich im Projekt FL+ Remote API Library_YYYYMMDD in der Funktion Sample4[FC4].



Hinweis!

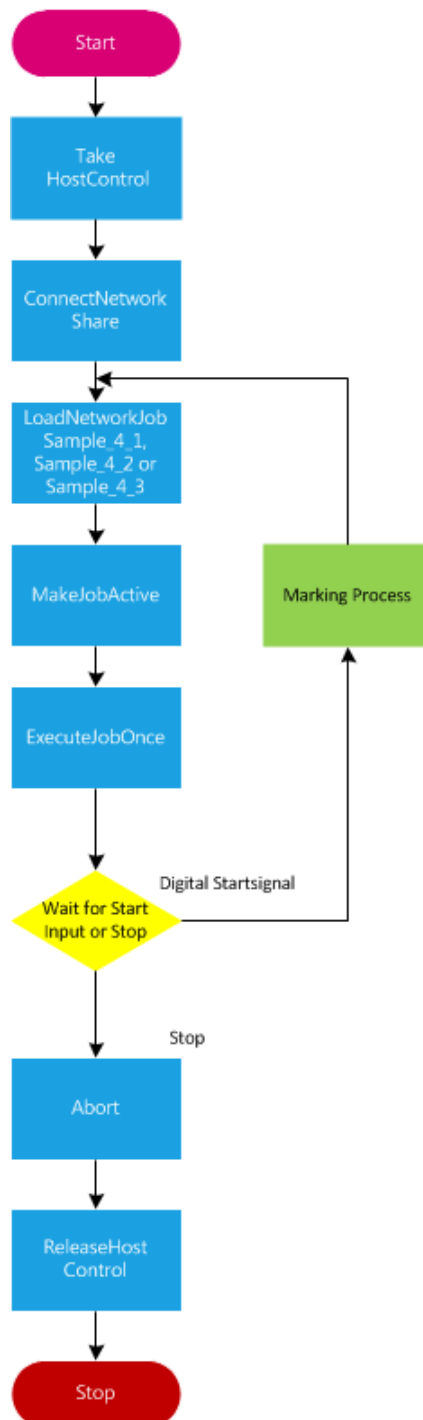
Beachten Sie, dass sich in Netzwerk4 die Eingabebedingung „Start Sample“ befindet. Diese muss aktiviert werden, um den Zugriff auf den Laser freizugeben und das Programmbeispiel anzuwenden.



Hinweis!

Nach Freigabe von „Start Sample“ ist ein Zugriff auf den Laser durch die Software cabLase 5 nicht möglich.

4.4.4 Flowchart



Hinweis!
Blaue Elemente entsprechen den Remote API Befehlen.

5.1 Auswahl des Netzwerkadapters

cabLase Editor 5 unterstützt mehrfache Netzwerkadapter im PC.

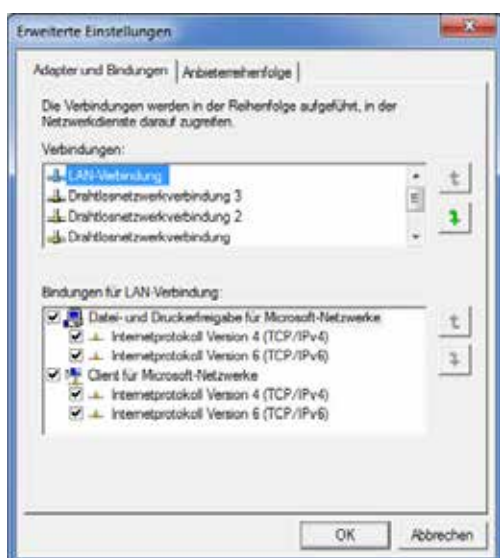


Hinweis!

Sind in einem PC mehrere Netzwerkadapter vorhanden, so ist die Verbindungsreihenfolge in den Windows-Einstellungen des Netzwerkadapters so zu wählen, dass die Verbindung zum Beschriftungslaser FL+ als Erstes durchgeführt wird! ▷ Windows-Dokumentation

Beispiel Windows 7

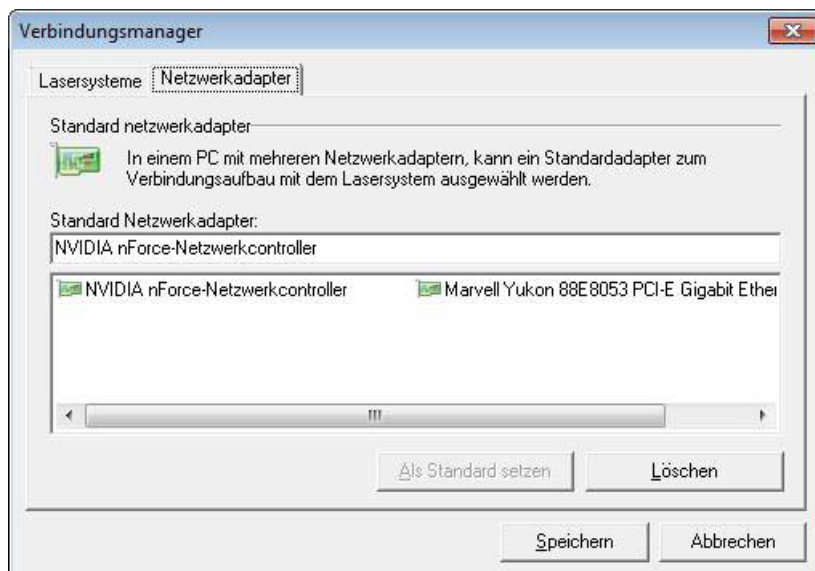
- ▶ "Systemsteuerung > Netzwerk- und Freigabecenter > Adaptereinstellungen ändern" wählen.
- ▶ Menü "Erweitert > Erweiterte Einstellungen" wählen.
- ▶ Im Fenster "Verbindungen" die für den Anschluss des FL+ verwendete Verbindung wählen.
- ▶ Gewählte Verbindung mit den Pfeiltasten neben dem Fenster in die oberste Zeile verschieben.
- ▶ "OK" klicken.



Einstellung der Priorität des Netzwerkadapters

Der mit dem Gerät verbundene Netzwerkanschluss muss in cabLase Editor 5 eingestellt werden.

- In der Hauptmenüleiste "Verbindungen > Einrichten" auswählen.
- Karteikarte "Netzwerkadapter" auswählen.



Auswahl des Netzwerkadapters

- Netzwerkadapter, an dem der FL+ angeschlossen ist, auswählen.
- Auf "Als Standard setzen" klicken.



Hinweis!

Ist die Subnetz-Einstellung inkompatibel, so weist der Beschriftungslaser FL+ dem Netzwerkadapter des PC eine temporäre IP-Adresse zu. Diese wird nach Neustart des PC gelöscht.

5.2 Manuelles Verbinden des Lasers mit cabLase Editor 5



Hinweis!

Vor Verbinden von cabLase Editor 5 mit dem Beschriftungslaser FL+ muss eine Netzwerkverbindung hergestellt sein.



Hinweis!

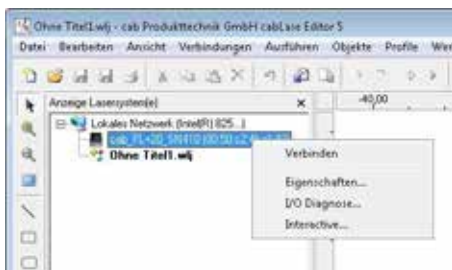
Es muss sichergestellt sein, dass der Beschriftungslaser FL+ am Netzschalter eingeschaltet wurde!

Nach dem Starten der Software wird im Feld "Anzeige Lasersysteme" die Meldung "Suche Lasersysteme" angezeigt



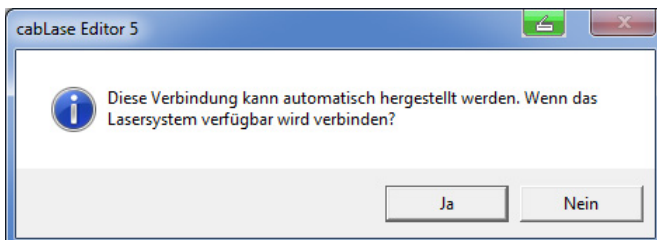
Suche nach Lasersystemen

- Im Bereich "Anzeige Lasersystem(e)" mit der rechten Maustaste auf den im Netzwerk erkannten Beschriftungslaser FL+ klicken und "Verbinden" wählen.



Verbinden des Lasersystems

Es folgt eine Abfrage zur Einrichtung einer automatischen Verbindung :



Abfrage automatische Verbindung

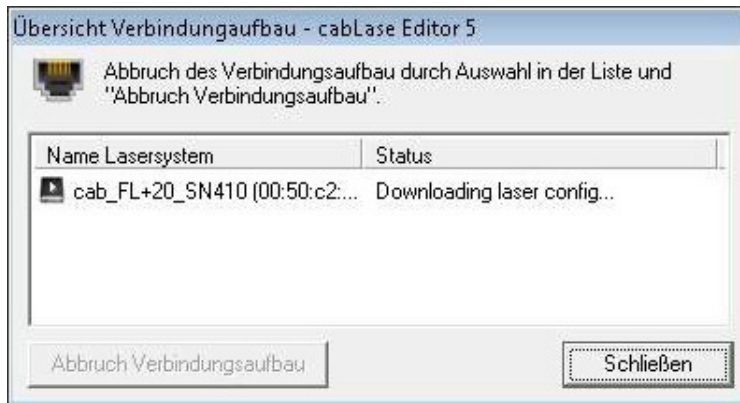
- Gewünschte Methode zum Verbindungsaufbau wählen.



Hinweis!

Die Einrichtung des automatischen Verbindungsaufbaus ist auch später möglich.

Anschließend werden die Konfigurationseinstellungen des Beschriftungslasers FL+ auf den lokalen PC übermittelt. Die Verbindungsanzeige gibt Auskunft über den Status.



Übersicht Verbindungsaufbau

Zur Anzeige unterschiedlicher Verbindungs-Zustände werden folgende Symbole verwendet:

Symbol	Status
	Im Netzwerk verfügbar
	Im Netzwerk nicht verfügbar
	Verbunden mit lokaler Installation cabLase Editor 5
	Fehlerhafte Firmware oder Lizenz
	PC Software inkompatibel
	Laser bereits mit anderem Netzwerkteilnehmer verbunden oder in Stand-alone Modus

Tabelle 1 Verbindungs-Zustände

Durch Bewegen des Mauszeigers über ein erkanntes Lasersystem wird dessen Verbindungsstatus angezeigt:



Anzeige des Verbindungsstatus'

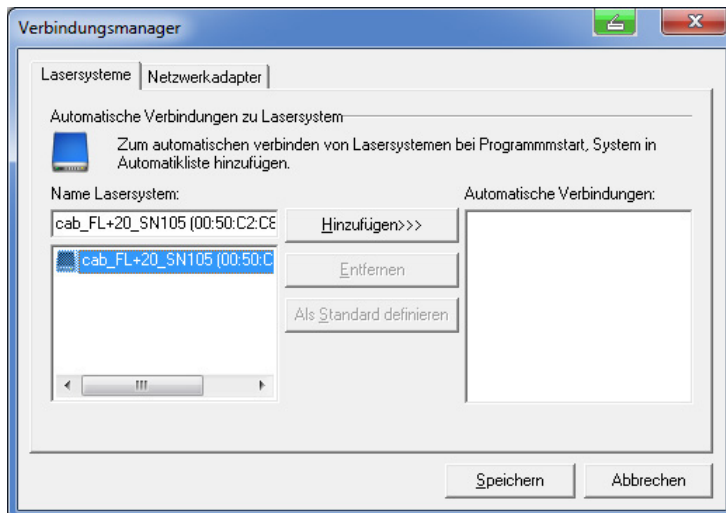


Hinweis!

Zum Beenden einer aktiven Verbindung auf "Trennen" klicken.

5.3 Automatisches Verbinden des Lasers mit cabLase Editor 5

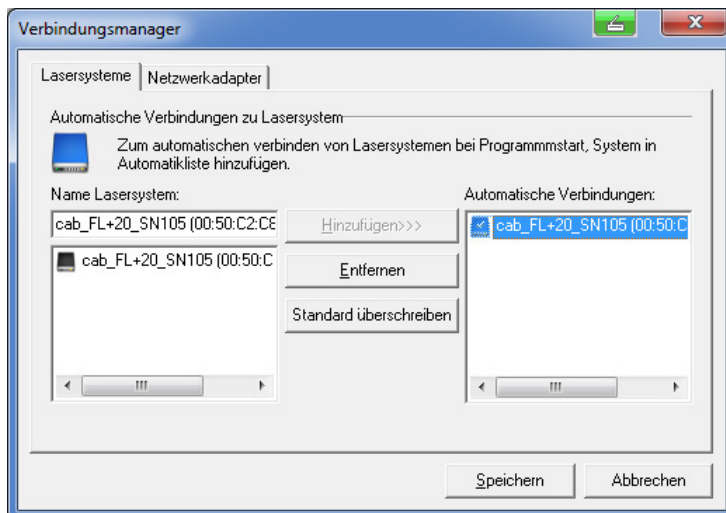
- Menü "Verbindungen > Einrichten" wählen. Es wird folgendes Ein-/Ausgabefenster erreicht :



Automatische Verbindung einrichten

Im linken Fenster werden alle im Netzwerk befindlichen Beschriftungssysteme FL+ mit ihren jeweiligen MAC-Adressen angezeigt.

- Auf den für die Verbindung gewünschten Beschriftungslaser FL+ klicken:
Der gewählte Beschriftungslaser FL+ wird blau hinterlegt.
- Auf "Hinzufügen" klicken.
Der Laser wird der Liste "Automatische Verbindungen" hinzugefügt.



Standardsystem festlegen

- Gewünschten Laser in der Liste "Automatische Verbindungen" markieren und Auswahl durch Klicken auf "Als Standard definieren" bestätigen.
Wurde die Auswahl durchgeführt, wird dies mit einem Häkchen im PC-Symbol bestätigt.
- Gewählte Einstellungen durch Klicken auf "Speichern" übernehmen.



Hinweis!

Beim Starten der Software verbindet sich cabLase Editor 5 jetzt immer automatisch mit dem ausgewählten Beschriftungslaser FL+.

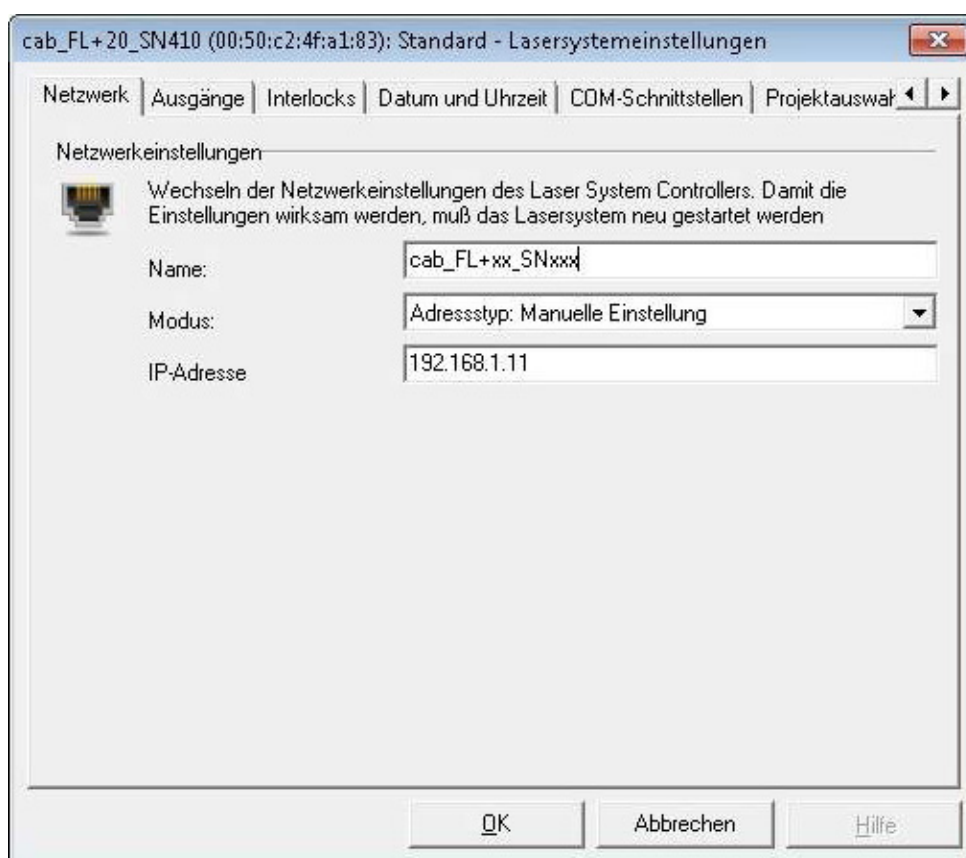
5.4 Ändern der IP-Adresse



Achtung!

Falsche Einstellungen der IP-Adresse können zum Verlust der Verbindung führen, die nur durch Änderung der Einstellungen am lokalen PC behoben werden können.

- ▶ In der Lasersystem Anzeige mit der rechten Maustaste erkannttes Lasersystem auswählen und auf "Voreinstellungen" klicken.
- ▶ "Netzwerk" auswählen.



Einstellen der IP-Adresse

- ▶ Unter Modus "Adresstyp: Manuelle Einstellung" wählen und die gewünschte IP-Adresse eingeben.

6.1 Allgemein

Der Beschriftungslaser FL+ unterstützt die Möglichkeit, Beschriftungsaufträge ohne Datenübertragung vom PC auszuführen (Stand-Alone-Betrieb). In dieser Betriebsart kann eine übergeordnete Steuerung (z.B. eine SPS) die vollständige Kontrolle über den Beschriftungslaser übernehmen. Hierfür stehen die "Remote Command API" (Application Programming Interface) und der "COM Automation Server" (⤵ Lieferumfang Software) zur Verfügung. Ebenso kann der Aufruf gespeicherter Dateien über das "Job Select"-Signal erfolgen.

Für Stand-Alone-Betrieb müssen Job-Dateien und Laser-Zeichenfonts in einem Binär-Format im Speicher des FL+-Controllers abgelegt werden und über ein Unterprogramm oder den System-Explorer geladen werden.



Hinweis!

Das Editieren und Modifizieren ist nur bei *.wlj-Dateien möglich.

Beim Übertragen in den Speicher des Beschriftungslasers FL+ werden die Daten durch cabLase Editor 5 automatisch in binäre *.dat-Job-Dateien konvertiert.

Zum Speichern vorhandener *.dat-Dateien vom Beschriftungslaser FL+ zu einem lokalen PC kann der Lasersystemexplorer verwendet werden.

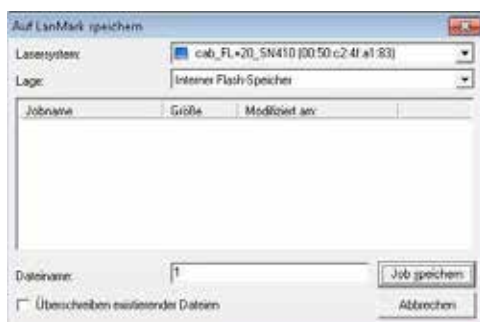
6.2 Speichern von *.wlj Dateien im Laserspeicher

Das Beispiel benötigt zwei Job-Dateien genannt 1.wlj und 2.wlj auf dem PC.



Auswahl des Ziels der Datenspeicherung

- ▶ Im Fenster "Anzeige Lasersysteme" mit rechter Maustaste auf einen geöffneten Job klicken.
- ▶ Mit "Zuordnen" aktives Lasersystem als Ziel der Datenspeicherung auswählen.
- ▶ Mit "Speichern nach Lasersystem" Job in den Laserspeicher übertragen.



Festlegung des Dateinamens

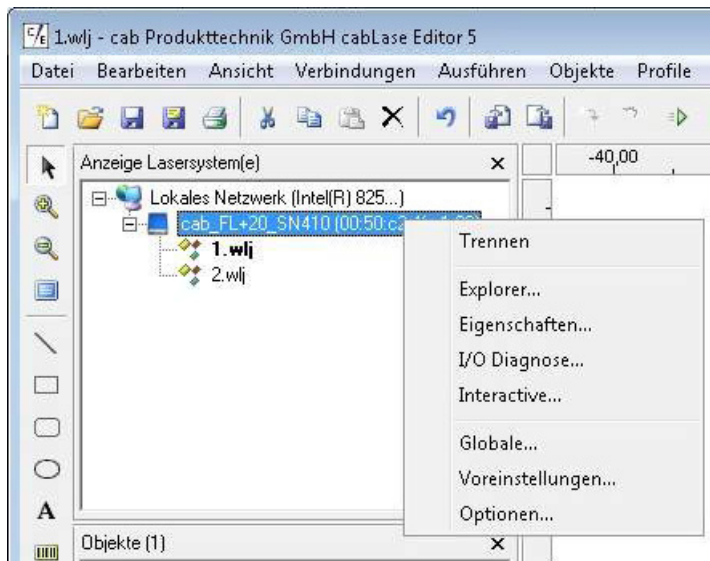
- ▶ Datei benennen und auf "Job speichern" klicken.
Die Beispiel-Dateien werden in ein Binärformat konvertiert und als 1.dat bzw. 2.dat im Laserspeicher abgelegt. Diese Dateien sind deutlich größer als die ursprünglichen *.wlj-Dateien.



Hinweis!

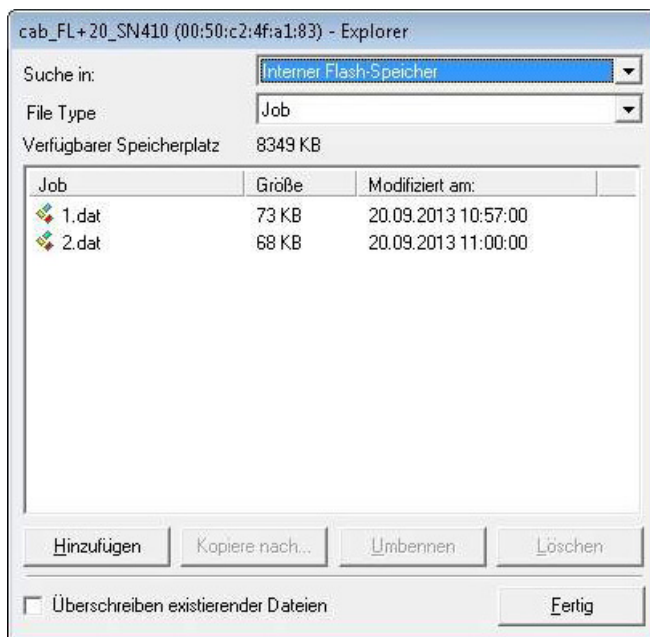
Um auf dem internen Speicher abgelegte Beschriftungslayouts über digitale Codierung aufzurufen, muss der Eingang "Job Select" auf der CON2 des Beschriftungslasers FL+ aktiviert werden!
Der Zugriff über "Remote Command API" und "COM Automation Server" ist immer möglich.

6.3 Verwalten von *.dat Dateien



Aufruf des System-Explorers

- Im Fenster "Anzeige Lasersysteme" auf aktives Gerät klicken und "Explorer" wählen.

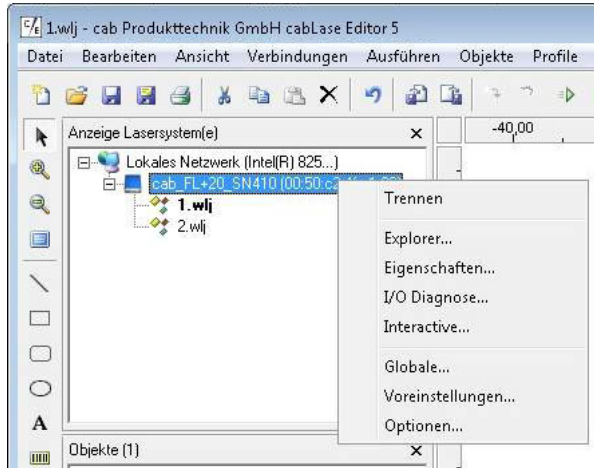


Anzeige der Job-Dateien

- "Internal flash memory" und Datei-Typ "Job" wählen, um gespeicherte .dat-Dateien anzuzeigen.
- "Kopieren nach..." wählen, um Dateien vom Beschriftungslaser FL+ an den PC zu übertragen.
- "Hinzufügen.." wählen, um bereits auf dem PC vorhandene .dat-Dateien vom PC auf den Beschriftungslaser FL+ zu übertragen.

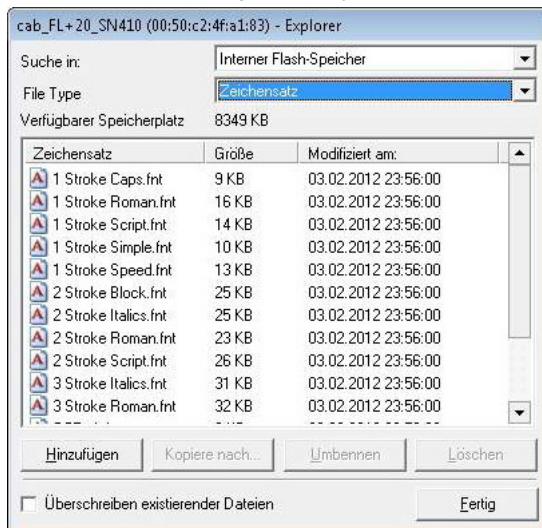
6.4 Speichern von Laser-Font-Dateien

Das beschriebene Beispiel verwendet in den Job Dateien 1.wlj und 2.wlj bzw. 1.dat und 2.dat Textobjekte die mit Arial-Font erstellt wurden. Dieser muss in den Laserspeicher geladen werden.



Aufruf des System-Explorers

- Im Fenster "Anzeige Lasersysteme" auf aktives Gerät klicken und "Explorer" wählen.



Anzeige der Font-Dateien

- "Internal flash memory" und Datei-Typ "Zeichensatz" wählen, um verfügbare Laserfonts anzuzeigen.
- "Hinzufügen.." wählen, um Schriftfont vom PC auf das Lasersystem zu übertragen.
- "Kopieren nach..." wählen, um Schriftfont vom Lasersystem an den PC zu übertragen.



Hinweis!

Das Standardverzeichnis für installierte Laser-Font-Dateien auf dem lokalen PC ist c:\marker\marker\fonts. Weitere Informationen zur Umwandlung von TrueType-Fonts in Laserfonts ► [Software-Dokumentation](#).



Hinweis!

Um ein Layout über die Remote API Schnittstelle auszuführen, muss im Vorfeld geprüft werden, ob sich die im Layout verwendeten Fonts auf der Lasersteuerung befinden.



Hinweis!

Um die neu geladenen Fonts verwenden zu können, muss das Lasersystem neu initialisiert werden. Dazu gibt es mehrere Möglichkeiten:

- Laser aus- und wieder einschalten
- Reset am Bedienfeld ausführen
- Reset von der externen Steuerung über CON3 - PIN10 auslösen

7.1 API Befehlssatz

Die Befehlsschnittstelle, welche von der Remote API unterstützt wird, basiert auf einem Textprotokoll. Alle Befehle müssen am Ende mit einem Carriage Return (0X0D) und Line Feed (0X0A) abgeschlossen werden, so dass eine Übertragung erfolgreich stattfinden kann.

Die nachfolgende Liste beschreibt eine Auswahl der wichtigsten Remote API Befehle zum Aufbau der üblichsten Applikationen. Eine vollständige Liste aller Befehle erhalten Sie in unserer englischsprachigen Remote API Library in einem separaten Dokument.

Befehle mit mehreren Eingabeparametern werden in kommasepariertem Format zum Host geschickt. Sofern Antworten mit mehreren Parametern zurückgeschickt werden, erfolgt dies ebenfalls in kommasepariertem Format.



Hinweis!

Alle Befehle müssen mit der Anweisung „Carriage Return“ und „Line Feed“ abgeschlossen werden (0X0D & 0X0A / <CR>&<LF>).



Hinweis!

Die Übertragung von Befehlen mit mehreren Parametern erfolgt in einem kommaseparierten Format.



Hinweis!

Antworten von der Lasersteuerung mit mehreren Parametern werden in kommasepariertem Format übertragen.



Hinweis!

Alle Antworten werden mit der Anweisung „Carriage Return“ und „Line Feed“ abgeschlossen (0X0D & 0X0A / <CR>&<LF>).

7.1 Abort

Befehl	Abort
Verwendung	Stoppt die Ausführung eines Beschriftungsauftrags
Programmcode	1 oder 1,blocking
Parameter	Parameter blocking: (Integerwert) 0 - für sofortigen Abbruch, bspw.: 1,0 1 - für Abbruch nach Abarbeitung des aktuellen Auftrags, bspw.: 1,1
API Rückmeldung	0 = Success, 1 = Idle
Bemerkung	Für den Abbruch Befehl gibt es generell zwei Möglichkeiten. Zum einen, dass die Abarbeitung des aktuellen Auftrags sofort gestoppt werden soll, zum anderen dass erst nach der vollständigen Abarbeitung und Verarbeitung aller Daten gestoppt wird. Wird der Parameter 0, für sofortigen Abbruch verwendet stoppt der Laser sofort ohne Rücksicht auf Datenverlust. Beim Parameter 1 wird auf die vollständige Verarbeitung Rücksicht genommen. Wird die Kurzform des Befehls 1 verwendet ist automatisch der Parameter 1 gesetzt.



Hinweis!

Verwenden Sie immer den Parameter blocking=1, wenn Sie einen zur Software synchronisierten Fertigungsablauf und eine eindeutige Datenzuordnung wünschen (Seriennummern / Tracking and Tracing).

7.2 ClearJobList

Befehl	ClearJobList
Verwendung	Löscht alle in den Auftragsspeicher geladenen Beschriftungsaufträge
Programmcode	200
Parameter	-
API Rückmeldung	0 = Success
Bemerkung	Wird verwendet, um alle sich im Auftragsspeicher befindlichen Beschriftungsaufträge zu löschen. Der Befehl wird auch auf den aktuell aktivierten Beschriftungsauftrag angewendet.



Hinweis!

Durch ClearJobList wird der aktuell durch MakeJobActive aktivierte Beschriftungsauftrag aus dem Speicher gelöscht.

7.3 ConnectNetworkShare

Befehl	ConnectNetworkShare
Verwendung	Stellt die Verbindung zu einem Netzlaufwerk her
Programmcode	523,Remotesharename,Username,Password
Parameter	Remotesharename: Spezifiziert das Netzlaufwerk, zu dem eine Verbindung aufgebaut werden soll (maximal 256 Zeichen). Der Remotesharename ist beispielhaft im Format \\ServerDaten\Laserstation einzugeben. Username: Spezifiziert den User für die Anmeldung (Maximal 49 Zeichen) Password: Spezifiziert das Passwort für die Anmeldung des Users (Maximal 63 Zeichen)
API Rückmeldung	Wenn der Befehl aufgrund eines Netzwerkfehlers nicht ausgeführt werden kann, wird ein Responsecode: 33,extendederrorinfo zurückgeschickt. Die Fehlermeldungen werden von der Windows API erzeugt. Wenn beispielsweise der Zugriff verweigert wird, ist die Antwort 33,5. Mehr Informationen erhalten Sie im Windows API Handbuch.
Bemerkung	Dieser Befehl wird nur von Netzwerken unter Microsoft Windows unterstützt. Das Zugriffsverzeichnis kann sich nicht direkt auf einer Domain befinden.



Hinweis!

Die komplette Funktionalität des Befehls im Bereich "extendederrorinfo" ist nur in mit Windows Betriebssystemen kompatiblen Netzwerken vorhanden.

7.4 Echo

Befehl	Echo
Verwendung	Liefert eine Rückmeldung der Steuerung auf eine beliebige Zeichenkettenübertragung
Programmcode	37,String
Parameter	String: Ist eine Zeichenkette, die nach dem Senden als Echo von der Lasersteuerung zurückgesendet wird (maximal 3000 Zeichen).
API Rückmeldung	Entspricht dem String, der gesendet wurde
Bemerkung	Dieser Befehl kann verwendet werden, um Kommunikationsbefehle zu verifizieren, oder bei aktiviertem Energiesparmodus die Steuerung „live“ zu halten.

7.5 EnableObject

Befehl	EnableObject
Verwendung	Ermöglicht die selektive An- oder Abwahl eines Objekts
Programmcode	106, Objectindex, State
Parameter	Objectindex: Spezifiziert die ganzzahlige Indexnummer des Objekts State: 0 bedeutet, dass das Objekt nicht markiert wird 1 bedeutet, dass das Objekt markiert wird
API Rückmeldung	0 = Success
Bemerkung	Um den Befehl durchführen zu können, darf sich der Beschriftungsauftrag nicht in Bearbeitung befinden.



Hinweis!

Wird eine große Anzahl von Objekten ausgeschaltet, kann es sinnvoller sein, an Stelle des Wiedereinschaltens der Objekte das Layout neu zu laden.



Hinweis!

Der Objektindex entspricht der in der Software cabLase visualisierten Objektnummer um 1 reduziert.

7.6 ExecuteJobContinuous

Befehl	EcecuteJobContinuous
Verwendung	Startet die Aktivierung eines aktivierten Beschriftungsauftrags zur dauerhaften Wiederholung
Programmcode	208, cacheobjects
Parameter	Cacheobjects: (Integerwert 0 oder 1) 0 bedeutet, dass die Beschriftungsobjekte erst nach einem Startsignal im internen FIFO Speicher aufbereitet werden 1 bedeutet, dass die Beschriftungsobjekte unmittelbar nach dem Befehl für die Beschriftung im FIFO Speicher aufbereitet werden
API Rückmeldung	0 = Success, 2 = Busy, 109 = NoActiveJob
Bemerkung	Der client muss sich im TakeHostControl Modus befinden, um diesen Befehl ausführen zu können. Der Beschriftungsauftrag muss sich im Auftragsspeicher der Lasersteuerung befinden und durch MakeJobActive aktiviert sein. Die Steuerung wartet auf ein digitales Start Marking Signal und wird den Beschriftungsprozess solange wiederholen, bis der Befehl Abort aufgerufen wird. Cacheobjects = 0 und cacheobjects = 1 unterscheiden sich voneinander durch den Zeitraum, zu dem Objekte und Anweisungen für den FIFO Speicher aufbereitet werden. Bei cacheobjects = 1 wird mit der Aufbereitung nicht auf das externe Startsignal gewartet. Die Ausführung mit cacheobjects = 0 macht Sinn, wenn Objekte im Layout, bspw. durch digitale Eingangssignale gesteuert werden sollen. Der Regelfall ist cacheobjects = 1.



Hinweis!

Bei der Verwendung von ExecuteJobContinuous muss bei der Erstellung des Beschriftungslayouts das externe Startsignal aktiviert werden.



Hinweis!

Bei wiederholter Beschriftung eines Beschriftungsauftrags muss das externe Startsignal wiederholt abgesetzt werden.

7.7 ExecuteJobOnce

Befehl	EnableObject
Verwendung	Startet die Beschriftung eines aktivierten Beschriftungsauftrags einmalig ohne Wiederholung
Programmcode	207,cacheobjects
Parameter	Cacheobjects: (Integerwert 0 oder 1) 0 bedeutet, dass die Beschriftungsobjekte erst nach einem Startsignal im internen FIFO Speicher aufbereitet werden 1 bedeutet, dass die Beschriftungsobjekte unmittelbar nach dem Befehl für die Beschriftung im FIFO Speicher aufbereitet werden
API Rückmeldung	0 = Success, 2 = Busy, 109 = NoActiveJob
Bemerkung	Nach dem Absetzen des Befehls wartet die Steuerung auf ein externes digitales Startsignal auf der Schnittstelle CON2, bevor mit der Beschriftung begonnen wird. Der Prozess kann an beliebiger Stelle durch den Befehl „Abort“ abgebrochen werden, auch wenn mit der Beschriftung noch nicht begonnen wurde. Die Ausführung mit cacheobjects = 0 macht Sinn, wenn Objekte im Layout, bspw. durch digitale Eingangssignale gesteuert werden sollen. Der Regelfall ist cacheobjects = 1.



Hinweis!

Bei der Verwendung von ExecuteJobOnce muss bei der Erstellung des Beschriftungslayouts das externe Startsignal aktiviert werden.



Hinweis!

Bei wiederholter Beschriftung eines Beschriftungsauftrags muss der Befehl wiederholt abgesetzt werden.

7.8 GetAllIOWords

Befehl	GetAllIOWords																																																				
Verwendung	Liefert den Status aller digitalen Ein- und Ausgänge des Systems																																																				
Programmcode	31																																																				
Parameter	-																																																				
API Rückmeldung	<p>StandardWord,ExtendedWord</p> <p>Die API liefert ein 18-Bit Wort gefolgt von einem 32-Bit-Wort Das StandardWord liefert die Ein- und Ausgänge der Grundfunktionen Das ExtendedWord liefert die Ein- und Ausgänge der Erweiterung</p> <p>StandardWord:</p> <table> <tr> <td>Bit 0: User In 1</td><td>Bit 10: User Out 1</td></tr> <tr> <td>Bit 1: User In 2</td><td>Bit 11: User Out 2</td></tr> <tr> <td>Bit 2: User In 3</td><td>Bit 12: User Out 3</td></tr> <tr> <td>Bit 3: User In 4</td><td>Bit 13: User Out 4</td></tr> <tr> <td>Bit 4: Start Marking</td><td>Bit 14: Marking in Progress</td></tr> <tr> <td>Bit 5: Job Load</td><td>Bit 15: Job Busy</td></tr> <tr> <td>Bit 6: Interlock 1</td><td>Bit 16: System Error</td></tr> <tr> <td>Bit 7: Interlock 2</td><td>Bit 17: Ready</td></tr> <tr> <td>Bit 8: Interlock 3</td><td></td></tr> <tr> <td>Bit 9: Interlock 4</td><td></td></tr> </table> <p>ExtendedWord:</p> <table> <tr> <td>Bit 0: User In 5</td><td>Bit 16: User Out 5</td></tr> <tr> <td>Bit 1: User In 6</td><td>Bit 17: User Out 6</td></tr> <tr> <td>Bit 2: User In 7</td><td>Bit 18: User Out 7</td></tr> <tr> <td>Bit 3: User In 8</td><td>Bit 19: User Out 8</td></tr> <tr> <td>Bit 4: User In 9</td><td>Bit 20: User Out 9</td></tr> <tr> <td>Bit 5: User In 10</td><td>Bit 21: User Out 10</td></tr> <tr> <td>Bit 6: User In 11</td><td>Bit 22: User Out 11</td></tr> <tr> <td>Bit 7: User In 12</td><td>Bit 23: User Out 12</td></tr> <tr> <td>Bit 8: User In 13</td><td>Bit 24: User Out 13</td></tr> <tr> <td>Bit 9: User In 14</td><td>Bit 25: User Out 14</td></tr> <tr> <td>Bit 10: User In 15</td><td>Bit 26: User Out 15</td></tr> <tr> <td>Bit 11: User In 16</td><td>Bit 27: User Out 16</td></tr> <tr> <td>Bit 12: User In 17</td><td>Bit 28: User Out 17</td></tr> <tr> <td>Bit 13: User In 18</td><td>Bit 29: User Out 18</td></tr> <tr> <td>Bit 14: User In 19</td><td>Bit 30: User Out 19</td></tr> <tr> <td>Bit 15: User In 20</td><td>Bit 31: User Out 20</td></tr> </table>	Bit 0: User In 1	Bit 10: User Out 1	Bit 1: User In 2	Bit 11: User Out 2	Bit 2: User In 3	Bit 12: User Out 3	Bit 3: User In 4	Bit 13: User Out 4	Bit 4: Start Marking	Bit 14: Marking in Progress	Bit 5: Job Load	Bit 15: Job Busy	Bit 6: Interlock 1	Bit 16: System Error	Bit 7: Interlock 2	Bit 17: Ready	Bit 8: Interlock 3		Bit 9: Interlock 4		Bit 0: User In 5	Bit 16: User Out 5	Bit 1: User In 6	Bit 17: User Out 6	Bit 2: User In 7	Bit 18: User Out 7	Bit 3: User In 8	Bit 19: User Out 8	Bit 4: User In 9	Bit 20: User Out 9	Bit 5: User In 10	Bit 21: User Out 10	Bit 6: User In 11	Bit 22: User Out 11	Bit 7: User In 12	Bit 23: User Out 12	Bit 8: User In 13	Bit 24: User Out 13	Bit 9: User In 14	Bit 25: User Out 14	Bit 10: User In 15	Bit 26: User Out 15	Bit 11: User In 16	Bit 27: User Out 16	Bit 12: User In 17	Bit 28: User Out 17	Bit 13: User In 18	Bit 29: User Out 18	Bit 14: User In 19	Bit 30: User Out 19	Bit 15: User In 20	Bit 31: User Out 20
Bit 0: User In 1	Bit 10: User Out 1																																																				
Bit 1: User In 2	Bit 11: User Out 2																																																				
Bit 2: User In 3	Bit 12: User Out 3																																																				
Bit 3: User In 4	Bit 13: User Out 4																																																				
Bit 4: Start Marking	Bit 14: Marking in Progress																																																				
Bit 5: Job Load	Bit 15: Job Busy																																																				
Bit 6: Interlock 1	Bit 16: System Error																																																				
Bit 7: Interlock 2	Bit 17: Ready																																																				
Bit 8: Interlock 3																																																					
Bit 9: Interlock 4																																																					
Bit 0: User In 5	Bit 16: User Out 5																																																				
Bit 1: User In 6	Bit 17: User Out 6																																																				
Bit 2: User In 7	Bit 18: User Out 7																																																				
Bit 3: User In 8	Bit 19: User Out 8																																																				
Bit 4: User In 9	Bit 20: User Out 9																																																				
Bit 5: User In 10	Bit 21: User Out 10																																																				
Bit 6: User In 11	Bit 22: User Out 11																																																				
Bit 7: User In 12	Bit 23: User Out 12																																																				
Bit 8: User In 13	Bit 24: User Out 13																																																				
Bit 9: User In 14	Bit 25: User Out 14																																																				
Bit 10: User In 15	Bit 26: User Out 15																																																				
Bit 11: User In 16	Bit 27: User Out 16																																																				
Bit 12: User In 17	Bit 28: User Out 17																																																				
Bit 13: User In 18	Bit 29: User Out 18																																																				
Bit 14: User In 19	Bit 30: User Out 19																																																				
Bit 15: User In 20	Bit 31: User Out 20																																																				
Bemerkung	Das ExtendedWord ist bei cab FL+ nur bis zum Eingang 8 und Ausgang 8 gültig.																																																				



Hinweis!

Bei FL+ sind die Bits 4-15 und Bits 20-31 beim ExtendedWord nicht aktiv.

7.9 GetFlashJobFileList

Befehl	GetFlashJobFileList
Verwendung	Übergibt eine kommaseparierte Liste aller sich im internen Flash Speicher der Lasersteuerung befindlichen Beschriftungsaufträge
Programmcode	203
Parameter	-
API Rückmeldung	0,job1,jobX job1 Entspricht dem 1ten Namen des Beschriftungsauftrags im Speicher jobX Entspricht dem Xten Namen des Beschriftungsauftrags im Speicher
Bemerkung	Beim internen Flash Speicher handelt es sich um den systeminternen Speicher auf der Lasersteuerung. Das Lesen und Beschreiben des Speichers kann über cabLase oder im FTP Modus erfolgen.



Hinweis!

Mehr Informationen zum Lesen und Speichern von Beschriftungsaufträgen auf den internen Flashspeicher finden Sie unter Kapitel 6.2.



Hinweis!

Sofern das Auslesen des internen Flash Speichers nicht möglich ist, antwortet die Steuerung mit einem ErrorCode.

7.10 GetFontFileList

Befehl	GetFontFileList
Verwendung	Liefert eine kommaseparierte Liste aller Fonts, welche auf der Lasersteuerung im Flash Memory installiert sind
Programmcode	48
Parameter	-
API Rückmeldung	0,Fontfile1,...,fontfileN Fontfile1 entspricht dem ersten Font in der Liste FontfileN entspricht dem letzten Font in der Liste
Bemerkung	Unter dem Flash Memory versteht man den internen Speicher der Lasersteuerung, auf welchen mit cabLase oder mit einem FTP Server Zugriff erfolgt.



Hinweis!

Mehr Informationen über den Zugriff auf den internen Flash Speicher erhalten Sie in Kapitel 6.2 oder in der Montageanleitung des Beschriftungslasers FL⁺.

7.11 GetKFactor

Befehl	GetKFactor
Verwendung	Liefert den Kalibrierungsfaktor der aktuellen F-Theta Linsenkonfiguration
Programmcode	10
Parameter	-
API Rückmeldung	KFactor: Liefert den Kalibrierungsfaktor in Feldeinheiten (Bits/mm)
Bemerkung	Verwenden Sie diesen Befehl, um eine Konvertierung zwischen der Maßeinheit mm und den laserspezifischen Feldeinheiten zu erhalten.



Hinweis!

Beachten Sie, dass Lasersteuerungen der Plattform 6 und Plattform 7 die Verwendung von nativen Maßangaben in µm unterstützen.

7.12 GetNetworkJobFileList

Befehl	GetNetworkJobFileList
Verwendung	Übergibt eine kommaseparierte Liste aller sich im Netzwerkpfad befindlichen Beschriftungsaufträge
Programmcode	221,Subfolder
Parameter	Subfolder: Ist die Pfadangabe für das Netzlaufwerk 221, übergibt die Dateien im Root Verzeichnis 221,\Arbeitsdateien übergibt die Dateien im Verzeichnis \Arbeitsdateien
API Rückmeldung	0,job1,jobX job1 Entspricht dem 1ten Namen des Beschriftungsauftrags im Netzwerk jobX Entspricht dem Xten Namen des Beschriftungsauftrags im Netzwerk
Bemerkung	Um den Befehl ausführen zu können, muss zuvor über den Befehl ConnectNetworkShare eine Verbindung zum Netzlaufwerk aufgebaut werden.



Hinweis!

Verwenden Sie zwingend den Befehl ConnectNetworkShare zum Initialisieren einer Netzwerkverbindung, um den Befehl GetNetworkFileList anzuwenden.

7.13 GetObjectCenter

Befehl	GetObjectCenter
Verwendung	Liefert die Koordinaten des Zentrums des Objekts
Programmcode	104,Objectindex
Parameter	Objectindex: Ist die ganzzahlige Nummer des indexierten Objekts
API Rückmeldung	X,Y X entspricht der X-Koordinate des Zentrums in Bits Y entspricht der Y-Koordinate des Zentrums in Bits
Bemerkung	Um den Befehl durchführen zu können, darf sich der Beschriftungsauftrag nicht in Bearbeitung befinden. Das Beschriftungsfeld ist als kartesisches Koordinatensystem aufgebaut, dessen Ursprung in der Mitte ist.

7.14 GetUSBJobFileList

Befehl	GetUSBJobFileList
Verwendung	Übergibt eine kommaseparierte Liste aller sich auf dem angeschlossenen USB Speichermedium befindlichen Beschriftungsaufträge
Programmcode	204
Parameter	-
API Rückmeldung	0,job1,jobX job1 Entspricht dem 1ten Namen des Beschriftungsauftrags im Speicher jobX Entspricht dem Xten Namen des Beschriftungsauftrags im Speicher
Bemerkung	Sofern kein USB Speichermedium angeschlossen wurde, antwortet die Steuerung mit dem Response: NoDrive. USB Speichermedien unterscheiden sich bezüglich Formatierung, Speichergröße und Adressierung. Sollten Probleme beim Zugriff bestehen, wechseln Sie zunächst das Speichermedium, oder setzen Sie sich mit dem Hersteller in Verbindung.


Hinweis!

Sofern das Auslesen des USB Speichermediums nicht möglich ist, antwortet die Steuerung mit einem ErrorCode.

7.15 HardwareReset

Befehl	HardwareReset
Verwendung	Führt einen Reset an der Steuerung des Lasers durch
Programmcode	8
Parameter	-
API Rückmeldung	keine Antwort, da Lasersteuerung neu initialisiert wird
Bemerkung	Durch diesen Befehl führt die Steuerung einen Softreset durch. Vor dem Bootvorgang wird die Socket Verbindung zum Client automatisch geschlossen und muss nach der Initialisierung erneut aufgebaut werden. Ebenfalls muss erneut mit dem Befehl TakeHostControl gestartet werden. Sofern Änderungen an der IP Adresseinstellung vorgenommen wurden, sind diese nach dem Neustart aktiv.


Hinweis!

Alle sich im Arbeitsspeicher befindlichen Aufträge werden gelöscht und müssen neu geladen und aktiviert werden. Übergebene Variablen gehen verloren.


Hinweis!

Während des Bootvorgangs kann eine Verbindung mit der Lasersteuerung aufgebaut werden, jedoch kein Datenaustausch erfolgen. Das Beenden des Bootvorgangs kann über eine Statusabfrage oder auf der digitalen Schnittstelle abgefragt werden.


Hinweis!

Die über den HardwareReset eingeleitete Initialisierung kann bis zu 90 Sekunden dauern.

7.16 LoadFlashJob

Befehl	LoadFlashJob
Verwendung	Lädt einen Markierungsauftrag aus dem Flashspeicher der Steuerung in den Arbeitsspeicher
Programmcode	205.jobname
Parameter	jobname: (Stringwert) Ist der Dateiname des Layouts mit der Dateierweiterung, bspw. Circle.dat.
API Rückmeldung	0 = Success, 3 = NoJob
Bemerkung	Es können gleichzeitig mehrere Beschriftungsaufträge in den Auftragsspeicher der Steuerung geladen werden. Bevor am Layout editiert oder modifiziert werden kann, ist es erforderlich, das Layout mit dem Befehl MakeJobActive zu aktivieren.



Hinweis!

Es können gleichzeitig mehrere Beschriftungsaufträge in den Auftragsspeicher geladen werden.



Hinweis!

Bevor ein Beschriftungsauftrag, bspw. hinsichtlich Variablen, Objekten oder sonstigen Eigenschaften editiert oder modifiziert werden kann, muss das Layout über den Befehl MakeJobActive aktiviert werden.

7.17 LoadNetworkJob

Befehl	LoadNetworkJob
Verwendung	Lädt einen Markierungsauftrag aus einer im Netzwerk verfügbaren Quelle in den Arbeitsspeicher
Programmcode	222.jobname
Parameter	jobname: (Stringwert) Ist der Pfad und Dateiname des Layouts mit der Dateierweiterung, bspw. \Arbeitsdateien\ Circle.dat.
API Rückmeldung	0 = Success, 3 = NoJob
Bemerkung	Der Jobname kann sämtliche Unterverzeichnisse relativ zum verbundenen Netzlaufwerk beinhalten. Das Netzlaufwerk wird mit dem Befehl ConnectNetworkShare definiert und muss im Befehl nicht explizit angegeben werden! Soll die Datei circle.dat vom Laufwerk server01 im Pfad \Arbeitsdateien\ geladen werden, gestaltet sich der Befehl wie folgt: 222,\Arbeitsdateien\circle.dat Der Zugriff erfolgt unter: \\Server01\Arbeitsdateien\circle.dat Es können gleichzeitig mehrere Beschriftungsaufträge in den Auftragsspeicher der Steuerung geladen werden. Bevor das Layout editiert oder modifiziert werden kann, ist es erforderlich, dieses mit dem Befehl MakeJobActive zu aktivieren.



Hinweis!

Netzlaufwerke werden außerhalb des Befehls LoadNetworkJob mit dem Befehl ConnectNetworkShare definiert.



Hinweis!

Laufwerksbezeichnungen und Server werden im jobname nicht angegeben, sondern nur die relativen Unterverzeichnisse.

**Hinweis!**

Es können gleichzeitig mehrere Beschriftungsaufträge in den Auftragsspeicher geladen werden.

**Hinweis!**

Bevor ein Beschriftungsauftrag, bspw. hinsichtlich Variablen, Objekten oder sonstigen Eigenschaften editiert oder modifiziert werden kann, muss das Layout über den Befehl MakeJobActive aktiviert werden.

7.18 LoadUSBJob

Befehl	LoadUSBJob
Verwendung	Lädt einen Beschriftungsauftrag über ein am USB angeschlossenes Speichermedium in den Arbeitsspeicher
Programmcode	206,jobname
Parameter	jobname: (Stringwert) Ist der Dateiname des Layouts mit der Dateierweiterung, bspw. Circle.dat
API Rückmeldung	0 = Success, 3 = NoJob
Bemerkung	Es können gleichzeitig mehrere Beschriftungsaufträge in den Auftragsspeicher der Steuerung geladen werden. Bevor am Layout editiert oder modifiziert werden kann, ist es erforderlich, das Layout mit dem Befehl MakeJobActive zu aktivieren.

**Hinweis!**

Es können gleichzeitig mehrere Beschriftungsaufträge in den Auftragsspeicher geladen werden.

**Hinweis!**

Bevor ein Beschriftungsauftrag, bspw. hinsichtlich Variablen, Objekten oder sonstigen Eigenschaften editiert oder modifiziert werden kann, muss das Layout über den Befehl MakeJobActive aktiviert werden.

7.19 MakeJobActive

Befehl	MakeJobActive
Verwendung	Aktiviert einen sich im Auftragsspeicher befindlichen Markierungsauftrag zu Bearbeitung
Programmcode	201,jobname
Parameter	Jobname: (Stringwert) Ist der Dateiname des Layouts mit der Dateierweiterung, bspw. Circle.dat.
API Rückmeldung	0 = Success, 2 = Busy, 3 = NoJob
Bemerkung	Bevor Veränderungen, bspw. Variablen in ein Layout übernommen werden können, muss das jeweilige Layout zunächst aktiviert werden. Sofern sich mehrere Layouts im Auftragsspeicher befinden, wird immer der Beschriftungsauftrag abgearbeitet, welcher durch den Befehl aktiviert wurde.


Tip!

Es können gleichzeitig mehrere Beschriftungsaufträge in den Auftragsspeicher geladen werden. Durch Umschalten mit MakeJobActive wird die Prozesszeit minimiert.

7.20 RemoveJob

Befehl	RemoveJob
Verwendung	Löscht den aktuellen Beschriftungsauftrag aus dem Speicher
Programmcode	202
Parameter	-
API Rückmeldung	0 = Success, 3 = NoJob
Bemerkung	Die Routine erlaubt das Löschen des aktuell im Auftragsspeicher befindlichen Beschriftungsauftrags, welcher über MakeJobActive aktiviert wurde. Sollen mehrere Beschriftungsaufträge gelöscht werden, so müssen diese einzeln aktiviert werden.


Hinweis!

RemoveJob löscht die aktivierten Beschriftungsaufträge aus dem Auftragsspeicher, jedoch nicht die Beschriftungslayouts, welche auf der Steuerung gespeichert sind.

7.21 RemoveObject

Befehl	RemoveObject
Verwendung	Löscht ein Objekt in dem aktuell aktivieren Beschriftungsauftrag
Programmcode	141,objectindex
Parameter	objectindex: (Integerwert) Ist die laufende Nummer des Objekts im Beschriftungslayout.
API Rückmeldung	0 = Success, 10 = ArgOutOfRange, 104 = NoObject
Bemerkung	Die Ausführung des Befehls ist nur dann möglich, wenn sich kein Beschriftungsauftrag in Bearbeitung befindet


Hinweis!

Zum wieder Einschalten der Objekte muss das Layout erneut geladen und aktiviert werden.

7.22 ReleaseHostControl

Befehl	ReleaseHostControl
Verwendung	Schließt die bestehende Verbindung eines Clients mit dem Host und setzt die Steuerung zurück in den Stream Mode
Programmcode	3
Parameter	-
API Rückmeldung	0 = Success
Bemerkung	Wenn ein Remote API Client die Verbindung zur Steuerung unterbricht, wird diese automatisch in den Stream Mode umgeschaltet. Diese Art von Schnittstellendatenübertragung wird unter Verwendung der Beschriftungssoftware cabLase verwendet.



Hinweis!

Sollen mehrere Clients mit einer Lasersteuerung Daten austauschen so ist darauf zu achten, dass die jeweiligen Instanzen mit ReleaseHostControl beendet werden.

7.23 ResetObject

Befehl	ResetObject
Verwendung	Löscht die Vektorliste eines Objekts nach einer Transformation und stellt die ursprünglichen initialen Vektoren wieder her
Programmcode	111,objectindex
Parameter	Objectindex: Entspricht dem ganzzahligen Objektindex
API Rückmeldung	0 = Success, 104 = NoObject
Bemerkung	Um den Befehl durchführen zu können, darf sich der Beschriftungsauftrag nicht in Bearbeitung befinden. Der Befehl löscht die Vektoren des transformierten Objekts und generiert die Vektorliste eines sich im Ursprungszustand befindlichen Objekts ohne Transformation.



Hinweis!

Der Befehl löscht die nach der Transformation generierten Vektoren, basierend auf dem aktuellen Wert des Objekts, bspw. aktueller Seriennummer. Soll der ursprüngliche Zustand wieder hergestellt werden, muss der Beschriftungsauftrag neu geladen werden.

7.24 ResetUserTransform

Befehl	ResetUserTransform
Verwendung	Hebt eine Objekttransformation auf und stellt den ursprünglichen Zustand her
Programmcode	112,objectindex
Parameter	Objectindex: Entspricht dem ganzzahligen Objektindex
API Rückmeldung	0 = Success, 104 = NoObject
Bemerkung	Um den Befehl durchführen zu können, darf sich der Beschriftungsauftrag nicht in Bearbeitung befinden. Der Befehl hebt die Transformation auf. Um zusätzlich die dem Objekt entsprechende, generierte Vektorliste zu löschen, muss zusätzlich der Befehl ResetObject angewendet werden.



Hinweis!

Um die aktive Vektorliste eines bereits transformierten Objekts zu löschen, muss nach dem Befehl ResetUserTransform auch der Befehl ResetObject verwendet werden.

7.25 SetExternalStartMode

Befehl	SetExternalStartMode
Verwendung	Legt den Modus fest wie ein externer Start zu erfolgen hat
Programmcode	215,Mode
Parameter	Mode: 0 = Startet, wenn der Eingang auf High steht 1 = Startet, wenn der Eingang auf Low steht 2 = Startet nach einem Flankenwechsel von Low auf High 3 = Startet nach einem Flankenwechsel von High auf Low
API Rückmeldung	0 = Success, 3 = NoJob
Bemerkung	Um den Befehl durchführen zu können, darf sich der Beschriftungsauftrag nicht in Bearbeitung befinden. Der Befehl legt fest, mit welcher Logik auf ein Start Process Signal an der digitalen Schnittstelle CON2 des Lasers reagiert wird.



Hinweis!

In der Regel wird das digitale Startsignal auf der Schnittstelle CON2 verwendet, um einen synchronisierten Ablauf zu starten. Wird der Mode = 1 verwendet, startet die Beschriftung des Auftrags bereits durch Verwenden des Befehls ExecuteJobOnce.

7.26 SetObjectString

Befehl	SetObjectString
Verwendung	Wird verwendet, um einem Zeichenketten basierenden Beschriftungsobjekt variable Daten zuzuweisen
Programmcode	100,objectindex,newstring
Parameter	objectindex: (Integerwert) Gibt die jeweilige Indexnummer des Objects aus dem Layout an newstring: (Stringwert) Ist die variable Information, die dem Objekt zugewiesen werden soll in einem Bereich von 1-2999 Zeichen
API Rückmeldung	0 = Success, 10 = ArgOutOfRange
Bemerkung	Das zu verändernde Objekt muss zwingend ein Zeichenketten basierendes Objekt sein. Der Objektindex muss dem zugehörigen Objekt entsprechen und Bestandteil der Objektliste sein. Der Beschriftungsauftrag darf sich nicht in Ausführung befinden. Der Befehl ExecuteJobOnce darf erst nach der Zuweisung der Variablen abgesetzt werden.



Hinweis!

Für die Zuweisung von Variablen in ein Objekt ist nicht der Objektname, sondern dessen Indexnummer relevant. Die Indexnummer kann in der Software cabLase ermittelt werden.



Hinweis!

Bei der Erstellung von Layouts ist zu beachten, dass das Einfügen von Objekten die jeweilige Indexnummer von nachfolgenden Objekten verändert.



Hinweis!

Die Objektindizierung beginnt mit 0. Die Indexnummer ist jeweils um 1 reduziert und startet mit "0".



Hinweis!

Der Beschriftungsauftrag darf sich zum Verändern von Variablen nicht in Ausführung befinden. Der Befehl ExecuteJobOnce darf erst nach der Veränderung der Variablen abgesetzt werden.

**Hinweis!**

Sollen beim Objekttyp Datamatrixcode Kontrollfunktionen eingeschlossen werden, müssen diese mit einem ~ - Zeichen vor dem Kontrollzeichen abgetrennt sein. Um ein normales Zeichen zu übergeben, muss eine doppeltes ~~ - Zeichen vorangestellt werden. Soll eine ASCII 0 übergeben werden, so ist die Zeichenfolge ~@ zu verwenden. Mehr Informationen hierzu finden Sie auch im Softwaremanual cabLase Editor 5 unter "Formatted String".

7.27 TakeHostControl

Befehl	TakeHostControl
Verwendung	Erlaubt die alleinige Kontrolle der Lasersteuerung von extern
Programmcode	2
Parameter	-
API Rückmeldung	0 = Success, 4 = InControl
Bemerkung	Ist die Lasersteuerung bereits mit einem Client oder die Software cabLase aktiv mit der Steuerung verbunden, oder ist die Steuerung in der Verarbeitung eines Beschriftungsauftrages, kann keine Verbindung aufgebaut werden. In diesem Fall muss der weitere Client die Verbindung trennen, die Software cabLase geschlossen, oder gewartet werden, bis der aktuelle Beschriftungsauftrag abgearbeitet ist.

**Hinweis!**

Verwenden Sie ReleaseHostControl um eine bestehende Verbindung zu einem Client zu trennen.

**Hinweis!**

Verwenden Sie GetJobStatus, um den Bearbeitungsstand des aktuellen Beschriftungsauftrags zu erhalten.

**Hinweis!**

Beenden Sie die Software cabLase vor dem Start des Remote API Interface, so dass der gemeinsame Port freigegeben wird.

7.28 TransformObject

Befehl	TransformObject
Verwendung	Wird verwendet um ein Objekt zu verschieben, zu drehen oder zu skalieren
Programmcode	102, Objectindex, Rotation, Rotationscenterx, Rotationscentery, Xscale, Yscale, Xoffset, Yoffset
Parameter	<p>Objectindex: Entspricht dem ganzzahligen Objektindex des Objekts</p> <p>Rotation: Gibt den Drehwinkel in Grad an im Wertebereich -360 bis +360</p> <p>RotationcenterX: Gibt die X-Koordinate des Mittelpunkts in Bits an, um welche die Drehung erfolgen soll im Wertebereich von -2147483648 bis +2147483647.</p> <p>RotationcenterY: Gibt die Y-Koordinate des Mittelpunkts in Bits an, um welche die Drehung erfolgen soll im Wertebereich von -2147483648 bis +2147483647.</p> <p>Xscale: Ist der Streckungsfaktor des Objekts in X-Richtung. Der Wert muss größer als 0 sein.</p> <p>Yscale: Ist der Streckungsfaktor des Objekts in Y-Richtung. Der Wert muss größer als 0 sein.</p> <p>Xoffset: Ist ein Offsetwert in Bits, um welcher das Objekt in X-Richtung verschoben werden soll im Wertebereich von -2147483648 bis +2147483647.</p> <p>Yoffset: Ist ein Offsetwert in Bits, um welcher das Objekt in Y-Richtung verschoben werden soll im Wertebereich von -2147483648 bis +2147483647.</p>
API Rückmeldung	0 = Success, 10 = ArgOutOfRange
Bemerkung	Durch Anwendung dieses Befehls werden sowohl die Hüllkurve als auch die Füllung eines Objekts transformiert. Die Transformation bleibt auch dann erhalten, wenn bspw. ein SetObjectString Befehl angewendet wird oder wenn es sich bei dem Objekt um ein Seriennummernfeld handelt. Die wiederholte Anwendung des Befehls wirkt relativ zur zuletzt durchgeführten Transformation. Durch Transformationen kann es vorkommen, dass sich ein Objekt außerhalb des spezifizierten Markierungsfeldes befindet. Eine Überprüfung kann bspw. über den Befehl GetObjectRectangle in Verbindung mit der Beschriftungsfeldgröße erfolgen. Das Beschriftungsfeld ist als kartesisches Koordinatensystem aufgebaut, dessen Ursprung in der Mitte ist.



Hinweis!

Koordinaten für Verschiebung und Drehzentrum müssen in Bits umgerechnet werden.



Hinweis!

Bei der Verwendung dieses Befehls obliegt es der Verantwortung des Programmierers, zu überprüfen, ob sich das Objekt nach der Transformation noch im Markierungsfeld befindet.



Achtung!

Der wiederholte Aufruf von TransformObject führt zu einer relativen Transformation. Um den ursprünglichen Zustand des Layouts wieder herzustellen, muss der Befehl ResetUserTransform angewendet werden.

8.1 Steuerung

Befehl	Beschreibung
Steuerung	
1	Abort
2	TakeHostControl
3	ReleaseHostControl
4	GetHostControlStatus
5	GetHostInControl
6	EnableBroadcasting
7	LoadHardwareDefaults
8	HardwareReset
9	GetRemoteIP
10	GetKFactor
14	SetPerformanceGlobals
15	ResetPerformanceGlobals
16	OpenCOMPort
17	CloseCOMPort
18	COMWriteLine
19	GoToZ
20	GoToXYZ
21	SetMOTFEncoderRate
22	SetMemBuffer
23	GetMemBuffer
24	GetAvailableRAM
27	COMWriteChar
29	SetUserOutBit
30	GetUserInWord
31	GetAllIOWords
32	SetUserOutInitWord
33	GetUserOutInitWord
34	SampleMOTFEncoderCount
35	ClearMOTFEncoderCount
36	GetMOTFEncoderCount
37	Echo
38	GetLensFileList
39	LoadLensFile
40	SetUserOutPreferences
41	GetUserOutPreferences
42	SetUserOutWord
43	GetVersions
44	GetLaserFileList
45	LoadLaserFile
46	GetMotionFileList
47	LoadMotionFile
48	GetFontFileList
49	GetActiveLaser

Befehl	Beschreibung
Steuerung	
50	GetActiveLens
52	GetAvailableDiskSpace
56	ClearCommandCache
57	TurnLaserOn
58	TurnLaserOff
59	GetMotionDeviceNames
60	GetMotionCalFactors
61	SendMotionCommand
63	GetMotionErrorCodes
64	GetMotionHomedOnceFlags
65	GetMotionStatus
66	GetProfileFileList
67	GetLastInterlockWord
70	PulseUserOutBit
71	COMWriteBinarychar
72	COMWriteCarEx
73	COMWriteLineEx
74	COMReadLineEx
75	CloseCOMPortEx
76	SetZOffsetRWU
77	SendMotionCommandEx

**Hinweis!**

Die vollständige Dokumentation aller Remote API Befehle erhalten Sie in unserem separaten Handbuch.

8.2 Objekte

Befehl	Beschreibung
Objekte	
100	SetObjectString
102	TransformObject
103	GetObjectRect
104	GetObjectCenter
105	GetObjectType
106	EnableObject
107	GetObjectString
108	GetObjectName
109	SetObjectUserData
110	GetObjectUserData
111	ResetObject
112	ResetUserTransform
113	TransformObjectByName
114	TransformObjectByNameEx
115	SetObjectProfile
116	GetObjectProfile
117	SetObjectProfileFromFile
118	GetObjectNumPasses
119	SetObjectNumPasses
120	GetObjectMarkMode
121	SetObjectMarkMode
122	GetObjectNumMarkingPasses
123	AddObjectMarkingPass
124	DeleteObjectMarkingPass
125	SetObjectPassSettings
126	GetObjectPassSettings
127	TransformObjectNewFill
128	TrnsformObjectByNameNewFill
136	NewObject
137	SetObjectUnicodeString
138	GetObjectUnicodeString
139	GetObjectVectors
140	SetObjectVectors
141	RemoveObject
142	GetObjectExecuteTime
143	SetObjectName
144	SetObjectProperties
145	GetObjectProperties
146	SetObjectOutlineSettings
147	GetObjectOutlineSettings
148	SetObjectFillSettings
149	GetObjectFillSettings
151	GetObjectFontMetrics

8.3 Beschriftungsauftrag

Befehl	Beschreibung
Beschriftungsauftrag	
200	ClearJobList
201	MakeJobActive
202	RemoveJob
203	GetFlashJobFileList
204	GetUSBJobFileList
205	LoadFlashJob
206	LoadUSBJob
207	ExecuteJobOnce
208	ExecuteJobContinuous
209	GetJobStatus
210	GetLastError
211	GetObjectCount
214	GetJobExecutionStatus
215	SetExternalStartMode
216	GetExternalStartMode
218	GetActiveJob
219	SaveFlashJob
220	SaveUSBJob
221	GetNetworkJobFileList
222	LoadNetworkJob
223	SaveNetworkJob
224	GetLastMotionError
225	NewJob


Hinweis!

Die vollständige Dokumentation aller Remote API Befehle erhalten Sie in unserem separaten Handbuch.

8.4 Administration

Befehl	Beschreibung
Administration	
500	SetAdminPIN
501	GetAdminPIN
502	SetDHCPMode
503	GetDHCPMode
504	SetLocalGateway
505	GetLocalGateway
506	SetLocalIP
507	GetLocalIP
508	SetNodeFriendlyName
509	GetNodeFriendlyName
510	SetSubnetMask
511	GetSubnetMask
512	SetUserPIN
513	GetUserPIN
514	SetCOMPortSpeed
515	GetCOMPortSpeed
516	SetCOMPortAssignments
517	GetCOMPortAssignments
518	SetLocalTime
519	GetLocalTime
523	ConnectNetworkShare
524	SetCOMPortSpeedEx
525	GetCOMPortSpeedEx
526	GetLocalDeviceList
527	SetActiveLocalDevice
528	SetCOMPortMode
529	GetCOMPortMode



Hinweis!

Die vollständige Dokumentation aller Remote API Befehle erhalten Sie in unserem separaten Handbuch.

Im Folgenden werden die möglichen API Response Codes, welche vom Host an den Client zurückgeschickt werden können, aufgelistet. Die Response Codes dienen zur Fehlersuche bei Problemen in der Ablaufsteuerung.

Wert	Kurzbeschreibung	Beschreibung
0	Success	Der Befehl wurde erfolgreich abgeschlossen
1	Idle	Derzeit kein Beschriftungsauftrag in Bearbeitung
2	Busy	Es befindet sich ein Beschriftungsauftrag in Bearbeitung
3	NoJob	Der ausgewählte Beschriftungsauftrag kann nicht gefunden werden
4	InControl	Die Steuerung befindet sich bereits in der Remote Kontrolle eines anderen Clients im Netzwerk
5	NotInContrl	Die Steuerung befindet sich nicht in Remote Kontrolle eines anderen Clients im Netzwerk
6	LicenseUnavailable	Keine gültige Softwarelizenz gefunden
7	LicenseAccessDenied	Zugriff verweigert. Die installierte Softwarelizenz erlaubt keinen Zugriff auf die ausgewählte Funktion
8	BadCommand	API Befehl außerhalb Befehlssatz
9	BadArg	Ungültiger Parameter im API Befehl
10	ArgOutOfRange	Ein Parameterwert im API Befehl liegt außerhalb des gültigen Wertebereichs
11	UnkownTimeZone	Ungültige Eingabe der Zeitzone beim System Setup
12	Reserved	Frei
13	BadConversion	Übertragungsfehler. Fehler beim Konvertieren von Multibyte- und Unicodezeichen
14	RegistryError	Schreib- oder Lesefehler in der Registry des Windows CE Betriebssystems
15	TimeZoneFileError	Fehler beim Zugriff auf eine Zeitzonendatei
16	ResetInterlock	Eine Not-Halt-Bedingung wurde signalisiert und muss durch den Befehl Abort zurückgesetzt werden
17	ListNotOpen	Zugriffsfehler bei einer Liste. Die Liste muss vor dem Zugriff geöffnet werden
18	ListAlreadyOpen	Zugriffsfehler auf eine geöffnete Liste. Die Liste ist bereits geöffnet
19	BadData	Formatfehler bei der Übergabe von Parametern in einem Befehl
20	APIException	API Fehler. Fehler im Ablauf des API Programms
21	JobAborting	Der Beschriftungsauftrag befindet sich im Abbruch eines zuvor gesendeten Abort Befehls
22	FPGALoadFail	Fehler beim Laden einer Datei aus dem FPGA der Lasersteuerung
23	JobManagerInitFail	Fehler beim Initialisieren der Beschriftungsaufträge
24	LaserLoadFail	Fehler beim Laden der Laserkonfigurationsdatei
25	LensLoadFail	Fehler beim Laden der Linsenkonfigurationsdatei
26	PMLoadFail	Fehler beim Laden der Leistungsparameter
27	MotionLoadFail	Fehler beim Initialisieren der Achsenkonfigurationsdatei
28	HostManagerInitFail	Fehler beim Initialisieren der Hostverwaltung
29	InvalidIPAddress	Die eingegebene IP-Adresse entspricht nicht den Standards von IPv4
30	DataUnknown	Das aktuelle Datenformat konnte nicht erkannt werden
31	BadChecksum	Fehler in der Checksumme der übertragenen Daten
32	NetworkShareNotConnected	Fehler beim Zugriff auf eine Datenquelle im Netzwerk. Keine Netzwerkverbindung eingerichtet oder vorhanden

Wert	Kurzbeschreibung	Beschreibung
33	NetworkConnectFail	Fehler beim Zugriff auf eine Netzwerkverbindung
34	UnknownNetworkError	Ein nicht spezifizierter Netzwerkfehler ist aufgetreten
35	APICommandTimeout	Timeout Fehler bei der Übertragung eines API Befehls
36	ExternalProcessFail	Interne Verwendung
37	DLLLoadFail	Interne Verwendung
38	NoAdapter	Es konnte kein Netzwerkadapter gefunden werden
39	AddIPAddressFailure	Fehler bei der Übergabe der IP-Adresse
40	BadAPIResponse	Der API Baustein antwortet mit einer nicht spezifizierten Rückmeldung
41	CannotCreateSocket	Interne Verwendung
42	CannotConnectSocket	Interne Verwendung
43	CannotGetFPGABufInfo	Interne Verwendung
44	CannotGetFPGABuf	Interne Verwendung
45	CannotWriteFPGABuf	Interne Verwendung
46	FPGAException	Interne Verwendung
47	FTPConnectionError	Fehler beim Versuch sich mit einer FTP Datenquelle zu verbinden
48	FileAlreadyExists	Der zu ladende Beschriftungsauftrag existiert bereits im Speicher
49	UnknownOS	Interne Verwendung
50	SocketException	Interne Verwendung
51	ProcessTimeout	Timeout Fehler im Prozess
52	DeviceNotFound	Das ausgewählte Gerät konnte nicht gefunden werden
53	LoginInProgress	Fehler bei der Verbindung mit einem externen Gerät. Das Gerät hat bereits eine aktive Verbindung
54	APIClientInControl	Es besteht bereits eine aktive Verbindung zu einem Remote API Client
55	StreamClientInControl	Es besteht bereits eine aktive Verbindung zur Beschriftungssoftware cabLase
56	CannotConnectToAPI	Fehler beim Zugriff auf ein Gerät durch den freigegebenen Port der Remote API
57	ReadFail	Interne Verwendung
58	StreamBufferFull	Interne Verwendung
59	NoConfigRecord	Der angegebene Eintrag in der Datenquelle kann nicht gefunden werden
60	OperationCanceled	Die Ausführung des Beschriftungsauftrags wurde durch den Anwender abgebrochen
61	NoData	Interne Verwendung
62	InitializationError	Interne Verwendung
63	FailToCreateServiceThread	Interne Verwendung
64	CannotOpenDevice	Interne Verwendung
65	SegmentFull	Interne Verwendung
66	MarkerLibraryNotInitialized	Es wurde versucht eine Operation durchzuführen, bevor die Systembibliotheken vollständig geladen wurden
67	RingBufferNotInitialized	Es wurde versucht eine Operation durchzuführen, bevor der Ringspeicher vollständig initialisiert wurde
68	AccessDenied	Zugriff zu einer Datenquelle verweigert
69	RequiresUACElevation	Die Operation kann nur durch eine Erhöhung der UAC ausgeführt werden

Wert	Kurzbeschreibung	Beschreibung
70	NotAllowed	Unerlaubte Operation
71	NoLaserConfig	Laserkonfigurationsdatei nicht gefunden
72	NoLensConfig	Linsenkonfigurationsdatei nicht gefunden
73	OutOfMemory	Speicherüberlauf
74	LensTableNotFound	Die Linsenkorrekturdatei konnte nicht gefunden werden
75	HostControlInitError	Keine Host Controller Routine beim Boot Vorgang geladen
76	NoBytesRead	Fehler beim Lesen von Daten. Keine Bytes gelesen
77	WritePending	Daten wurden zur Warteschlange hinzugefügt und so bald als möglich ausgeführt
84	NoPen	Ein in der Vektorliste angegebener Stift ist im System nicht mit einem Profil hinterlegt
100	NoFilesFound	Keine Dateien im spezifizierten Verzeichnis gefunden
101	NoDrive	Kein Laufwerk gefunden
102	JobOutOfMemory	Außerhalb des zulässigen Speicherbereichs
103	TooManyObjects	Systemfehler. Kontaktieren Sie den Hersteller
104	NoObject	Das spezifizierte Objekt existiert nicht
105	JobException	Der spezifizierte Beschriftungsauftrag kann nicht ausgeführt werden
106	NotInHostControl	Die Operation kann nicht ausgeführt werden. Es ist kein Remote API Client verbunden
107	WrongHostType	Die Operation kann mit der aktiven Remote API Host nicht ausgeführt werden
108	ErrorJobBusy	Die Operation kann nicht ausgeführt werden. Es befindet sich aktuell ein Beschriftungsauftrag in der Abarbeitung
109	NoActiveJob	Kein aktivierter Beschriftungsauftrag
110	ErrorSoftware	Systemfehler. Kontaktieren Sie den Hersteller
111	LoadFail	Fehler beim Laden eines Beschriftungsauftrags
112	NoObjects	Version der Beschriftungsdatei nicht kompatibel mit der aktuellen Firmwareversion
113	WriteFail	Fehler beim Schreiben der Beschriftungsauftragsdatei
114	JobFileFormat	Formatfehler bei der Beschriftungsauftragsdatei
115	FileException	Systemfehler bei der Ausführung des Beschriftungsauftrags
116	UnknownObject	Unbekannter Objekttyp
117	UnknownType	Unbekannter Typ
118	NotSupported	Operation wird nicht unterstützt
119	NotAvailable	Datenquelle nicht verfügbar
120	FPGADDataFail	Formatfehler im FPGA
121	FileNotFound	Die spezifizierte Datei wurde nicht gefunden
122	FileCreationError	Fehler beim Erstellen einer Datei
123	WriteFileFail	Unvollständige Datei. Fehler beim Schreiben der Datei
124	PathNotFound	Der angegebene Pfad konnte nicht gefunden werden
125	NotInCacheMode	Der Befehl erwartet, dass der Beschriftungsauftrag bereits im Cache Speicher des Systems gestartet wurde
126	NotWaitingForStartMark	Der FIFO Speicher wartet im Moment nicht auf ein Start Marking Signal
127	MotionNotHomed	Der Befehl ist nicht erlaubt, wenn die Bewegungsachsen nicht zuvor referenziert wurden
128	No3DModel	Die Operation konnte nicht durchgeführt werden. Es ist kein 3D Modell geladen

Wert	Kurzbeschreibung	Beschreibung
129	ProjectionError	Beim Projizieren eines 3D Modells ist ein Fehler aufgetreten
200	NoProperties	Objekt hat keine spezifizierten Eigenschaften
201	ObjectException	Interner Systemfehler beim Verarbeiten des Objekts
202	Abort	Ausführung wurde abgebrochen
203	NoFontResource	Der im Objekt spezifizierte Zeichensatz konnte nicht gefunden werden
204	NoOverride	Interner Objektfehler
205	ExternalEnableDenied	Ausführung von externer Steuerung nicht erlaubt
206	CannotCreatePort	Fehler beim Einstellen der Schnittstellendaten (Baudrate, Stop Bits, etc.)
207	CannotOpenPort	Fehler beim Öffnen der seriellen Schnittstelle
208	PortNotOpen	Serielle Schnittstelle muss vor Verwendung geöffnet werden
209	PortTimeout	Timeout Fehler an der seriellen Schnittstelle
210	WrongPortNumber	Ungültige serielle Schnittstelle
211	WrongObjectType	Ausführung wird bei dem verwendeten Objekttyp nicht unterstützt
212	AxisNotConfigured	Es wurde ein unspezifiziertes Objekt für eine externe Achse verwendet
213	TextBufferOverrun	Zu viele Zeichen im internen Pufferspeicher (maximal 3000 Zeichen)
214	InvBarcodeStringValue	Die Zeichenkette für ein Barcodeobjekt beinhaltet ungültige Zeichen
215	InvBarcodeStringLength	Die Länge der Zeichenkette für ein Barcodeobjekt hat eine ungültige Länge
216	InvBarcodeNarrowWidth	Der Barcode hat ein ungültiges Höhen-/Breitenverhältnis
217	InvBarcodeWidthReduce	Ungültige Breitenreduzierung bei einem Barcodeobjekt
218	InvBarcodeECC	Der ECC Wert für ein Datamatrixobjekt ist ungültig und kann nicht dekodiert werden
219	BarcodeOutOfMemory	Nicht ausreichend Speicherplatz vorhanden, um die Ausführung des Barcodeobjekts zu ermöglichen
220	BarcodeUnknownError	Systemfehler. Bitte kontaktieren Sie den Hersteller
221	BarcodeException	Falsche Daten. Falsches Ergebnis der Routine durch inkorrekte Daten
222	NoVectors	Beim Speichern eines Objekts wurden Außenkontur und Füllung nicht angewählt
223	BadMotionResponse	Der externe Achsenkontroller antwortet mit einem unerwarteten Fehler
224	MotionDriverNotFound	Ein Objekt greift auf einen Treiber für einen Achsenkontroller zu, welcher nicht existiert
225	AxisNotFound	Eine Operation greift auf die Indexnummer einer Achse zu, welche nicht existiert
226	EncoderNotFound	Eine Operation erwartet die Rückmeldung eines Encoders. Es wurde kein Decoder an der externen Achse gefunden
227	InvStringValue	Ungültige Zeichenkette
228	MotionControllerNotFound	Das Achssystem kann keinen zugehörigen Achskontroller finden
229	MotorNotProvisioned	Der verwendete Motor ist für den LEC Achskontroller nicht zulässig
230	RuntimeMotionError	Fehler bei einer Achsbewegung

Wert	Kurzbeschreibung	Beschreibung
231	ObjectOutOfBounds	Beschriftungsobjekt befindet sich außerhalb des Markierfelds
232	InvVersion	Interne Verwendung
233	NoOutline	Außenkontur wurde für die Markierung gewählt, ohne dass eine Kontur besteht

**Hinweis!**

In einigen Fällen kann es vorkommen, dass anstelle des erwarteten Zahlenwerts eine Fehlermeldung in Textform empfangen wird.

Im Folgenden werden die möglichen API Response Codes aufgelistet, welche aufgrund einer Systemabfrage über den API Befehl GetLaserError erfolgen können.

Wert	Kurzbeschreibung	
8001	QueueFul	Überlauf Spooler Warteschlange
9001	ProcessAbort	Prozess abgebrochen
9002	FIFOEmptyTimeout	Timeout FIFO Speicher – leer
9003	EventTimeout	Event Timeout
9004	BadOpcode	Ungültiger Befehl
9005	FirmwareBug	Fehler Firmware
9006	WriteDigitalBad	Fehler beim Schreiben von digitalen Informationen
9007	SetLaserPowerBad	Fehler Laserleistung
9008	SetCorrectionTableBad	Fehler Linsenkorrekturdatei
9009	SetLaserPulseBad	Fehler Laserpulsbreite
9010	WaitForIOBad	Fehler digitale IO Schnittstelle
9011	WaitForIOTimeout	Timeoutfehler digitale IO Schnittstelle
9012	SetLaserStandbyBad	Fehler Standby Laser
9013	CPLDTimeout	Timeoutfehler CPLD
9014	LaserActiveTimeout	Fehler Laser beim Einschalten des Lasers
9015	SetMotfOrientationBad	Fehler bei der Objektorientierung
9016	EnableMotfBad	Fehler beim Verwenden von Achsmodul
9020	ServoFault	Fehler Servoantrieb
9021	InterlockAssert	Not-Halt festgestellt
9022	interlockDeassert	Not-Halt Reset

In der Laserbeschriftungssoftware cabLase 5 werden unterschiedliche Objekttypen zur Gestaltung der Layouts verwendet. Bei der Adressierung über die Remote API Schnittstelle werden die unterschiedlichen Objekttypen nicht mit einem Namen, sondern als Integerwert übergeben.

Im Folgenden werden die Werte für unterschiedliche Objekttypen aufgelistet.

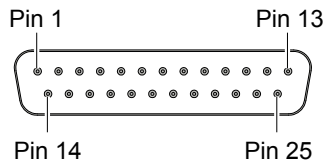
Wert	Kurzbeschreibung
0	Polyline
1	Barcode
2	Text
3	Bitmap
4	Vector Graphic
5	Point
6	Line
7	Polygon
8	Rectangle
9	Rounded Rectangle
10	Spiral
107	Laser Control
108	Set Port
109	Time Delay
110	Wati Port
111	Alignment
115	Rotary Motion
116	Linear Motion
117	XY Motion

**Hinweis!**

Beachten Sie, dass zum Ansprechen von Objekten über die Remote API Schnittstelle nicht der Typ oder der Name des Objekts, sondern dessen fortlaufende Indexnummer im Layout relevant ist.

12.1 Schnittstelle Extern I/O CON2

Für die Einbindung in einen übergeordneten Steuerungsablauf ist das Gerät mit einer I/O-Schnittstelle ausgerüstet, bei der je acht Ein- und Ausgänge z.B. über cabLase Editor frei programmiert werden können. Die Schnittstelle besitzt einen 25-poligen SUB-D-Stecker.



Schnittstelle Extern I/O

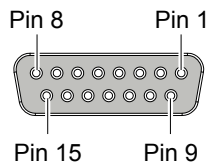
Pin	Signal	Name	Beschreibung	Aktivierung / Aktiver Zustand
1		FP24V	interne Betriebsspannung +24 V, Si T 500 mA zur Versorgung externer Verbraucher	
2	⊕→	Ready to Mark	Beschriftungsauftrag ist geladen, Gerät wartet auf Start der Beschriftung	Kontakt Pin 2 / Pin 12 geschlossen
3	⊕→	Busy	Beschriftungsauftrag läuft	Kontakt Pin 3 / Pin 12 geschlossen
4	⊕→	User Out1	frei programmierbar	Kontakt Pin 4 / Pin 12 geschlossen
5	⊕→	User Out2	frei programmierbar	Kontakt Pin 5 / Pin 12 geschlossen
6	⊕→	User Out3	frei programmierbar	Kontakt Pin 6 / Pin 12 geschlossen
7	⊕→	User Out4	frei programmierbar	Kontakt Pin 7 / Pin 12 geschlossen
8	⊕→	User Out5	frei programmierbar	Kontakt Pin 8 / Pin 12 geschlossen
9	⊕→	User Out6	frei programmierbar	Kontakt Pin 9 / Pin 12 geschlossen
10	⊕→	User Out7	frei programmierbar	Kontakt Pin 10 / Pin 12 geschlossen
11	⊕→	User Out8	frei programmierbar	Kontakt Pin 11 / Pin 12 geschlossen
12	⊕→	Common for Outputs	Gemeinsames Potenzial für alle Ausgänge	
13		GND	0 V intern	
14	⊖←	Start Process	Start des Beschriftungsvorgangs gemäß Einstellung in der Software	Zuschalten +24V zwischen Pin 14 und Pin 25
15	⊖←	Job Select	aktiv : Beschriftungsdaten werden von interner Datei geladen und können über eine Binärcodierung mit Hilfe der digitalen Eingänge aufgerufen werden ▷ Speichern von Jobfiles für den Stand-alone-Betrieb inaktiv : Beschriftungsdaten werden vom Rechner empfangen	Zuschalten +24V zwischen Pin 15 und Pin 25
16	⊕→	Error	Fehlermeldung der Steuerung	Kontakt Pin 16 / Pin 12 geschlossen
17	⊖←	User In1	frei programmierbar	+24V zwischen Pin 17 und Pin 25
18	⊖←	User In2	frei programmierbar	+24V zwischen Pin 18 und Pin 25
19	⊖←	User In3	frei programmierbar	+24V zwischen Pin 19 und Pin 25
20	⊖←	User In4	frei programmierbar	+24V zwischen Pin 20 und Pin 25
21	⊖←	User In5	frei programmierbar	+24V zwischen Pin 21 und Pin 25
22	⊖←	User In6	frei programmierbar	+24V zwischen Pin 22 und Pin 25
23	⊖←	User In7	frei programmierbar	+24V zwischen Pin 23 und Pin 25
24	⊖←	User In8	frei programmierbar	+24V zwischen Pin 24 und Pin 25
25	⊖←	Common for Inputs	Gemeinsames Potenzial für alle Eingänge, 0 V extern	Hinweis : Bei Nutzung der internen +24V (Pin 1) zum Schalten der Eingänge muss Pin 25 mit Pin 13 verbunden werden.

Tabelle 2 Pinbelegung der Schnittstelle Extern I/O

12.2 Schnittstelle Remote CON3



Über diese Schnittstelle wird die Hardware des FL+ gesteuert.

Die Schnittstelle besitzt eine 15-polige SUB-D-Buchse.

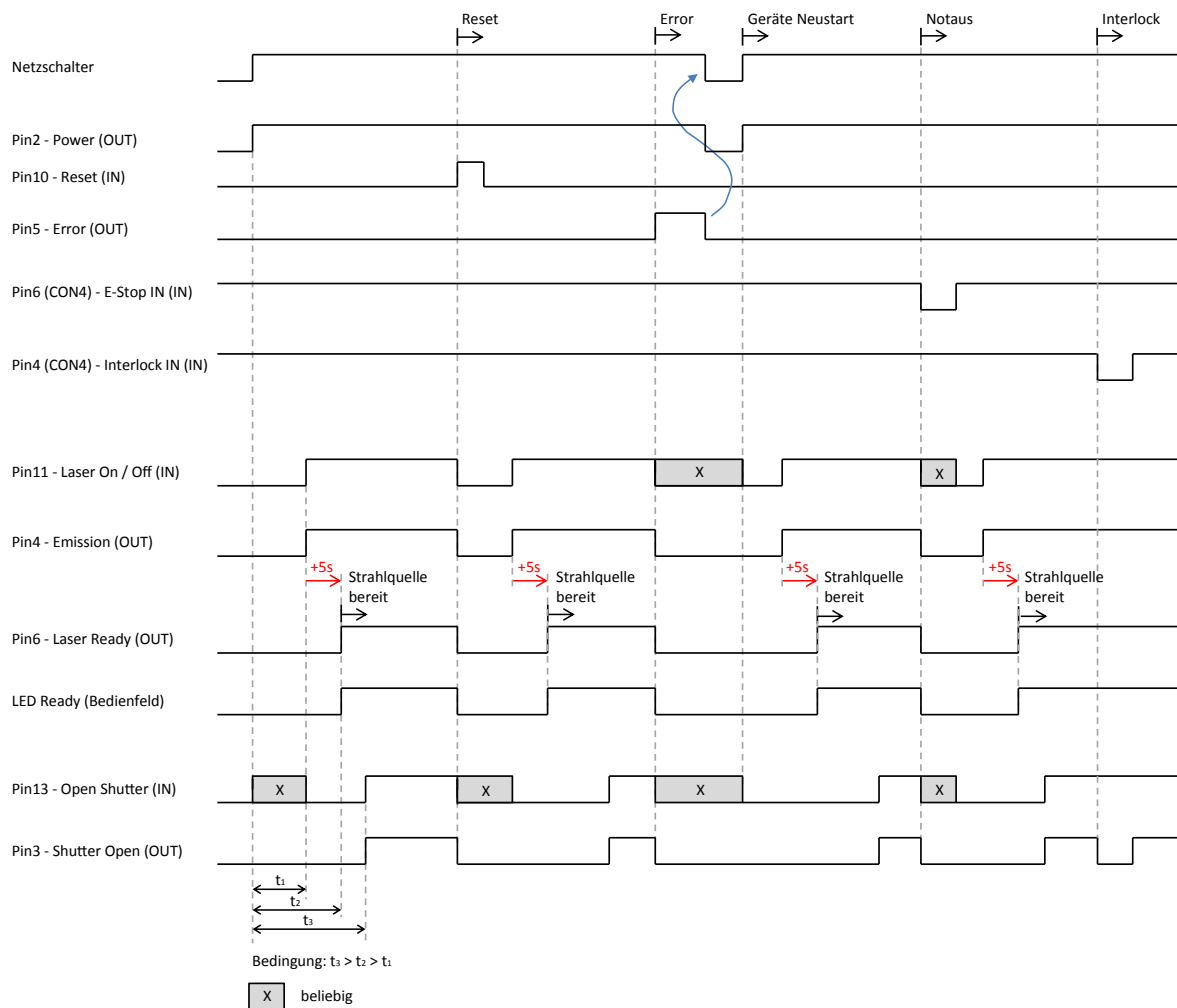


Schnittstelle Remote

Pin	Signal	Name	Beschreibung	Aktivierung / Aktiver Zustand
1		FP24V	interne Betriebsspannung +24 V, max. 500 mA zur Versorgung externer Verbraucher	
2	⊖→	Power	Netzspannung liegt an.	Kontakt zwischen Pin 2 und Pin 7 ist geschlossen
3	⊖→	Shutter Open	Shutter ist offen	Kontakt zwischen Pin 3 und Pin 7 ist offen
4	⊖→	Emission	Strahlquelle eingeschaltet und bereit zum Beschriften ► Unter den Bedingungen der Laserklasse 4 spezielle Sicherheitsvorschriften einhalten.	Kontakt zwischen Pin 4 und Pin 7 ist geschlossen
5	⊖→	Error	Steuerung ist nach dem Einschalten noch nicht bereit oder es ist ein Fehler in der Steuerung aufgetreten	Kontakt zwischen Pin 5 und Pin 7 ist geschlossen
6	⊖→	Laser Ready	Steuerung und Strahlquelle sind eingeschaltet und bereit zum Beschriften	Kontakt zwischen Pin 6 und Pin 7 ist geschlossen
7	⊖→	Common for Outputs	Gemeinsames Potenzial für alle Ausgänge	
8		GND	0V intern	
9	⊖←	Job Select Register	aktiv : User In/Out 1 bis User In/Out 4 an CON2 werden als digitale Eingänge/Ausgänge 1 bis 4 in cabLase verwendet inaktiv : User In/Out 1 bis User In/Out 8 an CON2 werden als digitale Eingänge/Ausgänge 5 bis 12 in cabLase verwendet Hinweis : Die Beschaltung von Pin 9 wird nur bei der Initialisierung des Beschriftungslasers (Einschalten oder Rücksetzen) ausgewertet! Im inaktiven Zustand können die User In 1 bis 8 zur Auswahl von Layouts, nicht jedoch zur Objektauswahl innerhalb eines Layouts verwendet werden.	aktiv, wenn +24V zwischen Pin 9 und Pin 15 anliegt
10	⊖←	Reset	Rücksetzen des Beschriftungslasers Hinweis : Der nach dem Rücksetzen erfolgende Neustart und die Initialisierung der Lasersteuerung können bis zu 30 Sekunden in Anspruch nehmen!	aktiv, wenn +24V zwischen Pin 10 und Pin 15 anliegt
11	⊖←	Laser On/ Off	aktiv : Einschalten der Strahlquelle Bedingungen zum Einschalten : NOT-AUS nicht aktiv, Shutter geschlossen inaktiv : Ausschalten der Strahlquelle	aktiv, wenn +24V zwischen Pin 11 und Pin 15 anliegt
12	⊖→	Pilot Laser is On	Pilotlaser ist eingeschaltet	Kontakt zwischen Pin 12 und Pin 7 ist geschlossen
13	⊖←	Open Shutter	Anforderung zum Öffnen des Shutters Bedingungen zum Öffnen: NOT-AUS nicht aktiv, Interlock geschlossen	aktiv, wenn +24V zwischen Pin 13 und Pin 15 anliegt

Pin	Signal	Name	Beschreibung	Aktivierung / Aktiver Zustand
14		Pilot Laser On/Off	aktiv : Einschalten des Pilotlasers Bedingungen zum Einschalten : NOT-AUS nicht aktiv, Shutter geschlossen inaktiv : Ausschalten des Pilotlasers	aktiv, wenn +24V zwischen Pin 14 und Pin 15 anliegt
15		Common for Inputs	Gemeinsames Potenzial für alle Eingänge Hinweis: Bei Nutzung der internen +24V (Pin 1) zum Schalten der Eingänge muss Pin 15 mit Pin 8 (GND) verbunden werden.	

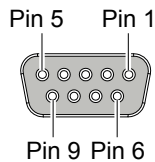
Signaldiagramm Remote-Schnittstelle



12.3 Schnittstelle Interlock CON4

Die Schnittstelle dient der Einbindung des Beschriftungslasers in externe Sicherheitskreise und zum Anschluss eines externen NOT-AUS-Schalters.

Die Schnittstelle besitzt eine 9-polige SUB-D-Buchse.



Schnittstelle Interlock / E-Stop



Hinweis!

Ohne Beschaltung der Eingänge Interlock In (Pin 4) und E-Stop In (Pin 6) kann keine Beschriftung durchgeführt werden.

Pin	Signal	Name	Beschreibung	Aktivierung / Aktiver Zustand
1		FP24V	interne Betriebsspannung +24 V, max. 500 mA	
2	⊕→	E-Stop Meldekontakt Anschluss A	Status des NOT-AUS-Relais' Gerät betriebsbereit, wenn Signal aktiv	Kontakt zwischen Pin 2 und Pin 7 ist geöffnet, wenn NOT-AUS nicht aktiviert d.h. das NOT-AUS-Relais bestromt ist
3	⊕→	Interlock Meldekontakt Anschluss A	Status des Interlock-Relais' Gerät betriebsbereit, wenn Signal aktiv	Kontakt zwischen Pin 3 und Pin 8 ist geöffnet, wenn der Sicherheitskreis geschlossen d.h. das Interlock-Relais bestromt ist
4	⊖←	Interlock IN	Relais Interlock Anschluss für Sicherheitsschalter Gerät betriebsbereit, wenn Signal aktiv	aktiv, wenn +24V zwischen Pin 4 und Pin 9 anliegt
5		GND INT	0 V intern	
6	⊖←	E-Stop IN	Relais NOT-AUS Anschluss für NOT-AUS-Schalter Gerät betriebsbereit, wenn Signal aktiv	aktiv, wenn +24V zwischen Pin 6 und Pin 9 anliegt
7	⊕→	E-Stop Meldekontakt Anschluss B	Status des NOT-AUS-Relais' (Rückleitung)	▷ Pin 2
8	⊕→	Interlock Meldekontakt Anschluss B	Status des Interlock-Relais' (Rückleitung)	▷ Pin 3
9		GND EXT	Gemeinsames GND Potential für Interlock IN und E-Stop IN	

Tabelle 3 Pinbelegung der Schnittstelle Interlock / E-Stop

13.1 Referenzdokumente

	Bezeichnung Dokument	Herkunft
1	20.015 LEC Remote API Manual.pdf	LC
2	cab_ma_flplus_400_de.pdf	cab

13.2 Revisionierung der Beispielprogramme

Revision	Änderung	Datum
FL+ Remote API Library_160315.zip	Auslieferungszustand	15.03.15

13.3 Kontakt

Name	Firma	Funktion	email
Mehmet Seker	cab GmbH & Co. KG	Service / Support	m.seker@cab.de
Martin Föll	cab GmbH & Co. KG	Service / Support	m.foell@cab.de
Hans Löhner	cab GmbH & Co. KG	Software / Support	h.loehner@cab.de