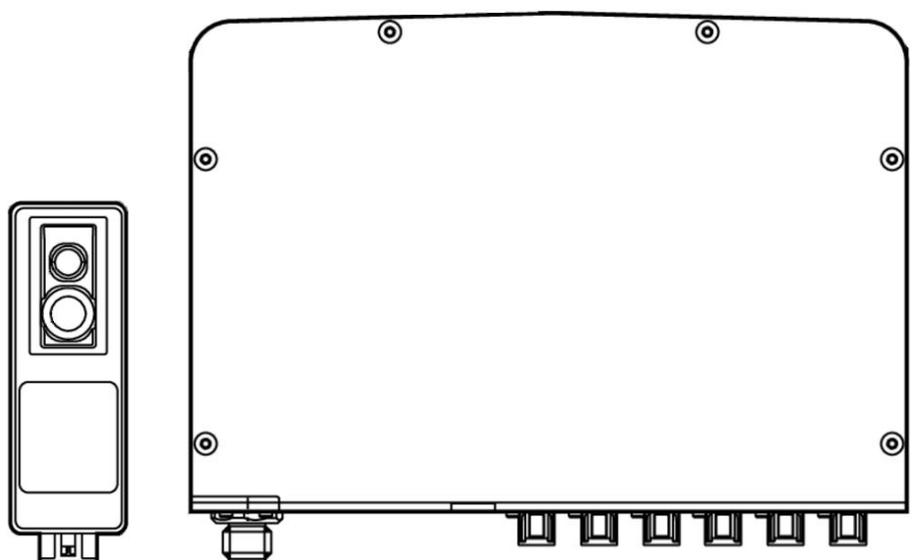




Description
Siemens S7-1500
function block
OVP81x-ProtV2

Version 1.0

Obstacle Detection System (ODS)
and
Pick and Drop System (PDS)
with device interface protocol-version V2.x
via TCP





Contents

| | |
|--|----|
| 1. Preface | 2 |
| I. Disclaimer | 2 |
| II. Demo Software and Templates | 2 |
| III. Customer-specific software | 2 |
| 2. Requirements..... | 3 |
| 3. Function Block Description | 4 |
| 4. Integration of the Function Block into the User Program..... | 12 |
| 4.1. Inserting the Function Block into the User Program | 12 |
| 4.2. Setting the Connection Parameters..... | 12 |
| 4.3. Calling the Instances of the Function Block in the User Program..... | 13 |
| 5. Control of the ODS and PDS Applications of the VPU with the help of the Function Block <i>OVP81x-ProtV2</i> | 14 |
| 5.1. Connect to the VPU | 14 |
| 5.2. ODS Application Commands | 14 |
| 5.3. PDS Application Commands..... | 15 |
| 6. Appendix..... | 17 |
| 6.1. Display in the Output Parameter <i>ComStatus</i> | 17 |
| 6.2. Display in the Command Error Info Outputs in <i>ODS_Result</i> and <i>PDS_Result</i> Structures..... | 18 |
| 6.3. Setting the IP Address of the CPU in the TIA Portal | 18 |
| 6.4. Displaying the Hardware ID of the CPU Ethernet Interface in the TIA Portal | 18 |



1. Preface

In the user program of an S7 1500 controller, the function block *OVP81x-ProtV2* is intended to simplify data exchange via the TCP interface with the PLC server application of a Video Processing Unit (VPU) series OVP81x from ifm electronic gmbh. The block was created in SCL by means of the TIA Portal software V16 in the editing language English (USA).

This document

- describes the function block and its parameters in detail
- describes the use of the block in the S7 program

The function block is made available as a demo program. It is only intended to serve as an example to the user. Any use of this function block for machine control is at your own risk. The authors and rights holders of this code package disclaim any liability for the functionality or compatibility of this demo version.

For copyright reasons, only the free distribution of the code package is allowed.

The function block is unencrypted. Its code can be extended and adapted to own needs.

I. Disclaimer

ifm disclaims to the fullest extent authorized by law any and all warranties, whether express or implied, including, without limitation, any implied warranties of title, non-infringement, quiet enjoyment, integration, merchantability or fitness for a particular purpose.

Without limitation of the foregoing, ifm expressly does not warrant that:

- (a) the software will meet your requirements or expectations
- (b) the software or the content of the software is free from errors, viruses or other defects
- (c) Results, outputs or data provided or generated by the Software are accurate, up-to-date, complete or reliable
- (d) the Software is compatible with third party software
- (e) Errors in the software can be corrected

II. Demo Software and Templates

Demo Software and Templates are provided "as is" and "as available" without any express or implied warranty. The user acknowledges and agrees to use the software at his own risk. Under no circumstances can ifm be held liable for any direct, indirect, incidental or consequential damages arising from the use of the software or inability to use the software. The user may only use the software for demonstration purposes and to assess the functionalities and capabilities of the software.

III. Customer-specific software

1. The software created and used has been compiled by ifm especially for the customer with the help of modular software components, which have been created by ifm for numerous applications (standard software blocks) and adapted to the required contractual service (customer-specific application program).
2. Upon full payment of the purchase price for the customer-specific application program, ifm transfers the non-exclusive right of use to the customer without the customer acquiring any rights to the standard software block on which the individual or customer-specific adaptation is based. Without prejudice to these provisions, ifm reserves the right to create and offer customer-specific software solutions of the same type to other customers on the basis of other reference conditions. In any case, ifm retains a simple right of use of the customer-specific solution for internal purposes.
3. By accepting the program, the user acknowledges and agrees to use the software at his own risk. By accepting the program, the user also acknowledges and agrees that the software meets the requirements of the agreed specifications. ifm excludes all liability, in particular with regard to the suitability of the software for a specific purpose.



2. Requirements

- IFM Video Processing Unit (VPU) of the OVP81x series with PLC server application and protocol version V2.x
- Siemens S7-1500 CPU with firmware V2.6 or higher
- S7 CPU cycle time < 20ms
- Siemens TIA Portal Software V16 or higher



3. Function Block Description

Version

Version V1.0

Use

Simplification of the interface in the V2.x protocol of a Video Processing Unit (VPU) of the OVP81x series in the PLC user program

Features

- Automatically adjusts to the endianness used
- Establishing a TCP connection to the VPU PLC server
- Monitoring the connection to the PLC server
- Receiving the result data and displaying it at the outputs
- Sending commands to an ODS application in the VPU
- Sending commands to two PDS applications in the VPU

Required additional program blocks

- Siemens system function block TRCV_C
- Siemens system function block TSEND

Layout

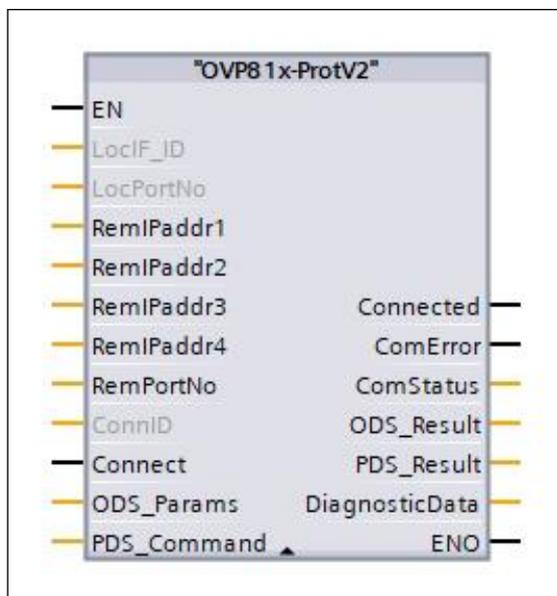


Figure 1



Input parameters

Table 1

| Name | Data type | Description |
|----------------------|-----------------------------|--|
| <i>LocIF_ID</i> | HW_INTERFACE | Hardware identifier of the Ethernet interface used by the PLC (see chapter 6.4) <ul style="list-style-type: none"> • preset with 64 and hidden if no other value or variable is assigned |
| <i>LocPortNo</i> | UInt | TCP port No. of the CPU for this TCP connection <ul style="list-style-type: none"> • 0 = the FB uses <i>LocPortNo</i> = 3000 + <i>RemIPAddr4</i>, otherwise [2000..5000] recommended • preset with 0 and hidden if no other value or variable is assigned |
| <i>RemIPAddr1..4</i> | USInt | IP address of the Video Processing Unit (VPU) |
| <i>RemPortNo</i> | UInt | TCP port No. of the PLC-server-application of the Video Processing Unit (VPU) |
| <i>ConnID</i> | CONN_OUC | Unique number for internal identification of this connection <ul style="list-style-type: none"> • 0 = the FB uses <i>ConnID</i> = 3000 + <i>RemIPAddr4</i>, otherwise [1..4095] valid • preset with 0 and hidden if no other value or variable is assigned |
| <i>Connect</i> | Bool | <ul style="list-style-type: none"> • TRUE establishes a connection to the PLC server of the VPU • FALSE terminates the connection to the PLC server of the VPU |
| <i>ODS_Params</i> | Struct | ODS parameters |
| <i>Enable</i> | Bool | Enable ODS commands |
| <i>TimeoutTime</i> | Time | Command response timeout time |
| <i>PresetID</i> | USInt | Zone Preset ID to activate |
| <i>MaxHeight</i> | UInt | Maximum height to activate |
| <i>OverhangMask</i> | Word | Overhanging load mask to activate |
| <i>PDS_Command</i> | Array[0..1] of Struct | PDS commands |
| <i>Select</i> | USInt | Selection of the PDS command to be executed <ul style="list-style-type: none"> 1 = Get Rack 2 = Get Pallet 3 = Get Item 4 = Volume Check |
| <i>TimeoutTime</i> | Time | Command response timeout time |
| <i>GetRack</i> | see Table 2 | Get Rack command parameters |



| | | |
|------------------|-----------------------------|---------------------------------|
| <i>GetPallet</i> | see Table 3 | Get Pallet command parameters |
| <i>GetItem</i> | see Table 4 | Get Item command parameters |
| <i>VolCheck</i> | see Table 5 | Volume Check command parameters |

Get Rack command parameters structure

Table 2

| Name | Data type | Description |
|-------------------------------|-----------|---|
| <i>GetRack</i> | Struct | Get Rack command parameters |
| <i>HorizontalDropPosition</i> | UInt | Selection of the horizontal drop setting 0 = left 1 = center 2 = right |
| <i>VerticalDropPosition</i> | UInt | Selection of the vertical drop setting 0 = interior 1 = floor |
| <i>DepthHint</i> | Int | Estimated distance between rack and coordinate system center [mm] |
| <i>ZHint</i> | Int | Estimated z-coordinate of the rack shelf [mm] |
| <i>ClearingVolumeXMin</i> | Int | Minimum x-coordinate [mm] of the volume which will be checked for obstacles |
| <i>ClearingVolumeXMax</i> | Int | Maximum x-coordinate [mm] of the volume which will be checked for obstacles |
| <i>ClearingVolumeYMin</i> | Int | Minimum y-coordinate [mm] of the volume which will be checked for obstacles |
| <i>ClearingVolumeYMax</i> | Int | Maximum y-coordinate [mm] of the volume which will be checked for obstacles |
| <i>ClearingVolumeZMin</i> | Int | Minimum z-coordinate [mm] of the volume which will be checked for obstacles |
| <i>ClearingVolumeZMax</i> | Int | Maximum z-coordinate [mm] of the volume which will be checked for obstacles |



Get Pallet command parameters structure

Table 3

| Name | Data type | Description |
|--------------------|-----------|---|
| <i>GetPallet</i> | Struct | Get Pallet command parameters |
| <i>DepthHint</i> | Int | Estimated distance between pallet and calibrated coordinate system center [mm]. <=0 for automatic detection |
| <i>PalletIndex</i> | Int | Index of the pallet parameter set [0..9] |
| <i>PalletOrder</i> | Int | 0 = scoreDescending 1 = Descending 2 = Ascending |

Get Item command parameters structure

Table 4

| Name | Data type | Description |
|------------------|-----------|---|
| <i>GetItem</i> | Struct | Get Item command parameters |
| <i>DepthHint</i> | Int | Estimated distance between pallet and calibrated coordinate system center [mm]. <=0 for automatic detection |
| <i>ItemIndex</i> | Int | Index of the item parameter set [0..9] |
| <i>ItemOrder</i> | Int | 0 = scoreDescending 1 = Descending 2 = Ascending |

Volume Check command parameters structure

Table 5

| Name | Data type | Description |
|-------------------|-----------|--|
| <i>VolCheck</i> | Struct | Volume Check command parameters |
| <i>VolumeXMin</i> | Int | Minimum x-coordinate [mm] of the volume which will be checked for pixels |
| <i>VolumeXMax</i> | Int | Maximum x-coordinate [mm] of the volume which will be checked for pixels |
| <i>VolumeYMin</i> | Int | Minimum y-coordinate [mm] of the volume which will be checked for pixels |
| <i>VolumeYMax</i> | Int | Maximum y-coordinate [mm] of the volume which will be checked for pixels |
| <i>VolumeZMin</i> | Int | Minimum z-coordinate [mm] of the volume which will be checked for pixels |
| <i>VolumeZMax</i> | Int | Maximum z-coordinate [mm] of the volume which will be checked for pixels |



Output parameters

Table 6

| Name | Data type | Description |
|-------------------------|------------------------------|--|
| <i>Connected</i> | Bool | TRUE = TCP connection established to the PLC server of the VPU |
| <i>ComError</i> | Bool | TRUE = Communication error occurred |
| <i>ComStatus</i> | String[12] | Communication status information (see Table 13) |
| <i>ODS_Result</i> | see Table 7 | ODS result data |
| <i>PDS_Result</i> | Array[0..1] of Struct | PDS result data |
| <i>CommandSuccess</i> | Bool | TRUE = PDS Command successfully completed |
| <i>CommandFailed</i> | Bool | TRUE = PDS Command failed |
| <i>CommandErrorInfo</i> | String[12] | PDS Command error information (see Table 14) |
| <i>GetRack</i> | see Table 8 | Get Rack result data |
| <i>GetPallet</i> | see Table 9 | Get Pallet result data |
| <i>GetItem</i> | see Table 10 | Get Item result data |
| <i>VolCheck</i> | see Table 11 | Volume Check result data |
| <i>DiagnosticData</i> | Array[0..19] of Struct | Diagnostic data |
| <i>Source</i> | UInt | Diagnose trigger source |
| <i>Severity</i> | UInt | Severity |
| <i>DiagnosticID</i> | UDInt | Diagnostic ID |

ODS result data structure

Table 7

| Name | Data type | Description |
|--------------------------------|------------|--|
| <i>ODS_Result</i> | Struct | ODS result data |
| <i>ActivatePresetSuccess</i> | Bool | TRUE = Activate Zone Preset successfully completed |
| <i>ActivatePresetFailed</i> | Bool | TRUE = Activate Zone Preset failed |
| <i>ActivatePresetErrorInfo</i> | String[12] | Activate Zone Preset error information (see Table 14) |
| <i>SetMaxHeightSuccess</i> | Bool | TRUE = Set Maximum Height successfully completed |
| <i>SetMaxHeightFailed</i> | Bool | TRUE = Set Maximum Height failed |
| <i>SetMaxHeightErrorInfo</i> | String[12] | Set Maximum Height error information (see Table 14) |



| | | |
|-----------------------------|--------------------------|---|
| <i>SetOverhangSuccess</i> | Bool | TRUE = Set Overhanging Load Mask successfully completed |
| <i>SetOverhangFailed</i> | Bool | TRUE = Set Overhanging Load Mask failed |
| <i>SetOverhangErrorInfo</i> | String[12] | Set Overhanging Load Mask error information (see Table 14) |
| <i>ResultAge</i> | UInt | Result age indicator (incremented with the age of the result data, max value: 255) |
| <i>Timestamp</i> | LDT | Time stamp of ODS algorithm result |
| <i>ZoneConfigID</i> | UDInt | Current zone configuration ID |
| <i>Zone1occ</i> | Bool | TRUE = Zone 1 occupied |
| <i>Zone2occ</i> | Bool | TRUE = Zone 2 occupied |
| <i>Zone3occ</i> | Bool | TRUE = Zone 3 occupied |
| <i>PolarOcc</i> | Array[0..674] of UInt | Polar occupancy grid values |

Get Rack result data structure

Table 8

| Name | Data type | Description |
|-----------------------|-----------|---|
| <i>GetRack</i> | Struct | Get Rack result data |
| <i>ResultAge</i> | UInt | Result age indicator (incremented with the age of the result data, max value: 255) |
| <i>Timestamp</i> | LDT | Time stamp of PDS algorithm result |
| <i>DetectionValid</i> | Bool | TRUE = valid results FALSE = invalid detection |
| <i>PositionX</i> | Int | X-coordinate of the rack position [mm] |
| <i>PositionY</i> | Int | Y-coordinate of the rack position [mm] |
| <i>PositionZ</i> | Int | Z-coordinate of the rack position [mm] |
| <i>Roll</i> | Int | Rotation of the rack about the x-axis [milliradians] |
| <i>Pitch</i> | Int | Rotation of the rack about the y-axis [milliradians] |
| <i>Yaw</i> | Int | Rotation of the rack about the z-axis [milliradians] |
| <i>NumPixels</i> | UDInt | Number of pixels inside the clearing volume |
| <i>AnchoredSide</i> | Int | 0 = left 1 = center 2 = right |
| <i>Flags</i> | Word | Bitmask encoding the detection status bits |



Get Pallet result data structure

Table 9

| Name | Data type | Description |
|-----------------------|-----------|---|
| <i>GetPallet</i> | Struct | Get Pallet result data |
| <i>ResultAge</i> | UInt | Result age indicator (incremented with the age of the result data, max value: 255) |
| <i>Timestamp</i> | LDT | Time stamp of PDS algorithm result |
| <i>DetectionValid</i> | Bool | TRUE = valid results FALSE = invalid detection |
| <i>PalletIndex</i> | Int | Index of the parameter set which was used to detect the pallet |
| <i>CenterX</i> | Int | X-coordinate of the center block [mm] |
| <i>CenterY</i> | Int | Y-coordinate of the center block [mm] |
| <i>CenterZ</i> | Int | Z-coordinate of the center block [mm] |
| <i>LeftPocketX</i> | Int | X-coordinate of the left pocket [mm] |
| <i>LeftPocketY</i> | Int | Y-coordinate of the left pocket [mm] |
| <i>LeftPocketZ</i> | Int | Z-coordinate of the left pocket [mm] |
| <i>RightPocketX</i> | Int | X-coordinate of the right pocket [mm] |
| <i>RightPocketY</i> | Int | Y-coordinate of the right pocket [mm] |
| <i>RightPocketZ</i> | Int | Z-coordinate of the right pocket [mm] |
| <i>Roll</i> | Int | Pallet's rotation about the x-axis [milliradians] |
| <i>Pitch</i> | Int | Pallet's rotation about the y-axis [milliradians] |
| <i>Yaw</i> | Int | Pallet's rotation about the z-axis [milliradians] |



Get Item result data structure

Table 10

| Name | Data type | Description |
|------------------|-----------|---|
| <i>GetItem</i> | Struct | Get Item result data |
| <i>ResultAge</i> | UInt | Result age indicator (incremented with the age of the result data, max value: 255) |
| <i>Timestamp</i> | LDT | Time stamp of PDS algorithm result |
| <i>Score</i> | UInt | Detection score. 0 = invalid detection |
| <i>ItemIndex</i> | Int | ID of the fitted item |
| <i>PositionX</i> | Int | X-coordinate of the item's principal point [mm] |
| <i>PositionY</i> | Int | Y-coordinate of the item's principal point [mm] |
| <i>PositionZ</i> | Int | Z-coordinate of the item's principal point [mm] |
| <i>Roll</i> | Int | Item's rotation about the x-axis [milliradians] |
| <i>Pitch</i> | Int | Item's rotation about the y-axis [milliradians] |
| <i>Yaw</i> | Int | Item's rotation about the z-axis [milliradians] |

Volume Check result data structure

Table 11

| Name | Data type | Description |
|------------------|-----------|---|
| <i>VolCheck</i> | Struct | Volume Check result data |
| <i>ResultAge</i> | UInt | Result age indicator (incremented with the age of the result data, max value: 255) |
| <i>Timestamp</i> | LDT | Time stamp of PDS algorithm result |
| <i>NumPixels</i> | UDInt | Number of pixels inside the volume |
| <i>NearestX</i> | DInt | Smallest x-coordinate inside the volume (derived from quantile) [mm] |



4. Integration of the Function Block into the User Program

4.1. Inserting the Function Block into the User Program

The function block *OVP81x-ProtV2* uses the blocks *TRCV_C* version V3.2 and *TSEND* version V4.0 from the integrated *Open user communication* library V7.0 (see Figure 2).

If you use other versions of these Siemens blocks in your project, this can lead to errors in the translation of the function block *OVP81x-ProtV2*. To avoid this, set your project to the version of the *Open user communication* library V7.0 before inserting the function block, or use a library version that includes the above versions of *TRCV_C* and *TSEND*. Note: The *Instructions* tab is visible only when you open any code block of your program.

Alternatively, you can edit the *OVP81x-ProtV2* function block after insertion and update the declarations and calls of the *TRCV_C* and *TSEND* blocks. Please note that other versions of these blocks may behave differently.

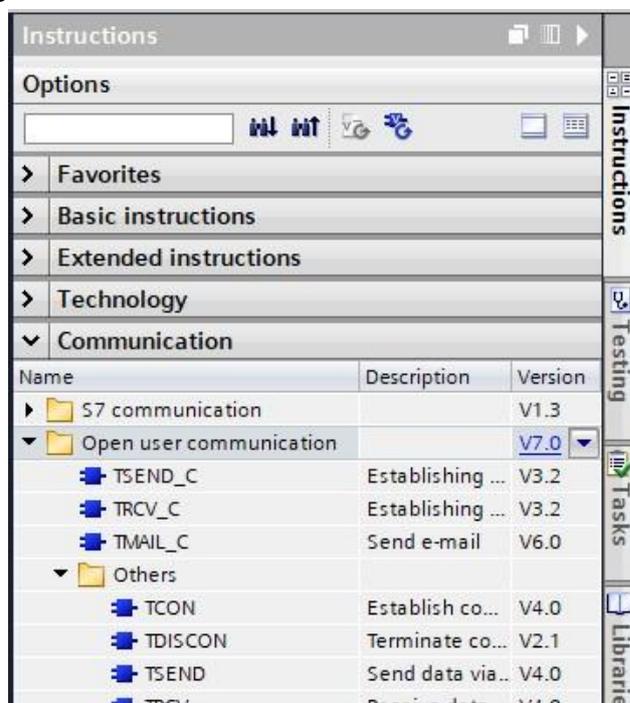


Figure 2

- Copy the function block *OVP81x-ProtV2* into your TIA Portal project.
- Compile the function block.
- Create an instance of the *OVP81x-ProtV2* function block. If you want to use several Video Processing Units, create a separate instance of the function block for each VPU in the user program.

4.2. Setting the Connection Parameters

The PLC and the Video Processing Unit (VPU) can only communicate on the network if they are on the same subnet. The most used subnet mask is 255.255.255.0. This means that the first three parts of the IP address must be the same for all devices on this subnet, and the last part must be different and unique in each device.

- Set the IP address and subnet mask in the Video Processing Unit. There are various ways to do this, for example with the help of the VisionAssistant software, which you can download from www.ifm.com.
- Set the IP address and subnet mask of the Ethernet interface of the S7 CPU (see chapter 6.3).



The TCP connection between PLC and VPU is only created and set up in the PLC user program. Do not configure this connection in the Devices & networks section of the TIA Portal project.

- Define the endpoints of the TCP connection between PLC and the PLC server application of the VPU. To do this, assign fixed values to the following connection parameters for each instance of the function block. Make sure that multiple instances are not communicating with the same VPU.

LocIF_ID

- Enter the hardware identifier of the Ethernet interface used by the PLC (see chapter 6.4).

This parameter of the function block is preset with the value 64. This corresponds to the hardware identifier of the first Ethernet interface of most standard CPUs. The parameter is hidden if the default assignment is not changed. If you are using a different CPU or interface, expand the block parameters display and change the assignment according to your hardware.



LocPortNo

- Specify the TCP port number that the PLC should use internally to connect to the VPU.

This parameter of the function block is preset with the value 0. With this setting, the function block calculates the locally used port number from 3000 + the value of the input parameter *RemIPAddr4*. However, the device cannot check whether the calculated local port number is still free. The parameter is hidden if the default assignment is not changed. If you want to set the local port yourself, expand the block parameters display and change the assignment according to your choice. For classic CPUs, Siemens has released the local port numbers 2000 to 5000 for use by the user. Specify a port number that has not yet been used.

Tip: If you are using a classic CPU, set *LocPortNo* = 0 on all instances of the function block *OVP81x-ProtV2*.

RemIPAddr1 .. RemIPAddr4

- Specify the IP address of the VPU.

RemPortNo

- Specify the TCP port number of the PLC server application of the VPU.

ConnID

- Assign a unique identification number to the connection. This is used in the S7 station as an internal reference. Keep in mind that connections to other devices can also occupy an ID.

This parameter of the function block is preset with the value 0. With this setting, the function block calculates the identification number from 3000 + the value of the input parameter *RemIPAddr4*. However, the block cannot check whether the calculated identification number is still free. The parameter is hidden if the default assignment is not changed. If you want to set the identification number yourself, expand the block parameters display and assign the parameter a unique value from the range between 1 and 4095.

Tip: Set *ConnID* = 0 on all instances of the function block *OVP81x-ProtV2*.

4.3. Calling the Instances of the Function Block in the User Program

- Call all instances of the function block *OVP81x-ProtV2* in the cyclic program once per PLC cycle.



5. Control of the ODS and PDS Applications of the VPU with the help of the Function Block *OVP81x-ProtV2*

5.1. Connect to the VPU

The function block automatically adjusts to the endianness used.

Establishing the connection

- Set the signal at input parameter *Connect* = TRUE.

The function block

- establishes a TCP connection with the set connection parameters
- shows output *Connected* = TRUE if the connection is established
- shows output *ComError* = TRUE if a communication error occurred
- displays communication status information in the *ComStatus* output (see [Table 13](#))
- automatically tries to reconnect in case of connection errors
- receives the data sent by the VPU and displays the individual values in the outputs

Terminating the connection

- Set the signal at input parameter *Connect* = FALSE.

The function block

- terminates the TCP connection
- displays *Connected* = FALSE, *ComError* = FALSE and *ComStatus* = NO_COM

5.2. ODS Application Commands

The function block supports following commands for one ODS application:

- Select Zone Preset ID
- Set Maximum Height
- Set Overhanging Load Mask

Only if *Enable* = TRUE in the *ODS_Params* inputs structure the function block sends ODS commands to the VPU.

For each of the ODS commands, the function block provides input parameters for the setting values in the *ODS_Params* inputs structure.

After establishing the connection to the VPU, the function block automatically sends the setting values to the VPU. The function block then monitors the setting values. If a value changes, the function block sends the corresponding command with the new setting value to the VPU.

If the command was not executed successfully, the function block repeats the command with the current setting value until the VPU confirms the successful execution of the command.

For each of the ODS commands, the function block displays the result of the command execution in the *ODS_Result* outputs structure. In addition, the function block displays the cyclic ODS data received from the VPU.

Activating a Zone Preset in the ODS Application

The zone presets are stored in the ODS application of the VPU and identified by an ID.

- Set *ODS_Params.Enable* = TRUE.
- At the input parameter *ODS_Params.TimeoutTime*, set the command response timeout time.
- At the input parameter *ODS_Params.PresetID*, set the ID of the zone preset you want to activate.

The function block

- sends the command to activate the specified zone preset every time the given ID is changed



- sends the command automatically after connecting to the VPU
- aborts the response reception every time the device does not respond within the set timeout time
- repeats sending the command periodically if it has previously failed
- displays *ActivatePresetSuccess* = TRUE in the output if the command was successfully executed
- shows *ActivatePresetFailed* = TRUE in the output if the command was not executed successfully
- displays in the *ActivatePresetErrorInfo* output which error occurred (see [Table 14](#))



A positive acknowledgement of the command does not mean that the zone preset is already active in the ODS. The configuration used to determine zone occupancy is always displayed in the *ODS_Result.ZoneConfigID* output.

Setting the Maximum Height in the ODS Application

- Set *ODS_Params.Enable* = TRUE.
- At the input parameter *ODS_Params.TimeoutTime*, set the command response timeout time.
- At the input parameter *ODS_Params.MaxHeight*, set the height value you want to set in the ODS application.

The function block

- sends the command to activate the specified maximum height every time the given height value is changed
- sends the command automatically after connecting to the VPU
- aborts the response reception every time the device does not respond within the set timeout time
- repeats sending the command periodically if it has previously failed
- displays *SetMaxHeightSuccess* = TRUE in the output if the command was successfully executed
- shows *SetMaxHeightFailed* = TRUE in the output if the command was not executed successfully
- displays in the *SetMaxHeightErrorInfo* output which error occurred (see [Table 14](#))

Setting the Overhanging Load Mask in the ODS Application

- Set *ODS_Params.Enable* = TRUE.
- At the input parameter *ODS_Params.TimeoutTime*, set the command response timeout time.
- At the input parameter *ODS_Params.OverhangMask*, set the bit mask you want to set in the ODS application.

The function block

- sends the command to activate the specified bit mask every time the given bit mask is changed
- sends the command automatically after connecting to the VPU
- aborts the response reception every time the device does not respond within the set timeout time
- repeats sending the command periodically if it has previously failed
- displays *SetOverhangSuccess* = TRUE in the output if the command was successfully executed
- shows *SetOverhangFailed* = TRUE in the output if the command was not executed successfully
- displays in the *SetOverhangErrorInfo* output which error occurred (see [Table 14](#))

5.3. PDS Application Commands

The function block supports two individual PDS applications.

For each of the PDS applications, the function block provides a separate entry for the input parameters in the *PDS_Command[]* array. The function block also displays the result data of each of the PDS applications in a separate entry in the *PDS_Result[]* output array.

The function block supports the commands listed in [Table 12](#) for each PDS application. You can select a command using the *Select* input parameter in the *PDS_Command[]* structure.



Table 12

| Select | Command |
|--------|--------------|
| 1 | Get Rack |
| 2 | Get Pallet |
| 3 | Get Item |
| 4 | Volume Check |
| Other | No command |

For each of the PDS commands, the function block provides individual substructures with input parameters for the specific setting values in the *PDS_Command[]* inputs structure.

After establishing the connection to the VPU, the function block automatically sends the selected PDS command of each PDS application with the actual setting values to the VPU. The function block then monitors the setting values as long the command is selected. If one of the values changes or a different command is selected, the function block sends the corresponding command with the new setting values to the VPU.

If the command was not executed successfully, the function block repeats the command with the current setting values until the VPU confirms the successful execution of the command or the command is deselected.

The function block displays in the *PDS_Result[]* output structure whether the command was executed successfully. The command result is displayed as long the command is selected. The command success is displayed only if the VPU sends the requested data. The function block displays this data for each command in a separate substructure. If another command is activated, the last received data of this command remains displayed.

Executing a PDS Command

- In the specific input parameter structure *PDS_Command[].<command>*, set the values of the command parameters.
- At the input parameter *PDS_Command[].TimeoutTime*, set the command response timeout time.
- Set the input parameter *PDS_Command[].Select* to the specific value of the command (see [Table 12](#)).

The function block sends the selected command to the specific PDS application of the VPU.

As long the command is still selected, the function block

- sends the command with current setting values if one of the setting values is changed
- aborts the response reception every time the device does not respond within the set timeout time
- repeats sending the command periodically with current setting values if it has previously failed
- displays *PDS_Result[].CommandSuccess* = TRUE if the command was successfully executed and the VPU sends the requested data
- shows *PDS_Result[].CommandFailed* = TRUE if the command was not executed successfully
- displays in the *PDS_Result[].CommandErrorInfo* output which error occurred (see [Table 14](#))
- displays the command result data of the VPU in the *PDS_Result[].<command>* substructure



6. Appendix

6.1. Display in the Output Parameter *ComStatus*

Table 13

| <i>ComStatus</i> | Description |
|------------------|---|
| NO_ERROR | No error |
| NO_COM | No communication with the PLC server of the VPU |
| CONNECTING | Establishing connection to the PLC server of the VPU |
| DEVICE_PROT | Device communication frame protocol error |
| PROT_VERSION | Wrong protocol version of the PLC server application of the VPU |
| TIMEOUT | Timeout of receiving cyclic data from VPU |
| TRCV_C_... | Siemens system function block TRCV_C displays an error. The reported error code follows as hexadecimal value. The complete list of error codes of the block can be found in the Siemens documentation of the block. Here is an excerpt of the most common error codes: |
| ... 8085 | Parameter LEN or DATA changed after first call or out of range |
| ... 8086 | Parameter <i>ConnID</i> out of range |
| ... 8087 | Maximum number of connections reached, no additional connection possible |
| ... 8091 | Maximum nesting depth exceeded |
| ... 809B | Parameter <i>HW_ID</i> does not point to the CPU Ethernet-interface |
| ... 80A1 | <ul style="list-style-type: none"> - Connection or port already being used by the user - The specified connection has not yet been established - The specified connection is being terminated - The interface is being re-initialized |
| ... 80A4 | IP-address at parameter <i>RemIPAddr</i> invalid or match the local IP-address |
| ... 80AA | A connection is currently being established with the same <i>ConnID</i> by another block |
| ... 80C3 | <ul style="list-style-type: none"> - All connection resources are in use - A block with this <i>ConnID</i> is already processed in a different priority group |
| ... 80C4 | <ul style="list-style-type: none"> - The connection cannot be established at this time - The connection is being established |
| ... 80C5 | Connection terminated by the communication partner |
| ... 80C6 | The remote partner cannot be reached (network error) |
| ... 80C7 | Execution timeout |
| ... 8922 | Parameter Data invalid target area, the area does not exist in the DB |
| ... 8932 | Parameter Data DB number is too high |
| ... 8A3A | Parameter Data no access to the data area (e.g. DB does not exist) |



6.2. Display in the Command Error Info Outputs in *ODS_Result* and *PDS_Result* Structures

Table 14

| ...ErrorInfo | Description |
|--------------|---|
| NO_ERROR | No error |
| NO_SELECTION | No command selected |
| NO_COM | No communication with the PLC server of the VPU |
| TIMEOUT | Timeout of receiving command response |
| INVALID_CMD | Unknown command |
| NOT_POSSIBLE | Device unable to execute the command due to parameter range or device state |
| RESPONSE_LEN | Unexpected response length received. Data cannot be evaluated. |

6.3. Setting the IP Address of the CPU in the TIA Portal

- Open the device configuration of the S7 station
- Switch to the *Device view* and select the CPU.
- By double-clicking on the Ethernet interface of the CPU used, you open the properties of the Ethernet interface.
- Open the *General* tab.
- In the *Ethernet addresses* folder, set the IP address and the subnet mask.

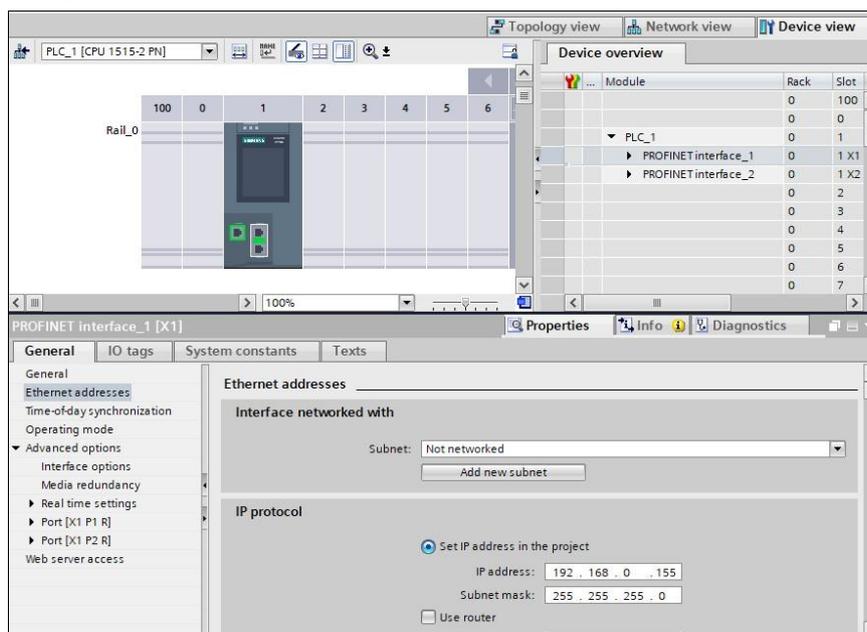


Figure 3

6.4. Displaying the Hardware ID of the CPU Ethernet Interface in the TIA Portal

- Open the properties of the Ethernet interface as described in Chapter 6.3
- Open the *System constants* tab.

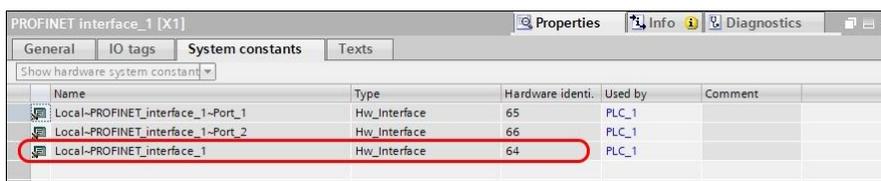


Figure 4

The hardware identifier of the Ethernet interface is displayed here. Specify this identifier at the instance of the function block *OVP81x-ProtV2* at the input parameter *LocIF_ID*.