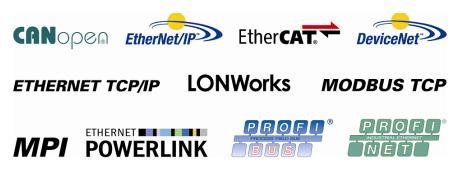
# THE RESULT MUST AGREE – BUS-TECHNOLOGY MADE BY DEUTSCHMANN!

# UNIGATE<sup>®</sup> CL

## A solution for all devices with serial interface



The intelligent protocol converter for:















ETHERNET TCP/IP

**LONWorks** 

**MODBUS ASCII** 

**MODBUS RTU** 

MODBUS TCP



PROFILI PUS

PROF	1
INDUSTRIAL ETHERNET	

RS 232

RK512

3964R

# The intelligent

## **Protocol converter UNIGATE® CL**

#### A solution for all devices with serial interface

The UNIGATE® Gateway by Deutschmann connects devices, such as automation components via their serial interface to Fieldbuses and Ethernet. Here UNIGATE® takes on the task of a protocol converter and converts the different interfaces.

UNIGATE® CL features the interfaces RS232, RS485 and RS422 (also the SSI-protocol is supported). With the type series UNIGATE® CL you can either configure the adaptation to your product or to the protocol or the capability of the UNIGATE® CL protocol converter can be expanded considerably through the generation of a Script. Our approved configuration tool WINGATE can be used for the configuration or you can generate your intelligent Script by means of the PC-tool "Protocol Developer", which is available free of charge. In both cases changes in the terminal equipment's Firmware are not necessary.



The Deutschmann Gateway UNIGATE® CL uses the complete data capacity of the respective Bus and supports all speeds. The design of the series UNIGATE® CL is meeting the standards. The Bus connection for Profibus + CAN is made via a D-SUB socket at the front, for all other Buses from the bottom side in accordance with the respective standard. The connection of the serial side is made through a screw-plug connector on the upper side. Numerous diagnosis LEDs signalize the state of the device on the Bus side as well as on the serial side.

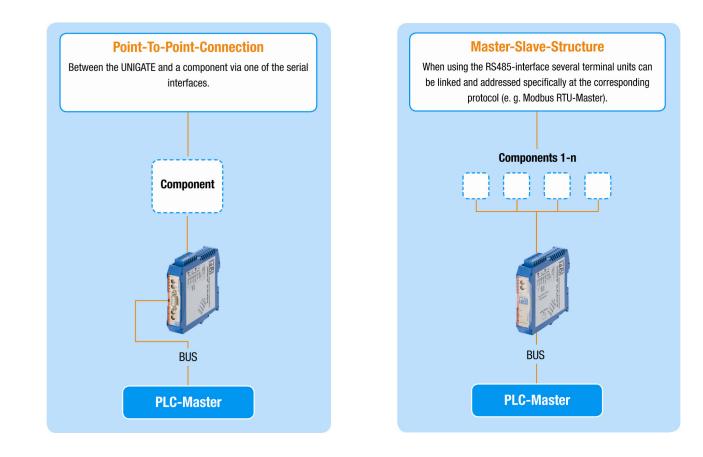
#### Your advantage

With Deutschmann UNIGATE® CL-modules you bring existing components (also older devices) into modern networks. As device manufacturer you save the development of the interfaces based on the corresponding Fieldbus or Ethernet. Due to the standardization of the Deutschmann UNIGATE® CL-series configurations and Scripts that have been generated once can be used practically without changes for other Fieldbuses or versions based on Ethernet from the UNIGATE® CL-series. The Firmware adaptation of your own terminal-units becomes unnecessary.

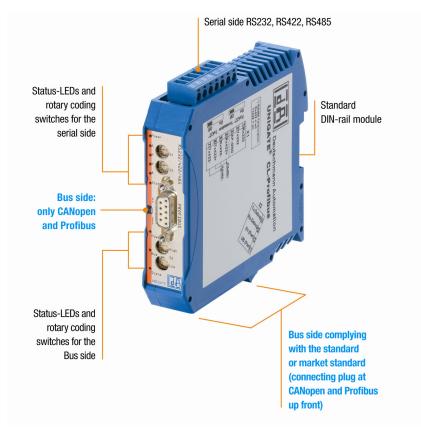
#### Facts that speak for the protocol converters UNIGATE® CL by Deutschmann:

- Available for most Fieldbuses and Industrial Ethernet-versions
- RS232, RS485- and RS485-interface on board
- Same structure on the serial side in all Bus-versions
- The Fieldbus- or Ethernet-side corresponds to the standards or the commercially available versions
- SSI-protocol is supported
- Integrated isolation on the Bus side; optionally also on the serial side
- Configuration of the module via configuration tool WINGATE®
- Free programming by means of the PROTOCOL DEVELOPER and the Deutschmann Script-language
- Adaptation to your Firmware not required
- Additional Debug-interface on board of interest in case the Script language is used
- Modern, slim DIN-rail module
- Identical dimensions for all Bus-versions
- Customized design possible, e.g. your own logo
- Extensive voltage range from 10.0 to 33.0 Volt
- If the RS485-interface is used, several terminal units can be operated in an own network (such as Modbus RTU)

### **Application example**



### **UNIGATE® CL design**



Cam Controls | Fieldbus Gateways | Industrial Ethernet Products



# **Configuration tool**

WINGATE

# DeviceNet<sup>®</sup>





#### ETHERNET **POWERLINK**

ETHERNET TCP/IP

#### **LONWorks**

**MODBUS ASCII** 

**MODBUS RTU** 

**MODBUS TCP** 



PROFI BUSI



**RS 232** RS 485/422

**RK512** 

3964R

The UNIGATE® CL-series can be configured with the comfortable configuration tool WINGATE. All models have the standard protocols 3964R and 3964, Modbus ASCII and Modbus RTU (Master- and Slave-operation possible) as well as a Universal 232-protocol for transparent data exchange available. The selection and configuration of the protocols is carried out via the WINGATEsoftware.

✓ more items visible	more <u>i</u> tems editable
Parameter	Value
Software revision	V 2.0
Device type	CANopen (Script)
Script revision	30
Serial Number	35540022
Store Serial Number	No
Protocol	Universal 232
232 Start character	00
232 Length	No length byte
232 End character	FF 25
232 RX Timeout (10ms)	20 No Checksum
232 Checksum	
Start bits	1 8
Data bits Char bits	8
Stop bits	Even
Parity Baudrate	19200
232 Interface	485
Blocklength fieldbus input	28
Blocklength fieldbus output	
Data exchange	On Event
	OILTCIL

Picture 1: WINGATE main window



Picture 2: Standard protocols



Picture 3: Sub-window: parameter selection

#### Our customers are looking for flexible solutions: With good reason.

You are rather inflexible with the usual configuration tools for Protocol Converters and Gateways and you have to stick to standard specifications. Adapting the Firmware to a certain protocol or observing a fixed way of proceeding is unwanted or too extensive, particularly in case of an existing software. Subsequent changes and customized adaptations are hardly possible. A profound knowledge of the buses is partially expected.

For that reason the Deutschmann Script language came into being in the year 1999 already. Our customer only has to process the data of the bus and he does not have to take care of the specific features of the Fieldbuses and Industrial Ethernet buses. For this programming he does not have to possess knowledge of programming languages, but he generates a Script by means of a Windows-tool 'PROTOCOL DEVELOPER'.

#### 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 Deutschmann Script language is strongly based upon tools, such as C++. In case you do not want to generate the Script yourself, we are also offering this service.

#### What can be done with a Script device?

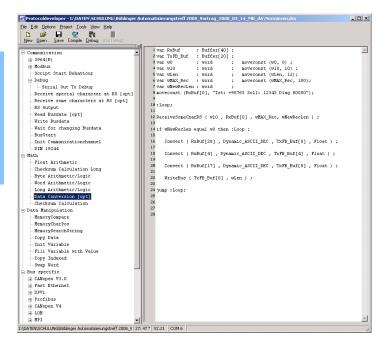
Our Script devices are in the position to process a lot of commands. In this case a command is always a small firmly outlined task. All commands can be put into classes or groups. A group of commands deals with the communication in general. This group's commands enable the Gateway to send and receive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

# Following please find the explanation of some command groups on the left side of the adjacent example window:

Declarations	Variable declaration
FlowControl	Subfunction calls, jumps, branches
Math	Mathematical functions, data conversions
Communication	Send and receive data
Device control	Set and read parameters. Exemplary the baud rate for
	the serial interface is mentioned here.
Bus specific	Here commands are placed that enter bus-specific values.

# The amount of tasks, that can be processed with a Script is virtually unlimited. Scripts, that are conceivable:

- automatically acquire data from one participant at the serial interface, edit this data and then present the edited data in the bus over and over again
- carry out actions only in case the bus data changes
- carry out time-controlled actions
- inform of communication states
- exchange data between 2 serial participants (RS485) and present the state in the bus



Picture 1: Protocol Developer main window

By means of this short enumeration it becomes clear that the Scripts are a flexible solution for your problems. Data can be processed, converted and arranged on both sides (on the RS-side and the Bus side as well). That way the Script basically offers the chance to cope with almost all requirements.



Picture 2: Function selection window













ETHERNET TCP/IP

LONWorks

MODBUS ASCII

MODBUS RTU

MODBUS TCP



PROF DUS





**RK512** 

3964R

#### The use of the Protocol Developer

On the right side of the window (picture 1) you see a Script, that shows the simple structure of the syntax.

The Protocol Developer is meant as a tool for an easy generation of a Script for our Script Gateways. Its operation is exactly aimed at this use.

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, is displayed, and after entering the values the right text is automatically added to the Script.

#### Compiling

Debugging

Break pointSingle-step

Error indication

be checked quickly.

further functions, such as

Display of the variables and their values

After the Script was generated it is compiled. The resulting code is loaded into the device afterwards. This can be carried out with the PROTOCOL DEVELOPER. A Script download tool is also available. The compiled code is very small and because of the large Script memory of the UNI-GATE® products also extensive Scripts are possible.

The Debugger enables the control of a UNIGATE® Gate-

way that is in the Debug-mode. For convenient Debugging

are made available. That way also extensive Scripts can

npilieren	
File: Editor	
Phase: 3	
Line 24	
Errors/Warnings/Notes 0/0/0 0,0 s	
Compilation successful, 124 Bytes code generated.	
v	
Close	7

Picture 3: Compilation window with error message and file size

→ Start	🗢 4 🔶 K		
	Stop Single Step Continue Reset		
	<pre>% moveconst (RxBuf[0], "Ist: +98765 Soll: 12345 Diag 80000");</pre>		_
	9		
	10 :Loop;		
	12 ReceiveSomeCharRS ( w10 , RxBuf[0] , wMAX Rec, wNewRecLen ) ;		
	13		
1	<sup>14</sup> if wNewRecLen equal w0 then :Loop ;		
	15		
	16 Convert ( RxBuf[28] , Dynamic_ASCII_DEC , TxFB_Buf[0] , Float ) 17	;	
	18 Convert ( RxBuf[4] , Dynamic &SCII DEC , TxFB Buf[4] , Float ) ;		
	19		
2	Convert ( RxBuf[17] , Dynamic_ASCII_DEC , TxFB_Buf[8] , Float )	;	
	21		
	22 WriteBus ( TxFB_Buf[0] , wLen ) ; 33		
	24 hump :Loop;		
	5		
2	26		
	27		
2	28		
variabl	e value	type	Size
RxBuf	lst: +98765 Soll: 12345 Diag 80000<00><00><00><00><00><00>	BUFFER	40
TxFB_E	Buf 0x47 0x9C 0x40 0x00 0x47 0xC0 0xE6 0x80 0x46 0x40 0xE4 0x00 0x00 0x00 0x00 0x00 0x0	BUFFER	20
11110_1			

Picture 4: Debug window with variables and their contents

	ReceiveSomeCharRS		Î
0	Systax		
	ReceiveSomeCharRS ( w_Timeout , a_ReceiveData[0] , w_NumberCharToReceive [, w	_CharsReceived]) ;	
0	description		
0	This commands waits for receipt of given number of characters. The time between any ch used. All Characters received are stored in variable destination. This variable must be big example	aracters must not exceed time milliseconds. If Time variable contains 0 no timeout enough to hold all incoming characters, for example a buffer variable.	control is
	vor w_Timeout : word; vor w_NumberCharToSeceive : word; vor s_buffer : Buffer[10]; //optional : word; //the number of chars	received up to a timeout occures	
	<pre>KoveConst ( w_NumberCharToReceive, 3 ) ; KoveConst ( w_Timeout, 1000 ) ; // Assume data string "Hello" not yet read in the gateway RoceiveGoncharHS (w_Timeout , a_ReceiveDists[0] , w_Numb</pre>		
•	<pre>// 3 bytes are road, data now contains "Hel", // KS input Suffer still contains data "lo" // which is the rest of the data "Hello" see also</pre>	CANoperV41.dss COV4_IC_Testmode.dss DeviceNet.dss Ether(AT_dss	OK Abbruch
	<pre>// Receive special character at RS: ReceiveSpecialCharRS (b_Char, w_Timeout, a_ReceiveData[0], w_NumberC // RS Output SendRS (c. 6 source[0], w_NumberBrites ) :</pre>		Hilfe
	sote	Interbus.dss	
	If you are using 9 databits every character received is stored as 2 bytes in the receive NumberToReceive to 8.	LON.dss     LON_62_SNVT_UNVT.dss     MPI_Siemens.dss	
0.4	2011. Tare optimes: OS, 2005 04 13 (Beldon, Substitute=BC3)	Profibus.dss Profibus_Tgl_FBLen.dss Profinet.dss	

#### Support

The PROTOCOL DEVELOPER has a context-sensitive help function, that provides an extensive description of each Script command.

Templates for different tasks and bus versions can directly be taken over and adjusted to your requirements.

Picture 6: Extract from the templates

Picture 5: Online help

#### **Example Scripts**

For each Script command the scope of delivery of the free PROTOCOL DEVELOPER also includes a commented example Script. More support is available from our free Hotline and on our website the current versions of our instruction manuals and the software tools are also available free of charge (www.deutschmann.de).

nen in: 🔁 Example			-	← 🗈 📸 📰•
DAVADEMO		Example_Custom0001.dss	example_IC_CO_set_ID_and_baudrate_with_basebord.dss	example_SC_CO_V3.dss
Server-Side-Include-Demo		example_databits_9_bit.dss	example_IC_functions_DirectAccess.dss	example_SendEMail.dss
t 🗀 Universal		example_define_memory.dss	example_IC_RS_ModbusMaster.dss	example_Set_HTML_String.dss
E D include_optional_alleBusse	dss	Example_delay.dss	example_if_BitSetdss	example_SetSerialInBufLen.dss
<ul> <li>_include_optional_Basisboa</li> </ul>	rd_Init-ShiftRegister.dss	example_DIN19244 (Master).dss	example_InitFastEthernet.dss	example_ShiftRegister_Basis_FB
<ul> <li>_include_optional_Basisboa</li> <li>_include_optional_DeviceNet</li> </ul>	t.dss	example_DN_IC_init.dss	example_InitFIFO.dss	example_ShiftRegister_Basis_IC.
Include_optional_EtherCA	f.dss	example_DN_Init.dss	example_LON.dss	example_ssi.dss
_include_optional_IC_CANc	pen_Busparameter.dss	example_DP_event_para.dss	example_math_conv_ASCII_Dyn.dss	example_SwapWord.dss
_include_optional_IC_Devic	eNet_Busparameter.dss	example_EIP.dss	example_math_conv_float_real.dss	example_Sys_Error_Mask.dss
_include_optional_IC_Ether	CAT_Busparameter.dss	Example_EmptyFile.dss	example_MemoryCharPos.dss	example_Transparent_IC.dss
teioninclude_optional_IC_Profit	us_Busparameter.dss	example_EN_Loop.dss	example_MemoryCompare.dss	example_Transparent_IC_CO.ds
include optional Profibus.	dss	example Errorcode.dss	example MemorySearchString.dss	example Transparent IC DN.ds
include optional Profinet.	dss	example EthernetFE ModbusTCP withTimeoutContr.dss	example ModBus Slv.dss	example Transparent IC Profib
CO V30 IC example 16Rs	PDO_16TxPDO.dss	example_eventhandler.dss	Example_Modbus-ASCII (Master).dss	Example_VersionInfo.dss
Convert_PrintfANSI_C.dss	-	example FB-AN.dss	Example_Modbus-RTU (Master).dss	example WaitBusDataChanged.
EvaKit.dss		example FB-FB autodetect.dss	Example Modbus-RTU (Slave) für IC Basisb.dss	example_WaitBusDataChanged_
example 25 Reset.dss		example FB-FB-CO.dss	Example Modbus-RTU (Slave).dss	Rexample Watchdog.dss
example_3964.dss		example FB-FB-CO-V4 32Byte IO.dss	example ModbusTCP.dss	example XY Testmode.dss
example 3964R.dss		example FB-FB-DN.dss	example MPI RS-3964R.dss	▶ ICDPInitConfiaBytes.dss
example_ASCII_IP.dss		Example FB-FB-EN.dss	example MPI RS Trans.dss	MPI ModSlv01.dss
Rexample Basisboard Set R	S EB Statel ED.dss	Example FB-FB-FE ModbusTCP-IP.dss	example ParamBusStateLED.dss	MPI ModSly02.dss
Rexample Basisboard Set R		Example FB-FB-IB32.dss	example Printf ANSI C.dss	MPI ModSly03.dss
example CalcChecksum.ds		Example FB-FB-IB32 mit Tgl-FBLenByte.dss	example Profibus DiagnoseData.dss	MPIexample.dss
example Calculate.dss		Pexample FB-FB-PB.dss	example Profibus DPV1.dss	SignedIntToFloat.dss
example Calculate GetBit	31 vml dec	example FBRS Init.dss	example Profibus GetProfibusConfig.dss	Transparent422.dss
example_Calculate_long.ds		example FE ModbusTCP.dss	example_Profibus_t8M_Functions.dss	Transparent_SC_Profibus_V_1_0
example CAN Laver 2.dss		example FE WebServer.dss	example Profibus IC DPV1.dss	UNI232_SC_DP.dss
example_CAN_Layer_2.uss		example FieldbusID.dss	example Profibus ParameterData.dss	■ UNI232_5C_DP.USS
example_CC_RSH00B05k10		example_reduusio.uss  example_float_math.dss	example_Profinet-CL.dss	
Example_CO_IC_V4_NewPA	ara_2007_12.dss	example_noac_matn.dss  example_Get_FieldbusLen.dss	Example_Profinet-CL.dss	
Example_CO_V4.dss		<pre>example_uec_melodusten.dss</pre>	Example_Promec-IC.dss     example_ReadWriteEEROM.dss	
example_CO_V30_IC_defa	JIC. OSS	example_gw_control_set_RS_StateLED.dss	example_R5485.dss	
example_Convert.dss		example_gw_control_set_select_ID.dss	example_RS_CL_ModBusRTUSIv.dss	
Example_copy_data.dss		example_gw_control_set_warningTime.dss	Example_RS_output.dss	
example_CopyIndex.dss		example_IB_32_ModbusMaster.dss	example_RSModBusRTUSIv.dss	
•				<u>•</u>
Dateiname: Normieren.	dss			▼ Ŭffnen

Picture 7: Extensive library for example Scripts

#### That speaks for the Script language by Deutschmann:

- No changes required in your own Firmware
- · Flexible and powerful Script language; especially generated for the bus communication
- Easy to handle. Based upon current tools, such as C++
- On request customized commands, e. g. in case functions are missing or an optimization is required for time-critical applications
- You can generate the Scripts yourself or Deutschmann is also offering the Script-generation as service
- Extensive support by means of help functions, templates, examples, hotline and workshops
- In the initial state devices can already be equipped with your Script

## UNIGATE® CL Starterkit

Deutschmann Starterkits are arranged in a way, that enables you to connect your product to the selected Fieldbus or Industrial Ethernet at the laboratory bench fast and low-priced. In order to meet the customers' requests our kits are split in two:

- The **Starterkit** contains the Gateway UNIGATE® CL in accordance with the selected Bus, the corresponding cables, the voltage supply as well as a CD with the software tools and a manual.
- The **Add-On** contains a Master simulation, so that the Bus side can also be tested in a fast and cost-saving way. Depending on the Bus technical literature on the selected Bus might also be included in the package (deviations are possible). Certainly already existing Buses can be used instead of the Add-On.



Picture 1: Starterkit (example ProfibusDP)



Picture 2: Add-On (example ProfibusDPV0)

You receive the configuration tool WINGATE, the Protocol Developer for Script generation as well as extensive Script examples, device description files and much more free of charge. For further information and the mentioned tools please log on to our website **www.deutschmann.com** 

General technical data for all versions					
• Fixing	DIN-rail module with integrated grounding				
Interfaces	RS232, RS422, RS485 and SSI on board				
<ul> <li>Baud rates of the serial interfaces</li> </ul>	110 Baud up to 625 KBaud (partially up to 2 MBaud)				
<ul> <li>RS-connection and supply voltage</li> </ul>	Through screw-plug connector				
Input buffer RS-Seite	1024 bytes input- and output-buffer each				
• Diagnosis	6 LEDs for diagnosis of the serial application interface				
• Bus ID	Adjustable				
Operating voltage	10 to 33 Volt				
Protection type	DIN-rail module IP24				
Dimensions	23 x 100 x 115 (B x H x D), connector not included				
Temperature range	0°C to +55°C (-20°C to +75°C on request)				
Certificates	CE and bus-specific certifications				
Galvanic division	Optional for the serial side of all versions available, standard on the bus side				
Bus termination resistor	Connectable for the Fieldbuses and RS				

Bus-specific technical data									
UNIGATE® CL	Bus connection	Bus data	Bus baud rates	Bus ID	UNIGATE® CL	Bus connection	Bus data	Bus baud rates	Bus ID
CANopen	9 pin D-SUB connector	255 bytes I/O	Adjustable via DIP-switch	Adjustable via DIP-switch	MPI	9 pin D-SUB connector	240 bytes I/O	Adjustable via Script	Adjustable via rotary switch
DeviceNet	5 pin screw-plug connector	255 bytes I/O	Adjustable via DIP-switch	Adjustable via DIP-switch	Ethernet Powerlink	2xRJ45	1541 bytes I/O	100 Mbit	IP-address adjustable via rotary switch
EtherCAT	2xRJ45	1486 bytes I/O	100 MBaud	Permanent MAC-address, is automatically assigned	Profibus	9 pin D-SUB connector	244 bytes I/O	Automatic detection	Adjustable via rotary switch
Ethernet/IP	2xRJ45	500 bytes I/O	10/100 Mbit	IP-address adjustable via WINGATE	Profinet	2xRJ45	1500 bytes I/O	100 MBaud	IP-address is automatically assigned via the device's name
Fast Ethernet Modbus TCP	RJ45	1024 Byte I/O	10/100 Mbit	IP-address via WINGATE or Script	RS (Modbus, 3964R, RK512)	7 pin screw-plug connector	512 bytes I/O	Adjustable via Script	Adjustable via DIP-switch
LONWorks62	4 pin screw-plug connector	62 In und Out SNVTs 1024 bytes I/O	FTT-10A, 78 kBit/s	Permanent Neuron ID	LONWorks512	4 pin screw-plug connector	512 bytes I/O 256 In and 256 Out SNVTs	FTT-10A, 78 kBit/s	Permanent Neuron ID

Subject to technical changes. We do not accept liability for any misprints or errors.

Fieldbus Gateways

avs I Industri