LioN-P µDCU
Distributed Control Unit

LioN-P Digital
LDMicro

0980 ESL 393-121-DCU1

µDCU Programming Manual

Table of contents

1. Introduction

The LioN-P µDCU is a multiprotocol fieldbus slave module based on the 0980 ESL 393-121 module. It provides all the functionality of the base module, but has an additional integrated programmable logic unit. This unit can execute user programs created with a small external tool, called *LDMicro*. These programs are created in a ladder logic manner and are called “DCU programs”.

This allows the user to add additional control logic which is stored directly in the slave module itself and is independent from fieldbus or plc. This ranges from simple Boolean operations of input and outputs to fully autonomous (without any plc) programs.

To a plc the module appears as a normal slave module with 8 bit output data (consuming) and 16 bit input data (producing) of cyclic data.

I/Os used by the DCU program are no more controllable by the plc directly, but can be used to communicate with the plc, because the corresponding cyclic bits can be read and written by the DCU program.

2. Cyclic Data and Parameters

2.1. Input Data

Input	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

The following applies here:

- 1A ... 4A: Actual status of input channel A (contact pin 4) of the M12 socket connections 1 to 4.
- 1B ... 4B: Actual status of input channel B (contact pin 2) of the M12 socket connections 1 to 4.
- 5A ... 8A: Actual status of output channel A (contact pin 4) of the M12 socket connections 5 to 8.
- 5B ... 8B: Actual status of output channel B (contact pin 2) of the M12 socket connections 5 to 8.

Note: Depending on the selected fieldbus, the module can provide additional cyclic data bytes (e.g. diagnostic information). Please refer to the 0980 ESL 393-121 manual for information about fieldbus specific details.

2.2. Output Data

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

- 5A ... 8A: Target status of output channel A (contact pin 4) of the M12 socket connections 1 to 8.
- 5B ... 8B: Target status of output channel B (contact pin 2) of the M12 socket connections 1 to 8.

2.3. Module Parameters

The LioN-P µDCU has one additional plc parameter which controls the DCU startup behaviour.

2.3.1. DCU startup parameter

Disabled	The DCU starts in DISABLED state.
Locked	The DCU is disabled and cannot be started by web interface.
Run	The DCU starts in RUN state and executed the DCU program, IF there is a valid program loaded.

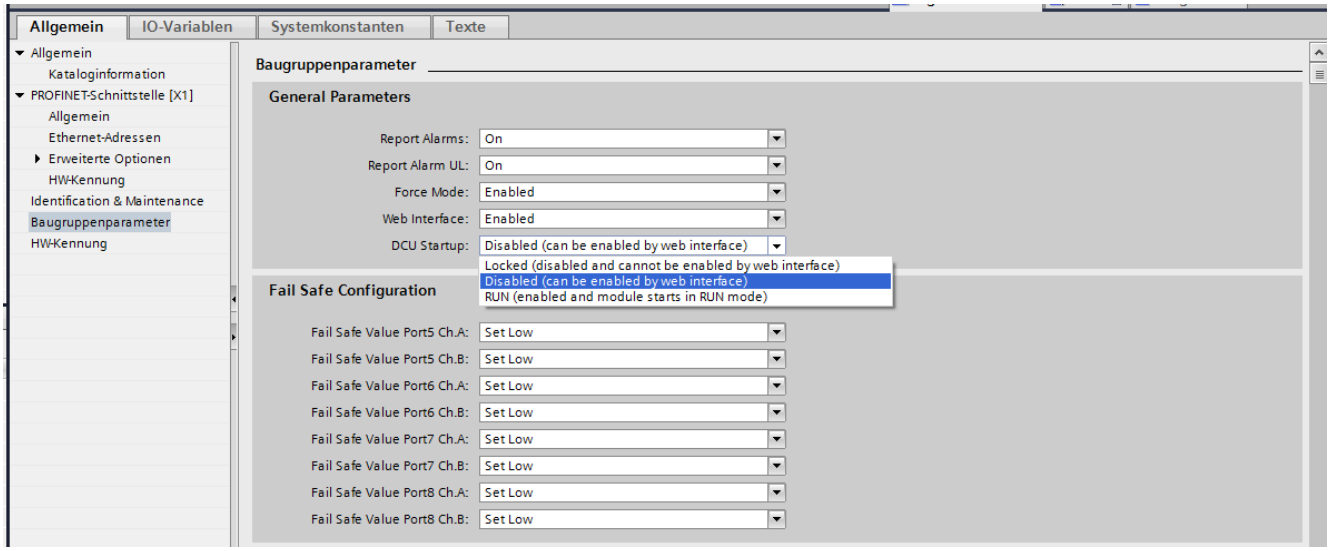
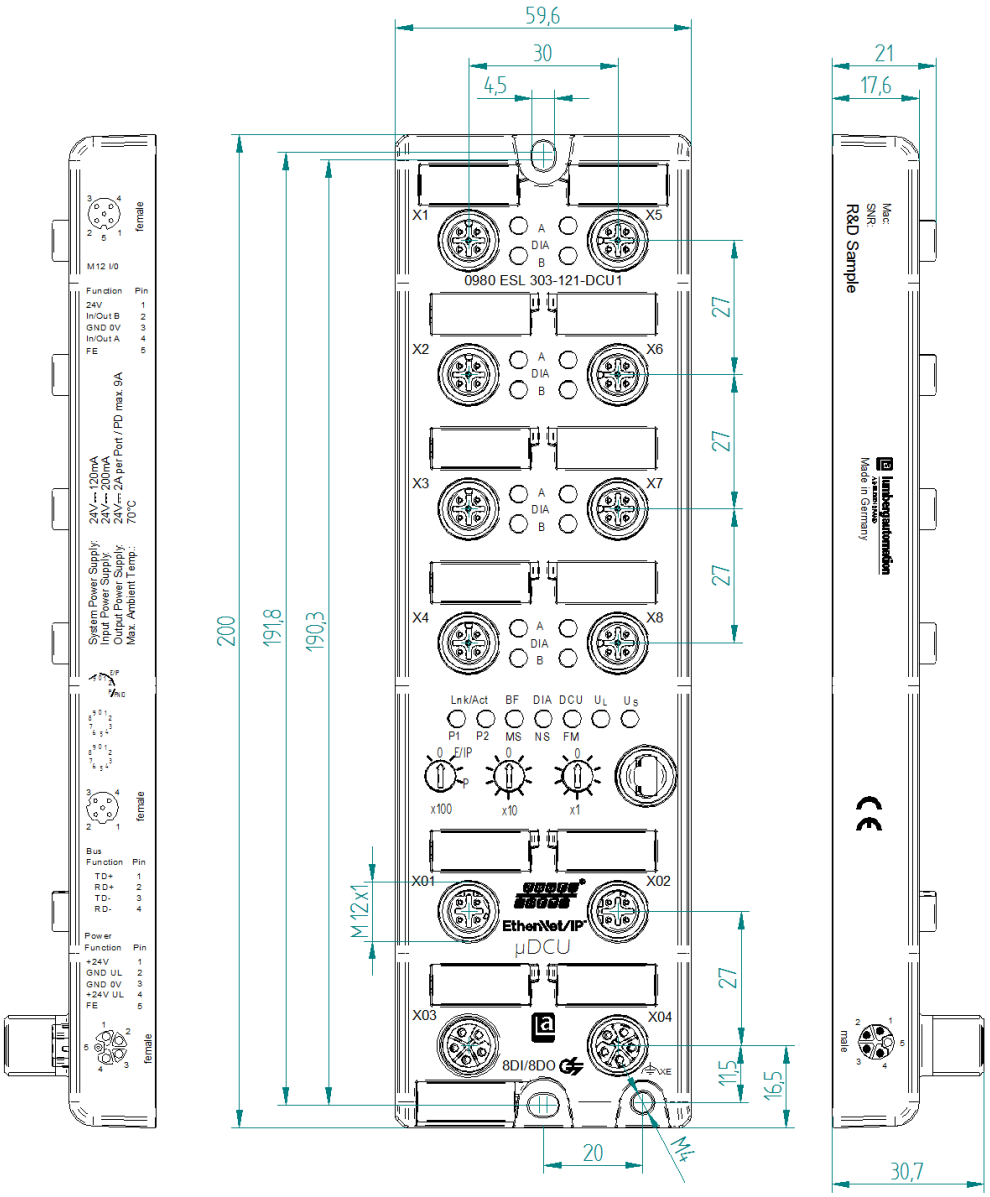


Figure 1 - DCU Startup Parameter in TIA Portal

3. Technical Data

3.1. Mechanical Data



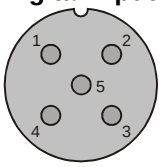
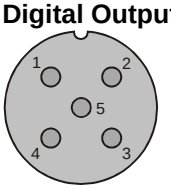
3.2. DCU / Forcemode LED description

The DCU/FM LED on the module indicates the DCU and force mode status.

LED Color	Meaning
Off	DCU Disabled / No Forcemode active.
Blue Flashing	DCU is running
Blue On	DCU is stopped
Red Flashing	DCU error
Blue/Red Flashing	Force Mode ON

3.3. Port types

The µDCU has 8 digital inputs and 8 digital outputs organized in 8 ports with 2 channels each.
For more details please refer to the *0980 ESL 393-121* datasheet and manual.

X1…X4	**Digital Input**  PIN 1: +24Vdc Sensor Supply (200 mA max.) PIN 2: Input Channel B PIN 3: GND PIN 4: Input Channel A PIN 5: FE / Earth
X5…X8	**Digital Output / 2A max.**  PIN 1: n.c. PIN 2: Output Channel B PIN 3: GND PIN 4: Output Channel A PIN 5: FE / Earth

3.4. Electrical Specifications

Please refer to 0980 ESL 393-121 Datasheet.

4. DCU Programming Details

4.1. µDCU / LDMicro Limitations

Max. Rungs	99
Max. Bits	99
Max. Integers	99
Min. µDCU Cycle Time	10 ms

4.2. LDMicro

Open source ladder logic programming tool:
LDMicro download: http://cq.cx/ladder.pl#dl

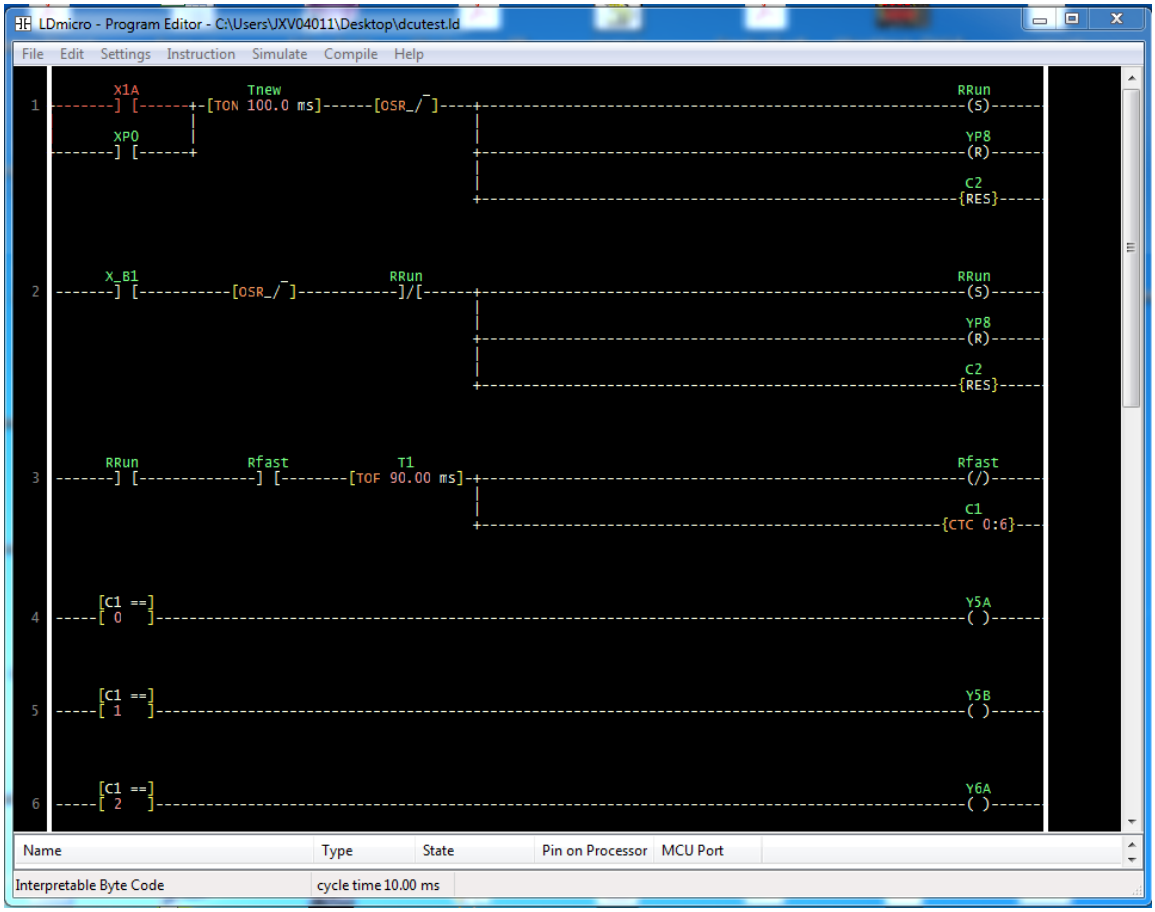


Figure 2 - LDMicro user interface

4.2.1. LDMicro Introduction

With LDMicro the user can create programs in a Ladder Diagram style according to EN 61131-3. Here all elements of the program are arranged on horizontal lines (Rungs). Rungs are always executed from left to right without a guaranteed Rung order. This concept is derived from hardwired relay circuits.

LDMicro offers a large number of instructions such as:

- Bit operations such as contacts, coils, set/reset
- Edge Detection
- Timers and turn on/off delays.
- Up/Down/Circular counters.
- 16 bit signed arithmetic operations.

DCU programs created with LDMicro are able to:

- Control all inputs and outputs of the module.
- React to diagnostic events (short-circuit, undervoltage etc.)
- Communicate with a connected PLC.

- Share information on the network.

4.2.2. Filetypes

Program files for LDMicro are named with **.ld**. Those files can be loaded, edited and saved via LDMicro applicaton.
To compile a program for the DCU, first select the correct target type under *"Settings"* -> *"Microcontroller"* -> *Interpretable Bytecode.*
It is also possible to set the cycle time (*Settings -> MCU Parameters*). A cycle time of 10ms or above is recommended.
Then choose from menu "*Compile" -> "Compile as…*", and select a location and name where the compiled program should be stored. The program will then be compiled. The result is an **.int** file.
This file can now be uploaded into the DCU via the web interface.

4.3. Datatypes

LDMicro knows the following datatypes:

Bit	0 or 1
Int	16 bit signed integer (−32768 to +32767)
T	Timer
C	Counter

4.4. Naming conventions

4.4.1. LDMicro conventions

There are 3 types of bits with a mandatory naming convention:

Type	**Convention**	**Example**
Input Bit	Must start with "X"	`X1A, X5P`
Output Bit	Must start with "Y"	`Y2B, Y3P`
Internal Relay	Must start with "R"	`R1,RRun,RStart`

4.4.2. LioN-P µDCU conventions

Type	**Convention**	**Example**
Physical IO Input Data	X followed by port number and channel	`X1A, X5B`
Physical IO Output Data	Y followed by port number and channel	`Y2B, Y7A`
Cyclic data to PLC	Y followed by "P" and bit number	`YP5, YP15`
Cyclic data from PLC	X followed by "P" and bit number	`XP0, XP6`
Special bits	"X" or "Y" followed by _(underline) and a name	`X_DIA, Y_STOP`
Integer values for IOs	IN or OUT followed by byte no.	`IN1,IN2,OUT1,OUT2`
Integer values for special information	_(underline) followed by a name	`_SCS, _CE1`

4.5. Available Data

This data is available directly in LDMicro programs. Just name a bit or integer variable in LDMicro according to the following list.

Symbol	Direction	Typ	Description
Basic Input/Output Data			
X*n*A	Input	Bit	Digital Inputs Channel A. n=Port. Range: 1…8
X*n*B	Input	Bit	Digital Input Channel B. n=Port. Range: 1…8
Y*n*A	Output	Bit	Digital Output Channel A.n=Port. Range: 1…8
Y*n*B	Output	Bit	Digital Output Channel B.n=Port. Range: 1…8
XP*n*	Input	Bit	Cyclic Input Data from PLC (Consuming Data). n=0…7
YP*n*	Output	Bit	Cyclic Output Data to PLC (Producing Data). n=0…15
OUT1	Output	Int	1st output byte
OUT2	Output	Int	2nd output byte
IN1	Input	Int	1st input byte
IN2	Input	Int	2nd input byte
Diagnostic Information			
X_DIA	Input	Bit	Diagnosis Master Bit
X_SCS	Input	Bit	Sensor Diagnosis Bit
X_SCA	Input	Bit	Actuator Diagnosis Bit
X_LVS	Input	Bit	Sensor Supply Voltage fault
X_LVA	Input	Bit	Actuator Supply Voltage fault
X_COMM	Input	Bit	Cyclic connection to PLC established
_SCS	Input	Int	Sensor short circuit information per Port
_CE1	Input	Int	Channel Error LSB
_CE2	Input	Int	Channel Error MSB
DCU Control			
Y_STOP	Output	Bit	Causes the DCU to STOP
Y_DIS	Output	Bit	Causes the DCU to DISABLE itself.
Special			
_MSG	Output	Int	Show Message with corresponding number on web gui.
X_B*n*	Input	Bit	Virtual Button on web gui pressed. n=button number 1…10.
_P*n*	Output	Int	Data to publish. n=0..31

4.1. Physical and Logical I/O data

The symbols XnA/XnB allows the DCU program to read directly the corresponding physical input. A *contact* named with this symbol would be interpreted as *closed* if the corresponding input pin is shorted with +24VDC (e.g. Pin 1).
The symbols XnA/XnB can be used to control directly a physical digital output. A *coil* which is named with this symbol would activate the corresponding output pin which is set on +24VDC.

I/Os which are used in a DCU program are disconnected from the corresponding cyclic data to and from the plc. However this cyclic data can still be read and manipulated by the DCU program, in order to communicate or exchange information with the plc.
I/Os which are NOT used can still be directly controlled by a plc.

The module provides 16 bit of cyclic input data to the plc (producing data), which is represented in the DCU program by the symbol YP*n*, where n is the bit number of the cyclic bit ranging from 0 to 15. A *coil* which is named with this symbol would control the corresponding cyclic bit in den module's producing data.
Only cyclic bits which are disconnected from physical I/Os (because they are used in a DCU program) can be manipulated in this way.

Likewise the 8 bit of cyclic output data from the plc (consuming data) can be read by a DCU program with the XP*n* symbol. Where n is the cyclic bit number ranging from 0 to 7.

This allows the DCU program to react on events triggered by the plc.

4.1.1. Cyclic input (consuming) process data and their corresponding bit names in LDMicro

The module provides one byte of consuming data. Normally they control directly the 8 physical outputs of the module. However if the DCU is a physical outputs, the consuming data can still be read by the DCU program to react on PLC events.

Byte 0 - Cyclic bit no	0	1	2	3	4	5	6	7
Corresponding physical output	Port 5 Ch. A	Port 5 Ch. B	Port 6 Ch. A	Port 6 Ch. B	Port 7 Ch. A	Port 7 Ch. B	Port 8 Ch. A	Port 8 Ch. B
LDMicro bit name for controlling the physical output	Y5A	Y5B	Y6A	Y6B	Y7A	Y7B	Y8A	Y8B
LDMicro bit name for reading cyclic data from PLC (when the output is controlled by DCU)	XP0	XP1	XP2	XP3	XP4	XP5	XP6	XP7

4.1.2. Cyclic output (producing) process data and their corresponding bit names in LDMicro

The module provides two byte of producing data. Normally they represent the state of the 16 physical inputs of the module. The DCU program can manipulate the producing data to send information to the PLC.

Byte 0 - Cyclic bit no.	0	1	2	3	4	5	6	7
Corresponding physical input.	Port 1 Ch. A	Port 1 Ch. B	Port 2 Ch. A	Port 2 Ch. B	Port 3 Ch. A	Port 3 Ch. B	Port 4 Ch. A	Port 4 Ch. B
LDMicro bit name for reading the physical input state.	X1A	X1B	X2A	X2B	X3A	X3B	X4A	X4B
LDMicro bit name for writing cyclic data to PLC.	YP0	YP1	YP2	YP3	YP4	YP5	YP6	YP7

Byte 1 - Cyclic bit no.	0	1	2	3	4	5	6	7
Corresponding physical input.	Port 5 Ch. A	Port 5 Ch. B	Port 6 Ch. A	Port 6 Ch. B	Port 7 Ch. A	Port 7 Ch. B	Port 8 Ch. A	Port 8 Ch. B
LDMicro bit name for reading the physical input state.	X5A	5B	X6A	X6B	X7A	X7B	X8A	X8B
LDMicro bit name for writing cyclic data to PLC (the cyclic bit is then disconnected from the physical input).	YP8	YP9	YP10	YP11	YP12	YP13	YP14	YP15

4.2. HTTP state information and public data

Integer variables named by `_Pn`, where n is a number ranging from 0 to 31, are considered to be published. They can be read by a HTTP request as part of the DCUs state information, provided in JSON format.
HTTP Request: `http://[ip-address]/r/state.dcu`

The request contains the following information:

Object	Datatype	Name	Description
Root	Boolean	`supported`	True if the module has a DCU functionality
Root	String	`state`	Textual representation of the DCUs current state
Root	Boolean	`autostart`	True if DCU autostart is enabled by web interface
Root	Integer	`color`	Color of the textual state string
Root	Boolean	`runAllowed`	True if the DCU can be set on run mode
Root	Boolean	`reloadAllowed`	True if a new DCU program can be uploaded
Root	Boolean	`disableAllowed`	True if the DCU can be disabled
Root	Boolean	`warnRunStop`	True if a state change to RUN or STOP affects the I/O control.
Root	Boolean	`warnDisable`	True if a state change to DISABLED affects the I/O

			control.
Root	Object PROGRAM	program	Contains information about the current DCU program.
Root	Object GUI	gui	Information about the virtual GUI.
PROGRAM	Boolean	valid	True if a valid DCU program is loaded
PROGRAM	Integer	maxlines	Maximal number of lines allowed in a DCU programs
PROGRAM	Integer	bitcount	Number of bits used by the current DCU program
PROGRAM	Integer	intcount	Number of integers used the current DCU program
PROGRAM	Integer	cycle	Cycle time in ms of the current DCU program
PROGRAM	String Array	channels	Textual representation of the I/O channels used by the DCU program.
PROGRAM	Array PUBLISH	Publish	Published Integer values
PUBISH	Integer	PNr	Number of the public integer (0…31]
PUBISH	Integer	PVal	Value of the public integer

4.3. DCU Web Interface

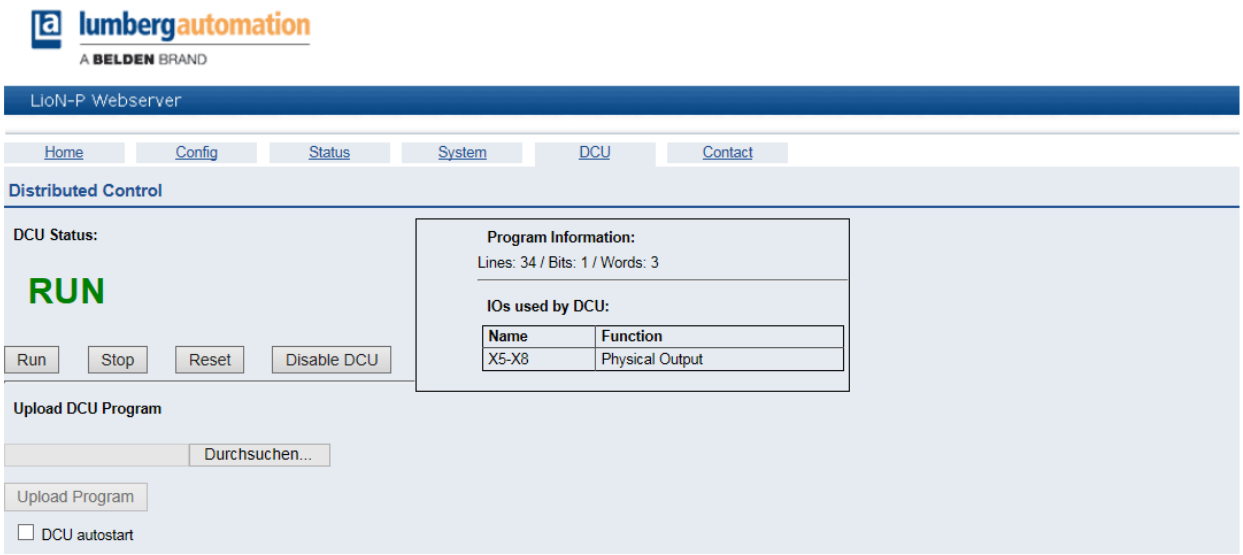


Figure 3 - DCU web interface

The DCU web interface allows the user to upload programs to the DCU and to control the DCU state.

4.3.1. Username and Password

To change the DCU state or upload programs, “WRITE” or “ADMIN” privileges was needed.
The default password for user “admin” is “private”.

4.3.2. DCU States

There are the following DCU states:

State on web site	Description
NO PROGRAM	There is no program loaded or the uploaded file is not a valid program.
LOCKED	The DCU is locked by the master (PLC) configuration.
DISABLED	The DCU is disabled. No program is running and the DCU has no

	control over the inputs and outputs. **The module acts as a normal digital I/O module.**
STOP	The DCU controls the inputs and outputs that are used in the loaded program, but the program is stopped. All other inputs and outputs can still controlled by the master.
RUN	The DCU controls the inputs and outputs that are used in the loaded program, and the program is executed. All other inputs and outputs can still controlled by the master.

4.3.3. Uploading a program into the DCU

Programs which are created and compiled with LDMicro can be directly uploaded into the plc.
Choose the program file (.int) and press upload. Program upload is NOT allowed when the DCU is in RUN mode. WRITE or ADMIN user rights are needed to upload a DCU program.

4.3.4. Program Information

The box on the right shows some information about the currently loaded program.
The IO table shows all physical IOs that are used by the plc program. Only those IOs are controlled by the DCU. All other IOs can still be controlled by the fieldbus master (if there is one).

The public table shows all public variables used in the program and there corresponding values.
The public variables can be obtained via a JSON request to the modules ip-address:
`http://[ip-address]/r/state.dcu`

4.3.5. Autostart

If the autostart checkbox is checked, the DCU will automatically start in RUN mode if the module is powered on and if there is a valid program loaded.

5. Appendix

5.1. Process data Ethernet/IP

5.1.1. Assembly instance ID 101 (input data with diagnosis)

INPUT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	X4-B	X4-A	X3-B	X3-A	X2-B	X2-A	X1-B	X1-A
Byte 1	X8-B	X8-A	X7-B	X7-A	X6-B	X6-A	X5-B	X5-A
Byte 2	MI-IME	0	0	0	MI-SCA	MI-SCS	MI-LVA	MI-LVS
Byte 3	0	0	0	0	SCS-X4	SCS-X3	SCS-X2	SCS-X1
Byte 4	0	0	0	0	0	0	0	0
Byte 5	CE-X8B	CE-X8A	CE-X7B	CE-X7A	CE-X6B	CE-X6A	CE-X5B	CE-X5A

Table 25: Bit assignment for assembly instance ID 101

5.1.2. Assembly instance ID 102 (input data without diagnosis)

INPUT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	X4-B	X4-A	X3-B	X3-A	X2-B	X2-A	X1-B	X1-A
Byte 1	X8-B	X8-A	X7-B	X7-A	X6-B	X6-A	X5-B	X5-A
Byte 2	0	0	0	0	MI-SCA	MI-SCS	MI-LVA	MI-LVS

Table 26: Bit assignment for assembly instance ID 102

5.1.3. Assembly instance ID 100 (output data)

INPUT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 0	X8-B	X8-A	X7-B	X7-A	X6-B	X6-A	X5-B	X5-A

Table 27: Bit assignment for assembly instance ID 100

5.1.4. Key

- X1-A...X4-A: Input status, channel A (contact pin 4) of slots X1 to X4
- X1-B...X4-B: Input status, channel B (contact pin 2) of slots X1 to X4
- X5-A...X8-A: Output status, channel A (contact pin 4) of slots X5 to X8
- X5-B...X8-B: Output status, channel B (contact pin 2) of slots X5 to X8
- MI-LVS: Module information byte – undervoltage for system/sensor power supply
- MI-LVA: Module information byte – actuator undervoltage
- MI-SCS: Module information byte – sensor short-circuit
- MI-SCA: Module information byte – actuator short-circuit
- MI-IME Module information byte – internal module error
- SCS-X1...SCS-X4: Sensor short-circuit at slot X1 to X4
- CE-X5A...CE-X8A: Channel error, channel A (contact pin 4) of slots X1 to X8
- CE-X5B...CE-X8B: Channel error, channel B (contact pin 2) of slots X1 to X8

5.2. Process data PROFINET

5.2.1. Input data

This module provides two bytes of input data that reflects the current status of the input and output channels.

Input	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

The following applies here:

- 1A ... 4A: Actual status of input channel A (contact pin 4) of the M12 socket connections 1 to 4.
- 1B ... 4B: Actual status of input channel B (contact pin 2) of the M12 socket connections 1 to 4.
- 5A ... 8A: Actual status of output channel A (contact pin 4) of the M12 socket connections 5 to 8.
- 5B ... 8B: Actual status of output channel B (contact pin 2) of the M12 socket connections 5 to 8.

5.2.2. Output data

This module requires two bytes of status information for control of the digital outputs.

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

The following applies here:

- 5A ... 8A: Target status of output channel A (contact pin 4) of the M12 socket connections 1 to 8.
- 5B ... 8B: Target status of output channel B (contact pin 2) of the M12 socket connections 1 to 8.

5.3. Process data EtherCAT

The PDO 0x1A00 or 0x1A01 can be selected for the input data of the module.

PDO 0x1A00, Input data in byte format

PDO		PDO Content			
Index	Size	Index	Size	Type	Name
0x1A00	2	0x6000:01	1.0	USINT	Port X1A..X4B (refer to table x1)

Table 1: Input data in byte format

Content of Port X1A..X4B and Port X5A..X8B:

Index	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x6000:01	X4B	X4A	X3B	X3A	X2B	X2A	X1B	X1A

Table 2: Bit assignment for input data in byte format

- X1A...X8A: Input status, channel A (contact pin 4) of slots X1 to X8
- X1B...X8B: Input status, channel B (contact pin 2) of slots X1 to X8

PDO 0x1A01, Input data in bit format

PDO		PDO Content			
Index	Size	Index	Size	Type	Name
0x1A01	2	0x6020:01	0.1	BIT	Port X1A
		0x6020:02	0.1	BIT	Port X1B
		0x6020:03	0.1	BIT	Port X2A
		0x6020:04	0.1	BIT	Port X2B
		0x6020:05	0.1	BIT	Port X3A
		0x6020:06	0.1	BIT	Port X3B
		0x6020:07	0.1	BIT	Port X4A
		0x6020:08	0.1	BIT	Port X4B

Table 3: Input data in bit format

For the input data direction (TxPDO of the device) the following flexible selectable PDO's are available:

PDO 0x1A02, Output status in byte format

This object delivers (optional selectable) the real outputs status as input data to the controller (Output Mirror):

PDO		PDO Content			
Index	Size	Index	Size	Type	Name
0x1A00	2	0x2200:01	1.0	USINT	Port X5A..X8B (refer to table x1)

Table 4: Input data in byte format

Content of Port X5A..X8B:

Index	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x2200:01	X8B	X8A	X7B	X7A	X6B	X6A	X5B	X5A

Table 5: Bit assignment for output status in byte format

- X5A...X8A: Output status, channel A (contact pin 4) of slots X5 to X8
- X5B...X8B: Output status, channel B (contact pin 2) of slots X5 to X8

PDO 0x1A03, Output status in bit format

This object delivers (optional selectable) the real outputs status as input data to the controller (Output Mirror):

PDO		PDO Content			
Index	Size	Index	Size	Type	Name
0x1A03	2	0x2020:01	0.1	BIT	Port X5A
		0x2020:02	0.1	BIT	Port X5B
		0x2020:03	0.1	BIT	Port X6A
		0x2020:04	0.1	BIT	Port X6B
		0x2020:05	0.1	BIT	Port X7A
		0x2020:06	0.1	BIT	Port X7B
		0x2020:07	0.1	BIT	Port X8A
		0x2020:08	0.1	BIT	Port X8B

Table 6: Input data in bit format

The PDO's 0x1A01 or 0x1A02, 0x1A02 or 0x1A03 can be combined flexible with the PDO's 0x1A04 (Error Register) and/or 0x1A05 (Diagnostic Register)

PDO 0x1A04, Error register

PDO		PDO Content			
Index	Size	Index	Size	Type	Name
0x1A04	1	0x1001:01	1.0	USINT	Error Register

Table 7: Error Register

Content of the Error Register:

B7	B6	B5	B4	B3	B2	B1	B0	Error description
0	0	0	0	0	0	0	0	No error
0	0	0	0	0	0	1	1	Output overload error, MI-SCS or MI-SCA
0	0	0	0	0	1	0	1	Voltage error, MI-LVS
0	0	0	0	0	1	0	1	Voltage error outputs, MI-LVA
1	0	0	0	0	0	0	1	Additional function forcing, MI-FC
1	0	0	0	0	0	0	1	Additional function device diagn., MI-IME

Table 8: Bit Content of diagnostic register

PDO 0x1A04, Diagnostic register

PDO		PDO Content			
Index	Size	Index	Size	Type	Name
0x1A05	4	0x2001:01	4.0	UDINT	Diagnostic Register

Table 9: Diagnostic Register

Content of Diagnosis Register:

INPUT	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Byte 1	MI-IME	MI-FC	0	0	MI-SCA	MI-SCS	MI-LVA	MI-LVS
Byte 2	SCS-X8	SCS-X7	SCS-X6	SCS-X5	SCS-X4	SCS-X3	SCS-X2	SCS-X1
Byte 3	0	0	0	0	0	0	0	0
Byte 4	CE-X8B	CE-X8A	CE-X7B	CE-X7A	CE-X6A	CE-X6A	CE-X5B	CE-X5A

Table 10: Bit Content of diagnostic register

Key

- MI-LVS: Module information byte – undervoltage for system/sensor power supply
- MI-LVA: Module information byte – actuator undervoltage
- MI-SCS: Module information byte – sensor short-circuit at an M12 slot
- MI-SCA: Module information byte – actuator short-circuit
- MI-FC: Module information byte – Forcing activ
- MI-IME Module information byte – internal module error
- CE-X5A...CE-X8A: Channel error, channel A (contact pin 4) of slots 5 to X8
- CE-X5B...CE-X8B: Channel error, channel B (contact pin 2) of slots X5 to X8

The PDO's 0x1600 or 0x1601 can be selected for the output data of the module.

PDO 0x1600, Output data in byte format

PDO		PDO Content			
Index	Size	Index	Size	Type	Name
0x1600	2	0x6200:01	1.0	USINT	Port X5A..X8B (refer to table x1)

Table 11: Input data in byte format

Content of Port X1A..X4B and Port X1A..X4B:

Index	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x6200:01	X8B	X8A	X7B	X7A	X6B	X6A	X5B	X5A

Table 12: Bit assignment for input data in byte format

- X5A...X8A: Output data, channel A (contact pin 4) of slots X5 to X8
- X5B...X8B: Output data, channel B (contact pin 2) of slots X5 to X8

PDO 0x1601, Output data in bit format

PDO		PDO Content			
Index	Size	Index	Size	Type	Name
0x1601	2	0x6220:01	0.1	BIT	Port X5A
		0x6220:02	0.1	BIT	Port X5B
		0x6220:03	0.1	BIT	Port X6A
		0x6220:04	0.1	BIT	Port X6B
		0x6220:05	0.1	BIT	Port X7A
		0x6220:06	0.1	BIT	Port X7B
		0x6220:07	0.1	BIT	Port X8A
		0x6220:08	0.1	BIT	Port X8B

Table 13: Input data in bit format