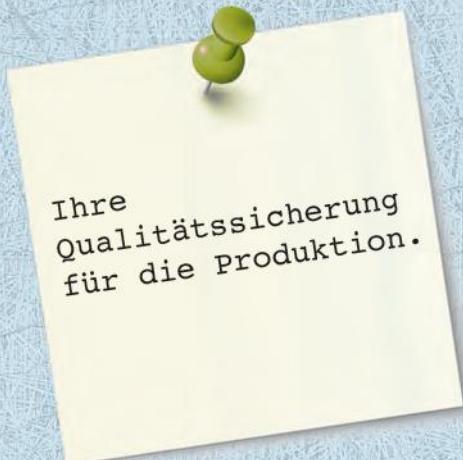


R.IZO[®] - einfach eine
gute Entscheidung.



R.IZO by light lässt sich auf verschiedene Weise ansprechen. Neben dem Betrieb mittels Barcode-Scanner, ist auch die Übertragung direkt über einen angeschlossenen PC möglich.

Grundsätzlich unterscheidet R.IZO by light die Eingabe auf Basis der RS-232 Funktionalität und der Eingabe auf Basis der CAN-Bus Befehle.

Die beiden Funktionen unterscheiden sich im Wesentlichen über die Eingabesprache der Befehle. Für die Verbindung mittels RS-232, können die Befehle im Klartext eingegeben werden, **im CAN-Bus** ist die Eingabe im **Hex-Format** vorgegeben.

Für beide Anschlussarten liegen die entsprechenden Adapter vor und können unter den folgenden Teilenummern bestellt werden:

111401 – FTDI-Bus → USB auf RS-232
101073 – CAN-BUS → USB auf CAN

Im nachfolgenden sind die möglichen Befehle aufgeführt, wobei sich die Programmierung an der Abfolge und den Befehlssätzen von **Terminalprogrammen** orientiert. Als Beispiel sei hier das Open-Source-Produkt **Tera Term** genannt.

Mit Tera Term lassen sich die Befehle als Einzelschritt und als vorher programmiertes Makro an R.IZO by light übermitteln. Gleichzeitig dient es auch als Kommunikationsüberwachung, da sämtliche gesendeten Befehle, ein- und ausgehend von R.IZO sichtbar werden.

Um eine standardisierte Befehlsfolge zu ermöglichen, kann als Beispiel in jedem Modul eine spezifische Programmierung also „Kennungen“ hinterlegt werden.

Die Adressierung eines Moduls könnte hierbei wie folgt dargestellt und auf weitere Module angewendet werden:

Adresse		Farbkürzel		Würfelanzahl
10001		G(rün)		1
10001		G(rün)		2
10002		G(rün)		1
10002		G(rün)		2

Somit würde ein Leuchtmódul mit folgendem Code programmiert werden „10001G1“.

Mit dieser dargestellten Adressierung kann nun das Leuchtmódul eigenständig aufgerufen werden.

Hinweis:

Die oben erwähnten Inhalte, sofern nicht ab Werk vorhanden, müssen manuell in die gewünschten Module programmiert werden.

Nutzen Sie hierzu die Barcodekarte und passende Codes um die Einzelprogrammierung vorzunehmen!

Vor der Übertragung von Befehlen, muss die Com-Schnittstelle mit den folgenden Parametern initialisiert werden:

Baudrate:	9600
Data:	8 Bit
Parity:	Even
Stop:	1 Bit
Flow control:	none

Die Übertragung kann mittels Null-Modemkabel oder dem zu Anfang beschriebenen FTDI-BUS erfolgen.

Zur Steuerung der Leuchtmodule können die vorherig erwähnten „Einzel“-Leuchtmodule(10001G1) oder eine komplette Greifreihenfolge (siehe Technische Dokumentation) aufgerufen werden.

Für die Kommunikation stehen folgende Meldungen als auch Befehle zur Verfügung:

Meldungen der Leuchtmodule über den seriellen RS-232:

- **CANID064** - FAILGRIP - Fehlgriff detektiert - ein inaktives Modul hat den Eingriff registriert und sendet diese Information. Gleichzeitig ist rotes Blinken erkennbar.
- **CANID065** - NEXT STEP - Einleiten des Folgeschrittes in der Arbeitsabfolge - sofern der Auftrag mehr als einen Arbeitsschritt enthält.
- **CANID066** - LAST STEP - Letzter Schritt der programmierten Arbeitsfolge und Meldung, das keine weiteren aktiven Module mehr vorhanden sind.

Befehle zur Programmierung der Leuchtmodule:

- **\$RIZO+B%ABRT** – Abbruch – Abbruch einer aktiven Arbeitsreihenfolge und Abschalten des Glimmmodus im Leuchtmkul. Dieser ist nach dem Abwählen des aktiven Modul aktiv. Wobei hier die Abwahl des Moduls bedeutet, dass ein Eingriff des Mitarbeiters durch die Sensorik erkannt wurde.
- **\$RIZO+B%OKAY** – Übernehmen – Übernahme von einer neuen Programmierung oder der Änderung einer bestehenden Arbeitsabfolge. Dieser Befehl kann auch zur Wiederholung der letzten aktiven Arbeitsreihenfolge eingesetzt werden.
- **\$RIZO+B-PRESTP** – Vorhergehender Schritt – Zurückspringen innerhalb der Arbeitsfolge, zum vorhergehenden Leuchtmkul
- **\$RIZO+B-NXTSTP** – Nachfolgender Schritt – Vorspringen innerhalb der Arbeitsfolge, zum nachfolgenden Leuchtmkul
- **\$RIZO+B-SEQOFF** – Abschalten der Greifdetektion
- **\$RIZO+B-SEQUON-** Einschalten der Greifdetektion

Für ein besseres Verständnis der Einzelaufrufe von Leuchtmotulen ist im Folgenden eine Beispielprogrammierung aufgeführt welche auf Basis des vorherig erwähnten Tera Term Softwaremoduls geschrieben ist.

Hinweis:

Um das folgende Makrobeispiel zu nutzen sollte folgende Programmierung vorgenommen werden:

*1 Leuchtmotul mit Barcode „10001G1“ programmieren
Also Grün in Stückzahl 1.*

*Alle Leuchtmotule mit Barcode „Rot“ programmieren
Also rote Farbe ohne Stückzahl.*

Allgemeine Befehle zur Programmierung der Leuchtmotule:

- **send** - Sendebefehl zur Übermittlung von Einzelbefehlen
- **#13** - Enter
- **wait** - Warten auf bestimmte Befehle oder Meldungen
- **mpause** - Kurzpause nach gesendeten Befehlen – zwingend notwendig!
- **pause** - Frei einzusetzende Pause, um Zustände über einen festgelegten Zeitraum hinweg zu erhalten.
- **End** - Ende der Programmierung
- **If/goto** - Schleifenprogrammierung innerhalb des Programms

Beispielmakro für das Anleuchten eines einzelnen Moduls und aktiver Fehlgriffüberwachung.

```
; Start des Makro
; Einzelmodul
;
; File: Makro 3
; Beschreibung: Aufleuchten eines Moduls mit Würfel 1 -
; Abwahl möglich. Bei Fehlgriff leuchten alle anderen Module
; in Rot und bleiben für 5 Sekunden stehen - in dieser Zeit
; ist keine Abwahl möglich. Nach 5 Sekunden erfolgt der Reset
; danach ist der normale Ablauf möglich

; CANID064 = FEHLGRIFF
; CANID065 = NEXT STEP
; CANID066 = LAST STEP

send '$RIZO+B%ABRT' #13
; Bus reset mit Abbruch

send '10001G1' #13
; Anwahl an Leuchtmittel: Modul 10001, Grün, Würfel 1

wait 'CANID066' 'CANID064'
; Warten auf Rückmeldung des Moduls -
; Last Step oder Fehlgriff

if result=1 goto abort1
if result=2 goto seqoff1
; Auswertung der Modulrückmeldung
; Result 1 entspricht im String 1 dem Last Step / Weiter
; Result 2 entspricht im String 2 dem Fehlgriff

:abort1
; Schleife für das Weiterschalten nach der korrekten Entnahme

send '$RIZO+B%ABRT' #13
; Bus reset mit Abbruch

goto END
; Verweis auf Programmende
```

```
:seqoff1
; Schleife aufgrund des Eingriffs in den falschen Behälter

send '$RIZO+B-SEQOFF' #13
; Abschalten der Sensorik, Eingreifen möglich

mpause 250
; Pause wegen Befehslänge

send 'Rot' #13
; Sende Warnlicht in Rot an alle Module

send '+10001G1' #13
; Zusätzlich zu den roten Leuchtmotulen - Anwahl des
; Leuchtmotuls mit „+“: Modul 10001, Grün, Würfel 1

pause 5
send '$RIZO+B%ABRT' #13
; Bus reset mit Abbruch

send '+10001G1' #13
; Anwahl an Leuchtmotul: Modul 10001, Grün, Würfel 1

send '$RIZO+B-SEQUON' #13
; Einschalten der Sensorik

mpause 250
; Pause wegen Befehslänge

wait 'CANID066' 'CANID064'
; Warten auf Rückmeldung des Moduls - Last Step oder
; Fehlgriff

if result=1 goto abort1
if result=2 goto seqoff1
; Auswertung der Modulrückmeldung - Fehlerschleife
; Result 1 entspricht im String 1 dem Last Step / Weiter
; Result 2 ent spricht im String 2 dem Fehlgriff

pause 1
send '$RIZO+B%ABRT' #13
; Bus reset mit Abbruch

:END
; Ende des Makros
```

Notizen:

Motto:lean!

Karlsruher Str. 21
78048 Villingen-Schwenningen
Tel: 07721/94429-42

info@mottolean.de
www.mottolean.de