

# WinAC Serial driver

Operator manual

V 1.7.11 • May 2013

Applikationen & Tools

Answers for industry.

**SIEMENS**

## **Industry Automation and Drives Technologies Service & Support Portal**

This article is taken from the Service Portal of Siemens AG, Industry Automation and Drives Technologies. The following link takes you directly to the download page of this document.

<http://support.automation.siemens.com/WW/view/de/48174374>

If you have any questions concerning this document please e-mail us to the following address:

[online-support.automation@siemens.com](mailto:online-support.automation@siemens.com)

[applications.aud.koe.nrh.rd@siemens.com](mailto:applications.aud.koe.nrh.rd@siemens.com)

# SIEMENS

## SIMATIC WinAC Serial driver

Operator manual

**Automation Task**

**1**

**Installation**

**2**

**Overview**

**3**

**The user interface**

**4**

**Error Codes**

**5**

**History**

**6**

## Warranty and Liability

### Note

The Application Examples are not binding and do not claim to be complete regarding the circuits shown, equipping and any eventuality. The Application Examples do not represent customer-specific solutions. They are only intended to provide support for typical applications. You are responsible for ensuring that the described products are used correctly. These application examples do not relieve you of the responsibility to use safe practices in application, installation, operation and maintenance. When using these Application Examples, you recognize that we cannot be made liable for any damage/claims beyond the liability clause described. We reserve the right to make changes to these Application Examples at any time without prior notice. If there are any deviations between the recommendations provided in these application examples and other Siemens publications – e.g. Catalogs – the contents of the other documents have priority.

We do not accept any liability for the information contained in this document.

Any claims against us – based on whatever legal reason – resulting from the use of the examples, information, programs, engineering and performance data etc., described in this Application Example shall be excluded. Such an exclusion shall not apply in the case of mandatory liability, e.g. under the German Product Liability Act (“Produkthaftungsgesetz”), in case of intent, gross negligence, or injury of life, body or health, guarantee for the quality of a product, fraudulent concealment of a deficiency or breach of a condition which goes to the root of the contract (“wesentliche Vertragspflichten”). The damages for a breach of a substantial contractual obligation are, however, limited to the foreseeable damage, typical for the type of contract, except in the event of intent or gross negligence or injury to life, body or health. The above provisions do not imply a change of the burden of proof to your detriment.

Any form of duplication or distribution of these Application Examples or excerpts hereof is prohibited without the expressed consent of Siemens Industry Sector.

# Table of Contents

<b>Warranty and Liability</b> .....	<b>4</b>
<b>1 Automation Task</b> .....	<b>6</b>
1.1 Overview .....	6
1.2 Needed knowledge .....	6
1.3 Required Hardware and Software Components .....	7
<b>2 Installation</b> .....	<b>8</b>
2.1 Quickstart .....	8
2.2 Install / Uninstall WinAC driver on runtime system .....	8
2.3 Installation WinAC driver on engineering system .....	9
<b>3 Overview</b> .....	<b>10</b>
<b>4 The user interface</b> .....	<b>11</b>
4.1 Initialization FB 7 – COM_INIT_WINAC .....	13
4.1.1 Parameter setting with the DB_COM_PARAMS .....	15
4.2 Send data FB3 – P_SEND_WINAC .....	17
4.3 Data receive FB2 – P_RCV_WINAC .....	18
4.4 Delete reception buffers FB12 – P_RESET_WINAC .....	19
4.5 Get V24-signals FB5 – V24_STAT_WINAC .....	20
4.6 Put V24 signals FB6 – V24_SET_WINAC .....	21
4.7 Variable table / Watch table .....	21
<b>5 Error Codes</b> .....	<b>22</b>
5.1 Error codes of WinAC serial driver .....	22
5.2 Error codes of WinAC ODK functionality .....	24
<b>6 History</b> .....	<b>26</b>

# 1 Automation Task

## 1.1 Overview

### Introduction

WinAC RTX runs on PCs which typically are equipped with serial interfaces (COM ports). The WinAC RTX standard functionality does not support a communication with these interfaces.

The WinAC serial driver enables the serial communication of WinAC with other communication partners with these interfaces.

The FBs of the WinAC driver use the FBs of the board CP 340-RS 232 C as template.

At the moment only the ASCII protocol is supported.

### Precondition

The WinAC serial driver is realized as Windows driver. It supports the operating systems Windows XP Prof., Windows Embedded Standard 2009 (XP Embedded), Windows 7 and WES7.

The advantage of this approach is that the WinAC serial driver supports all serial interfaces (on board, plug-in card etc.) known to the Windows operating system.

Windows is not a real-time operating system. Thus a jitter in the range of milliseconds is a typical value. This must be taken into account (typical 5-10ms) for the projecting of the sign delay time. The jitter depends on the PC hardware. The WinAC driver is designed for applications in the low or middle performance area.

## 1.2 Needed knowledge

To understand this document the following knowledge is needed:

- SIMATIC WinAC RTX 2010
  - SIMATIC Manager STEP7 V5.5
- or*
- TIA-Portal V11, SP2, Update4 / TIA-Portal V12

## 1.3 Required Hardware and Software Components

The described application is based on the following components:

### Hardware components

- Simatic Microbox IPC 427D  
with WES2009 (Windows XP embedded) or WES7 (Windows 7 embedded).

### Standard software components

- SIMATIC WinAC RTX 2010
- SIMATIC Manager V5.5, SP2
- or*
- TIA-Portal V11, SP2, Update4 / TIA-Portal V12

## 2 Installation

### 2.1 Quickstart

#### Run-time system

- Execute the batch file **Install.bat**

#### Engineering system

- Retrieve the demo project.
- Copy the following components from the demo project in the user project
  - FBs 2, 3, 5, 6, 7, 12 including their Instance-DBs 2, 3, 5, 6, 7, 12
  - DB10
  - SFB 65001 and SFB 65002
- Configure the connection in DB10
- Call FB7 COM\_INIT with the configuration
- If the connection exists with FB3 /2 data send/receive.

#### NOTICE

The WinLC\_SER\_COM DLL needs the DLL **msvcrt.dll**. It is part of the standard image of the Simatic Microbox PC427B/C. For other computer configurations it must be installed additionally, probably.

### 2.2 Install / Uninstall WinAC driver on runtime system

The installation of the WinAC driver is limited to the calling of the **install.bat**. It copies the driver DLL to the system32 directory

To uninstall the driver DLL the user has to execute **uninstall.bat**. It removes the driver from the above-mentioned directory.

#### Installation under Windows XP (embedded)

Under Windows XP one can start the **install.bat**. This works from USB stick, too.

#### Installation under Windows 7

For copying a file to system32 Administrator privileges are needed. Thus the **install.bat** has to be started with Administrator rights (right click – Run as Administrator). The Windows 7 UAC has to be confirmed with “Yes”.

This works from USB stick, too.



## 2.3 Installation WinAC driver on engineering system

On the engineering system the following components are needed:

- Documentation
- Example project Step7 / TIA Portal

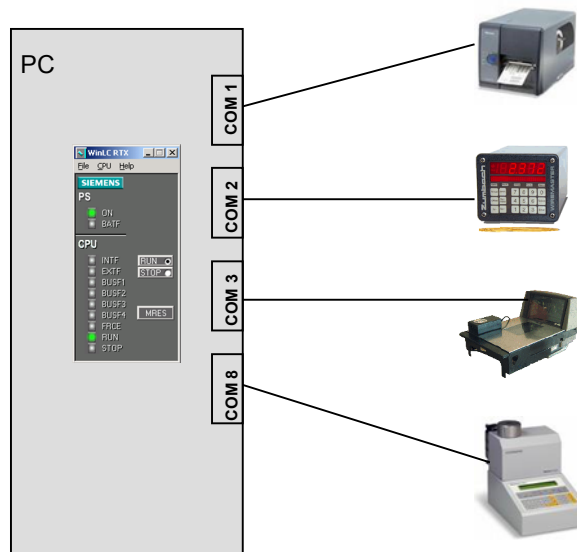
The installation package contains a Step7 and a TIA portal demo project. The corresponding function blocks have to be copied into your own project.

There is **no** installation needed of the driver on the engineering station.

### 3 Overview

The **WinAC serial driver** enables the WinAC application to communicate with up to eight communication partners simultaneously over serial interfaces of the PC. The communication partners are identified over the COM port connection (1 – 8)

Figure 3-1



This application does not guarantee real time behavior, because the driver DLL runs under Windows operating system.

## 4 The user interface

One goal of the development of the **WinAC serial driver** was to be as compatible as possible to the known FBs and FCs of the module CP 340-RS 232 C (**P\_SEND**, **P\_RCV**, **V24\_STAT**, **V24\_SET** and **P\_RESET**).

The function blocks of the WinAC driver are suffixed with prefix (WINAC) to avoid confusion with the CP340 library functions.

In addition to these FBs there is another FB 7 COM\_INIT\_WINAC. The parameter setting of the connections is stored in the DB\_COM\_PARAMS\_WINAC.

The documentation of the FBs 2,3,5,6 and 12 can be checked in the operator manual "SIMATIC S7 300 PtP coupling and configuration of CP 340" (04/2005, A5E00369892 01). The **WinAC serial driver** only supports the ASCII protocol.

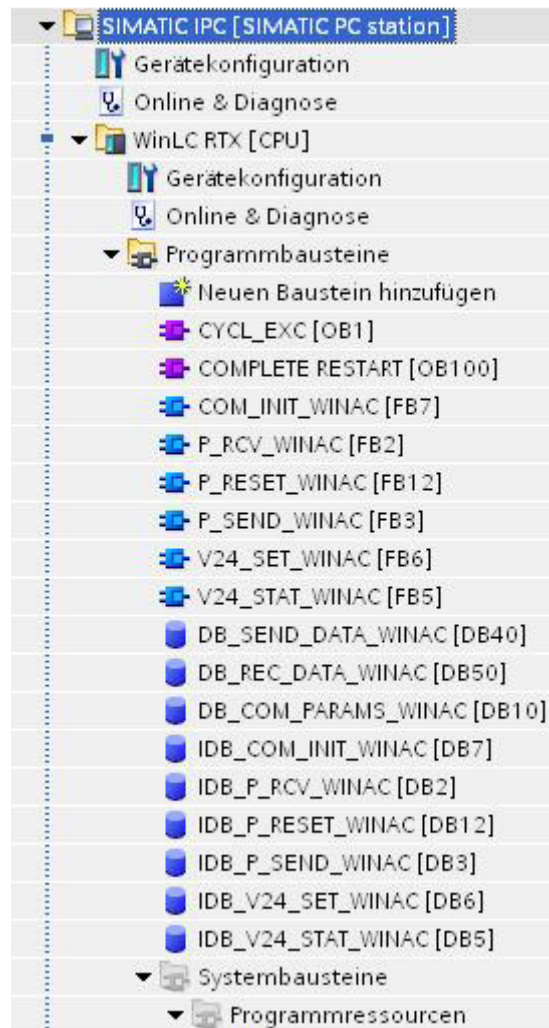
Beside the interfaces of the function blocks, the instance data blocks contain additional information.

The following figure shows an overview about the used components of the demo project.

Figure 4-1 Driver FBs in Step7 V5.5

Object name	Symbolic name	Created in language
Systemdaten	---	---
OB1	CYCL_EXC	FBD
OB100	COMPLETE RESTART	FBD
FB2	P_RCV_WINAC	STL
FB3	P_SEND_WINAC	STL
FB5	V24_STAT_WINAC	STL
FB6	V24_SET_WINAC	STL
FB7	COM_INIT_WINAC	STL
FB12	P_RESET_WINAC	STL
DB2	IDB_P_RCV_WINAC	DB
DB3	IDB_P_SEND_WINAC	DB
DB5	IDB_V24_STAT_WINAC	DB
DB6	IDB_V24_SET_WINAC	DB
DB7	IDB_COM_INIT_WINAC	DB
DB10	DB_COM_PARAMS_WINAC	DB
DB12	IDB_P_RESET_WINAC	DB
DB40	DB_SEND_DATA_WINAC	DB
DB50	DB_REC_DATA_WINAC	DB
VAT_Test	VAT_Test	
SFB65001	CREA_COM	STL
SFB65002	EXEC_COM	STL
SFC46	STP	STL
SFC64	TIME_TCK	STL

Figure 4-1 Driver FBs in TIA Portal V11



Caution

The FBs of WinAC serial driver need an instance-DB of their own. These FBs are not multi-instance-capable!

Note

The numbers of the FBs/DB 2,3,5,6 and 12 can be changed by the user freely. It has to be checked that the FB **COM\_INIT\_WINAC** gets the right numbers of the instance-DB.

## 4.1 Initialization FB 7 – COM\_INIT\_WINAC

This FB initializes the ODK-DLL. It must be called once (e.g. in the OB100). Only after a successful call of the **COM\_INIT\_WINAC** the other FBs of the WinAC serial driver can be used.

Figure 4-2 Driver load in Step7 V5.5

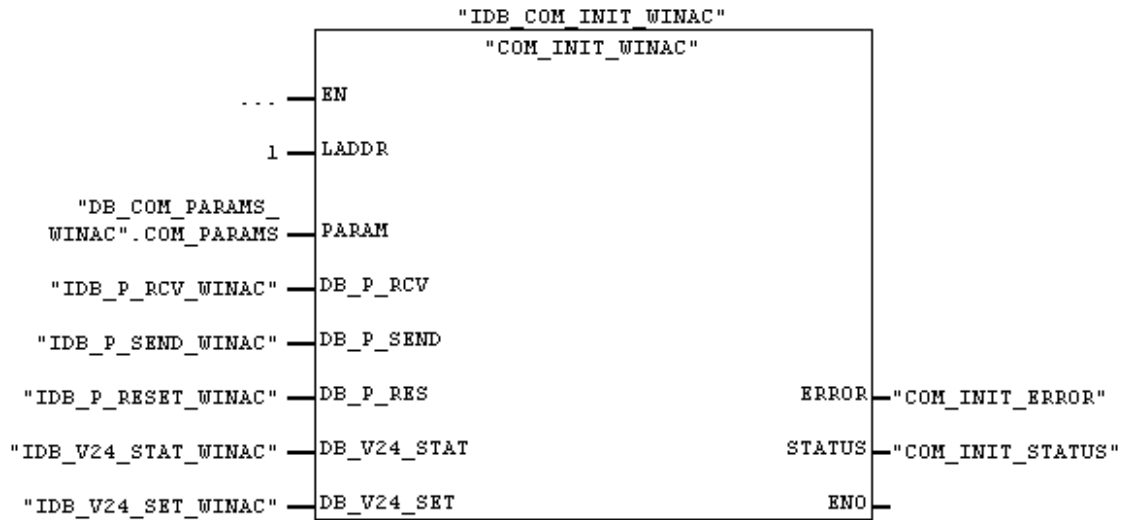
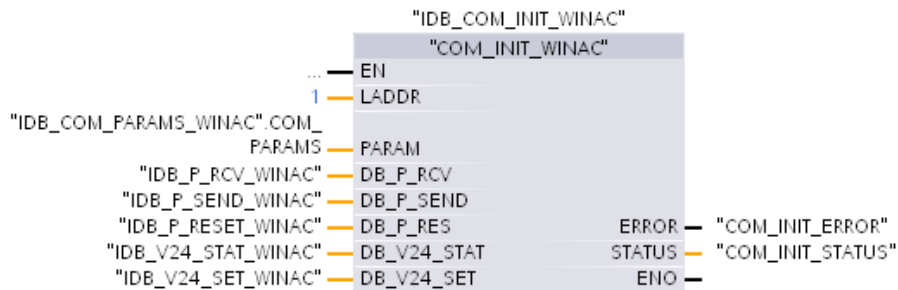


Figure 4-3 Driver load in TIA Portal V11



### Interface

Table 4-1 Parameter of the FBs COM\_INIT\_WINAC

Parameter	In/Out	Typ	Description
LADDR	In	INT	COM port which shall be initialized
PARAM	In	ANY	Pointer on DB_COM_PARAMS (DB10)
DB_P_RCV	In	BLOCK_DB	Instance-DB of P_RCV
DB_P_SEND	In	BLOCK_DB	Instance-DB of P_SEND
DB_P_RESET	In	BLOCK_DB	Instance-DB of P_RESET
DB_V24_STAT	In	BLOCK_DB	Instance-DB of V24_STAT

Parameter	In/Out	Typ	Description
DB_V24_SET	In	BLOCK_DB	Instance-DB of V24_SET
ERROR	Out	BOOL	Fault has appeared
STATUS	Out	WORD	Status of the call (evaluate if set ERROR)

### Return information

Besides the information of the FBs COM\_INIT\_WINAC interface, also additional information is stored in instance DB.

E.g. the DLL version is provided

(`"IDB_COM_INIT_WINAC".C_IF.DLL_VERSION`). The last character indicates the variant: debug (D) / release (A)

Table 4-2 Examples for versions

<code>&lt;&lt; IDB_COM_INIT_WINAC &gt;&gt;.C_IF.DLL_VERSION</code>	DLL-Version
DW#16#0001991D	V 1.9.9.1 Debug
DW#16#0002010A	V 2.0.1.0 Release

### 4.1.1 Parameter setting with the DB\_COM\_PARAMS

The parameter setting is stored in a DB\_COM\_PARAMS\_WINAC. This approach is different to the CP340 232 C.

Figure 4-5: DB\_COM\_PARAMS\_WINAC Step7 V5.5



Address	Name	Type	Initial value
0.0		STRUCT	
+0.0	COM_PARAMS	STRUCT	
+0.0	DB_Kennung	WORD	W#16#232C
+2.0	COM_PROTOCOL	STRUCT	
+0.0	TelegrammEndeKennung	BYTE	B#16#1
+2.0	Zeichenverzugszeit	DINT	L#50
+6.0	EndeZeichen_1	BYTE	B#16#2
+7.0	Enable2EndeZeichen	BYTE	B#16#0
+8.0	EndeZeichen_2	BYTE	B#16#0
+10.0	Telegrammlaenge	DINT	L#240
=14.0		END_STRUCT	
+16.0	COM_BAUDRATE_CHARFRAME	STRUCT	
+0.0	Baudrate	DINT	L#5
+4.0	CharSize	BYTE	B#16#8
+5.0	StopBits	BYTE	B#16#1
+6.0	EnableParity	BYTE	B#16#1
+7.0	Parity	BYTE	B#16#2
+8.0	BreakUeberwachung	BYTE	B#16#0
=10.0		END_STRUCT	
+26.0	COM_FLOWCONTROL	STRUCT	
+0.0	Flusskontrolle	BYTE	B#16#0
+1.0	XonChar	BYTE	B#16#11
+2.0	XoffChar	BYTE	B#16#13
+4.0	Wartezeit_1	DINT	L#2000
+8.0	Wartezeit_2	DINT	L#70
+12.0	Wartezeit_3	DINT	L#10
=16.0		END_STRUCT	
+42.0	COM_RECEIVEBUFFER	STRUCT	
+0.0	PufferLoeschen	BYTE	B#16#1
+2.0	GepufferteTelegramme	DINT	L#10
+6.0	UeberschreibenVerhindern	BYTE	B#16#1

Figure 4-6 DB\_COM\_PARAS\_WINAC TIA Portal V11

DB_COM_PARAS_WINAC				
	Name	Data type	Offset	Start value
1	▼ Static			
2	▼ COM_PARAMS	Struct	0.0	
3	DB_Kennung	Word	0.0	W#16#232C
4	▼ COM_PROTOCOL	Struct	2.0	
5	TelegrammEndeKe...	Byte	0.0	B#16#1
6	Zeichenverzugszeit	DInt	2.0	L#50
7	EndeZeichen_1	Byte	6.0	B#16#2
8	Enable2EndeZeich...	Byte	7.0	0
9	EndeZeichen_2	Byte	8.0	0
10	Telegrammlaenge	DInt	10.0	L#240
11	▼ COM_BAUDRATE_CHAR...	Struct	16.0	
12	Baudrate	DInt	0.0	L#5
13	CharSize	Byte	4.0	B#16#8
14	StopBits	Byte	5.0	B#16#1
15	EnableParity	Byte	6.0	B#16#1
16	Parity	Byte	7.0	B#16#2
17	BreakUeberwachung	Byte	8.0	0
18	▼ COM_FLOWCONTROL	Struct	26.0	
19	Flusskontrolle	Byte	0.0	0
20	XonChar	Byte	1.0	B#16#11
21	XoffChar	Byte	2.0	B#16#13
22	Wartezeit_1	DInt	4.0	L#2000
23	Wartezeit_2	DInt	8.0	L#70
24	Wartezeit_3	DInt	12.0	L#10
25	▼ COM_RECEIVEBUFFER	Struct	42.0	
26	PufferLoeschen	Byte	0.0	B#16#1
27	GepufferteTelegra...	DInt	2.0	L#10
28	UeberschreibenVer...	Byte	6.0	B#16#1

A detailed description of the individual parameters can be found in the operator manual “SIMATIC S7 300 PtP coupling and configuration of CP 340” (04/2005, A5E00369892 01).

**NOTE**

The smallest adjustable time for the character delay time differs from the value as described in the above-mentioned operator manual.

For the WinAC driver the value is 20 ms.



## 4.2 Send data FB3 – P\_SEND\_WINAC

This FB transfers a data block over a COM port to the communication partner.

The FB P\_SEND is documented in the equipment manual "SIMATIC S7 300 PtP coupling and configuration of CP 340" (04/2005, A5E00369892 01).

The DONE bit of the P\_SEND\_WINAC is set as soon as the telegram has been sent.

### Note

In contrast to the CP 340 FBs the parameter LADDR does not contain the base address of the CP 340 but the number (1-8) of the COM port.

Abbildung 4-7: Send data with FB3 P\_SEND\_WINAC Step7 V5.5

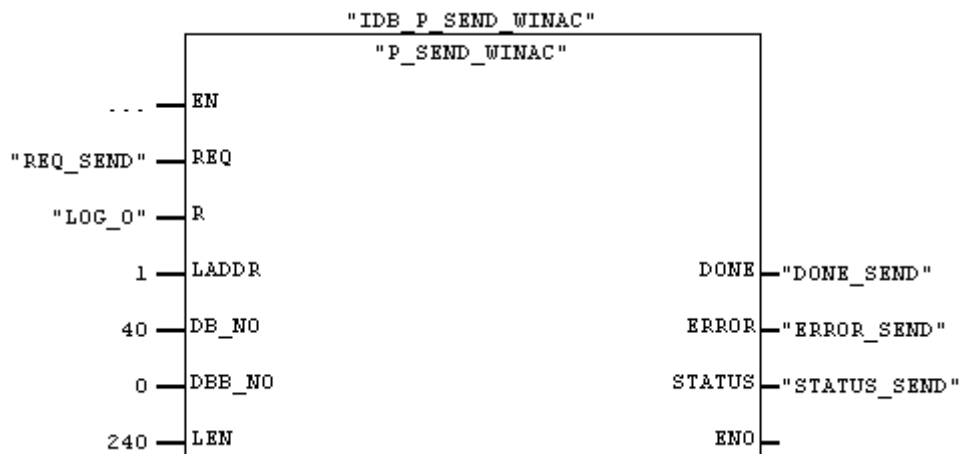
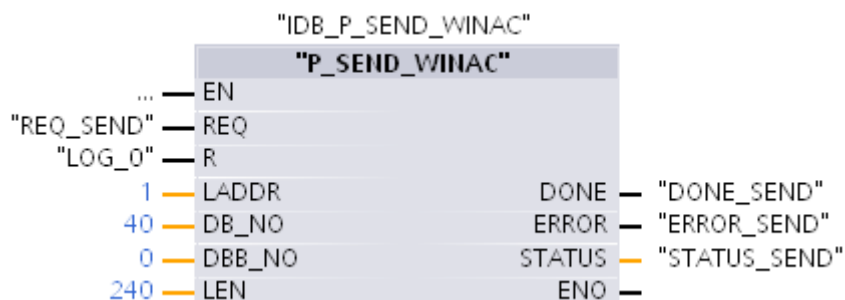


Abbildung 4-8 Send data with FB3 P\_SEND\_WINAC in TIA Portal V11



### 4.3 Data receive FB2 – P\_RCV\_WINAC

The WinAC driver receives data from communication partner over serial interface into an internal buffer. The function block P\_RCV\_WINAC provides this received data to the Step7 application.

The FB P\_RCV\_WINAC is documented in the operator manual “SIMATIC S7 300 PtP coupling and configuration of CP 340” (04/2005, A5E00369892 01).

**Note**

In contrast to the CP 340 FBs the parameter LADDR does not contain the base address of the CP 340 but the number (1-8) of the COM port.

Figure 4-9: Receive data with FB2 P\_RCV\_WINAC in Step7 V5.5

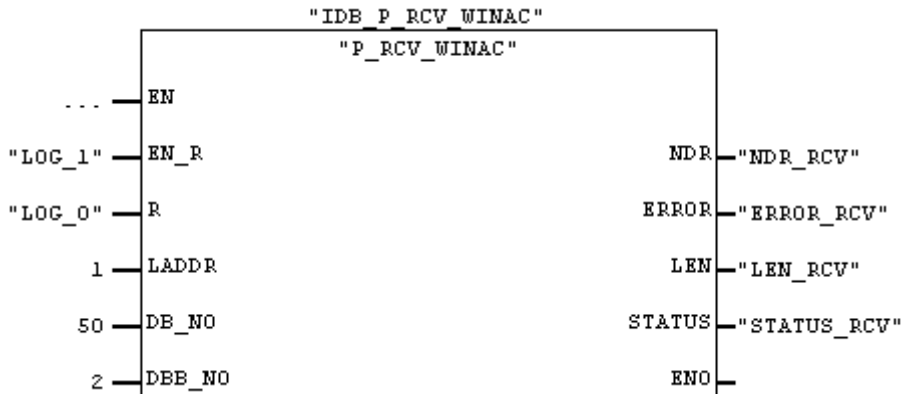
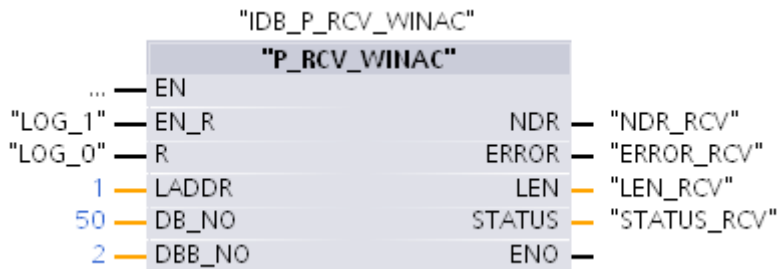


Figure 4-10 Receive data with FB2 P\_RCV\_WINAC in TIA Portal V11



## 4.4 Delete reception buffers FB12 – P\_RESET\_WINAC

This FB deletes the complete receiving buffer of a COM port. All stored telegrams are rejected.

The FB P\_RESET\_WINAC is documented in the operator manual “SIMATIC S7 300 PtP coupling and configuration of CP 340” (04/2005, A5E00369892 01).

**Note**

In contrast to the CP 340 FBs the parameter LADDR does not contain the base address of the CP 340 but the number (1-8) of the COM port.

## 4.5 Get V24-signals FB5 – V24\_STAT\_WINAC

This FB retrieves the RS232 secondary signals of the communication partner and provides the values to the user application. In contrast to the CP340 implementation as function (FC) the WinAC driver provides a function block (FB). The FC V24\_STAT is documented in the operator manual “SIMATIC S7 300 PtP coupling and configuration of CP 340” (04/2005, A5E00369892 01).

### Note

In contrast to the CP 340 FBs the parameter LADDR does not contain the base address of the CP 340 but the number (1-8) of the COM port.



The parameters DTR\_OUT and RTS\_OUT were kept for reasons of compatibility. These values stay always at 0. They do not provide information concerning statuses about V24 signals.

## 4.6 Put V24 signals FB6 – V24\_SET\_WINAC

The user can set or reset the RS232 secondary signals with this FB V24\_SET\_WINAC.

In contrast to the CP340 implementation as function (FC) the WinAC driver provides a function block (FB).

The FC V24\_STAT is documented in the operator manual "SIMATIC S7 300 PtP coupling and configuration of CP 340" (04/2005, A5E00369892 01).

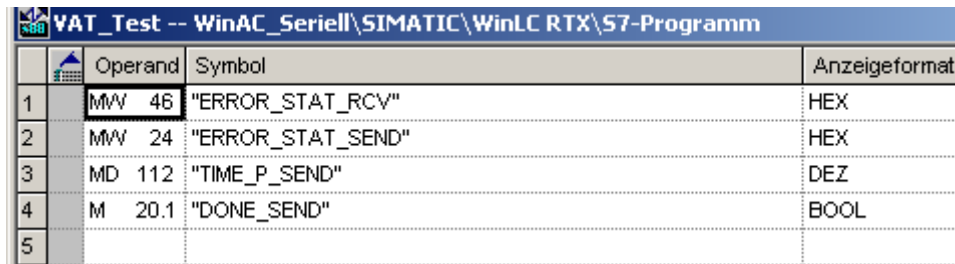
### Note

In contrast to the CP 340 FBs the parameter LADDR does not contain the base address of the CP 340 but the number (1-8) of the COM port.

## 4.7 Variable table / Watch table

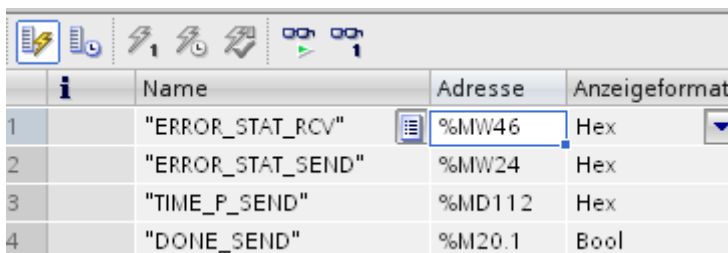
The example project includes variable/watch tables. The user can monitor important status information of the WinAC serial driver.

Figure 4-4 Variable table in Step7 V5.5



	Operand	Symbol	Anzeigeformat
1	MW 46	"ERROR_STAT_RCV"	HEX
2	MW 24	"ERROR_STAT_SEND"	HEX
3	MD 112	"TIME_P_SEND"	DEZ
4	M 20.1	"DONE_SEND"	BOOL
5			

Figure 4-5 Watch table in TIA Portal V11



	Name	Adresse	Anzeigeformat
1	"ERROR_STAT_RCV"	%MW46	Hex
2	"ERROR_STAT_SEND"	%MW24	Hex
3	"TIME_P_SEND"	%MD112	Hex
4	"DONE_SEND"	%M20.1	Bool

## 5 Error Codes

The WinAC serial driver can deliver two different categories of error codes:

- Special error codes of the WinAC serial driver
- Error codes of the WinAC ODK functionality

### 5.1 Error codes of WinAC serial driver

Table 5-1 Driver error codes

Error Code	Description
0x8200	Error at ReadFile != ERROR_IO_PENDING
0x8202	DSR == false at read operation
0x8300	GetRing
0x8301	GetRLSD
0x8400	SetRTS
0x8401	SetDTR
0x8502	Order is not permitted to the CP in this operating state.
0x850e	Error: Telegram length
0x8708	The wait time on XON or CTS = ON has passed
0x8806	Sign delay time exceeded: two signs succeeding were not receives with sign delay time
0x8807	Telegram length occurred: a telegram was received with the length 0
0x880A	A free reception buffer is not available: at the reception no empty reception buffer was available.
0x880C	Transmission error: a transmission error (parity error, Stopp bit error, overrun error) was recognized.
0x880D	BREAK Reception pipe to the partner is interrupted
0x8810	Parity error
0x8811	Character box fault
0x8812	Have sent the CP XOFF or have put CTS on off, were occurred border signs.
0x8818	DSR=OFF or CTS=OFF
0x8900	COMport was not initialized
0x8901	ThreadEvent, V24_STAT, P_SEND
0x8902	ThreadEvent, ThreadRead, V24_STAT
0x8A01	COMPORT-Range
0x8A02	COMPORT-Init
0x8A03	COM_PORTOCOL.TelegrammEndeKennung
0x8A04	COM_PORTOCOL.Zeichenverzugszeit
0x8A05	COM_PORTOCOL.EndeZeichen_1
0x8A06	COM_PORTOCOL.Enable2Endezeichen

Error Code	Description
0x8A07	COM_PORTOCOL.EndeZeichen_2
0x8A08	COM_PORTOCOL.Telegrammlaenge
0x8A09	COM_BAUDRATE_CHARFRAME.Baudrate
0x8A0A	COM_BAUDRATE_CHARFRAME.CharSize
0x8A0B	COM_BAUDRATE_CHARFRAME.StopBits
0x8A0C	COM_BAUDRATE_CHARFRAME.EnableParity
0x8A0D	COM_BAUDRATE_CHARFRAME.Parity
0x8A0E	COM_BAUDRATE_CHARFRAME.BreakUeberwachung
0x8A0F	COM_FLOWCONTROL.Flusskontrolle
0x8A10	COM_FLOWCONTROL.XonChar
0x8A11	COM_FLOWCONTROL.XoffChar
0x8A12	COM_FLOWCONTROL.Wartezeit_1
0x8A13	COM_FLOWCONTROL.Wartezeit_2
0x8A14	COM_FLOWCONTROL.Wartezeit_3
0x8A15	COM_RECEIVEBUFFER.PufferLoeschen
0x8A16	COM_RECEIVEBUFFER.GepufferteTelegramm
0x8A17	COM_RECEIVEBUFFER.UeberschreibenVerhindern
0x8A18	CreateFile return invalid handle
0x8A19	Commport not exists
0x8A1A	SetCommMask
0x8A1B	GetCommState
0x8A1C	SetCommState
0x8A1D	SetCommTimeouts
0x8A1E	Invalid DB-Kennung (INIT)
0x8A1F	Invalid DB-Size
0x8A20	Error SetupComm
0x8B00	Error write medium
0x8B01	Errori WriteFile()
0x8B02	SetEvent
0x8B03	SEND: Data Length too big; DB too small
0x8B04	SEND: Datenblock too big
0x8B05	Auto flow control CTS within the output curre time (Wartezeit_3) changed to OFF.

## 5.2 Error codes of WinAC ODK functionality

The driver had been developed with the WinAC ODK (Open Development Kit). The ODK can generate error codes, which are returned from **STATUS** of the FBs.

Table 5-2 WinAC ODK System error codes

ODK Code (HEX)	Description
0	Success
8001	An exception occurred.
8002	Input: the ANY pointer is invalid.
8003	Input: the ANY pointer range is invalid.
8004	Output: the ANY pointer is invalid.
8005	Output: the ANY pointer range is invalid.
8006	More bytes were written into the output buffer by the extension object than were allocated.
8007	ODK system has not been initialized: no previous call to SFB65001 (CREA_COM).
8008	The supplied handle value does not correspond to a valid extension object.
8009	More bytes were written into the input buffer by the extension object than were allocated.
807F	An internal error occurred.
80C3	Maximum number (32) of parallel jobs/instances exceeded.
8102	The call to CLSIDFromProgID failed.
8103	The call to CoInitializeEx failed.
8104	The call to CoCreateInstance failed.
8105	The library failed to load.
8106	A Windows response timeout occurred.
8107	Controller is in an invalid state for scheduling an OB.
8108	Schedule information for OB is invalid.
8109	Instance ID for SFB65001 call is invalid.
810A	Controller could not load proxy DLL.
810B	The WinAC controller could not create or initialize shared memory area.
810C	Attempt to access unavailable option occurred.
8201	The Execute command index could not be found
8250	No more available positions in the job list
8252	The count is invalid
8253	A data type of an item in the list is invalid
8254	The count specified is invalid
8255	A memory area of an item in the list is invalid
8256	A DB number of an item in the list is invalid
8257	A bit number of an item in the list is invalid
8258	A pBuff of an item in the list is invalid
8259	A data quantity is invalid



<b>ODK Code (HEX)</b>	<b>Description</b>
825A	The area offset parameter is invalid for this type
825B	The frequency value is invalid
825C	The callback pointer is invalid
825D	The job ID pointer is invalid
825E	The job ID is invalid
825F	Job could not be completed because address is incorrect
8260	Job could not be completed because of protection level
8261	Job could not be completed because of hardware issues
8301	Invalid Thread Execution Priority
8401	Invalid Asynchronous Event
8402	Asynchronous Processor Queue is empty
8403	Asynchronous Processor Queue is full

## 6 History

Table 6-1 History

Version	Date	Editor	Remark
V1.0	30.10.07	Chaudhary	First version
V1.1	27.02.08	Chaudhary	Error-Code conformed
V1.2	02.04.09	Cajlan	Tested with WinAC RTX 2008
V1.6.0	08.01.10	Chaudhary	Tested with WinAC RTX 2009, IPC 427C
V1.6.1	09.07.10	Chaudhary	character delay time 20-65530 ms
V1.6.10	16.12.11	Happ	Customization of the versioning
V1.7.0	13.07.12	GD	Tested with WinAC RTX 2010 Tested with TIA Portal V11 Tested with Windows 7
V1.7.10	18.12.12	GD	Mistakes occurred during the migration In TIA Portal corrected. SIMATIC project improved.
V1.7.11	08.05.13	AH	Tested with: <ul style="list-style-type: none"> <li>- WinAC RTX 2010</li> <li>- TIA Portal V12</li> <li>- WES7</li> <li>- IPC 427D</li> </ul>