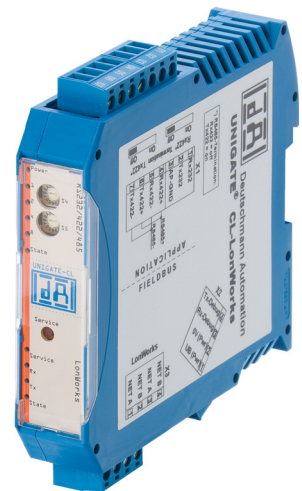


Deuschmann

your ticket to all buses

**Instruction Manual
Universal Fieldbus-Gateway
UNIGATE® CL - LONWorks**



Deuschmann Automation GmbH & Co. KG
www.deuschmann.com | wiki.deuschmann.de

1	Information on CE marking of the module	8
1.1	EU Directive EMC	8
1.2	Scope of application	8
1.3	Note installation guidelines	8
1.4	Installation of the unit	8
1.5	Working on switch cabinets	8
2	Information for the machine manufacturers	9
2.1	Introduction	9
2.2	EU Machinery Directive	9
3	Introduction	10
3.1	Difference in the hardware version	10
3.2	UNIGATE® CL software flow-chart	11
3.3	UNIGATE® block diagram	12
3.4	UNIGATE® application diagram	12
4	Operation modes of the Gateway	13
4.1	Configuration mode (config mode)	13
4.2	Test mode	13
4.3	Data exchange mode	14
5	RS-interface	15
5.1	RS-interfaces at the UNIGATE® CL	15
5.2	Buffer sizes at the UNIGATE® CL	15
5.3	Framing Check	15
6	SSI-interface	16
6.1	Initiation of the SSI-interface	16
6.2	Hardware-wiring	17
7	The Debug-interface	18
7.1	Overview of the Debug-interface	18
7.2	Starting in the Debug-mode	18
7.3	Communication parameter for the Debug-interface	18
7.4	Possibilities with the Debug-interface	18
7.5	Commands of the Debug-interface	18
8	Mode of operation of the system	19
8.1	General explanation	19
8.2	Interfaces	19
8.3	Data exchange	19
8.4	Possible network variables and their data lengths	19
8.5	Run-up phase	19
9	Generating a Script	20
9.1	What is a Script?	20
9.2	Memory efficiency of the programs	20
9.3	What can you do with a Script device?	20
9.4	Independence of buses	20
9.5	Further settings at the Gateway	20
9.6	The use of the Protocol Developer	21
9.7	Accuracies of the baud rates	21

9.8	Script processing times	22
10	Hardware ports, switches and LEDs	23
10.1	Device labeling	23
10.2	Connectors	23
10.2.1	Connector to the external device (RS-interface)	23
10.2.2	Connector supply voltage and DEBUG-interface	24
10.2.3	LONWorks-connector	24
10.2.4	Power supply	24
10.3	LEDs	24
10.3.1	LED "Service"	24
10.3.2	LED "Rx" (at LONWorks 62 only)	25
10.3.3	LED "Tx" (at LONWorks 62 only)	25
10.3.4	LED "(Bus) State"	25
10.3.5	LED "Power"	25
10.3.6	LED "State"	25
10.3.7	LEDs 1 / 2 / 4 / 8 (Error No. / Select ID)	25
10.4	Switches	25
10.4.1	Termination Rx 422 + Tx 422 (serial interface)	25
10.4.2	Rotary coding switches S4 + S5 (serial interface)	26
10.4.3	Pushbutton "Service"	26
10.5	The Debug cable for UNIGATE® CL	26
11	Error handling	27
11.1	Error handling at UNIGATE® CL	27
12	Installation guidelines	28
12.1	Installation of the module	28
12.1.1	Mounting	28
12.1.2	Removal	28
12.2	Wiring	28
12.2.1	Connection systems	28
12.2.1.1	Power supply	28
12.2.1.2	Equipotential bonding connection	28
12.2.2	LONWorks communication interface	29
12.2.2.1	Bus line with copper cable	29
12.2.3	Line routing, shield and measures to combat interference voltage	29
12.2.4	General information on line routing	29
12.2.4.1	Shielding of lines	29
13	LONWorks	31
13.1	Nodes	31
13.2	Network variables (NVTs)	31
13.2.1	Implementing the network variables into the UNIGATE-LONWorks	32
13.2.2	Installation	32
13.2.3	Technical data (characteristics) for LONWorks	32
13.3	Script examples for the initialization of LONWorks	33
13.4	LON-specific parameters	34
13.5	Darstellung der Daten in LONWorks	35
13.5.1	Network variables (NVTs)	36

13.5.2	Determination of a LON-Gateway's update time	36
14	Technical data	37
14.1	Device data	37
14.1.1	Interface data	38
15	Commissioning guide	39
15.1	Note	39
15.2	Components	39
15.3	Installation	39
15.4	Dimensional drawing UNIGATE® CL - LONWorks	39
15.5	Commissioning	39
15.6	LONWorks connection	39
15.7	Connection to the process device	40
15.8	Connecting the supply voltage	40
15.9	Shield connection	40
15.10	Project planning	40
15.11	Literature	40
16	Servicing	41
16.1	Returning a device	41
16.2	Downloading PC software	41
17	Annex.	42
17.1	Explanations of the abbreviations	42
17.2	Hexadecimal table	43

Disclaimer of liability

We have checked the contents of the document for conformity with the hardware and software described. Nevertheless, we are unable to preclude the possibility of deviations so that we are unable to assume warranty for full compliance. The information given in the publication is, however, reviewed regularly. Necessary amendments are incorporated in the following editions. We would be pleased to receive any improvement proposals which you may have.

Copyright

Copyright (C) Deutschmann Automation GmbH & Co. KG 1997 – 2020. All rights reserved. This document may not be passed on nor duplicated, nor may its contents be used or disclosed unless expressly permitted. Violations of this clause will necessarily lead to compensation in damages. All rights reserved, in particular rights of granting of patents or registration of utility-model patents.

1 Information on CE marking of the module

1.1 EU Directive EMC

The following applies to the module described in this User Manual:

Products which bear the CE mark comply with the requirements of EU Directive „Electromagnetic Compatibility“ and the harmonized European Standards (EN) listed therein.

The EU Declarations of Conformity are available at the following location for perusal by the responsible authorities in accordance with the EU Directive, Article 10:

Deutschmann Automation GmbH & Co. KG, Carl-Zeiss-Straße 8, 65520 Bad Camberg, Germany.

1.2 Scope of application

The modules are designed for use in the industrial sector and comply with the following requirements.

Scope of application	Requirement applicable to	
	Emitted interference	Interference immunity
Industry	EN 55011, cl. A (2007)	EN 61000-6-2 (2005)

1.3 Note installation guidelines

The module complies with the requirements if you

1. comply with the installation guidelines described in the User Manual when installing and operating the module.
2. also follow the rules below on installation of the equipment and on working on switch cabinets.

1.4 Installation of the unit

Modules must be installed in electrical equipment rooms/areas or in enclosed housings (e.g. switch boxes made of metal or plastic). Moreover, you must earth the unit and the switch box (metal box) or at least the top-hat rail (plastic box) onto which the module has been snapped.

1.5 Working on switch cabinets

In order to protect the modules against static electrical discharge, the personnel must discharge themselves electrostatically before opening switch cabinets or switch boxes.

2 Information for the machine manufacturers

2.1 Introduction

The UNIGATE® module does not constitute as a machine as defined by the EU "Machinery" Directive. Consequently, the module does not have a Declaration of Conformity in relation to the EU Machinery Directive.

2.2 EU Machinery Directive

The EU Machinery Directive stipulates the requirements applicable to a machine. The term "machine" is taken to mean a totality of connected parts or fixtures (see also EN 292-1, Paragraph 3.1)

The module is a part of the electrical equipment of the machine and must thus be included by the machine manufacturer in the Declaration of Conformity process.

3 Introduction

The module UNIGATE® CL-LONWorks 62 or CL-LONWorks 512 (*Version UNIGATE® CL-LONWorks 512 is no longer available!*) serves to adapt a serial port to the LON-bus. In this application, it functions as Gateway. It can be operated at any LON-bus with FTT-10A.

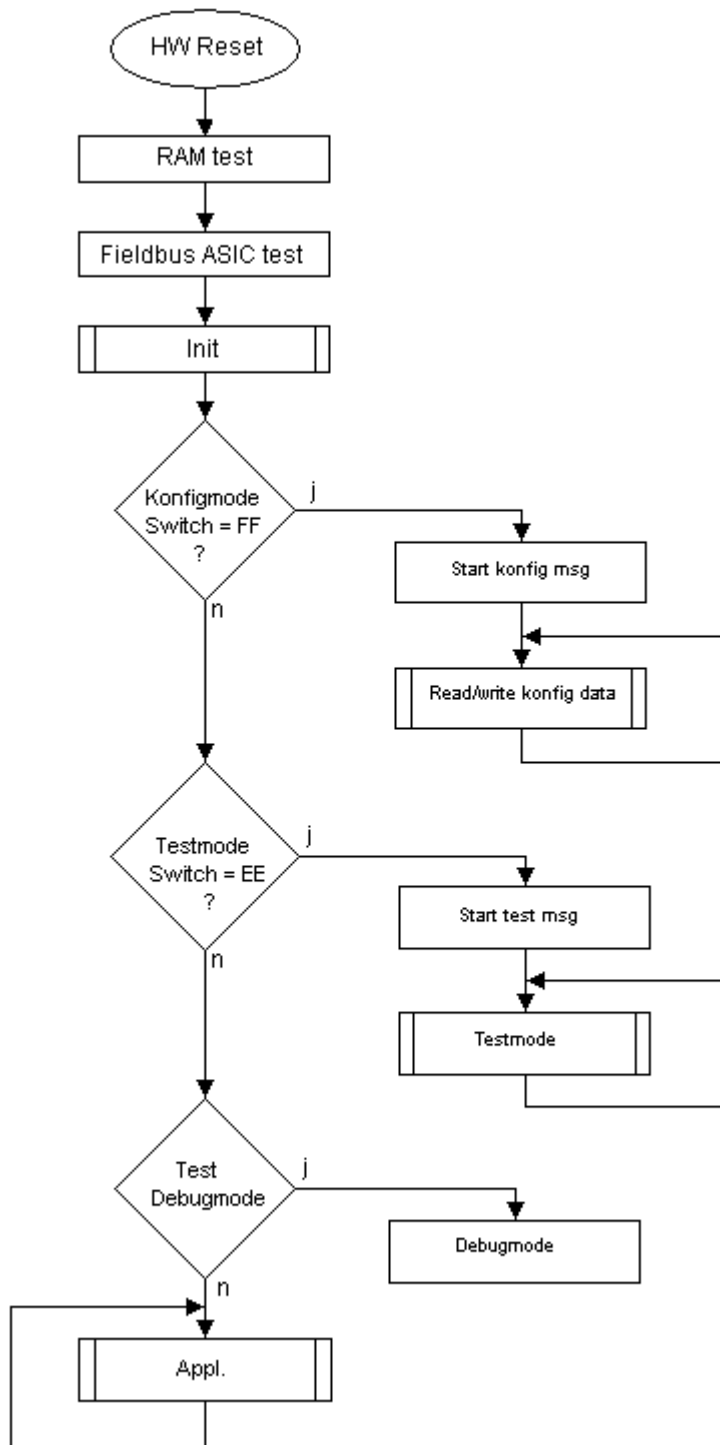
The module CL-LONWorks essentially consists of the following hardware components:

- FTT-10A interface
- LONWorks ECHOLON ASIC
- Microprocessor T89 C51RD2
- RAM and EPROM
- Optionally electrically isolated RS-interface
- Serial interface (RS232, RS485 and RS422) to the device connected externally

3.1 Difference in the hardware version

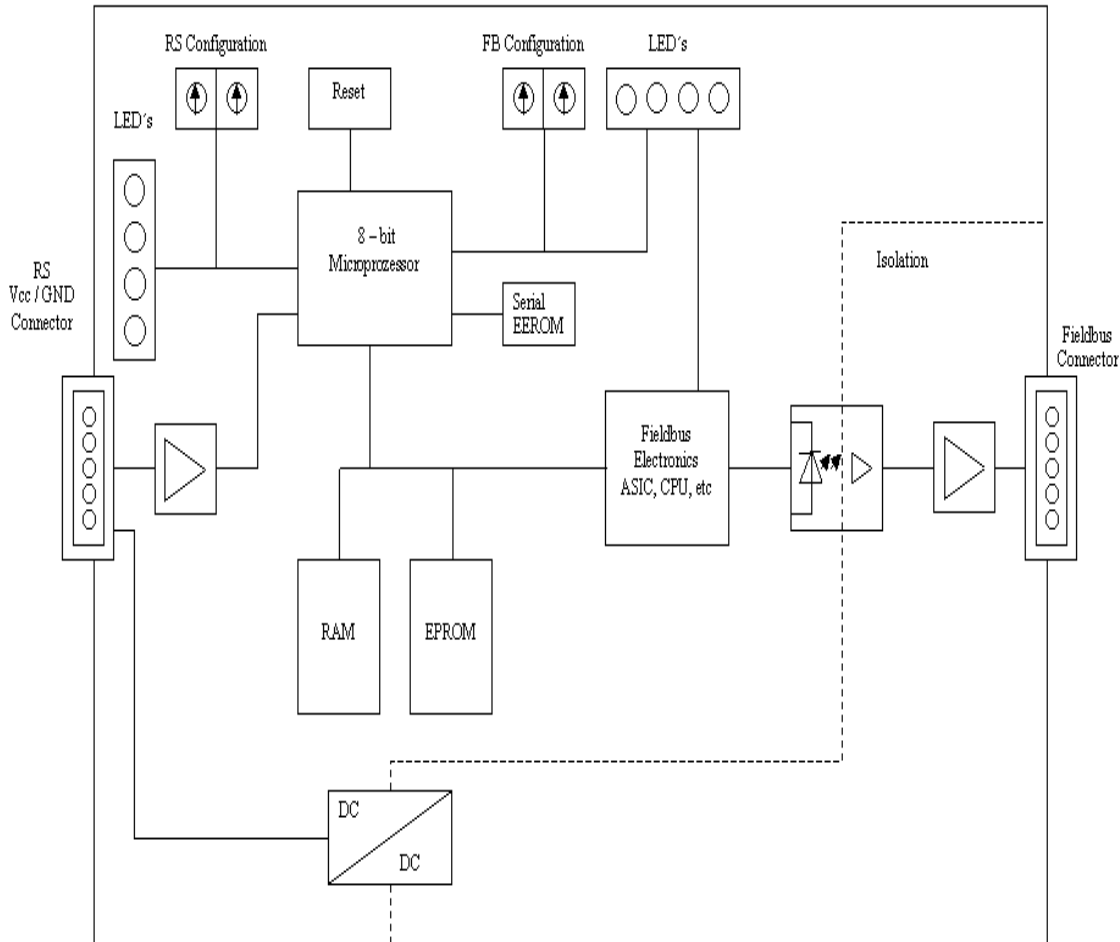
LONworks 512	up to 256 input SNVTs and 256 output SNVTs
LONWorks 62	up to 62 SNVTs, input and output distributed in a variable way

3.2 UNIGATE® CL software flow-chart



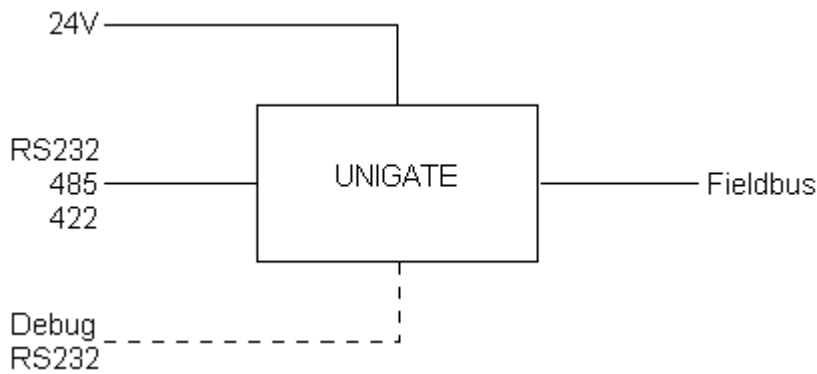
3.3 UNIGATE® block diagram

The following picture shows a typical UNIGATE®-module design.



3.4 UNIGATE® application diagram

The following graph shows a typical connection scheme.



4 Operation modes of the Gateway

4.1 Configuration mode (config mode)

The configuration mode serves to configure the Gateway. The following adjustments are possible in this mode.

- Loading a Script
- Updating the firmware

The Gateway will be starting in this mode in case both switches S4 as well as S5 are set on position "F" when switching on the Gateway. Right after switching on the Gateway in the configuration mode it will be sending its starting message, that looks analog with the following message: "RS-LN(62)-CL (232/422/485) V2.0 [30] (c)dA Switch=0xFF Script(8k)="Leer" Author="Deutschmann Automation GmbH" Version="1.0" Date=21.08.2001 SN=47110001 Neuron-ID=04 36 EB 13 02 00".

In the configuration mode the Gateway always operates with the settings 9600 Bauds, no Parity, 8 databits and 1 stopbit, the RS-State LED will always be flashing red, the "Error No/Select ID" LEDs are of no account for the user. All software revisions contain the configuration mode.

4.2 Test mode

Setting of the test mode

The test mode is set by bringing the switches S4 and S5 in position "E". All other switches will not be taken into consideration for the setting of the test mode. Now the Gateway has to be restarted with these settings (by a short disconnection from the power supply).

In the test mode the Gateway always operates with the settings 9600 baud, no parity, 8 databits and 1 stopbit.

The test mode may be helpful to integrate the Gateway in the relevant environment, for instance to test the parameters of the RS-interfaces.

Mode of operation of the test mode

After the restart in the test mode the Gateway will be sending the values 0-15 in hexadecimal representation ("0".."F") in ASCII-coding on the serial side every second. Simultaneously the same values are issued binary on the fieldbus-interface.

In this mode the State-LED on the RS-side will be flashing red, the "Error No/Select ID" LEDs will be displaying the value in a binary way, that is issued that moment. Additionally each character that is received at one of the interfaces will also be output at the same interface as a local echo. On the fieldbus-side only the first byte will be used for the local echo, that means on receiving as well as on transmitting only the first byte of the bus data is looked at, the other bus data do not change compared to the last data.



Attention:

In the testmode the Gateway UNIGATE-LONWorks 62 automatically generates a variable In and Out of the type SNVT_count. The Out-variable is incremented by one every second.

The program ID is changed to "TM004711".

„Testmode“ stands in the selfdocstring.

Existing bindings are deleted!

4.3 Data exchange mode

The Gateway has to be in the data exchange mode, so that a data exchange between the RS-side of the Gateway and the fieldbus is possible. As long as the Gateway is not in the configuration mode or the test mode, the data exchange mode is active. In the data exchange mode the Gateway will execute the downloaded Script.

5 RS-interface

5.1 RS-interfaces at the UNIGATE® CL

The UNIGATE® CL - LONWorks has the interfaces RS232, RS422 and RS485 available. The hardware always features a DEBUG-interface, see chapter 7.

5.2 Buffer sizes at the UNIGATE® CL

UNIGATE® CL features at the serial side a buffer with the size of 1024 bytes for input data and output data each.

The FIFO of the application interface (RS-interface) can be changed in any Gateway form Script revision 26 on, that is capable for Script. For it please check in the Protocol Developer under "Device Control" - "Hardware".

5.3 Framing Check

The length of the stop bit received by the Gateway is checked through the function "Framing Check". Here the stop bit generated by the Gateway is always long enough, so that connected participants can evaluate the stop bit.

Please be aware that the function "Framing Check" becomes effective only in case of 8 data bit and the setting "No parity".

An error is detected and indicated by the Error LEDs in case the stop bit does not show the length 1 bit during the activated check.

The possible setting for this parameter can be controlled by the Script (see online help from Protocol Developer). The presetting for the "Stop Bit Framing Check" is "enabled".

6 SSI-interface

The UNIGATE[®] also supports the connection of applications or products, that communicate via SSI.

6.1 Initiation of the SSI-interface

The required Script (`example_SSI`), the firmware- (`Cust0023`) and Protocol Developer-extension (`Cust_ssi.xml`) are available free of charge from our website at www.deutschmann.de, as well as the softwaretool PROTOCOL DEVELOPER and the configuration software WINGATE.

- In the Protocol Developer (see chapter 7, The Debug-interface) the ConfigFile "`Cust_ssi.xml`" has to be added. At Options → Settings → ConfigFiles.
- Load the Script "`example_SSI.dss`" into the Protocol Developer.
- The encoder type and the clock frequency has to be defined in the Script itself under "Set number of bits" and "Set type and clock stretch value" (default = 12-Bit-Single-Turn-Gray, max. clock stretch):

```
// Set number of bits
// 1..16 = Single Turn
// 17..32 = Multi Turn
moveconst (bNumBits, 12); // i.e. 12 bit single turn
// MT SSI 4096 x 4096 = 16777216 = 0b100000000000000000000000 => 24 bit

//-----
// Set type and clock stretch value
// Type (low nibble):
// 0 = Reserved
// 1 = output value as is (i.e. binary encoder)
// 2 = convert Gray encoded output value to binary (i.e. Gray encoder)
// >2 = Reserved
//
// Clock stretch value (high nibble):
// Please note that the given frequency values are only a rough estimate. The
// exact frequency varies depending on the devices underlying architecture.
// 0 = No Stretch --> ~300 kHz
// 1 = ~185 kHz
// 2 = ~150 kHz
// 3 = ~125 kHz
// 4 = ~110 kHz
// 5 = ~100 kHz
// 6 = ~ 88 kHz
// 7 = ~ 80 kHz
// 8 = ~ 72 kHz
// 9 = ~ 67 kHz
// A = ~ 62 kHz
// B = ~ 58 kHz
// C = ~ 54 kHz
// D = ~ 50 kHz
// E = ~ 48 kHz
// F = ~ 45 kHz
//moveconst ( wTyp, 0x02); // i.e. Gray encoder, no clock stretch (High-Nibble=0)
```


moveconst (wTyp, 0xF2); // i.e. Gray encoder, max clock stretch (High-Nibble=F)

- Load the Script into the device. Open WINGATE and activate the device in the configuration mode (see chapter 4.1, Configuration mode (config mode)) - an actuation message appears, that looks in line with the following (example CL-PB):
Special Firmware (23) not loaded
RS-PBV1-CL (232/422/485) V7.31[30] (c)dA Switch=0x02FF Script(8k)="SSI"
Author="Deutschmann Automation" Version="V 1.0" Date=20.03.2008 SN=47110002 ID=2
Konfigmode...
The note "Special Firmware (23) not loaded" means that the firmware-extension is not yet loaded. The extension is loaded through Extras → Firmware Script Extension. Select the file "Cust0023 (Cmd 23 + 24 for SSI).hex" and choose "write extension".
- Re-start the device → now only the device's actual actuation message appears and not the note any more.
- Bring the device into the data exchange mode (see chapter 4.3, Data exchange mode) → DONE!

6.2 Hardware-wiring

The clock wires of the SSI-interface are placed onto the Tx-wires of the RS422-interface and the data wires onto the Rx-wires at the UNIGATE® CL.

X1 (3-pin + 4-pin screw-plug-connector):

Pin no.	Name	Function at SSI
1	Rx 232	n. c.
2	Tx 232	n. c.
3	AP-GND	n. c.
4	Rx 422+	SSI DAT+
5	Rx 422-	SSI DAT-
6	Tx 422+	SSI CLK+
7	Tx 422-	SSI CLK-

7 The Debug-interface

7.1 Overview of the Debug-interface

The UNIGATE® IC features a Debug-interface, that allows a step-by-step processing of a Script. Normally this interface is only required for the development of a Script.

7.2 Starting in the Debug-mode

When applying power to the UNIGATE® (power up) the firmware will output the binary character 0 (0x00) after a self-test was carried out on this interface. If the UNIGATE® receives an acknowledgement via this interface within 500 ms, it is in the Debug-mode. The acknowledgement is the ASCII-character O (0x4F).

With the start in the Debug-mode the further execution of Script commands will be put to a stop.

7.3 Communication parameter for the Debug-interface

The Debug-interface is always operating with 9600 baud, no parity, 8 data bit, 1 stop bit. It is not possible to change this parameter in the Protocol Developer. Please consider the fact that these settings have to be in accordance with those of the PC-COM-interface and that the flow control (protocol) has to be set on „none“ there.

7.4 Possibilities with the Debug-interface

Usually the Protocol Developer is connected to the Debug-interface. With it a step-by-step processing of a Script, monitoring jumps and decisions and looking at memory areas is possible. Moreover breakpoints can be set. It basically possesses all characteristics a software-development tool is typically supposed to have. However, it is also possible to carry out a Scrip-update via this interface.

From Script version [27] on you can also output data with the Script command "SerialOutputToDebugInterface". Please also pay attention to the remark in the manual 'Protocol Developer'.

7.5 Commands of the Debug-interface

The commands for the use of the Debug-interface are described in the instruction manual Protocol Developer.

8 Mode of operation of the system

8.1 General explanation

Communication can be split into seven layers, layer 1 to layer 7, in accordance with the ISO/OSI-model.

The DEUTSCHMANN AUTOMATION LONWorks-Gateways convert all layers in accordance with the LONTalk-protocol. Layer 7 is realized according to chapter 8.3.

8.2 Interfaces

The Gateway features the interfaces RS232, RS485 and RS422.

8.3 Data exchange

The UNIGATE[®] has to be connected to the LON-bus through a projecting tool. In the Gateway the data received from the bus is transferred to the external device depending on the loaded Script. The external device responds in accordance with the protocol conventions.

The data received from the external device is written by the Gateway into the internal RAM to the fixed network variable.

All data is transferred consistently by the Gateway in both directions. The maximum data length of consistent data must be considered in the case of data exchange between the participants.

8.4 Possible network variables and their data lengths

At present the UNIGATE supports all network variables from the SNVT master list of May 2002.

UNIGATE	Maximum amount of SNVTs		Total byte	
	Input	Output	Input	Output
LONWorks 512	256	256	512	512
LONWorks 62	62 altogether		512	512

Aliases

UNIGATE	Amount of alias table entries
LONWorks 512	62
LONWorks 62	62

8.5 Run-up phase

The Gateway is programmed and configured in the run-up phase. Data exchange with the external device does not occur until after the run-up phase has been completed with no errors.

9 Generating a Script

9.1 What is a Script?

A Script is a sequence of commands, that are executed in that exact order. Because of the fact that also mechanisms are given that control the program flow in the Script it is also possible to assemble more complex processes from these simple commands.

The Script is memory-oriented. It means that all variables always refer to one memory area. While developing a Script you do not have to take care of the memory management though. The Protocol Developer takes on this responsibility for you.

9.2 Memory efficiency of the programs

A Script command can carry out e. g. a complex checksum like a CRC-16 calculation via data. For the coding of this command only 9 byte are required as memory space (for the command itself). This is only possible when these complex commands are contained in a library.

A further advantage of this library is, that the underlying functions have been in practical use for a couple of years and therefore can be described as 'void of errors'. As these commands are also present in the native code for the controller, at this point also the runtime performance of the Script is favorable.

9.3 What can you do with a Script device?

Our Script devices are in the position to process a lot of commands. In this case a command is always a small firmly outlined task. All commands can be put into classes or groups. A group of commands deals with the communication in general. This group's commands enable the Gateway to send and receive data on the serial side as well as on the bus-side.

9.4 Independence of buses

Basically the Scripts do not depend on the bus, they are supposed to operate on. It means that a Script which was developed on a PROFIBUS Gateway can also be operated on an Interbus without changes, since the functioning of these buses is very similar. In order to also process this Script on an Ethernet Gateway, perhaps further adjustments have to be made in the Script, so that the Script can be executed reasonably.

There are no fixed rules how which Scripts have to operate properly. When writing a Script you should take into account on which target hardware the Script is to be executed, so the necessary settings for the respective buses can be made.

9.5 Further settings at the Gateway

Most devices require no further adjustments, except for those made in the Script itself. However, there are also exceptions to it. These settings are made by means of the software WINGATE. If you know our UNIGATE[®]-series, you are already familiar with the proceeding with it. An example is the adjustment of the IP-address and the net-mask of an Ethernet-Gateway. These values have to be known as fixed values and are not available for the runtime. Another reason for the configuration of the values in WINGATE is the following: After an update of the Script these values remain untouched, i. e. the settings that were made once are still available after a change of the Script.

Only this way it is also possible that the same Script operates on different Ethernet-Gateways, that feature different IP-addresses.

9.6 The use of the Protocol Developer

The Protocol Developer is a tool for an easy generation of a Script for our Script Gateways. Its operation is exactly aimed at this use. After starting the program the Script that was loaded the last time is loaded again, provided that it is not the first start.

Typical for Windows Script commands can be added by means of the mouse or the keyboard. As far as defined and required for the corresponding command, the dialog to the corresponding command is displayed, and after entering the values the right text is automatically added to the Script. The insertion of new commands by the Protocol Developer is carried out in a way that existing commands will not be overwritten. Generally a new command is inserted in front of the one where the cursor is positioned. Of course the commands can also be written by means of the keyboard or already written commands can also be modified.

9.7 Accuracies of the baud rates

The baud rate of the serial interface is derived from the processor's crystal frequency.

Meanwhile all Script-Gateways are working with a crystal frequency of 40 MHz.

You can enter any desired integer baud rate into the Script. After that the firmware adjusts the baud rate, that can be derived the most precisely from the crystal frequency.

The baud rate the Gateway is actually working with (BaudIst) can be determined as follows:

$$\begin{aligned} \text{BaudIst} &= (\text{F32} / \text{K}) \\ \text{F32} &= \text{Crystal frequency [Hz]} / 32 \\ \text{K} &= \text{Round}(\text{F32} / \text{BaudSoll}); \\ &\quad \text{Round}() \text{ is a commercial roundoff} \end{aligned}$$

Example:

The actual baud rate is to be calculated, when 9600 baud are pre-set, where the Gateway is operated with 40 MHz:

$$\begin{aligned} \text{F32} &= 40000000 / 32 = 1250000 \\ \text{K} &= \text{Round}(1250000 / 9600) = \text{Round}(130.208) = 130 \\ \text{BaudIst} &= 1250000 / 130 = 9615.38 \end{aligned}$$

I. e.: The baud rate actually adjusted by the Gateway is 9615.38 baud

The resulting error in per cent can be calculated as follows:

$$\text{Error}[\%] = (\text{abs}(\text{BaudIst} - \text{BaudSoll}) / \text{BaudSoll}) * 100$$

In our example the following error results:

$$\text{Error} = (\text{abs}(9615.38 - 9600) / 9600) * 100 = 0.16\%$$

In practise errors below 2% can be tolerated!

In the following please find a listing of baud rates at a 40 MHz-crystal frequency with the corresponding errors:

4800 baud:	0.16%
9600 baud:	0.16%
19200 baud:	0.16%
38400 baud:	1.35%
57600 baud:	1.35%
62500 baud:	0%
115200 baud:	1.35%
312500 baud:	0%
625000 baud:	0%

9.8 Script processing times

The Script is translated by the Protocol Developer and the consequently generated code is loaded into the Gateway. Now the processor in the Gateway interprets this code. In this case, there are commands that can be processed very fast (e. g. "Set Parameter"). There are also commands, however, that take longer (e. g. copying 1000 bytes). Consequently, for one thing the processing time differs due to the kind of Script command. But the processing time of the Script commands is considerably more determined by the processor time that is available for this process. Since the processor has to carry out several tasks simultaneously (multitasking system) only a part of the processor's capacity is available for the Script processing. The following tasks - in the order of priority - are executed on the processor:

- Sending and receiving data at the Debug-interface (provided that the PROTOCOL DEVELOPER has been started on the PC)
- Sending and receiving data at the RS-interface
- Sending and receiving data at the Fieldbus-interface
- Tasks controlled via internal clock (1 ms) (e. g. flashing of an LED)
- Processing of the Script

From experience approximately 0.5 ms can be calculated for each Script line. This value confirmed itself again and again in many projects as a standard value. He is always quite right if the processor has enough time available for the Script processing.

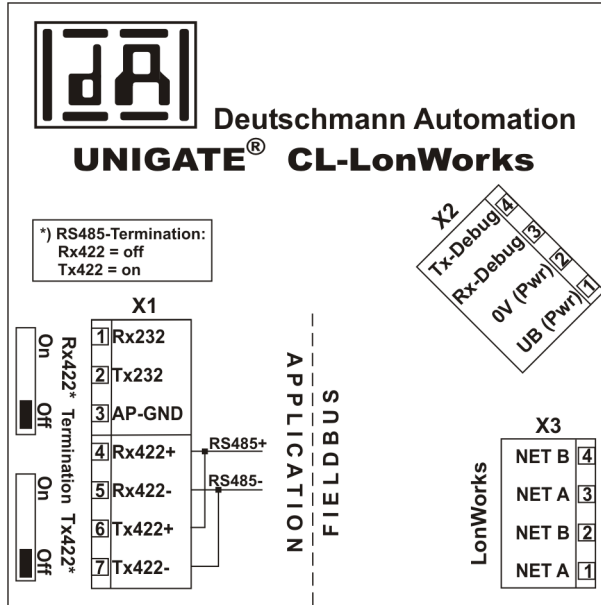
By means of the tasks mentioned above, the following recommendation can be formulated in order to receive a rather fast Script processing:

- Deactivate the Debug-interface (it is the normal case in the serial use)
- Keep the data length at the RS-interface as small as possible. The baud rate is not the problem here, but the amount of characters which are transferred per second.
- Do not unnecessarily extend the data length at the Fieldbus side. Especially at acyclical bus data, if possible do only send them when changes were made. The data length at buses that are configured to a fixed length (e. g. Profibus) should not be longer than absolutely necessary.

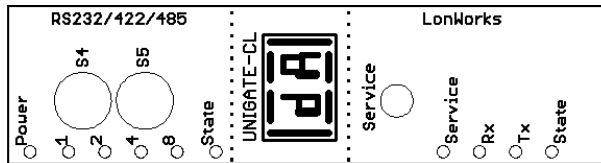
If the processing time should be too large in spite of these measures, there is the possibility to generate a customized Script command, that executes several tasks in one Script command. Please contact our support department for this purpose.

10 Hardware ports, switches and LEDs

10.1 Device labeling



Picture 1: Terminal labeling and termination



Picture 2: Front panel: Rotary switches, pushbutton and LEDs

10.2 Connectors

10.2.1 Connector to the external device (RS-interface)

The serial interface is available at the plug accessible on the upper side of the device.

Pin assignment X1 (3-pole and 4-pole screw-type plug connector)

Pin No.	Name	Function
1	Rx 232	Receive signal
2	Tx 232	Transmit signal
3	AP-GND	Application Ground
4	Rx 422+ (485+)	Receive signal
5	Rx 422- (485-)	Receive signal
6	Tx 422+ (485+)	Transmit signal
7	Tx 422- (485-)	Transmit signal



For the operation at a 485-interface the two pins labeled "485-" have to be connected together. Also the two pins "485+".

10.2.2 Connector supply voltage and DEBUG-interface

Pin assignment X2 (4-pole screw-plug connector, on the bottom side, at the back)

Pin No.	Name	Function
1	UB (Pwr)	10..33 V supply voltage / DC
2	0 V (Pwr)	0 V supply voltage / DC
3	Rx-Debug	Receive signal Debug
4	Tx-Debug	Transmit signal Debug



Attention:

At isolated devices (option GT) Ground for the DEBUG-Interface must be connected with pin 3 (AP-GND) of the RS-interface!

At devices that are not isolated also the 0V (Pwr)-signal can be used as reference.

10.2.3 LONWorks-connector

The plug (labeled: LONWorks) for the connection to the LON-bus is available on the lower side of the device.

Pin assignment X3 (4-pole screw-plug connector)

Pin No.	Name	Function
1	NET A-	Data signal LONWorks
2	NET B	Data signal LONWorks (connected with pin 1)
3	NET A	Data signal LONWorks (connected with pin 2)
4	NET B	Data signal LONWorks

10.2.4 Power supply

The device must be powered with 10-33 VDC, The voltage supply is made through the 4-pole screw-plug connector at the device's bottom side.

Please note that the devices of the series UNIGATE® should not be operated with AC voltage.

10.3 LEDs

The Gateway UNIGATE® CL - LONWorks features 10 LEDs with the following significance:

LED Service	green	Control service pushbutton
LED Rx	green	LON Rx communication
LED Tx	green	LON Tx communication
LED (Bus) State	red/green	Interface state LONWorks
LED Power	green	Supply voltage serial interface
LED State	red/green	User-defined / general Gateway error
LEDs 1 / 2 / 4 / 8 (Error No. / Select ID)	green	User-defined / general Gateway error

10.3.1 LED "Service"

For the version LONWorks 512

Lights green	No application, not configured or service pin button
Blinks green	Network variables not connected
Off	Configured and connected with network

For the version LONWorks 62

Lights green	Service pin button pressed
Blinks green	Not configured
Off	Configured

10.3.2 LED "Rx" (at LONWorks 62 only)

This LED is jittering when data is received.

10.3.3 LED "Tx" (at LONWorks 62 only)

This LED is jittering when data is sent.

10.3.4 LED "(Bus) State"

For the version LONWorks 512

Lights green	Hardware init o. k.
Blinks green/red	Software error or wink command
Lights red	Hardware error or reset

For the version LONWorks 62

Off	initialization error
Blinks green	Operational
Green on	Fieldbus data exchange o. k. , i. e. wink or poll received
Red/green	Wink: blinks for 10 seconds in turns

10.3.5 LED "Power"

This LED is connected directly to the (optionally also electrically isolated) supply voltage of the serial interface (RS232/422/485).

10.3.6 LED "State"

Lights green	Controllable via Script
Flashes green	Controllable via Script
Flashes green/red	Controllable via Script
Lights red	General Gateway error (see LEDs Error No.), controllable via Script
Flashes red	UNIGATE is in the configuration / test mode, controllable via Script

10.3.7 LEDs 1 / 2 / 4 / 8 (Error No. / Select ID)

If these 4 LEDs flash and LED "State" simultaneously lights red, the error number is displayed in binary notation (conversion table, see Annex) in accordance with the table in chapter "Error handling". Additionally there LEDs are controllable via Script.

10.4 Switches

The Gateway features 5 switches with the following functions:

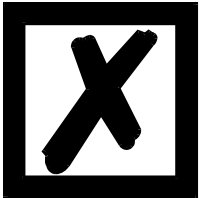
Termination Rx 422	switchable Rx 422-terminating resistor for the serial interface
Termination Tx 422	switchable Tx 422- or RS485-terminating resistor for the serial interface
Rotary coding switch S4	ID High for serial interface i. e. configmode
Rotary coding switch S5	ID Low for serial interface i. e. configmode
Pushbutton "Service"	LON net login

10.4.1 Termination Rx 422 + Tx 422 (serial interface)

If the Gateway is operated as the physically first or last device in an RS485-bus or as 422, there must be a bus termination at this Gateway. In order to do this the termination switch is set to position ON. The resistor (150 Ω) integrated in the Gateway is activated. In all other cases, the switch remains in position OFF.

Please refer to the general RS485 literature for further information on the subject of bus terminations.

If the integrated resistor is used, please allow for the fact that this also activates a pull-down resistor ($390\ \Omega$) to ground and a pull-up resistor ($390\ \Omega$) to VCC.



**At RS48 only the Tx 422-switch must be set to ON.
The Rx 422-switch has to be on OFF.**

10.4.2 Rotary coding switches S4 + S5 (serial interface)

These two switches can be read out through the Script command "Get (RS_Switch, Destination)" and the value can be used for further functions. This value is read in when the Gateway is switched on or always after a Script command has been executed. The switch positions "EE" (testmode) and "FF" (config mode) are reserved.

Note: The switch position "DD" (ie, S4 and S5 in position "D") is reserved for internal purposes.

10.4.3 Pushbutton "Service"

When pressing that pushbutton the device sends a service pin message to the LON network. With it the device can be logged in the network.

10.5 The Debug cable for UNIGATE® CL

As accessory a pre-configured Debug cable is available. The Debug cable connects the Gateway to Debug and RS.

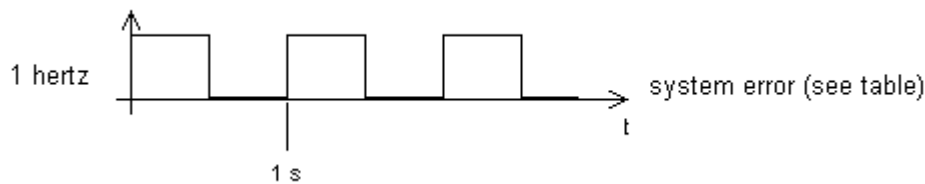
11 Error handling

11.1 Error handling at UNIGATE® CL

If the Gateway detects an error, the error is signalled by the “State“ LED lighting red and, simultaneously, the error number being indicated by means of LEDs “Error No.“ (flashing frequency 1 hertz) as shown in the table below. A distinction can be made between two error categories:

Serious errors (1-5): In this case, the Gateway must be switched off and switched back on again. If the error occurs again, the Gateway must be exchanged and returned for repair.

Warnings (6-15): These warnings are displayed for one minute simply for information purposes and are then automatically reset. If such warnings occur frequently, please inform After-Sales Service.

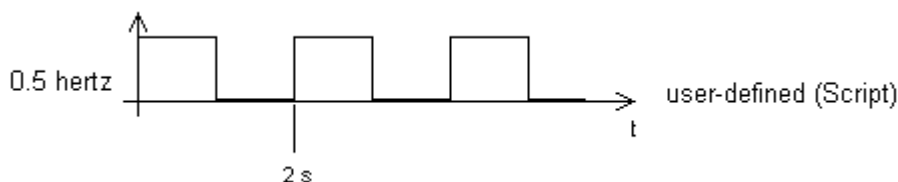


In the configuration mode these displays are not valid and only meant for internal use.

LED8	LED4	LED2	LED1	Error no. resp. ID	Error description
0	0	0	0	0	Reserved
0	0	0	1	1	Hardware fault
0	0	1	0	2	EEROM error
0	0	1	1	3	Internal memory error
0	1	0	0	4	Fieldbus hardware error or wrong Fieldbus-ID
0	1	0	1	5	Script error
0	1	1	0	6	Reserved
0	1	1	1	7	RS-transmit buffer overflow
1	0	0	0	8	RS-receive buffer overflow
1	0	0	1	9	RS timeout
1	0	1	0	10	General fieldbus error
1	0	1	1	11	Parity-or frame-check-error
1	1	0	0	12	Reserved
1	1	0	1	13	Fieldbus configuration error
1	1	1	0	14	Fieldbus data buffer overflow
1	1	1	1	15	Reserved

Table 1: Error handling at UNIGATE® CL

For user-defined errors the flash frequency is 0.5 hertz. The error is displayed as long as defined by "Set Warning Time".



12 Installation guidelines

12.1 Installation of the module

The module with the dimensions 23 x 115 x 116 mm (W x D x H) has been developed for switch cabinet use (IP 20) and can thus be mounted only on a standard mounting channel (deep DIN-rail to EN 50022).

12.1.1 Mounting

- Engage the module from the top in the top-hat rail and swivel it down so that the module engages in position.
- Other modules may be rowed up to the left and right of the module.
- There must be at least 5 cm clearance for heat dissipation above and below the module.
- The standard mounting channel must be connected to the equipotential bonding strip of the switch cabinet. The connection wire must feature a cross-section of at least 10 mm².

12.1.2 Removal

- First disconnect the power supply and signal lines.
- Then push the module up and swivel it out of the top-hat rail.

Vertical installation

The standard mounting channel may also be mounted vertically so that the module is mounted turned through 90°.

12.2 Wiring

12.2.1 Connection systems

The following connection systems must resp. may be used when wiring the module:

- Standard screw-type/plug connection (power supply + RS + LONWorks)
- a) In the case of standard screw-type terminals, one lead can be clamped per connection point. It is best to then use a screwdriver with a blade width of 3.5 mm to firmly tighten the screw.

Permitted cross-sections of the line:

- Flexible line with wire-end ferrule: 1 x 0.25 ... 1.5 mm²
- Solid conductor: 1 x 0.25 ... 1.5 mm²
- Tightening torque: 0.5 ... 0.8 Nm

b) The plug-in connection terminal strip is a combination of standard screw-type terminal and plug connector. The plug connection section is coded and can thus not be plugged on the wrong way round.

12.2.1.1 Power supply

The device must be powered with 10..33 V DC.

- Connect the supply voltage to the 4-pole plug-in screw terminal in accordance with the labelling on the device.

12.2.1.2 Equipotential bonding connection

The connection to the potential equalization automatically takes place if it is put on the DIN-rail.

12.2.2 LONWorks communication interface

12.2.2.1 Bus line with copper cable

This interface is located on the module in the form of a 4-pin screw-plug-connector on the lower side of the housing.

- Plug the LONWorks connecting plug onto the socket labelled "LONWorks".

12.2.3 Line routing, shield and measures to combat interference voltage

This chapter deals with line routing in the case of bus, signal and power supply lines, with the aim of ensuring an EMC-compliant design of your system.

12.2.4 General information on line routing

- Inside and outside of cabinets

In order to achieve EMC-compliant routing of the lines, it is advisable to split the lines into the following line groups and to lay these groups separately.

- ⇒ Group A:
 - shielded bus and data lines (e. g. for ProfibusDP, RS232C and printers etc.)
 - shielded analogue lines
 - unshielded lines for DC voltages ≥ 60 V
 - unshielded lines for AC voltage ≥ 25 V
 - coaxial lines for monitors
- ⇒ Group B:
 - unshielded lines for DC voltages ≥ 60 V and ≥ 400 V
 - unshielded lines for AC voltage ≥ 24 V and ≥ 400 V
- ⇒ Group C:
 - unshielded lines for DC voltages > 400 V

The table below allows you to read off the conditions for laying the line groups on the basis of the combination of the individual groups.

	Group A	Group B	Group C
Group A	1	2	3
Group B	2	1	3
Group C	3	3	1

Table 3: Line laying instructions as a function of the combination of line groups

- 1) Lines may be laid in common bunches or cable ducts.
- 2) Lines must be laid in separate bunches or cable ducts (without minimum clearance).
- 3) Lines must be laid in separate bunches or cable ducts inside cabinets but on separate cable racks with at least 10 cm clearance outside of cabinets but inside buildings.

12.2.4.1 Shielding of lines

Shielding is intended to weaken (attenuate) magnetic, electrical or electromagnetic interference fields.

Interference currents on cable shields are discharged to earth via the shielding bus which is connected conductively to the chassis or housing. A low-impedance connection to the PE wire is particularly important in order to prevent these interference currents themselves becoming an interference source.

Wherever possible, use only lines with braided shield. The coverage density of the shield should exceed 80%. Avoid lines with foil shield since the foil can be damaged very easily as the result of tensile and compressive stress on attachment. The consequence is a reduction in the shielding effect.

In general, you should always connect the shields of cables at both ends. The only way of achieving good interference suppression in the higher frequency band is by connecting the shields at both ends.

The shield may also be connected at one end only in exceptional cases. However, this then achieves only an attenuation of the lower frequencies. Connecting the shield at one end may be more favorable if

- it is not possible to lay an equipotential bonding line
- analogue signals (a few mV resp. mA) are to be transmitted
- foil shields (static shields) are used.

In the case of data lines for serial couplings, always use metallic or metallized plugs and connectors. Attach the shield of the data line to the plug or connector housing.

If there are potential differences between the earthing points, a compensating current may flow via the shield connected at both ends. In this case, you should lay an additional equipotential bonding line.

Please note the following points when shielding:

- Use metal cable clips to secure the shield braiding. The clips must surround the shield over a large area and must have good contact.
- Downstream of the entry point of the line into the cabinet, connect the shield to a shielding bus. Continue the shield as far as the module, but do not connect it again at this point!

13 LONWorks

The UNIGATE® LONWorks module communicates according to the LONTalk record. That means that it can communicate with all LONWorks nodes which correspond to this record. For further information on LONWorks technology see www.echelon.com and www.lonmark.org.

13.1 Nodes

Nodes are devices that interact with physically fixed control units and communicate via a net, that features the LONTalk-protocol. Every node in the net contains stored intelligence that realizes and carries out control functions (handling of the I/O signals). No central service or master-slave construction is required and therefore a LONWorks net is often called "distributed I/O net".

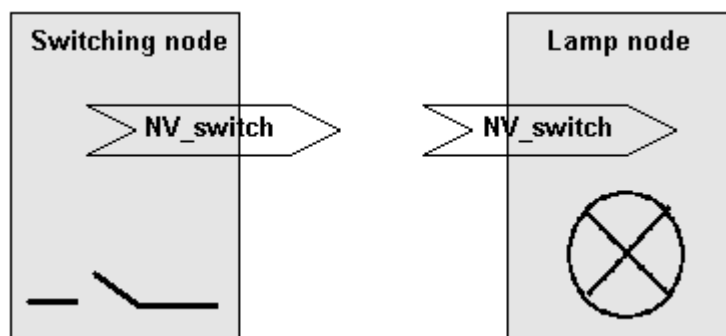
Every node includes a physical interface, a transmitter to a communication media. The UNIGATE®-LON-module uses the FTT-10A (Free Topology Twisted pair) from the Echelon association. This is the medium usually mostly used in automation and this construction supports the star, bus and loop wiring. The FTT-10A-transmitter is connected to the twisted bus cable and has a baud rate of 78 kbit/s. The FTT-10A has a high impedance if no voltage is supplied and it does not disturb the network communication if the operating voltage breaks down.

13.2 Network variables (NVTs)

A network variable is an object on a node that can be connected with one or several network variables at one or several nodes. The network variable of a node defines its „input“ and its „output“ from a network position and allows the sharing of the data in a distributed application. Every time an application writes in one of its network output variables, the new value of the net variables is then distributed via the net to all nodes with the network input variables that were connected with those network output variables. For example: In order to turn on the light via a LONWorks-net, the net output variable of a switching node has to be logically connected to the net input node of the lamp node. When the switch is activated, the network variable is distributed via the net and it is received in the lamp node. The light is turned on. See picture below.

The LONMark Organisation accepted several net variables, that represent different ways of standard data illustrations e. g. temperature, pressure, percentage, voltage. They are called standard network variables or SNVT's and they are listed in the "SNVT master and the programming manual" of the Echelon association.

A SNVT includes information on the type, resolution and size. The LONMark association defines objects that are used for a specific application as a group of SNVT's. Please consider that the customer can also define network variables as not standard and that they are generally called network variables or NVs.



13.2.1 Implementing the network variables into the UNIGATE-LONWorks

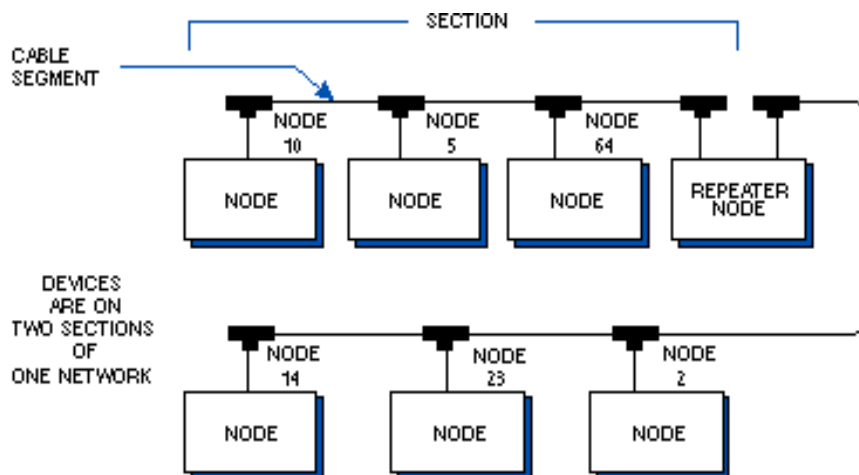
A LONWorks-node is application-specific - which means that it is impossible to have one universal group of network variables for all kinds of different applications. For that reason the UNIGATE®-LON has to be pre-configured with a fixed group of network variables, that might be required for a certain kind of application.

13.2.2 Installation

The network variables can be connected with the nodes by means of an installation tool. This bonding connects the network variables between different nodes. As soon as the bonding is completed, the installation tool can be taken from the net. During the installation time every node receives a unique net identity. For more information on LONWorks please read LONWorks-technology data from Motorola or the LONWorks documentation from Echelon.

13.2.3 Technical data (characteristics) for LONWorks

Technical transmission	Free topology (FTT-10A) twisted pair cable, transmission rate 78 kbits/s
Max. bus length	Bus up to 2000 m (with bus intensifiers)
Max. node-to-node distance	500 m
Max. number of nodes	32.385 nodes divided into 127 nodes/subnet 255-subnets/domain
Kinds of data	Peer to peer Communication



13.3 Script examples for the initialization of LONWorks

```

//***** LON init
//*****
// Type No from SNVT Master-List
// http://www.tsd.org/ieee1473/SNVT.pdf      [LON_SNVT_MasterList_
May2002]
// UG62 max count of SNVT's = 62 In and Out SNVT's together
// UG512 max count of SNVT's = 256 In and 256 Out-SNVT's

// 36=31byte, 37=31byte, 83=2byte, 88=29byte, 93=6byte =>99byte
// set In SNVT's
var InMapTable: buffer[5];          moveConst( InMapTable[0],
#36#37#83#88#93 );
var InMapsize : word;              moveConst( InMapsize, 5); // count of
In SNVT's

// set Out SNVT's
var OutMapTable: buffer[5];          moveConst( Out-
MapTable[0],#36#37#83#88#93 );
var OutMapsize : word;              moveConst( OutMapsize, 5); // count of
Out SNVT's

// init SNVT's
LonInMapping( InMapTable[0], InMapsize);
LonOutMapping(OutMapTable[0], OutMapsize);

// set SelfDocString
var SelfDocBuffer: buffer[20];      moveConst( SelfDocBuffer[0], "Vorlage
LON SC");
var SelfDocsize : word ;            moveConst( SelfDocsize, 14);
// init SelfDocString
LonSelfDocString( SelfDocBuffer[0], SelfDocsize);
// note: new init will activate only if ProgramID or SNVT types are
changed,
// like Configmode or Testmode. In this case also bindings will be lost!

// define ProgramID, default = SC000000 (SD... for UG512 V1.82 with new
LON-module
Set ( LONProgramID , 123456 ) ; // SC123456, SC = normal mode
// CM006969 Configmode
// TM004711 Testmode

// define BusMethod for fieldbusIN
Set ( BusdataTriggerMethod , DataChange ) ; // new fieldbus IN data
only if changed
moveconst (OnChange, 1); // 1 = TRUE, 0 = FALSE
if OnChange equal TRUE then :LonInitEnd;

Set ( BusdataTriggerMethod , BusEvent ) ; // new fieldbus IN data on
BusEvent

```

```

// note: BusEvent - you receive the same data and you can read out from
wich index
:LonInitEnd;
//***** LON init end
*****
Set ( RS_State_LED , StaticRed ) ;
//***** start fieldbus
*****
BusStart ;
Wait ( Bus_Active ) ;

Get ( BusInputLen16 , wBusInSize ) ;
Get ( BusOutputLen16 , wBusOutSize ) ;

Set ( RS_State_LED , RedGreenFlashing ) ;
//***** main
*****
:start ;

```

```

Get ( BusDataChanged , bDestination ) ;// sind geänderte Busdaten
gekommen ?

```

```

if bDestination equal b0 then :CheckRS_In;

```

```

ReadBus ( aBusInBuf[0] , wBusInSize ) ;

```

This routine is an extract from the example „LON_SC.dss“, that can be found in the Protocol Developer under menu „New“.

More LON Script examples are to be found in the folder „Examples“.

13.4 LON-specific parameters

```

// Bus specific -
// LON
// Lon Input SNVT Mapping
LonInMapping ( a_Source[0] , w_NumberChar ) ;
// Lon Output SNVT Mapping
LonOutMapping ( a_Source[0] , w_NumberChar ) ;
// Lon Self Documentation String
LonSelfDocString ( a_Source[0] , w_NumberChar ) ;

Set ( BusdataTriggerMethod , BusEvent ) ; // DataChange

// BusStart:
BusStart ;
// Device Control -
// Wait State:
Wait ( Bus_Active ) ;
//*****
*****

Get ( AvailableBusData , w_word ) ; // - 0..BusInputSize
Get ( BusDataChanged , b_byte ) ;

```

```

// Read Busdata:
  ReadBus ( a_Destination[0] , w_NumberChar ) ;
// Write Busdata:
  WriteBus ( a_Source[0] , w_NumberChar ) ;
// Wait for changing Busdata:
  WaitBusDataChanged ( w_Timeout , w_NumberChar ) ;

// Get Parameter:                               Set Parameter
// =====                                     =====
Get ( BusInputSize , b_byte ) ;                 // 0..255
Get ( BusOutputSize , b_byte ) ;               // 0..255

// supported from ScriptRev 26 and higher
Get ( BusInputLen16 , w_word ) ;               // 0..512
Get ( BusOutputLen16 , w_word ) ;             // 0..512

//.....
.....
// LON only:
// -----
Get ( LONProgramID , l_long ) ;                 Set ( LONProgramID , 0 ) ; // 0
.. 999999
Get ( CHANGED_SNVT_INDEX , wDestination ) ;
SetByVar ( CHANGED_SNVT_INDEX , wDestination ) ; // -1..255 (UG512)
                                                // -1..61 (UG 62)

//*****
*****

```

13.5 Darstellung der Daten in LONWorks

Any standard-compliant LONWorks participant can exchange data with the Gateway. It is also possible to use very "simple" master connections owing to the data structure.

After programming, the participants are connected through network variables. If the user has set the Flag 'Length byte', then it is the maximum length of the user data.

The user also configures the configuration telegram in the "Binding".

13.5.1 Network variables (NVTs)

The SNVTs are assigned by the user as desired in the Script.

```
var InMapsize : word;
var OutMapsize : word;
var InMapTable: buffer[62];
var OutMapTable: buffer[62];

moveConst( InMapsize, 1);
moveConst( OutMapsize, 1);
moveConst( InMapTable[0], #83 );
moveConst( OutMapTable[0], #83 );

LonInMapping( InMapTable[0], InMapsize);
LonOutMapping(OutMapTable[0], OutMapsize);

var SelfDocsize : word ;           moveConst( SelfDocsize, 10);
var SelfDocBuffer: buffer[60];
moveConst( SelfDocBuffer[0], "SC UNIGATE" );
LonSelfDocString( SelfDocBuffer[0], SelfDocsize);

Set ( LONProgramID , 943325 ) ;
```

13.5.2 Determination of a LON-Gateway's update time

In order to determine the average update time of a LON-Gateway, the following experimental setup was made:

Two LON-Gateways with 30 Input- and 30 Output-SNVTs each were connected with each other via LON (FTT10A system with 78 kbit/s) and a Binding between the 30 Input-SNVTs of the one Gateway and the 30 Output-SNVTs of the other Gateway was created. At this Binding for all SNVTs "Sending without acknowledgement of receipt" was selected since it is the only way to measure the plain Gateway-delay time. All SNVTs had a length of 2 bytes.

The measuring was started by a simultaneous change of all 30 SNVTs with the Script command "WriteBus()" and by means of a Protocol Analyser the period between the single SNVT-Updates on the LON-Bus were observed.

With it the average update time for the LONWorks modules UNIGATE 512 and UNIGATE 62 is 10 ms for the unconfirmed service.

At these measurings neither the times caused by the Script are taken into consideration nor the transmission times via the serial interface since these times are application specific.

Furthermore in a real environment the utilization capacity of the bus must be considered as well as potential response times with confirmed connections.

For more detailed information on the topic "Benchmarks" we like to refer to the book "LON-Technologie" [Hüthig-Verlag] (chapter 14).

14 Technical data

14.1 Device data

The technical data of the module is given in the table below.

No.	Parameter	Data	Explanations
1	Location	Switch cabinet	DIN-rail mounting
2	Enclosure	IP20	Protection against foreign bodies and water to IEC 529 (DIN 40050)
3	Service life	10 years	
4	Housing size	23 x 115 x 116 mm (screw-plug-connector included) 23 x 115 x 100 mm (screw-plug connector not included)	W x D x H
5	Installation position	Any	
6	Weight	130 g	
7	Operating temperature	-40°C ... +85°C	
8	Storage/transport temperature	-40 °C ... +85 °C	
9	Atmospheric pressure during operation during transport	795 hPa ... 1080 hPa 660 hPa ... 1080 hPa	
10	Installation altitude	2000 m 4000 m	Unrestricted Restricted - Ambient temperature ≤ 40°C
11	Relative humidity	Max. 80 %	No condensation, no corrosive atmosphere
12	External power supply	10..33 V DC	Standard power supply unit to DIN 19240
13	Current consumption at 24 VDC	Typ. 30 mA max 40 mA	At 10V. typ. 85 mA
14	Reverse voltage protection	Yes	But does not function!
15	Short-circuit protection	Yes	
16	Overload protection	Poly-switch	Thermal fuse
17	Undervoltage detection (USP)	≤ 9 V DC	
18	Emergency power supply	≥ 5 ms	Device fully operable

Table: Technical data of the module

14.1.1 Interface data

The table below lists the technical data of the interfaces and ports on the device. The data has been taken from the corresponding Standards.

	Interface designation	LONWorks	RS232-C	RS485/RS422
No.	Physical interface	FTT-10A	RS232-C	RS485/RS422
1	Standard	Echelon	DIN 66020	EIA Standard
2	Transmission mode	→ Difference signal	Asymmetrical asynchronous serial full duplex → Level	Symmetrical asynchronous serial half-duplex full duplex at RS422 → Difference signal
3	Transmission method	Master / master	Master / slave	Master / slave
4	Number of users: - Transmitters - Receivers	32.385 32.385 split in 127 nodes/subnet, 255 subnets/domain	1 1	32 32
5	Cable length: - Maximum - Baud rate-dependent	2000 m (with repeater) max. nodes 500 m	15 m no	1200 m <93.75 kBd → 500 m 625 kBd → 250 m
6	Bus topology	Free	Point-to-point	Line
7	Data rate: - Maximum - Standard values	78 kbit/s 78kbit/s	120 kBit/s 2.4 k/B 4.8 k/B 9.6 kBit/s 19.2 kBit/s 38.4 kBit/s	625 kBaud 2.4 kBit/s 4.8 kBit/s 9.6 kBit/s 19.2 kBit/s 57.6 kB 312.5 kB 625 kB
8	Transmitter: - Load - Maximum voltage - Signal, unloaded - Signal, loaded		3 ... 7 kΩ ± 25 V ± 15 V ± 5 V	54 Ω - 7 V ... 12 V ± 5 V ± 1.5 V
9	Receiver: - Input resistance - Max. input signal - Sensitivity		3 ... 7 Ω ± 15 V ± 3 V	12 Ω - 7 V ... 12 V ± 0.2 V
10	Transmit range (SPACE): - Voltage level - Logic level		+ 3 ... + 15 V 0	- 0.2 ... + 0.2 V 0
11	Transmit pause (MARK): - Voltage level - Logic level		- 3 ... -15 V 1	+ 1.5 ... +5 V 1

Table: Technical data of the interfaces and ports on the module

15 Commissioning guide

15.1 Note

Only trained personnel following the safety regulations may commission the UNIGATE®.

15.2 Components

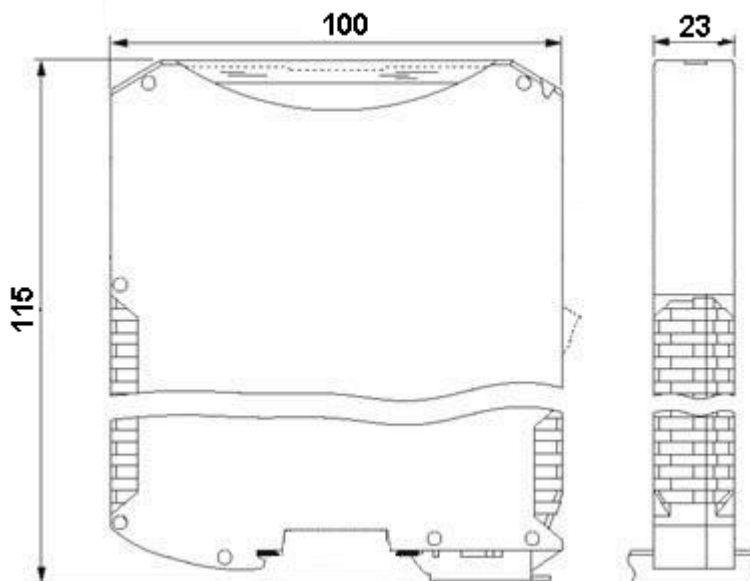
You will require the following components to commission the UNIGATE®:

- UNIGATE®
- Connection cable from gateway to the process
- Connector for the LON-bus-connection to the Gateway
- LONWorks (twisted pair) cable (generally this cable is already installed on site!)
- 10..33 V DC power supply (DIN 19240)
- XIF-file and user manual can be ordered separately or downloaded free of charge from our homepage at www.deutschmann.com.

15.3 Installation

The UNIGATE® CL - LON module features protection type IP20 and is thus suitable for switch cabinet use. The device is designed for snapping onto a 35 mm DIN-rail.

15.4 Dimensional drawing UNIGATE® CL - LONWorks



15.5 Commissioning

It is essential that you perform the following steps during commissioning in order to ensure that the module operates correctly.

15.6 LONWorks connection

Connect the device to the LONWorks at the interface labelled "LONWorks".

15.7 Connection to the process device

Please also read the manual for the process device when commissioning the process device.

15.8 Connecting the supply voltage

Please connect 10..33 DC voltage to the terminals provided for this.

15.9 Shield connection

Earth the top-hat rail onto which the module has been snapped.

15.10 Project planning

Use a project planning tool for Binding, i. e. the Pathfinder from TLON (www.tlon.de).

15.11 Literature

In order to get into the subject LON bus fast and intensively as well as for the operating mode of the available ASICs, we recommend the book "Verteilte Systeme in der Anwendung der LON-Technologie", ISBN: 3-7785-2581-6 (Hüthig Verlag) or "LONWorks Technology - An introduction) by F. Tiersch DESOTRON publishing company Dr. Günter Hartmann & Partner, GbR ISBN: 3-932875-11-7 (English version) or LONWorks 'Installationshandbuch', ISBN 3-8007-2575-4.

16 Servicing

Should questions arise that are not covered in this manual you can find further information in our

- FAQ/Wiki area on our homepage www.deutschmann.com or directly in our Wiki on www.wiki.deutschmann.de

If your questions are still unanswered please contact us directly.

Please note down the following information before calling:

- Device designation
- Serial number (S/N)
- Article number
- Error number and error description

Your request will be recorded in the Support center and will be processed by our Support Team as quickly as possible (Usually in 1 working day, rarely more than 3 working days.).

The technical support hours:

Monday to Thursday from 8 am to midday and from 1 pm to 4 pm, Friday from 8 am to midday. (CET)

Deutschmann Automation GmbH & Co. KG
Carl-Zeiss-Straße 8
D-65520 Bad-Camberg
Germany

Central office and sales department +49 6434 9433-0
Technical support +49 6434 9433-33

Fax sales department +49 6434 9433-40
Fax technical support +49 6434 9433-44

E-mail technical support support@deutschmann.de

16.1 Returning a device

If you return a device, we require as comprehensive a fault/error description as possible. We require the following information in particular:

- What error number was displayed?
- What is the supply voltage (± 0.5 V) with Gateway connected?
- What were you last doing or what last happened on the device (programming, error on power-up,...)?

The more precise information a fault/error description you provide, the more exactly we will be able to pinpoint the possible causes.

16.2 Downloading PC software

You can download current information and software free of charge from our Internet server. <http://www.deutschmann.com>

17 Annex

17.1 Explanations of the abbreviations

General

CL	=	Product group CL (Compact Line)
CM	=	Product group CM (CANopen Line)
CX	=	Product group CX
EL	=	Product group EL (Ethernet Line)
FC	=	Product group FC (Fast Connect)
GT	=	Galvanic separation RS-side
GY	=	Housing color gray
MB	=	Product group MB
RS	=	Product group RS
SC	=	Product group SC (Script)
232/485	=	Interface RS232 and RS485 switchable
232/422	=	Interface RS232 and RS422 switchable
DB	=	Additional RS232 DEBUG-interface
D9	=	Connection of the RS through 9-pin D-SUB instead of 5-pin screw-plug connector
PL	=	Board only without DIN-rail module and without housing cover
PD	=	Board only without DIN-rail module and with housing cover
AG	=	Gateway installed in a die-cast aluminum housing
EG	=	Gateway installed in a stainless steel housing
IC	=	Product group IC (IC-design DIL32)
IC2	=	Product group IC2 (IC-design DIL32)
IO8	=	Option I/O8
16	=	Script memory expanded to 16KB
5V	=	Operating voltage 5V
3,.3V	=	Operating voltage 3.3V

Fieldbus

CO	=	CANopen
C4	=	CANopen V4
C4X	=	CANopen V4-version X (see comparison table UNIGATE [®] IC for the respective product)
DN	=	DeviceNet
EC	=	EtherCAT
EI	=	Ethernet/IP
FE	=	Ethernet 10/100 MBit
FEX	=	Ethernet 10/100 MBit-version X (see comparison table UNIGATE [®] IC for the respective product)
IB	=	Interbus
IBL	=	Interbus
LN62	=	LONWorks62
LN512	=	LONWorks512
ModTCP	=	ModbusTCP
MPI	=	Siemens MPI [®]
PL	=	Powerlink
PN	=	Profinet-IO
PBDP	=	ProfibusDP
PBDPL	=	ProfibusDP-version L (see comparison table UNIGATE [®] IC for the respective product)

product)
PBDPX = ProfibusDP-version X (see comparison table UNIGATE® IC for the respective product)
PBDPV0 = ProfibusDPV0
PBDPV1 = ProfibusDPV1
RS = Serial RS232/485/422

17.2 Hexadecimal table

Hex	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

