

Handbuch

**LioN-P μ DCU
Distributed Control Unit**

**LioN-P Digital
LDMicro
0980 ESL 393-121-DCU1
0980 ESL 390-121-DCU1
 μ DCU Programming Manual**

Inhalt

1 Zu diesem Handbuch	5
1.1 Allgemeine Informationen	5
1.2 Versionsinformationen	6
2 Einführung	7
3 Begriffserläuterungen	8
4 Zyklische Daten und Parameter	9
4.1 Consuming-Daten	9
4.2 Producing-Daten	9
4.3 Datenaustausch	10
4.3.1 PROFINET	10
4.3.2 EtherNet/IP	11
4.3.3 EtherCAT	13
4.4 Modulparameter	14
4.4.1 DCU-Start-Parameter	14
5 Technische Daten	15
5.1 DCU-/Force-Mode-LED-Beschreibung	15
5.2 Port-Typen	16
5.2.1 0980 ESL 393-121-DCU1 (LioN-P 8DI8DO mit DCU)	16
5.2.2 0980 ESL 390-121-DCU1 (LioN-P 16DIO mit DCU)	17
5.3 Elektrische Spezifikationen	17

6	Details zur DCU-Programmierung	18
6.1	Beschränkungen in Bezug auf μ DCU/LDMicro	18
6.2	LDMicro	19
6.2.1	Einführung: LDMicro	19
6.2.2	Dateitypen	20
6.2.3	Datentypen	21
6.2.4	Namenskonventionen	21
6.2.4.1	LDMicro-Konventionen	21
6.2.4.2	Konventionen in Bezug auf LioN-P μ DCU	21
6.2.5	Verfügbare Daten	22
6.2.6	Physische Ein- und Ausgänge	23
6.2.7	Direktzugriff auf zyklische Bits	23
6.2.8	Lesen und Bearbeiten von Consuming- und Producing-Daten nach Kanal	24
6.2.9	Datenaustausch	25
7	DCU-Web-Interface	26
7.1	Übersicht über die Elemente des Web-Interface	26
7.2	Benutzername und Passwort	27
7.3	DCU-Status	28
7.4	Hochladen eines Programmes in die DCU	28
7.5	Programminformationen	29
7.6	Autostart	29
7.7	Benutzerdefinierte Zuordnung	29
7.7.1	Vornehmen der Zuordnung über den Zuordnungsdialog	30
7.7.2	Manuelle Erstellung einer Zuordnungsdatei	31
7.8	Variablen-Forcing	32
8	Batch-Upload von DCU-Programm und Zuordnungsdateien	34

8.1 POST-Anforderung zum Hochladen von Dateien	34
8.2 Verwendung des Perl-Skripts	35
9 Standardmäßige JSON-Modulinformationen	36
9.1 Beispiel für eine JSON-Antwort	36
9.2 JSON-Antwort-Objektstruktur	37
9.3 JSON-Antwort	38

1 Zu diesem Handbuch

1.1 Allgemeine Informationen

Bitte lesen Sie die Erklärungen und Einrichtungsanleitungen in diesem Handbuch sorgfältig, bevor Sie die LioN-P-Module in Betrieb nehmen. Bewahren Sie dieses Handbuch an einem Ort auf, der für alle Benutzer zugänglich ist.

Die Texte, Abbildungen, Diagramme und Beispiele dienen zur Erläuterung der Funktionalität der LioN-P- μ DCU-Module.

Bei weitergehenden Fragen zur Installation und Inbetriebnahme der Geräte sprechen Sie uns bitte an.

Belden Deutschland GmbH
– Lumberg Automation™ –
Im Gewerbepark 2
D-58579 Schalksmühle
Deutschland
support-automation@belden.com
www.lumberg-automation.com

Belden Deutschland GmbH – Lumberg Automation™ – behält sich vor, jederzeit technische Änderungen oder Änderungen dieses Handbuches ohne besondere Hinweise vorzunehmen.

1.2 Versionsinformationen

Index	Erstellt	Geändert
Versionsnummer	Version 1.0	Version 1.1
Datum		Oktober 2017
Name/Abteilung		

Index	Geändert	Geändert
Versionsnummer	Version 1.2	
Datum	Januar 2020	
Name/Abteilung	JGA/R&D	

Tabelle 1: Übersicht der Handbuch-Revisionen

2 Einführung

Die LioN-P μ DCU ist ein Multiprotokoll-Feldbus-Slave-Modul, das auf dem Modul 0980 ESL 393-121/0980 ESL 390-121 basiert. Die LioN-P μ DCU stellt alle Funktionen des Basismoduls bereit, verfügt jedoch über eine integrierte programmierbare Logikeinheit. Diese Einheit kann mit einem kleinen externen Tool, d. h. LDMicro, erstellte Programme ausführen. Diese Programme werden mittels der Ladder-Logic-Methode (Kontaktplanmethode) erstellt und als „DCU-Programme“ bezeichnet.

Dies bietet dem Benutzer die Möglichkeit, eine zusätzliche Steuerungslogik hinzuzufügen, die direkt im Slave-Modul selbst gespeichert wird und unabhängig vom Feldbus oder der SPS funktioniert. Dies reicht von einfachen booleschen Operationen für Eingänge und Ausgänge bis hin zu vollständig autonomen Programmen (ohne SPS).

Eine SPS nimmt das Modul als normales Slave-Modul mit 8-Bit-Ausgangsdaten (Consuming) und 16-Bit-Eingangsdaten (Producing) von zyklischen Daten wahr.

Vom DCU-Programm verwendete Eingänge/Ausgänge sind nicht mehr direkt über die SPS steuerbar, können jedoch für die Kommunikation mit der SPS verwendet werden, da die entsprechenden zyklischen Bits vom DCU-Programm gelesen und geschrieben werden können.

3 Begriffserläuterungen

DCU	Distributed Control Unit (dezentrale Steuerung). Eine programmierbare Logik für Feldbus-Module.
DCU-Programm	Ein vom Benutzer geschriebenes Programm, das von der DCU (Modul) ausgeführt werden kann.
LDMicro	Eine Windows-Software zum Erstellen von DCU-Programmen mittels der Kontaktplanmethode („Ladder Logic“).
Port	Physische Buchse an einem E/A-Modul (M12/M8, 5-polig), z. B. mit dem Namen X1, X6. Die Module können über 8 Ports verfügen.
Kanal	Ein Pin an einem Port. Definiert durch einen Port und einen Kanal-Buchstaben, z. B. X1A, X4B. Jeder Port kann bis zu 2 Kanäle aufweisen. Die Kanäle befinden sich für jeden Port an Pin 4 (A) und Pin 2 (B).
Consuming-Daten	Zyklische Daten, die durch einen Feldbus von einer SPS an das E/A-Modul übertragen werden, z. B. zur Steuerung eines Ausgangs.
Producing-Daten	Zyklische Daten, die durch einen Feldbus von einem E/A-Modul an die SPS übertragen werden, z. B. zum Lesen eines Eingangs.

4 Zyklische Daten und Parameter

Festgelegt für 0980 ESL 393-121-DCU1 und Grundeinstellungen für 0980 ESL 390-121-DCU1.

4.1 Consuming-Daten

Ausgang	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte n	4B	4A	3B	3A	2B	2A	1B	1A
Byte n + 1	8B	8A	7B	7A	6B	6A	5B	5A

Hier gilt Folgendes:

- ▶ 1A ... 8A: Ausgang von Kanal A (Kontaktstift 4) der M12-Steckverbindungen 1 bis 8.
- ▶ 1B ... 8B: Ausgang von Kanal B (Kontaktstift 2) der M12-Steckverbindungen 1 bis 8.

4.2 Producing-Daten

Eingang	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte n	4B	4A	3B	3A	2B	2A	1B	1A
Byte n + 1	8B	8A	7B	7A	6B	6A	5B	5A

- ▶ 1A ... 8A: Aktueller Status von Kanal A (Kontaktstift 4) der M12-Steckverbindungen 1 bis 8.
- ▶ 1B ... 8B: Aktueller Status von Kanal B (Kontaktstift 2) der M12-Steckverbindungen 1 bis 8.



Achtung: Abhängig vom ausgewählten Feldbus kann das Modul zusätzliche zyklische Daten-Bytes bereitstellen (z. B. Diagnose-Informationen). Informationen zu feldbuspezifischen Details finden Sie im Handbuch des Basismoduls.

4.3 Datenaustausch

Nur in 0980 ESL 390-121-DCU1 (16-DIO-Universal-DCU) verfügbar

- ▶ Das Modul stellt zusätzliche zyklische Daten für den alleinigen Zweck des Datenaustauschs zwischen einer SPS und dem DCU-Programm bereit. Das DCU-Programm kann beispielsweise Befehle und Daten von der SPS entgegennehmen und mit Ausführungsergebnissen antworten.
- ▶ Der Datenaustauschbereich umfasst 16 Bit plus 8 Wörter (als 16-Bit-Ganzzahl mit Vorzeichen) für jede Richtung.
- ▶ Die Datenaustausch-Bits können über die YEn-Bit-Variable in LDMicro geschrieben werden.
- ▶ Die Datenaustausch-Bits können über die XEn-Bit-Variable in LDMicro gelesen werden.
- ▶ Die mit EIn und EOn gekennzeichneten Integer-Variablen ermöglichen das Schreiben und Lesen von Austauschdatenwörtern.

4.3.1 PROFINET

Geräteübersicht							
Baugruppe	...	Baugr...	Steck...	E-Adresse	A-Adres...	Typ	
▼ DCU16 - 0980-ESL-390-121...		0	0			0980 ESL 390-121...	
▶ PN-IO		0	0 X1			0980-ESL-390-121...	
8 DI/8 DO-1,6A with DCU1_1		0	1	7	7	8 DI/8 DO-1,6A wit...	
16 Bit I/O DCU Extension_1		0	2	20...21	20...21	16 Bit I/O DCU Exte...	
8 INT I/O DCU Extension_1		0	3	256...271	256...271	8 INT I/O DCU Exten...	

Abb. 1: DCU-Austauschbereich im TIA-Portal

In PROFINET besteht der Austauschbereich aus 2 zusätzlichen Steckplätzen (2 und 3). Steckplatz 2 enthält die 16-Bit-Austauschdaten und Steckplatz 3 die 16-Byte-Austauschwörter.

4.3.2 EtherNet/IP

Die Austauschdaten werden innerhalb der zyklischen EtherNet/IP-Daten bereitgestellt. Das Format dieser Daten kann sich in Abhängigkeit von der zum jeweiligen Zeitpunkt ausgewählten Auslegung (16DI/DO, 8DI/8DI, 16DI usw.) ändern.

Die folgenden zyklischen Daten gelten für die 16DI/16DO-Auslegung mit DCU. Weitere Informationen finden Sie im Handbuch für LioN-P EtherNet/IP.

Byte	Funktion
0	Eingangsdaten
1	Eingangsdaten
2	Allgemeine Diagnose
3	Sensordiagnose
4	Reserviert
5	Aktuatordiagnose 1
6	Aktuatordiagnose 2
7	DCU-Bit-Austausch-Byte 1
8	DCU-Bit-Austausch-Byte 2
9	DCU-Integer-Austausch 1 MSB
10	DCU-Integer-Austausch 1 LSB
11	DCU-Integer-Austausch 2 MSB
12	DCU-Integer-Austausch 2 LSB
:	:
24	DCU-Integer-Austausch 8 MSB
25	DCU-Integer-Austausch 8 LSB

Tabelle 2: Zyklische Producing-Daten

Byte	Funktion
0	Ausgangsdaten
1	Ausgangsdaten
2	DCU-Bit-Austausch-Byte 1
3	DCU-Bit-Austausch-Byte 2
4	DCU-Integer-Austausch 1 MSB
5	DCU-Integer-Austausch 1 LSB
6	DCU-Integer-Austausch 2 MSB
7	DCU-Integer-Austausch 2 LSB
:	:
20	DCU-Integer-Austausch 8 MSB
21	DCU-Integer-Austausch 8 LSB

Tabelle 3: Zyklische Consuming-Daten

4.3.3 EtherCAT

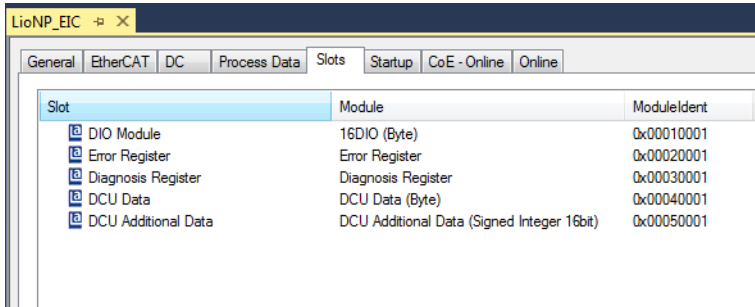


Abb. 2: DCU-Austauschbereich-Steckplätze in TwinCAT3

Name	Online	Type	Size	> Addr...	In/Out
Diagnosis Register		UDINT	4.0	42.0	Input
DCU Inputs 0..7		USINT	1.0	46.0	Input
DCU Inputs 8..15		USINT	1.0	47.0	Input
DCU Additional Input 0		INT	2.0	48.0	Input
DCU Additional Input 1		INT	2.0	50.0	Input
DCU Additional Input 2		INT	2.0	52.0	Input
DCU Additional Input 3		INT	2.0	54.0	Input
DCU Additional Input 4		INT	2.0	56.0	Input
DCU Additional Input 5		INT	2.0	58.0	Input
DCU Additional Input 6		INT	2.0	60.0	Input
DCU Additional Input 7		INT	2.0	62.0	Input
WcState		BIT	0.1	1522.1	Input
InputToggle		BIT	0.1	1524.1	Input

Abb. 3: DCU-Austauschvariablen in TwinCAT3

In EtherCAT sind die 2 Austauschbereiche als zusätzliche Steckplätze organisiert. Die 8 Integer-Werte (Ganzzahlen) mit Vorzeichen werden direkt als Variablen vom Typ INT mit einer Größe von 2 angezeigt.

4.4 Modulparameter

Die μ DCU verfügt über 1 zusätzlichen SPS-Parameter, der das Startverhalten der DCU steuert.

4.4.1 DCU-Start-Parameter

Disabled (Deaktiviert)	Die DCU startet mit dem Status DISABLED.
Lock (Gesperrt)	Die DCU ist deaktiviert und kann nicht über das Web-Interface gestartet werden.
Run (In Betrieb)	Die DCU startet mit dem Status RUN und führt das DCU-Programm aus, SOFERN ein gültiges Programm geladen ist.

The screenshot shows the TIA-Portal configuration window for the DCU. The left sidebar contains a tree view with 'Baugruppenparameter' selected. The main area is titled 'Baugruppenparameter' and contains two sections: 'General Parameters' and 'Fail Safe Configuration'. In the 'General Parameters' section, the 'DCU Startup' dropdown menu is open, showing four options: 'Disabled (can be enabled by web interface)', 'Locked (disabled and cannot be enabled by web interface)', 'Disabled (can be enabled by web interface)', and 'RUN (enabled and module starts in RUN mode)'. The 'Fail Safe Configuration' section contains eight 'Fail Safe Value' settings for ports 5, 6, 7, and 8, each with 'Ch.A' and 'Ch.B' sub-parameters, all set to 'Set Low'.

Abb. 4: DCU-Start-Parameter im TIA-Portal

5 Technische Daten

5.1 DCU-/Force-Mode-LED-Beschreibung

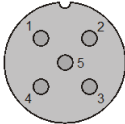
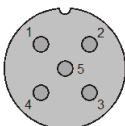
Die DCU-/FM-LED gibt den DCU- und Force-Mode-Status an.

LED-Farbe	Bedeutung
Aus	DCU deaktiviert/Force Mode nicht aktiv
Blaues Blinken	DCU in Betrieb
Blaues Leuchten	DCU angehalten
Rotes Leuchten	DCU-Fehler
Blaues/rotes Blinken	Force Mode EIN

5.2 Port-Typen

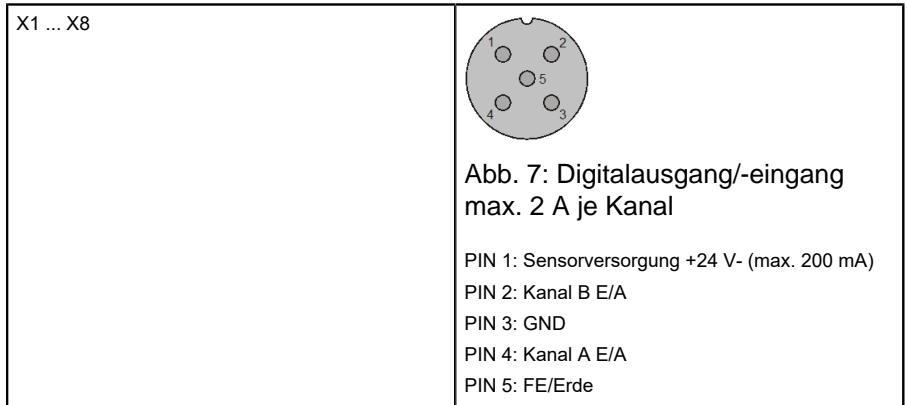
5.2.1 0980 ESL 393-121-DCU1 (LioN-P 8DI8DO mit DCU)

Die μ DCU verfügt über 8 Digitaleingänge und 8 Digitalausgänge, die in 8 Ports mit jeweils 2 Kanälen organisiert sind. Weitere Details finden Sie im Datenblatt und Handbuch zu 0980 ESL 393-121.

X1 ... X4	 <p>Abb. 5: Digitaleingang</p> <p>PIN 1: Sensorversorgung +24 V- (max. 200 mA) PIN 2: Eingangskanal B PIN 3: GND PIN 4: Eingangskanal A PIN 5: FE/Erde</p>
X5 ... X8	 <p>Abb. 6: Digitalausgang/max. 2 A</p> <p>PIN 1: nicht angeschlossen PIN 2: Ausgangskanal B PIN 3: GND PIN 4: Ausgangskanal A PIN 5: FE/Erde</p>

5.2.2 0980 ESL 390-121-DCU1 (LioN-P 16DIO mit DCU)

Dieses Modul verfügt über 16 Universal-Digitalkanäle, die in 8 Ports mit jeweils 2 Kanälen organisiert sind. Jeder Kanal kann als Digitaleingang oder Digitalausgang verwendet werden.



5.3 Elektrische Spezifikationen

Diesbezügliche Informationen finden Sie auf dem Datenblatt zu 0980 ESL 393-121-DCU1 oder 0980 ESL 930-121-DCU1.

6 Details zur DCU-Programmierung

6.1 Beschränkungen in Bezug auf μDCU/ LDMicro

Max. Anzahl von Rungs	99
Max. Anzahl von Bits	99
Max. Anzahl von Integer-Werten	99
Max. Anzahl von Zeilen (kompiliertes Programm)	4096
Durchschnittliche μDCU-Zykluszeit	10 ms

6.2 LDMicro

Open-Source-Kontaktplan-Programmierungs-Tool:

Download von LDMicro unter: <http://cq.cx/ladder.pl#dl>

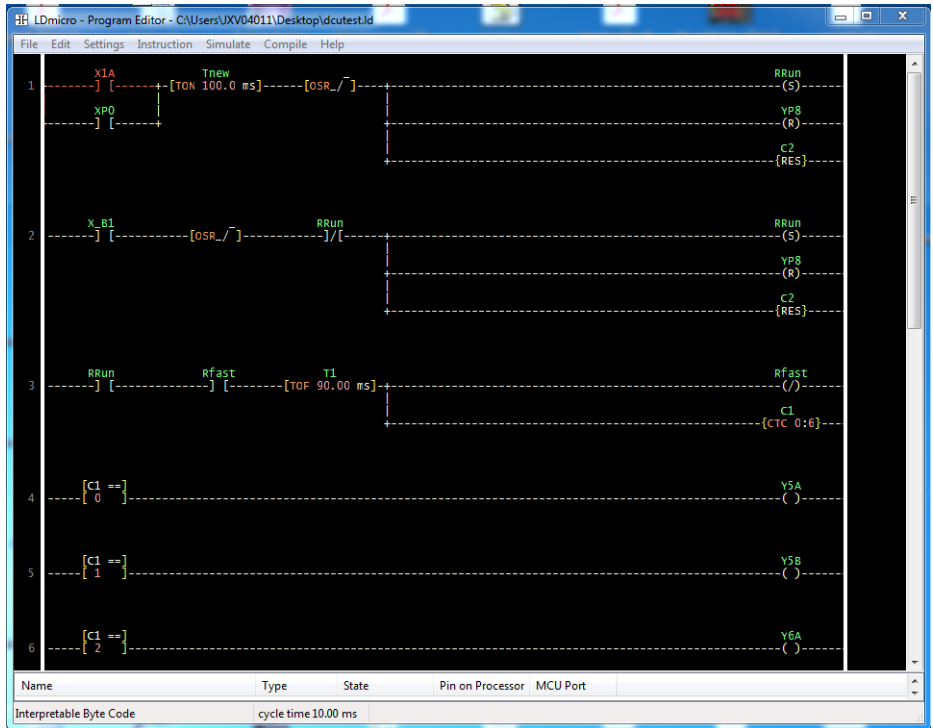


Abb. 8: LDMicro-Benutzeroberfläche

6.2.1 Einführung: LDMicro

Mit LDMicro kann der Benutzer Programme im Stil eines Kontaktplans („Ladder Diagram“) gemäß EN 61131-3 erstellen. Hier sind alle Elemente des Programms an horizontalen Linien („Rungs“) angeordnet. Rungs werden immer von links nach rechts ohne eine garantierte Rung-Reihenfolge ausgeführt. Dieses Konzept ist an verdrahtete Relaisschaltungen angelehnt.

LDMicro stellt umfassende Anweisungen bereit, z. B. zu:

- ▶ Bit-Operationen, z. B. Kontakte, Spulen, Setzen/Rücksetzen
- ▶ Flankenerkennung
- ▶ Zeitschalter und Aktivieren/Deaktivieren von Verzögerungen
- ▶ Vor-/Rückwärtszähler/zirkulierender Zähler
- ▶ Arithmetische Operationen (16 Bit, mit Vorzeichen)

Mit LDMicro erstellte DCU-Programme können:

- ▶ Jeden Eingang und Ausgang des Moduls steuern
- ▶ Auf Diagnose-Ereignisse (Kurzschluss, Unterspannung usw.) reagieren
- ▶ Mit einer verbundenen SPS kommunizieren
- ▶ Informationen innerhalb des Netzes weitergeben

6.2.2 Dateitypen

Programmdateien für LDMicro enthalten „.ld“ im Namen. Diese Dateien können über die LDMicro-Anwendung geladen, bearbeitet und gespeichert werden.

Um ein Programm für die DCU zu kompilieren, wählen Sie zunächst den korrekten Zieltyp unter **Settings > Microcontroller > Interpretable Bytecode** (Einstellungen -> Microcontroller -> Interpretierbarer Byte-Code) aus.

Außerdem besteht die Möglichkeit, die Zykluszeit festzulegen (**Settings > MCU Parameters**) (Einstellungen -> MCU-Parameter). Eine Zykluszeit von 10 ms oder höher wird empfohlen.

Wählen Sie anschließend im Menü **Compile > Compile as...** (Kompilieren -> Kompilieren als...). Wählen Sie einen Speicherort und den Namen, unter dem das kompilierte Programm gespeichert werden soll. Das Programm wird dann kompiliert. Das Ergebnis ist eine .int-Datei.

Diese Datei kann nun in die DCU hochgeladen werden.

6.2.3 Datentypen

LDMicro kennt die folgenden Datentypen:

Bit	0 oder 1
Int	16-Bit-Ganzzahl mit Vorzeichen (-32768 bis +32767)
T	Zeitschalter
C	Zähler

6.2.4 Namenskonventionen

6.2.4.1 LDMicro-Konventionen

Es gibt 3 Typen von Bits mit einer obligatorischen Namenskonvention:

Typ	Konvention	Beispiel
Eingangs-Bit	Muss mit „X“ beginnen	X1A, X5P
Ausgangs-Bit	Muss mit „Y“ beginnen	Y2B, Y3P
Internes Relais	Muss mit „R“ beginnen	R1, RRun, RStart

6.2.4.2 Konventionen in Bezug auf Lion-P µDCU

Typ	Konvention	Beispiel
Eingangsdaten des physischen E/A	X gefolgt von Port-Nummer und Kanal	X1A, X5B
Ausgangsdaten des physischen E/A	Y gefolgt von Port-Nummer und Kanal	Y2B, Y7A
Zyklische Daten an die SPS	Y gefolgt von „P“ und Bit-Nummer	YP5, YP15
Zyklische Daten von der SPS	X gefolgt von „P“ und Bit-Nummer	XP0, XP6
Spezielle Bits	„X“ oder „Y“ gefolgt von „_“ (Unterstrich) und einem Namen	X_DIA, Y_STOP
Integer-Werte für E/A	IN oder OUT gefolgt von Byte-Nummer	IN1, IN2, OUT1, OUT2
Integer-Werte für spezielle Informationen	„_“ (Unterstrich) gefolgt von einem Namen	_SCS, _CE1

6.2.5 Verfügbare Daten

Diese Daten sind direkt in LDMicro-Programmen verfügbar. Benennen Sie eine Bit- oder Integer-Variable in LDMicro gemäß der folgenden Liste.

Symbol	Richtung	Typ	Beschreibung
Grundlegende Eingangs-/Ausgangsdaten			
Xn[A/B]	Eingang	Bit	Digitaleingangsstatus von Port n (1–8) lesen. Kanal A oder B
Yn[A/B]	Ausgang	Bit	Digitalausgangsstatus in Port n (1–8) schreiben. Kanal A oder B
OUT[0/1]	Ausgang	Int	8 Ausgangsstatus von INT schreiben. (0 = X1–X4, 1 = X5–X8)
IN[0/1]	Eingang	Int	8 Eingangsstatus als INT lesen. (0 = X1–X4, 1 = X5–X8)
Datenaustausch (mit SPS)			
XPh	Eingang	Bit	Consuming-Bit aus SPS lesen. n = 0–15
YPn	Ausgang	Bit	Producing-Bit in SPS schreiben. n = 0–15
XCn[A/B]	Eingang	Bit	Consuming-Daten aus SPS für Port n (1–8) lesen. Kanal A oder B
YPn[A/B]	Eingang	Bit	Producing-Daten in SPS für Port n (1–8) schreiben. Kanal A oder B
XEn [*]	Eingang	Bit	Datenaustausch-Bit n (0–15) lesen
YEn [*]	Ausgang	Bit	Datenaustausch-Bit n (0–15) schreiben
EIn [*]	Eingang	Int	Datenaustauschwert von SPS. n = 0–7
EOn [*]	Ausgang	Int	Datenaustauschwert an SPS. n = 0–7
Diagnose-Informationen			
X_DIA	Eingang	Bit	Diagnose-Master-Bit
X_SCS	Eingang	Bit	Sensordiagnose-Bit
X_SCA	Eingang	Bit	Aktuatordiagnose-Bit
X_LVS	Eingang	Bit	Fehler Sensorversorgung
X_LVA	Eingang	Bit	Fehler Aktuatorversorgung
X_COMM	Eingang	Bit	Zyklische Verbindung zu SPS hergestellt
_SCS	Eingang	Int	Informationen zu Sensorkurzschluss je Port
_CE1	Eingang	Int	Kanalfehler LSB
_CE2	Eingang	Int	Kanalfehler MSB

DCU-Steuerung			
Y_STOP	Ausgang	Bit	Veranlasst die DCU, zu stoppen
Y_DIS	Ausgang	Bit	Veranlasst die DCU, sich selbst zu deaktivieren (DISABLE)
Speziell			
_Pn	Ausgang	Int	Zu veröffentlichende Daten. n = 0–31
_MSG	Ausgang	Int	Meldung mit entsprechender Nummer auf dem grafischen Web-Interface anzeigen
X_Bn	Eingang	Bit	Virtuelle Taste auf grafischem Web-Interface gewählt. n = Tastennummer 1–10
X_FIRST	Eingang	Bit	Wird nur für den 1. Programmzyklus nach Einschalten oder Rücksetzen gesetzt.

* Nur in 0980 ESL 390-121-DCU1 (16 DIO Universal) verfügbar

6.2.6 Physische Ein- und Ausgänge

Die mit dem Symbol X_{nA}/X_{nB} gekennzeichnete Variable bietet dem DCU-Programm die Möglichkeit, den entsprechenden physischen Eingang direkt zu lesen. Ein mit diesem Symbol gekennzeichneteter *Kontakt* würde bei Kurzschließen des entsprechenden Eingangspins mit 24 V- (z. B. Pin 1) als *geschlossen* interpretiert.

Die mit dem Symbol Y_{nA}/Y_{nB} gekennzeichnete Variable kann für die direkte Steuerung eines physischen Digitalausgangs verwendet werden. Eine *Spule* (auch „Coil“), die mit diesem Symbol gekennzeichnet, ist, würde den entsprechenden Ausgangspin aktivieren, der auf 24 V- gesetzt ist.

Eingänge/Ausgänge, die in einem DCU-Program verwendet werden, werden von den entsprechenden zyklischen Daten und der SPS getrennt. Allerdings ist das Lesen und Bearbeiten dieser Daten durch das DCU-Programm weiterhin möglich, um Informationen an die SPS zu kommunizieren oder mit ihr auszutauschen.

Für Eingänge/Ausgänge, die NICHT verwendet werden, ist weiterhin die direkte Steuerung durch die SPS möglich.

6.2.7 Direktzugriff auf zyklische Bits

Das Modul stellt 16 Bit an zyklischen Eingangsdaten für die SPS (Producing-Daten) bereit, die im DCU-Programm durch das Symbol Y_{Pn} dargestellt

werden. Dabei steht „n“ für die Bit-Nummer des zyklischen Bits, die in einem Bereich zwischen 0 und 15 liegt. Eine *Spule* (auch „Coil“), die mit diesem Symbol gekennzeichnet ist, steuert das entsprechende zyklische Bit in den Producing-Daten des Moduls.

Ausschließlich zyklische Bits, die von den physischen Eingängen/Ausgängen getrennt sind (da sie in einem DCU-Programm verwendet werden), können auf diese Weise bearbeitet werden.

Dementsprechend können die 8 Bit an zyklischen Ausgangsdaten von der SPS (Consuming-Daten) von einem DCU-Programm mit dem Symbol XP_n gelesen werden. Hierbei steht „n“ für die zyklische Bit-Nummer, die zwischen 0 und 7 liegt.

Dies bietet dem DCU-Programm die Möglichkeit, auf von der SPS angestoßene Ereignisse zu reagieren.

6.2.8 Lesen und Bearbeiten von Consuming- und Producing-Daten nach Kanal

Über die mit dem Symbol XC_nA/XC_nB gekennzeichnete Variable wird das zyklische Consuming-Bit gelesen, das von der SPS empfangen wird, die den angegebenen Ausgangskanal steuert. Selbst, wenn der Kanal von der DCU und daher nicht direkt von der SPS gesteuert wird, ist das DCU-Programm in der Lage, auf den Status dieses Bits zu reagieren.

Über die mit YP_nA/YP_nB gekennzeichneten Variablen ist die Bearbeitung der Producing-Daten für den angegebenen Kanal möglich, die mittels der zyklischen Daten an die SPS gesendet werden. Auf diese Weise kann ein DCU-Programm gegenüber der SPS einen Eingangstatus unabhängig von dem tatsächlichen Eingangstatus des Kanals simulieren (Eingangssimulation).

Physischer Port/Kanal	Port 1/A	Port 1/B	Port 2/A	Port 2/B	Port 3/A	Port 3/B	Port 4/A	Port 4/B
Consuming-Bit lesen	XC1A	XC1B	XC2A	XC2B	XC3A	XC3B	XC4A	XC4B
Producing-Bit bearbeiten	YP1A	YP1B	YP2A	YP2B	YP3A	YP3B	YP4A	YP4B

Physischer Port/Kanal	Port 5/A	Port 5/B	Port 6/A	Port 6/B	Port 7/A	Port 7/B	Port 8/A	Port 8/B
Consuming-Bit lesen	XC5A	XC5B	XC6A	XC6B	XC7A	XC7B	XC8A	XC8B
Producing-Bit bearbeiten	YP5A	YP5B	YP6A	YP6B	YP7A	YP7B	YP8A	YP8B

6.2.9 Datenaustausch

Nur in 0980 ESL 390-121-DCU1 (16-DIO-Universal-DCU) verfügbar

- ▶ Das Modul stellt zusätzliche zyklische Daten für den alleinigen Zweck des Datenaustauschs zwischen einer SPS und dem DCU-Programm bereit. Das DCU-Programm kann beispielsweise Befehle und Daten von der SPS entgegennehmen und mit Ausführungsergebnissen antworten.
- ▶ Der Datenaustauschbereich umfasst 16 Bit plus 8 Wörter (als 16-Bit-Ganzzahl mit Vorzeichen) für jede Richtung.
- ▶ Die Datenaustausch-Bits können über die YEn-Bit-Variable in LDMicro geschrieben werden.
- ▶ Die Datenaustausch-Bits können über die XEn-Bit-Variable in LDMicro gelesen werden.
- ▶ Die mit EIn und EOn gekennzeichneten Integer-Variablen ermöglichen das Schreiben und Lesen von Austauschdatenwörtern.

7 DCU-Web-Interface

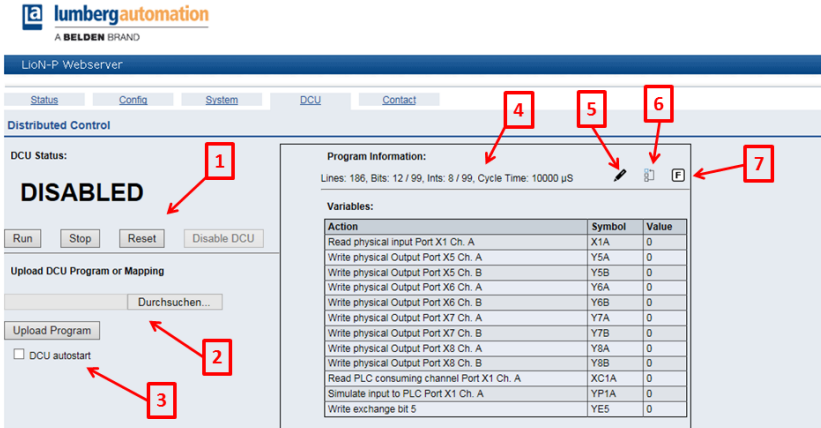


Abb. 9: DCU-Web-Interface

Das DCU-Web-Interface bietet dem Benutzer die Möglichkeit, Programme in die DCU hochzuladen und den DCU-Status zu steuern.

7.1 Übersicht über die Elemente des Web-Interface

1. Zeigt den aktuellen DCU-Status und stellt Schaltflächen zur Steuerung der DCU bereit.
2. Hochladen eines DCU-Programms oder einer Zuordnungsdatei.

3. Wenn das Kontrollkästchen für Autostart markiert ist, wird die DCU nach dem Einschalten automatisch betrieben, wenn ein gültiges Programm vorhanden ist.
4. Zeigt grundlegende Programminformationen und eine Liste aller Modulfunktionen, die zum jeweiligen Zeitpunkt vom DCU-Programm verwendet werden, anhand der Variablen an. Zeigt auch den aktuellen Wert jeder Variable an.
5. Ändern des Zahlenformats der Wertespalte für die INT-Variablen
6. Öffnet den Zuordnungsdialo g für die benutzerdefinierte Variablenzuordnung.
7. Aktivieren/Deaktivieren von Variablen-Forcing.

7.2 Benutzername und Passwort

Um den DCU-Status zu ändern oder Programme hochzuladen, sind Schreibberechtigungen („WRITE“) oder Administratorrechte („ADMIN“) erforderlich. Das Standardpasswort für den Benutzer „admin“ lautet „private“.

7.3 DCU-Status

Folgende DCU-Status sind verfügbar:

Status auf der Website	Beschreibung
NO PROGRAM (KEIN PROGRAMM)	Es liegt kein geladenes Programm vor oder die hochgeladene Datei ist kein gültiges Programm.
LOCKED (GESPERRT)	Die DCU ist durch die Master-(SPS-)Konfiguration gesperrt.
DISABLED (DEAKTIVIERT)	Die DCU ist deaktiviert. Kein Programm wird ausgeführt, und die DCU hat keine Kontrolle über die Eingänge und Ausgänge. Das Modul fungiert als normales digitales E/A-Modul.
STOP	Die DCU steuert die Eingänge und Ausgänge, die im geladenen Programm verwendet werden. Das Programm wird jedoch angehalten. Alle anderen Eingänge und Ausgänge können weiterhin vom Master gesteuert werden.
RUN	Die DCU steuert die Eingänge und Ausgänge, die im geladenen Programm verwendet werden. Das Programm wird ausgeführt. Alle anderen Eingänge und Ausgänge können weiterhin vom Master gesteuert werden.

7.4 Hochladen eines Programmes in die DCU

Programme, die mit LDMicro erstellt und kompiliert werden, können direkt in die SPS hochgeladen werden. Wählen Sie die Programmdatei (.int) aus, und wählen Sie „Upload“ (Hochladen). Der Programm-Upload ist NICHT zulässig, wenn die DCU im RUN-Modus betrieben wird. Um ein DCU-Programm hochzuladen, sind Schreibberechtigungen („WRITE“) oder Administratorrechte („ADMIN“) erforderlich.

Es ist auch möglich, eine Variablenzuordnungsdatei (.map) hochzuladen.

7.5 Programminformationen

Das Kästchen auf der rechten Seite zeigt Informationen zum aktuell geladenen Programm an. Die Variablen-tabelle zeigt alle im Programm enthaltenen Variablen an, die einer Modulfunktion zugeordnet sind. Hier finden Sie auch die Schaltflächen zum Ändern des Zahlenformats, des Zuordnungsdialogs und des Variablen-Forcing.

7.6 Autostart

Wenn das Kontrollkästchen für den Autostart markiert ist und sofern ein gültiges Programm geladen ist, startet die DCU beim Einschalten des Moduls automatisch im RUN-Modus.

7.7 Benutzerdefinierte Zuordnung

In der Regel wird auf die Zuordnung zwischen einer Variable und der entsprechenden Modulfunktion implizit im Variablennamen hingewiesen (siehe Tabelle in Kapitel 5.5). So wird eine Variable mit dem Namen Y5B automatisch mit der Modulfunktion „Physischen Ausgangsport 5, Kanal B setzen“ verknüpft. Bei Anwendung dieses Ansatzes ist keine weitere Zuordnung erforderlich.

Für eine bessere Lesbarkeit des DCU-Programms kann es nützlich sein, die Variablen entsprechend ihrer Funktion in der Anwendung zu benennen. Soll ein Ausgang zum Beispiel eine grüne LED steuern, kann der Variablenname Y_LEDGreen lauten.

Solch ein Variablenname ist dem Modul nicht bekannt und wird daher nicht mit einer Funktion verknüpft. Darüber hinaus wird diese Variable nicht in der Variablenliste angezeigt. Die Zuordnung zwischen dieser Variable und einem physischen Ausgang kann jedoch manuell über die Variablenzuordnung vorgenommen werden.

7.7.1 Vornehmen der Zuordnung über den Zuordnungsdialog

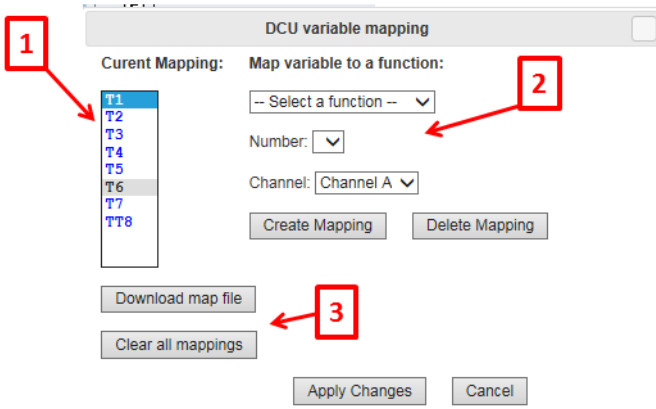


Abb. 10: Dialog für die Variablenzuordnung

Der Zuordnungsdialog zeigt eine Liste aller im aktuellen Programm enthaltenen Variablen (die nicht automatisch anhand ihrer Namen zugeordnet werden). [1]

Jede dieser Variablen kann durch die Auswahl der Variable mit einer Gerätefunktion verknüpft werden. Anschließend wird die Funktions-Dropdown-Liste mit möglichen Funktionen für den betreffenden Variablentyp befüllt.

Nach der Auswahl einer Funktion ist ggf. ihre Spezifizierung erforderlich. Beispielsweise ist hier die Auswahl einer Zahl oder eines Ports und Kanals erforderlich. [2]

Durch einen Klick auf **Create Mapping** (Zuordnung vornehmen) wird die Variable mit der betreffenden Funktion verknüpft. Diese Verbindung ist auch in der Variablenliste sichtbar.

Die Schaltfläche **Delete Mapping** (Zuordnung löschen) hebt eine bestehende Zuordnung auf.

Nachdem jede erforderliche Variable zugeordnet wurde, klicken Sie **Apply Changes** (Änderungen übernehmen), um die Änderungen an das Modul zu übermitteln.

Die Variablenliste auf der DCU-Seite zeigt nun auch die neu zugeordneten Variablen mit dem ursprünglichen Funktionsnamen in Klammern.

Zur Sicherung, Wiederverwendung oder externen Bearbeitung kann die aktuelle Zuordnung als Datei heruntergeladen werden. [3]

Die Schaltfläche **Clear all mappings** (Alle Zuordnungen löschen) löscht alle bestehenden Zuordnungen des Gerätes.

7.7.2 Manuelle Erstellung einer Zuordnungsdatei

Eine Variablenzuordnung kann auch einfach durch den Upload einer Zuordnungsdatei vorgenommen werden. Die Datei ist folgendermaßen strukturiert:

```
[CustomName];[Symbol]\r\n
```

Beispiel

```
XStart;X1A
```

```
YLED;Y5B
```

Der Dateiname sollte mit der Erweiterung „.map“ enden.

7.8 Variablen-Forcing

Program Information:

Lines: 186, Bits: 12 / 99, Ints: 8 / 99, Cycle Time: 10000 µS

Variables:

Action	Symbol	Value
Read physical input Port X1 Ch. A	X1A	0 <input type="checkbox"/>
Write physical Output Port X5 Ch. A	Y5A	1 <input checked="" type="checkbox"/>
Write physical Output Port X5 Ch. B	Y5B	0 <input type="checkbox"/>
Write physical Output Port X6 Ch. A	Y6A	0 <input type="checkbox"/>
Write physical Output Port X6 Ch. B	Y6B	0 <input type="checkbox"/>
Write physical Output Port X7 Ch. A	Y7A	0 <input type="checkbox"/>
Write physical Output Port X7 Ch. B	Y7B	0 <input type="checkbox"/>
Write physical Output Port X8 Ch. A	Y8A	1 <input checked="" type="checkbox"/>
Write physical Output Port X8 Ch. B	Y8B	0 <input type="checkbox"/>
Read PLC consuming channel Port X1 Ch. A	XC1A	0 <input type="checkbox"/>
Simulate input to PLC Port X1 Ch. A	YP1A	0 <input type="checkbox"/>
Write exchange bit 5	YE5	1 <input checked="" type="checkbox"/>

Abb. 11: Variablenliste mit aktivem Forcing

Alle Variablen eines DCU-Programms, die einer Modulfunktion zugeordnet sind, können vorgegeben werden (dies wird als „Forcing“ bezeichnet). Dies bedeutet, dass die manuelle Änderung des Wertes über das Web-Interface möglich ist.

Für Eingangsvariablen (Variablen, die mit Daten aus dem Modul befüllt sind) kann ein spezifischer Wert vorgegeben werden, der anschließend vom DCU-Programm gelesen wird. Auf diese Weise können Eingangsdaten für ein DCU-Programm bearbeitet werden, z. B. zu Testzwecken.

Für Ausgangsvariablen (Variablen, die von einem DCU-Programm geschrieben werden, um den Modulstatus zu ändern) kann ein spezifischer Wert vorgegeben werden, der direkt an die Modulfunktion weitergegeben wird, der die Variable zugeordnet ist. So lässt sich diese Funktion direkt bearbeiten. Die Bearbeitung von Ausgangsvariablen ist nur möglich, wenn die DCU den „RUN“-Status aufweist.

Um das Vorgeben (d. h. Forcing) von Variablen zu starten, klickt der Benutzer die mit „F“ gekennzeichnete Schaltfläche im Programminformationsbereich. Der Buchstabe auf den Schaltflächen ändert sich in „X“. Ein weiterer Klick auf die Schaltfläche beendet das Variablen-Forcing.

Alle Variablen in der Liste erhalten eine zusätzliche „F“-Schaltfläche in der Wertespalte. Ein Klick auf diese Schaltfläche öffnet ein kleines Dialogfenster, das dem Benutzer die Möglichkeit bietet, einen Force-Wert für diese Variable einzugeben. Bit-Variablen stellen nur eine Schaltfläche vom Typ „0“ und „1“ bereit, da Bit-Variablen ausschließlich auf einen Wert von „0“ oder „1“ gesetzt werden können.

Für Int-Variablen ist die Eingabe einer Zahl möglich.

Die mit „X“ gekennzeichnete Schaltfläche beendet sofort das Forcing für die Variable.

Eine vorgegebene Variable ist in der Liste gelb hinterlegt.

8 Batch-Upload von DCU-Programm und Zuordnungsdateien

Das DCU-Programm und die Zuordnungsdateien können auch mittels einer HTTP-POST-Anforderung direkt hochgeladen werden. Ein Perl-Beispielskript wird bereitgestellt, das direkt für den Batch-Upload verwendet werden kann.

8.1 POST-Anforderung zum Hochladen von Dateien

URI	/upload?cmd=store&fullpage=false	
Methode	POST	
MIME-Typ	multipart/form-data	
Formularfelder	Pfad	dcu
	submit	upload
	file	[hochzuladende Datei] (als „application/octet-stream“)
Dateiname	dc.int für Programmdatei dc.map für Zuordnungsdatei	

8.2 Verwendung des Perl-Skripts

Ein Perl-Skript (transfer.pl) für den Batch-Upload von Dateien in die DCU wird mit einer Batch-Beispieldatei bereitgestellt, die zeigt, wie das Skript verwendet werden kann.

Diese Zeile lädt eine DCU-Programmdatei (dc.int) in das Modul mit der IP-Adresse 192.168.1.20 hoch:

```
perl -w .\transfer.pl -s dc.int -t dcu 192.168.1.20 -a IO-Device:admin:private
```

Dasselbe gilt für die Zuordnungsdatei:

```
perl -w .\transfer.pl -s dc.map -t dcu 192.168.1.20 -a IO-Device:admin:private
```

Schließlich müssen der Benutzername und/oder das Passwort dahingehend angepasst werden, dass sie der tatsächlichen Modulkonfiguration entsprechen. Jeder Benutzer, der mindestens über Schreibberechtigungen („WRITE“) verfügt, kann verwendet werden.

IO-Device ist der tatsächliche Name und sollte nicht geändert werden.

9.2 JSON-Antwort-Objektstruktur

```
▼ object {11}
  name : 0980 ESL 300-121-DCU1
  fw-version : F10014
  hw-version : 4.1
  mac : 3C:B9:A6:00:17:00
  bus : 0
  ▶ inputs [2]
  ▶ outputs [2]
  ▶ consuming [2]
  ▶ producing [2]
  ▶ diag [4]
  ▼ dcu {7}
    state : 0
    autostart : 0
    ▶ public [6]
    ▶ consuming_bits [2]
    ▶ consuming_ints [8]
    ▶ producing_bits [2]
    ▶ producing_ints [8]
```

9.3 JSON-Antwort

Feldname	Datentyp	Beschreibung
name	Zeichenkette	Modulname
fw-version	Zeichenkette	Firmware-Version
hw-version	Zeichenkette	Hardware-Version
mac	Zeichenkette	MAC-Adresse des Moduls
bus	Zahl	0 = nicht an Feldbus angeschlossen 1 = an Feldbus angeschlossen
failsafe	Zahl	0 = normaler Ausgangsbetrieb 1 = Ausgänge im Failsafe-Betrieb
inputs	Zahl [2]	LSB = Status des physischen Eingangs, Port X1–X4 MSB = Status des physischen Eingangs, Port X5–X8
outputs	Zahl [2]	LSB = Status des physischen Ausgangs, Port X1–X4 MSB = Status des physischen Ausgangs, Port X5–X8
consuming	Zahl [2]	Verbrauch von Daten von der SPS
producing	Zahl [2]	Erzeugen von Daten für die SPS
diag	Zahl [4]	Enthält Diagnose-Informationen des Moduls Byte 0: <ul style="list-style-type: none"> ▶ Bit 0 = Fehler System-/Sensorversorgung (U_S) ▶ Bit 1 = Fehler Aktuatorversorgung (U_S) ▶ Bit 2 = Sensorkurzschluss erkannt ▶ Bit 3 = Aktuator-Überlast ▶ Bit 6 = Force Mode aktiv ▶ Bit 7 = Interner Modulfehler (E/A-Daten ungültig!) Byte 1 = Sensorkurzschluss, Port 1–8 Byte 2 = Aktuatorkurzschluss, Port 1–4 (Kanal A, B) Byte 3 = Aktuatorkurzschluss, Port 5–8 (Kanal A, B)
dcu	Objekt	(nur verfügbar auf DCU-Modulen)

Feldname	Datentyp	Beschreibung
dcu/state	Zahl	Aktueller DCU-Status: <ul style="list-style-type: none"> ▶ 0 = LOCKED (GESPERRT) ▶ 1 = NO PROGRAM (KEIN PROGRAMM) ▶ 2 = DISABLED (DEAKTIVIERT) ▶ 3 = STOP (STOPP) ▶ 4 = RUN (IN BETRIEB) ▶ 5 = ERROR (FEHLER)
dcu/autostart	Zahl	Lautet 1, wenn der lokale automatische Start aktiviert ist
dcu/public	Zahl [32]	Enthält alle Werte der öffentlichen DCU-Variablen _P0 - _P31
dcu/ consuming_bits	Zahl [2]	16 von der SPS gesetzte DCU-Austausch-Bits
dcu/producing_bit	Zahl [2]	16 vom DCU-Programm gesetzte DCU-Austausch-Bits
dcu/ consuming_ints	Zahl [8]	16 von der SPS gesetzte DCU-Austauschwörter (16-Bit-Ganzzahl mit Vorzeichen)
dcu/producing_ints	Zahl [8]	16 vom DCU-Programm gesetzte DCU-Austauschwörter