

# **SIMATIC NET**

**V1SL**

## **Getting Started**

(DPC31 Slave Firmware)

Version: 1.1

Date: 07/01

**Liability Exclusion**

We have tested the content of this document regarding agreement with the hardware and software described. Nevertheless, deviations can't be ruled out, and we are not guaranteeing complete agreement. However, the data in this document is checked periodically. Necessary corrections will be included in subsequent editions. We gratefully accept suggestions for improvement.

**Copyright**

Unless permission has been expressly granted, passing on this document, or using and copying it is not allowed. Offenders will be held liable. All rights reserved, in the event a patent is granted, or a utility model or design is registered.

The brand names SIMATIC, SINEC L2 are protected by law for Siemens through application or registration. All other product and system names are (registered) brand names of their respective proprietors, and are to be treated as such.

Subject to technical changes.

## Sample Slave with V1SL Standard Interface

For becoming familiar with all the details of V1SL, an example is provided that includes all necessary functionalities of a DPV1 slave. You can use this example as a basis for developing your slave.

The example provided includes all the presettings that are necessary for implementing a DPV1 slave with cyclical Class 1 services (C0), acyclical Class 1 services (C1), and acyclical Class 2 services (C2).

The necessary user settings (for example, PNO Ident No., station address, ...) are made in the module `sys_config.h` via `#defines`, and are automatically included to the global configuration file `"V1SL_CFG.H"`. For standard applications, the user doesn't need to make any changes to `V1SL_CFG.H`. However, it is up to user to adapt `V1SL_CFG.H` to its specific needs.

The file `"usr.c"` provides an example how to process the user data. The parts where the user has to add his own code for processing the periphery are marked with `"todo:..."`.

## 1 Installation

Install your development environment in the tool directory.

## 2 Modelling the Example

Feature	Sample Modelling
Parameterization	3 bytes user / DPV1 parameterization
Configuration	As configuration, 0x13 is permitted. This is also the default configuration.
Diagnosis	6 bytes standard diagnosis; Status PDU. Userdata[0] = TEMP_STATE.
Data Exchange	Temperature value is transferred to the input data 1+ 2.
DPV1 MSAC_C1	The slot number is ignored.
DPV1 MSAC_C2	Only Slot 0 is permissible.
DS_READ	Index 2: read current data set.  Else : NRS_PDU.
DS_WRITE	Index 2 : Use the transferred data set.  Else : NRS_PDU.
DATA_TRANSPORT	Output data = input data

## 3 Slave Settings in the File 'sys\_cfg.h'

In this module, the most important slave settings are included. The user can adjust these settings according to his requirements. Additional settings for details can, if needed, be made in the file 'v1sl\_cfg.h'.

#define	Meaning	for
SYS_C0_SUPPORT	1 = Cyclical Class 1 connection is supported 0 = Cyclical Class 1 connection is not supported	C0, C1
SYS_C1_SUPPORT	1 = Acyclical Class 1 services are supported 0 = Acyclical Class 1 services are not supported	C1
SYS_C2_SUPPORT	1 = Acyclical Class 2 services are supported 0 = Acyclical Class 2 services are not supported	C2
SYS_AL_SUPPORT	1 = Class 1 alarm processing is supported 0 = Class 1 alarm processing is not supported	C1
SYS_PBC_HARDWARE_ADDRESS	Start address of the DPC31	gen.
SYS_PBC_COM_MODE	Hardware settings of the DPC31	gen.
SYS_PBC_RESET_PIN	Address of the reset PIN	gen.
sys_pbc_reset_set_ sys_pbc_reset_clr_	Macros for setting/resetting the reset PIN. <b>Attention:</b> if the internal core of the DPC31 is used, the software is not allowed to trigger a reset!!!!	gen.
SYS_PBC_MIN_TSDR	Is to be set to 11 for pure DP slaves	gen.
SYS_PBC_USER_WD_VALUE	USER watchdog value of the DPC31	gen.
SYS_PBC_STATION_ADDRESS	Station address of the slave	gen.
SYS_PNO_IDENT_NUM_HIGH SYS_PNO_IDENT_NUM_LOW	PNO Ident Number	C0
SYS_C0_LEN_MAX_SSA	Max. length of the Set_Slave_Addr telegram	C0
SYS_C0_LEN_MAX_PRM	Max. length of the Set_Prm telegram	C0
SYS_C0_LEN_MAX_CFG	Max. length of the Chk_Cfg and Get_Cfg telegram	C0
SYS_C0_LEN_MAX_USER_DIAG	Max. length of user diagnostic data	C0
SYS_C0_LEN_MAX_INPUT	Max. input data length	C0
SYS_C0_LEN_MAX_OUTPUT	Max. output data length	C0
SYS_C0_NO_ADDRESS_CHANGE	TRUE: address changes via bus not permitted FALSE: address changes via but permitted	C0
SYS_C0_SYNC_SUPPORTED	TRUE: Sync is supported FALSE: Sync is not supported	C0
SYS_C0_FREEZE_SUPPORTED	TRUE: Freeze is supported FALSE: Freeze is not supported	C0
SYS_C1_USER_DATA_LEN	Max. user data length of a MSAC_C1 connection	C1
SYS_C2_USER_DATA_LEN	Max. user data length of a MSAC_C2 connection	C2
SYS_C2_CONNECT_COUNT	Max. number of MSAC_C2 connections	C2
SYS_C2_INITIATE_PDU_SIZE	Max. length of the MSAC_C2 Initiate-PDU	C2
SYS_C2_POLL_TIMEOUT	Min. poll timeout of the C2 slave	C2
SYS_C2_FINAL_COMM_POINT	TRUE: C2 connection is end point (not a router). FALSE: C2 connection is a router.	C2
SYS_C2_AUTO_INITIATE_RESP	TRUE: When the function C2_INITIATE was called, the Initiate-Req-PDU was automatically converted into a Response-PDU. FALSE: The user converts the Initiate-Req-PDU into a Response-PDU.	C2
SYS_C2_PROFILE_FEATURES_SUPPORTED_1 SYS_C2_PROFILE_FEATURES_SUPPORTED_2	Supported C2 profile features	C2
SYS_C2_PROFILE_IDENT_NUM_LOW SYS_C2_PROFILE_IDENT_NUM_HIGH	Ident number of the C2 profile used	C2
SYS_C2_SUBNET	1: C2_SUBNET_LOCAL	C2
SYS_IFA_CODE_ATTR	Code attribute of the interface (user)	gen.
SYS_INT_CODE_ATTR	Code attribute of the firmware	gen.
SYS_LL_DATA_ATTR	Data attribute of the ASIC	gen.
SYS_IFA_DATA_ATTR	Data attribute of the interface (user)	gen.
SYS_INT_DATA_ATTR	Data attribute of the firmware	gen.
SYS_DATA_ATTR_HUGE	Huge data attribute	gen.

A more detailed description of the settings is provided in the V1SL User Description.

## 4 Sample Code in the File 'usr.c'

While the settings in the file 'sys\_cfg.h' can be used with a few exceptions for most cases, the user has to replace the sample data with his own actual periphery data in the 'usr.c'.

To provide assistance, the relevant parts are marked with 'todo:...'.

Essentially, this concerns the following data:

in Front of the endless loop:

- Default configuration
- Initialization of the input data
- Initialization of the diagnostic state
- Initialization of the DPV1 data set buffer

In the endless loop:

- Set input data
- Process the diagnostic state

In the callback functions:

- Check the received parameterization
- Check the configuration sent by the master
- Handle the DPV1 data sets

Additional callback functions that are useful to certain applications exist as function bodies.