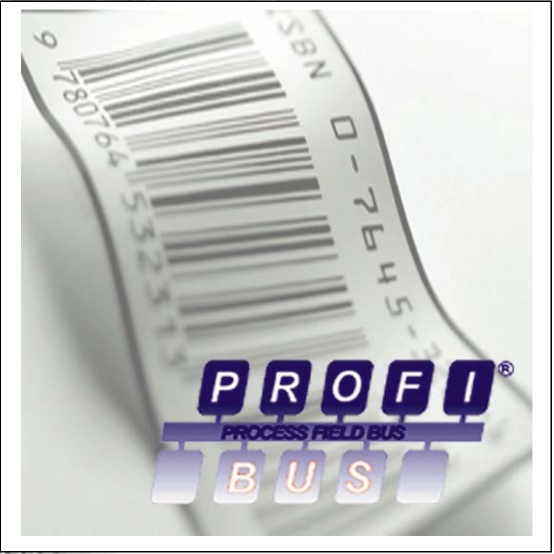
FACTORY AUTOMATION

Instruction Manual

PROFIBUS VB3X



CE

PEPPERL+FUCHS
VISOLUX

Subject to reasonable modifications due to technical advances.

General Information

This instruction manual contains information on the intended use of the product. It must be read and observed by all persons who implement, use, care for, maintain and monitor this product. This product can only fulfill the tasks for which it is intended if it is used, cared for, maintained and monitored in accordance with the instructions of Pepperl+Fuchs/Visolux.

The warrantee undertaken by Pepperl+Fuchs/Visolux for this product becomes null and void if it is not used, cared for, maintained and monitored in accordance with the instructions of Pepperl+Fuchs/Visolux.

Changes to the devices or components and the use of defective or incomplete devices or components are not permitted. Repairs to devices or components may only be performed by Pepperl+Fuchs/Visolux.

Repair tasks made on the product that are not performed by Pepperl+Fuchs/Visolux are not subject to influence on the part of Pepperl+Fuchs/Visolux.
Our liability is thus limited to repair tasks that are performed by Pepperl+Fuchs/Visolux.

The preceding information does not change information regarding warrantee and liability in the terms and conditions of sale and delivery of Pepperl+Fuchs.

Symbols used

This manual uses symbols to present important information on operating the VB3X and working safely with it. The meaning of these symbols is as follows:

Note

Recommendation for the user
Observing these notes will make it easier to place the system in operation and to work with the VB3X.

Subject to technical modifications.

Date of edition: 08/02/2004

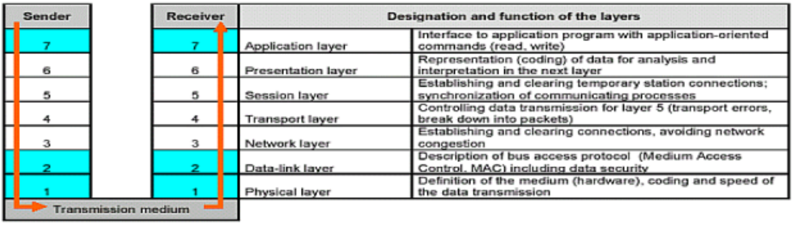
1 Overview

1.1 Description

Profibus is the world's most popular FieldBus.

Profibus is the most widely accepted international networking standard. Nearly universal in Europe and also popular in North America, South America and parts of Africa and Asia. Profibus can handle large amounts of data at high speed and serve the needs of the majority of automation applications.

Profibus was created under German Government leadership in co-operation with automation manufacturers (Siemens) in 1989. Today it is commonly found in Process Control, large assembly and material handling machines. Just a single-cable which is able to wire multi-input sensor blocks, pneumatic valves, complex intelligent devices, smaller sub-networks, operator interfaces and many other devices.



Sender	Receiver	Designation and function of the layers	
7	7	Application layer	Interface to application program with application-oriented commands (read, write)
6	6	Presentation layer	Representation (coding) of data for analysis and interpretation in the next layer
5	5	Session layer	Establishing and clearing temporary station connections; synchronization of communicating processes
4	4	Transport layer	Controlling data transmission for layer 5 (transport errors, break down into packets)
3	3	Network layer	Establishing and clearing connections, avoiding network congestion
2	2	Data-link layer	Description of bus access protocol (Medium Access Control, MAC) including data security
1	1	Physical layer	Definition of the medium (hardware), coding and speed of the data transmission

Figure 1.1 ISO/OSI reference model

The ISO/OSI reference model describes communications between the stations of a communication system: if a communication system does not require some specific functions, the corresponding layers have no purpose and are bypassed. Profibus uses layers 1, 2 and 7.

1.2 Profibus-DP

Basically Profibus is available in different versions:

- Profibus-DP (Decentralized Periphery)
  Multiple masters are possible with Profibus-DP, in which case each slave device is assigned to one master. This means that multiple masters can read inputs from the device but only one master can write outputs to that device.
- Profibus-FMS
  It is a peer to peer messaging format, which allows masters to communicate with one another. Just as in Profibus-DP, up to 126 nodes are available and all can be masters if desired. FMS messages consume more overhead than DP messages.
- Profibus-PA
  PA protocol is the same as the latest Profibus-DP except that voltage and current levels are reduced to meet the requirements of intrinsic safety (Class I div. II) for the process industry.

Pepperl+Fuchs supports Profibus-DP only, since this version has been specifically designed for factory automation. System version must be of this type.

Main features:

- Maximum Number of Nodes: 126
- Distance: 100m to 24 Km (with repeaters and fibre optic transmission)
- Baud rate: 9600 to 12M bps
- Messaging formats: Polling, Peer-to-Peer

From here on we will refer to Profibus-DP only.

2 Bus Communication

2.1 Data Exchange

Master Profibus is usually a PLC (Siemens S7 or others). Sometimes it could be a PC-based device as well.

VB3X-xxxx-P device is always ***Slave*** in the Profibus network.

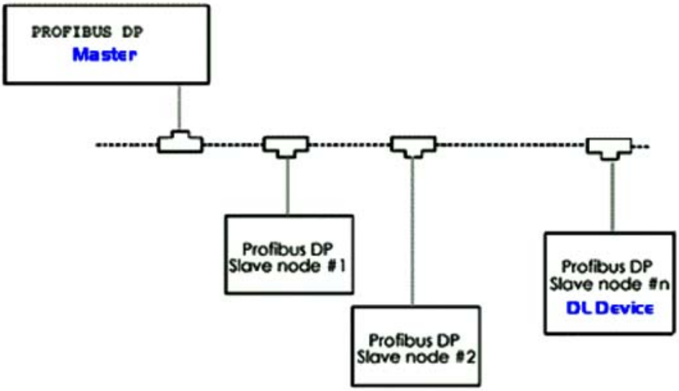


Figure 2.1 PROFIBUS network topology

Basically two shared memory areas (Exchange Areas) exist between SLAVE and MASTER so both devices provide information to each other. Exchange areas are physically placed on the VB3X Profibus card.

Date of edition: 08/02/2004

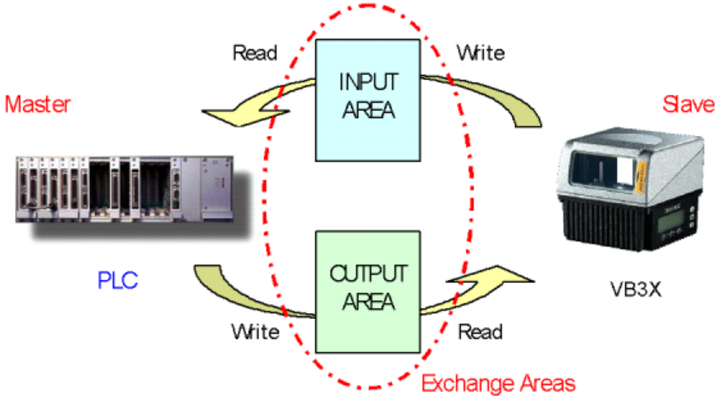


Figure 2.2 Data exchange

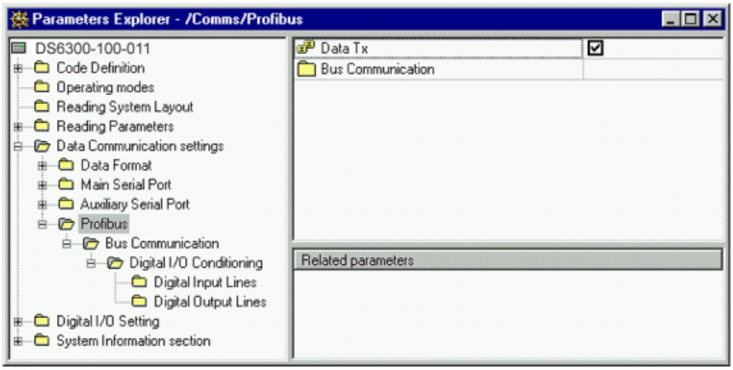
Input and output areas always refer to the Master: this means that the scanner writes to the Input buffer and the PLC writes to the Output buffer.

Dimensions of exchange areas can be set to different values by the PLC through the GSD file: the built-in Profibus models VB33/VB34 allow up to **128 bytes for the Input Area** and **32 bytes for the Output Area**.

2.2 Configuration Settings

The Configuration Tool allows complete parameterisation of Profibus communication.

In order to enable Profibus connectivity, the Data Tx parameter must be selected first (as default):



Data Tx ☑
Bus Communication

Figure 2.3 Data Tx parameterizing

2.2.1 Bus Communication Parameters

The Slave is identified over the Profibus network by setting its own address (Node Address parameter). This value, which is unique for each slave, should match the address specified while configuring your network, so that the PLC is able to detect the Slave properly.

2.2.2 Instructions for Profibus Address selection via Keypad:

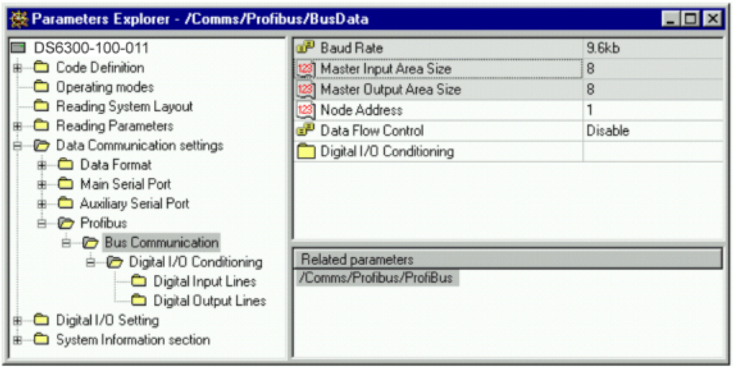
To enter the BUS submenu and configure the scanner follow the given procedure:

1. Press and hold both the ▲ (up arrow) and ▼ (down arrow) keys for about 2 seconds to enter the Main menu;
2. Use the ▲ (up arrow) or ▼ (down arrow) key to select the “Bus” item, then press the ENT (enter) key to confirm;
3. Use the ▲ (up arrow) or ▼ (down arrow) key to select the “BusAddrSel”” item, then press the ENT (enter) key to confirm;
4. Use the ▲ (up arrow) or ▼ (down arrow) key to select your Profibus address from 0 to 125 ; then, press the ENT (enter) key to confirm;
5. Use the ▲ (up arrow) or ▼ (down arrow) key to select the “Exit” item, then press the ENT (enter) key to confirm. Repeat this step again to exit the Main Menu and return to the scanner current operating mode.

Data Flow Control enables a powerful way to manage and optimise communication with the Profibus Master. A dedicated program running on the PLC is required to take advantage of this feature (see details in the " Flow Control Mode (FCM)" paragraph).

Final implementation should make use of Flow Control to obtain maximum reliability and optimal synchronisation between Master and Slave. As a first approach, Data Flow Control = Disable is the suggested way to check the HW/SW configuration of the Profibus network.

Within the Bus Communication section Baud Rate, Master Input Area Size and Master Output Data Size are read only, as these parameters are decided by the Profibus Master.



Baud Rate	9.6kb
Master Input Area Size	8
Master Output Area Size	8
Node Address	1
Data Flow Control	Disable
Digital I/O Conditioning	

Figure 2.4 Baud Rate parameterizing

2.3 Flow Control Mode (FCM)

The Flow Control Mode is a powerful way to manage and optimise the communication with the Profibus Master. By enabling the FCM a few bytes of the exchange areas are reserved for driver operations and the rest are used by the application layer.

The reserved bytes are used to implement many different features such as:

- Flow-control and corresponding buffering in both directions
- Fragmentation and reassembling of data longer than the exchange area sizes
- Synchronisation of flow control numbers
- Service Access Point oriented communication
- Length information

Note

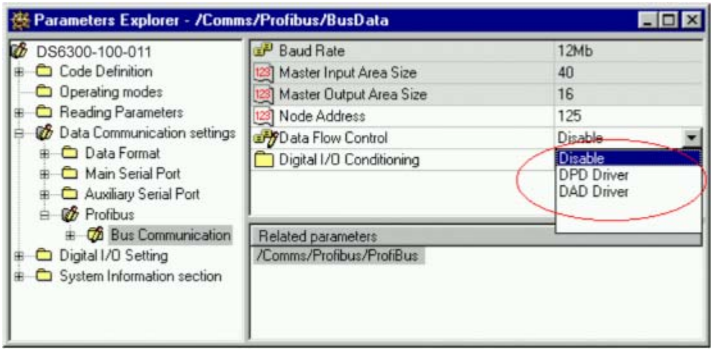
If the Flow Control is disabled, all the bytes of the exchange areas are used by the application layer. The input area is updated whenever a new reading event has to be transferred to the Master station.

➜ In this situation, the Master must read the input area before it changes due to a new message, typically new barcode occurrence.

➜ Moreover, two occurrences of the same barcode cannot be understood, since the input area does not change.

➜ In addition, if application data is longer than input area sizes, data is automatically truncated.

FCM can be selected by means of the Data Flow Control parameter.



Baud Rate	12Mb
Master Input Area Size	40
Master Output Area Size	16
Node Address	125
Data Flow Control	Disable
Digital I/O Conditioning	

Figure 2.5 Data Flow Control parameterizing

Basically three options are available:

- DAD Driver ➜ FCM compatible with "Flow Control = Anybus" used in new devices
- DPD Driver ➜ FCM compatible with "Flow Control = Profibus" used in multiplexer
- Disable ➜ No Flow Control

Although both drivers implement the same above features, two options are currently available to obtain the maximum compatibility towards different family devices.

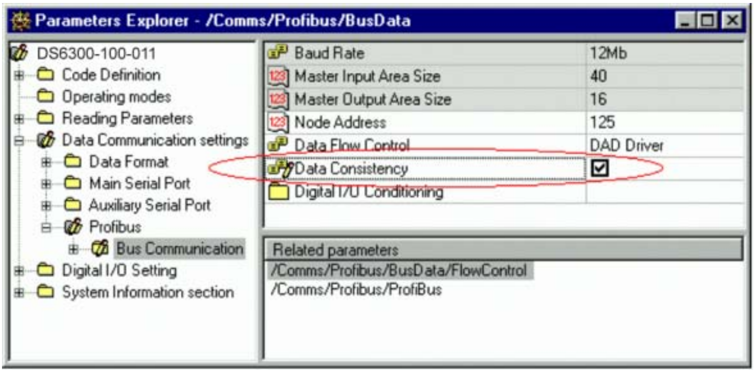
Date of edition: 08/02/2004

See the cross reference table below about supported FCM:

VB3x-Device	Configuration Software Identification Code-No.	DAD Driver	DPD Driver
VB33-2000-P	DS6300-100-011	x	x
VB33-2000-OM-P	DS6300-105-011	x	x
VB34-2500-P	DS6400-100-011	x	x
VB34-2500-OM-P	DS6400-105-011	x	x
Other Device			
MX4000-1100			x
MX4000-1100 SB2498 12MB/s		x	
CBOX-300 Profibus-DP		x	x

When FCM is utilised, the Data Consistency option can be enabled to improve the overall safety in data communication. This option is mandatory if Master Input Area size exceeds 32 bytes.

See details in the next paragraphs.



Parameters Explorer - /Comms/Profibus/BusData

Baud Rate	12Mb
Master Input Area Size	40
Master Output Area Size	16
Node Address	125
Data Flow Control	DAD Driver
Data Consistency	☑
Digital I/O Conditioning	

Related parameters
/Comms/Profibus/BusData/FlowControl
/Comms/Profibus/ProfiBus

Figure 2.6 Data Constistency

2.3.1 Flow Control drivers

The Flow Control driver is a layer that is built upon the intrinsic DP data exchange mechanism. Basically such a layer is required because the intrinsic DP Profibus mechanism is not message oriented.

In the following figure the complete Stack is represented:

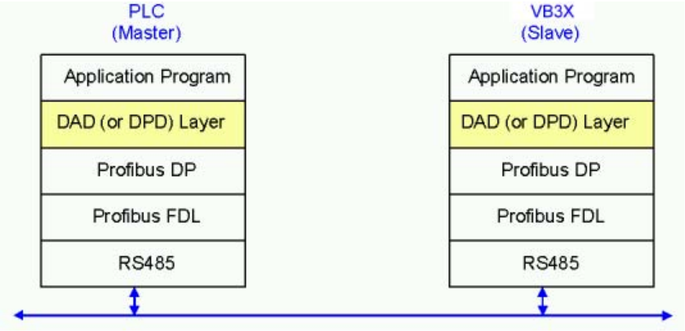


Figure 2.7 Flow Control driver

Two Flow Control drivers are available:

- DAD Driver à Compatible with Flow Control = Anybus
- DPD Driver à Compatible with Flow Control = Profibus

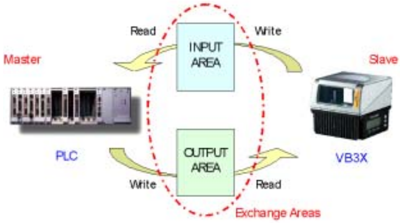
As described previously, they can be selected separately on VB3X by means of the Configuration tool. Obviously, the corresponding driver must be implemented on the PLC side.

The DAD driver should be the preferred solution for brand new installations.

2.3.2 FCM with DAD Driver

In order to implement the flow controlled version of the driver, exchange areas must be congruently compiled in both directions.

From now on we refer to the Input Area as a buffer made up of InputAreaSize bytes:



IN[0]	IN[1]	IN[2]	...	IN[InputAreaSize - 1]

and to the Output Area as a buffer made up of OutputAreaSize bytes:

OUT[0]	OUT[1]	OUT[2]	...	OUT[OutputAreaSize - 1]

Only the first three bytes are used by the DAD Driver layer in both buffers:

Control Field (byte 0)	used to issue and control the driver primitives such as flow-control, fragmentation and resynchronisation.
Service Access Point Field (byte 1)	used to distinguish among different services and to provide future expandability. (Since this SAP definition is introduced by the DAD Driver, it must not be confused with the SAP that is defined by the international standard).

Length Field

(byte 2)

contains the number of bytes used by the application layer:
Length Field = (InputAreaSize – 3) for the Input Area
Length Field = (OutputAreaSize – 3) for the Output Area.

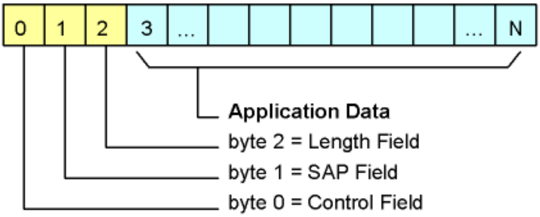


Figure 2.8 Exchange Input/Output Area buffer

The Application Data buffer holds useful information, typically the barcode messages, processed by the application program. IN[3] contains the first significant byte of the Application Data buffer (the same first byte you would see if VB3X transmitted the barcode buffer onto the Auxiliary port instead of the Profibus interface).

The structure of the application buffer and its length strictly depend on the selected data format on the VB3X. Barcode messages longer than (InputAreaSize – 3) will be split in pieces through an automatic fragmentation process (see details in the "Fragmentation and Reassembling" paragraph).

Control Field

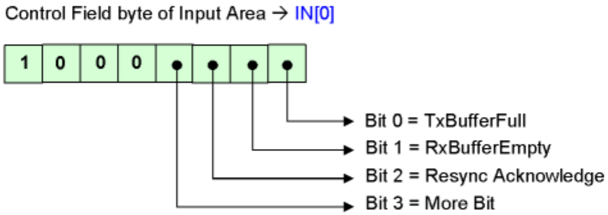
This is the core of the flow controlled communication.

The Input Area structure reserves bit 0 and bit 1 of IN[0] for handshake purposes while the Output Area structure, which is symmetrical, reserves bit 0 and 1 of OUT[0].

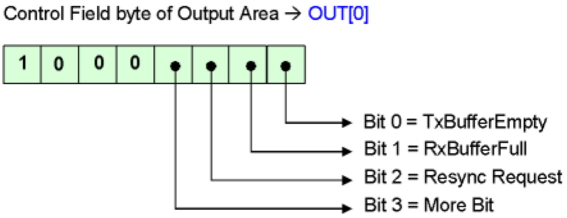
At any time the Master station can make a resynchronization request by means of bit 2 of the Output Area. This process, which resets the synchronization numbers (bit 0 and bit 1 of both Input and Output areas), has to be acknowledged by the Slave on bit 2 of the Input Area.

Bit 3 is used to control a fragmentation sequence in both directions.

Subject to reasonable modifications due to technical advances.


Control Field byte of Input Area → IN[0]
1
0
0
0
●
●
●
●
Bit 0 = TxBufferFull
Bit 1 = RxBufferEmpty
Bit 2 = Resync Acknowledge
Bit 3 = More Bit

IN[0].bit0	**TxBufferFull** toggles when Slave has made available new Input Area data
IN[0].bit1	**RxBufferEmpty** toggles when Output Area data has been read by Slave
IN[0].bit2	**Resync Acknowledge** set to 1 as an acknowledge to a resync request. With this bit, the master can detect a slave is on line.
IN[0].bit3	**More Bit** is 1 when this is not the last piece of a fragmentation sequence while it is 0 when this is the last piece
IN[0].bit4,5,6,7	Set to **0, 0, 0, 1** when DAD messaging protocol is used



OUT[0].bit0	**TxBufferEmpty** must toggle when Input Area data has been read by Master
OUT[0].bit1	**RxBufferFull** must toggle when Master makes available new Output Area data
OUT[0].bit2	**Resync Request** set to 1 for one second to resynchronize the slave. After resynchronization, all 4 handshake bits are set to 0
OUT[0].bit3	**More Bit** must be 1 when this is not the last piece of a fragmentation sequence and it must be 0 when this is the last
OUT[0].bit4,5,6,7	Set to **0, 0, 0, 1** when DAD messaging protocol is used

SAP Field

SAP (Service Access Point) is an identifier that is used to implement multiple services sharing the same communication channel between two remote stations.

The following values have been defined:

• SAP = 0	Used to transfer information messages between VB3X and PLC
• SAP = 255	Reserved for driver services (see details in the "SAP Services” paragraph)

All other SAP values are free and they could be used by dedicated application programs after agreement between the application programs themselves.

Length Field

The Application layer uses all or a part of the remaining bytes of the Exchange Area buffers that are not used by the DAD Driver. The Length Field is introduced to keep the information of how many bytes are really used by the Application Layer.

A fragment that is not the last one of a fragmentation sequence must fill this field with [InputAreaSize – 3] (or [OutputAreaSize - 3]), depending on whether it is an Input/ Output fragment. Otherwise this field gets a value that is less than or equal to [InputAreaSize – 3].

Data Transmission from VB3X to PLC

This paragraph describes how it is possible to exchange messages with flow control. The communication mechanism is simple:

- IN[0].bit0 [A] is used by VB3X to notify that "Slave has written a new data so Master can read it"
- OUT[0].bit0 [B] must be used by PLC to notify that "Master has read last data so Slave can send next message"

This happens each time bit A (or B) changes its state (toggles). Bit level doesn't matter, only the transition has to be considered.

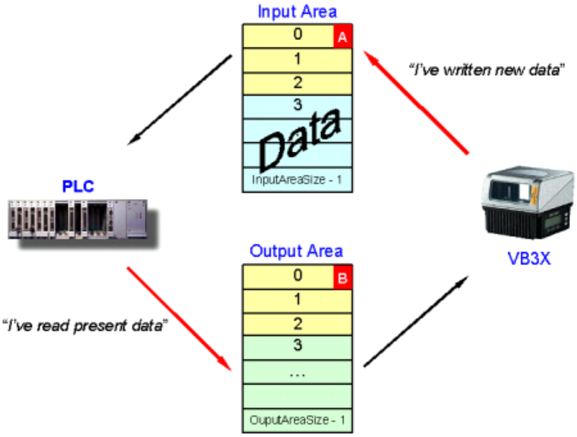


Figure 2.9 Message exchange with flow control

The following state machine shows data transmission from Slave to Master. Please note that each cycle transfers two data messages.

Date of edition: 08/02/2004

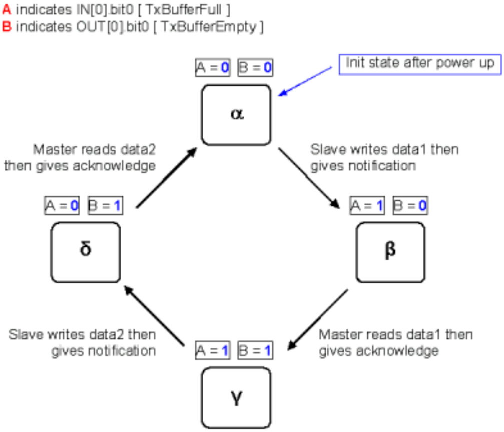

A indicates IN[0].bit0 [TxBufferFull]
B indicates OUT[0].bit0 [TxBufferEmpty]
A = 0 B = 0
Init state after power up
α
Slave writes data1 then gives notification
A = 1 B = 0
β
Master reads data1 then gives acknowledge
A = 1 B = 1
γ
Slave writes data2 then gives notification
A = 0 B = 1
δ
Master reads data2 then gives acknowledge

Figure 2.10 State machine

Let's analyse a typical data exchange based on the following settings:

- Flow Control = DAD Driver
- Input Area Size = 16
- Output Area Size = 8

- After power up Input and Output areas are generally filled by zero. According to DAD driver implementation, Input area has Control Field = 80h and SAP = 00h.

Input Area	80h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h
Output Area	00h 00h 00h 00h 00h 00h 00h 00h

- Also PLC must set the Control Field of Output area properly, as long as DAD messaging protocol is utilised.

Input Area	80h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h
Output Area	80h 00h 00h 00h 00h 00h 00h 00h

- VB3X reads a barcode "123456". Let's assume standard data formatting with <STX> as header and <CR><LF> as terminators. VB3X toggles bit A.

Input Area	81h 00h 09h 02h 31h 32h 33h 34h 35h 36h 0Dh 0Ah 00h 00h 00h 00h
Output Area	80h 00h 00h 00h 00h 00h 00h 00h

- PLC detects transition of bit A so now it can read incoming data (it copies 9 bytes in its memory from IN[3] on) then toggles bit B as acknowledge.
  Note: before the acknowledge, all further barcodes read by VB3X are buffered.

Input Area	81h 00h 09h 02h 31h 32h 33h 34h 35h 36h 0Dh 0Ah 00h 00h 00h 00h
Output Area	81h 00h 00h 00h 00h 00h 00h 00h

- VB3X reads a barcode "10DL" and toggles bit A.

Input Area	80h 00h 07h 02h 31h 30h 44h 4Ch 0Dh 0Ah 00h 00h 00h 00h 00h 00h

Output Area	81h 00h 00h 00h 00h 00h 00h 00h

- PLC reads new data message (it copies 7 bytes in its memory from IN[3] on) then toggles bit B as acknowledge.

Input Area	80h 00h 07h 02h 31h 30h 44h 4Ch 0Dh 0Ah 00h 00h 00h 00h 00h 00h
Output Area	80h 00h 00h 00h 00h 00h 00h 00h

- VB3X performs No Read and toggles bit A. Let's assume <CAN> as Global No Read character.

Input Area	81h 00h 04h 02h 18h 0Dh 0Ah 00h 00h 00h 00h 00h 00h 00h 00h 00h
Output Area	80h 00h 00h 00h 00h 00h 00h 00h

- PLC reads new data message (it copies 4 bytes in its memory from IN[3] on) then toggles bit B as acknowledge.

Input Area	81h 00h 04h 02h 18h 0Dh 0Ah 00h 00h 00h 00h 00h 00h 00h 00h 00h
Output Area	81h 00h 00h 00h 00h 00h 00h 00h

- Data exchange continues…

Data Transmission from PLC to VB3X

Analogous to the previous paragraph, flow control works even when data are coming from the Master towards the Slave. The communication mechanism is based on the same concepts:

- rOUT[0].bit1 [C] must be used by PLC to notify that "Master has written new data so Slave can read it"
- IN[0].bit1 [D] is used by VB3X to notify that "Slave has read data so Master can send next message"

This happens each time bit C (or D) changes its state (toggles). Bit level doesn't matter, only the transition has to be considered.

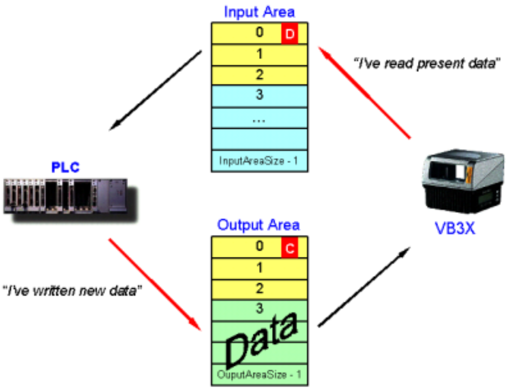


Figure 2.11 Message exchange with flow control

The following state machine shows data transmission from Master to Slave. Please

Date of edition: 08/02/2004

note that each cycle transfers two data messages.

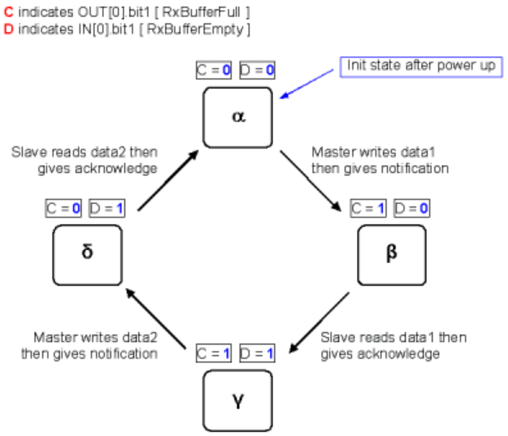

C indicates OUT[0].bit1 [RxBufferFull]
D indicates IN[0].bit1 [RxBufferEmpty]
C = 0 D = 0
Init state after power up
α
Master writes data1 then gives notification
C = 1 D = 0
β
Slave reads data1 then gives acknowledge
C = 1 D = 1
γ
Master writes data2 then gives notification
C = 0 D = 1
δ
Slave reads data2 then gives acknowledge

Figure 2.12 State machine

Resynchronisation

The resynchronisation process restarts the messaging protocol from a predefined state.

It may be used either at the Master startup to detect if the Slave is on line or during normal operations in case of errors requiring a protocol reset procedure.

The process is based on bit two:

- OUT[0].bit2 [E] must be used by PLC to request the Resynchronisation
- IN[0].bit2 [F] is used by VB3X to acknowledge the request

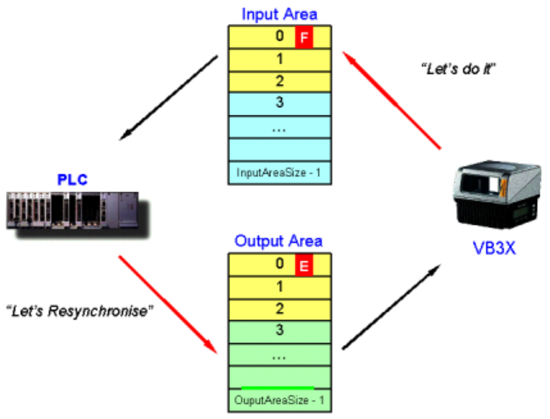


Figure 2.13 Resynchronization

Date of edition: 08/02/2004

The following state machine shows the resynchronisation cycle, requested by the PLC and performed together with VB3X:

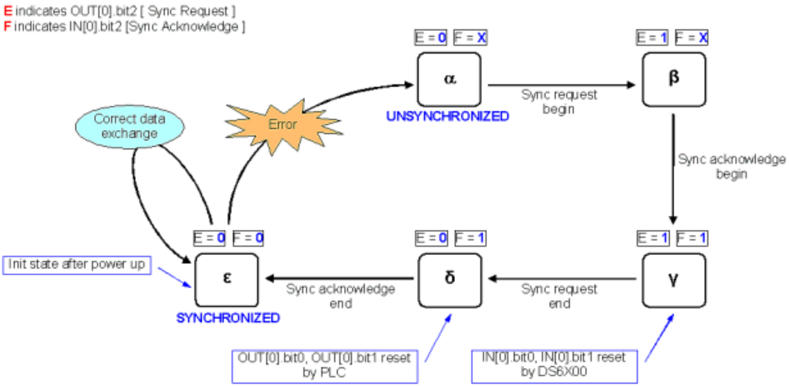


Figure 2.14 State machine

Let’s analyse the resynchronisation process, starting from the previous data exchange discussed in “Data Transmission from VB3X to PLC”…

Input Area	81h 00h 04h 02h 18h 0Dh 0Ah 00h 00h 00h 00h 00h 00h 00h 00h 00h
Output Area	81h 00h 00h 00h 00h 00h 00h 00h

- PLC requests resynchonisation by setting bit E = 1.

Input Area	81h 00h 04h 02h 18h 0Dh 0Ah 00h 00h 00h 00h 00h 00h 00h 00h 00h
Output Area	85h 00h 00h 00h 00h 00h 00h 00h

- VB3X detects the request and so it resets IN[0].bit0 and IN[0].bit1. Then it gives an acknowledge back to the PLC by means of bit F.

Input Area	84h 00h 04h 02h 18h 0Dh 0Ah 00h 00h 00h 00h 00h 00h 00h 00h 00h
Output Area	85h 00h 00h 00h 00h 00h 00h 00h

- PLC has to reset OUT[0].bit0 and OUT[0].bit1 before completing its request with bit E = 0.

Input Area	84h 00h 04h 02h 18h 0Dh 0Ah 00h 00h 00h 00h 00h 00h 00h 00h 00h
Output Area	80h 00h 00h 00h 00h 00h 00h 00h

- VB3X completes the acknowledge process by setting bit F = 0.

Input Area	80h 00h 04h 02h 18h 0Dh 0Ah 00h 00h 00h 00h 00h 00h 00h 00h 00h
Output Area	80h 00h 00h 00h 00h 00h 00h 00h

- Now Flow Control has been returned to a predefined state. All data exchange bits in the Control Field are surely zero and data transmission can proceed safely.

Fragmentation and Reassembling

The fragmentation process is activated whenever Application Data cannot be contained in the related exchange area. Basically long messages are split into pieces which are transmitted separately. Reassembling allows the reconstruction of the whole messages.

VB3X already implements these functions in the DAD layer, while the PLC needs a congruent management.

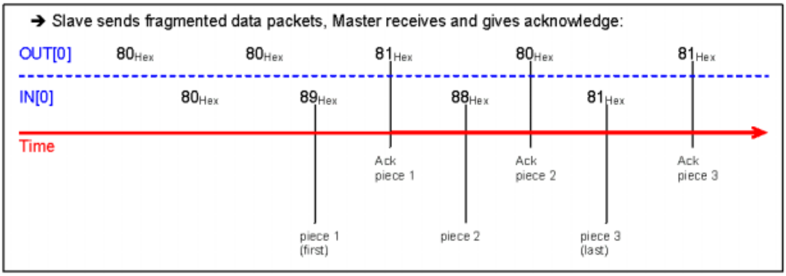
The fragmentation is based on the **More Bit** (bit 3) in the Control Field byte.

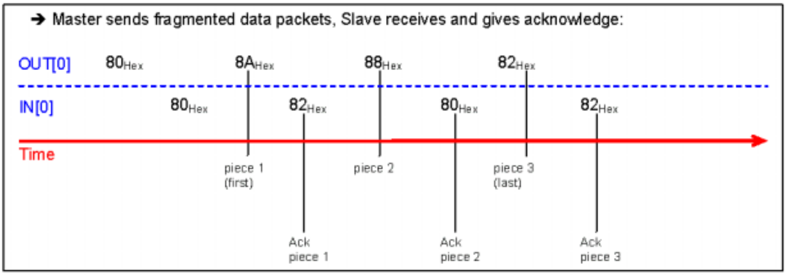
More Bit = 0 indicates that all the information is included within the current message. When Application Data is longer than (exchange area size – 3), the first partial message is transmitted having More Bit = 1. Following fragments keep More Bit = 1 and only the last piece will have More Bit = 0 again. Thanks to this mechanism, the receiver station may detect the last piece and so reassemble the entire information.

Some notes:

- VB3X can manage application messages up to 256 bytes
- Intermediate fragments have Length Field = (exchange area size – 3)
- Last fragment has Length Field = (exchange area size – 3)
- Bit0 and bit1 of both Input and Output areas are independently managed for any fragment

The following figures show how the Control Byte changes according to the fragmentation process. Both data flow directions are considered.


➔ Slave sends fragmented data packets, Master receives and gives acknowledge:


➔ Master sends fragmented data packets, Slave receives and gives acknowledge:

Let's analyse a fragmented data exchange based on the following settings:

- Flow Control = DAD Driver
- Input Area Size = 16
- Output Area Size = 8

- After power up Input and Output areas are generally filled by zero. According to DAD driver implementation, Input area has Control Field = 80Hex and SAP = 00Hex.

Input Area	80h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h
Output Area	00h 00h 00h 00h 00h 00h 00h 00h

- Also PLC must set the Control Field of Output area properly, as long as DAD messaging protocol is utilised.

Input Area	80h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h
Output Area	80h 00h 00h 00h 00h 00h 00h 00h

- VB3X reads a barcode with content “1234567890abcde1234567890abcde”. Let’s assume standard data formatting with <STX> as header and <CR><LF> as terminators. In this condition since the whole message cannot be included in Input Area, VB3X transmits first fragment "<STX>1234567890ab" only (setting More Bit = 1) then it toggles bit A.

Input Area	89h 00h 0Dh 02h 31h 32h 33h 34h 35h 36h 37h 38h 39h 30h 61h 62h
Output Area	80h 00h 00h 00h 00h 00h 00h 00h

- PLC detects transition of bit A so now it can read first incoming fragment (it copies 13 bytes in its memory from IN[3] on) then toggles bit B as acknowledge.

Input Area	89h 00h 0Dh 02h 31h 32h 33h 34h 35h 36h 37h 38h 39h 30h 61h 62h
Output Area	81h 00h 00h 00h 00h 00h 00h 00h

- VB3X detects transition of bit B so it sends second fragment "cde1234567890“ (still More Bit = 1) and toggles bit A.

Input Area	88h 00h 0Dh 63h 64h 65h 31h 32h 33h 34h 35h 36h 37h 38h 39h 30h
Output Area	81h 00h 00h 00h 00h 00h 00h 00h

- PLC reads second fragment (it copies 13 bytes in its memory from IN[3] on) then toggles bit B as acknowledge.

Input Area	88h 00h 0Dh 63h 64h 65h 31h 32h 33h 34h 35h 36h 37h 38h 39h 30h
Output Area	80h 00h 00h 00h 00h 00h 00h 00h

- VB3X sends third (last) fragment "abcde<CR><LF>“ (finally More Bit = 0) and toggles bit A.

Input Area	81h 00h 07h 61h 62h 63h 64h 65h 0Dh 0Ah 00h 00h 00h 00h 00h 00h
Output Area	80h 00h 00h 00h 00h 00h 00h 00h

- PLC reads last fragment (it copies 7 bytes in its memory from IN[3] on) and now the reassembling can be completed. Then it toggles bit B as acknowledge.

Input Area	81h 00h 07h 61h 62h 63h 64h 65h 0Dh 0Ah 00h 00h 00h 00h 00h 00h
Output Area	81h 00h 00h 00h 00h 00h 00h 00h

- Whole message has been completely transmitted.

SAP Services

FLUSH QUEUE is the unique driver service currently available. It performs flushing of the internal queues and may be issued at any time.

FLUSH QUEUE Service

Request:	Flush data buffers (issued by the Master station to VB3X)
Action:	Flush all information from previous decoding phases
Response:	Command accepted/Command rejected (generated by VB3X toward Master)

Application data areas must be formatted as follows:

Request Command	byte 3	byte 4
Flush data buffer	**'['** (5B Hex)	**'F'** (46 Hex)

Response Command	byte 3	byte 4
Command accepted	**'A'** (41 Hex)	**' '** (20 Hex)
Command rejected	**'B'** (43 Hex)	**' '** (20 Hex)

DAD internal queues

VB3X-10X-012 has two internal queues (one for each direction) to keep the application events: input queue and output queue.

The input queue is used when a new message (generally a barcode) has to be transmitted by D6X000 before the Master station has generated all the acknowledge handshakes for each previous transmission.

The output queue is rarely used at the moment.

The queues are sized in the following way:

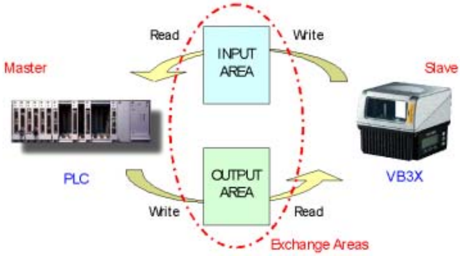
- 350 elements of 128 bytes each for input queue (data flow from Slave to Master)
- 26 elements of 32 bytes each for output queue (data flow from the Master to Slave)

The queues may be flushed by the Master station through the SAP = 255 primitive. This is generally done at the Master startup if the Master station wants to cancel all the previous buffers that were generated before its startup. However, the Master station is free to decide not to cancel them.

2.3.3 FCM with DPD Driver

In order to implement the flow controlled version of the driver, exchange areas must be congruently compiled in both directions.

This driver supports all the features listed in the “FCM with DAD Driver” paragraph and implements all the handshake mechanisms previously discussed.



Profibus Station Address Field is the only different requirement. The first four bytes are used by the DPD Driver layer in both buffers as follows:

Control Field
(byte 0)
used to issue and control the DPD Driver primitives such as flow-control, fragmentation and resynchronisation.

Profibus Station Address Field
(byte 1)
contains the information of the selected VB3X address of the Profibus network.

Service Access Point Field
(byte 2)
used to distinguish among different services and to provide future expandability. (Since this SAP definition is introduced by the DPD Driver, it must not be confused with the SAP that is defined by the international standard).

Length Field
(byte 3)
contains the number of bytes used by the application layer:Length Field = (InputAreaSize – 4) for the Input AreaLength Field = (OutputAreaSize – 4) for the Output Area.

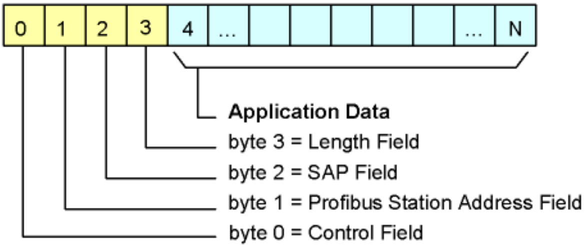


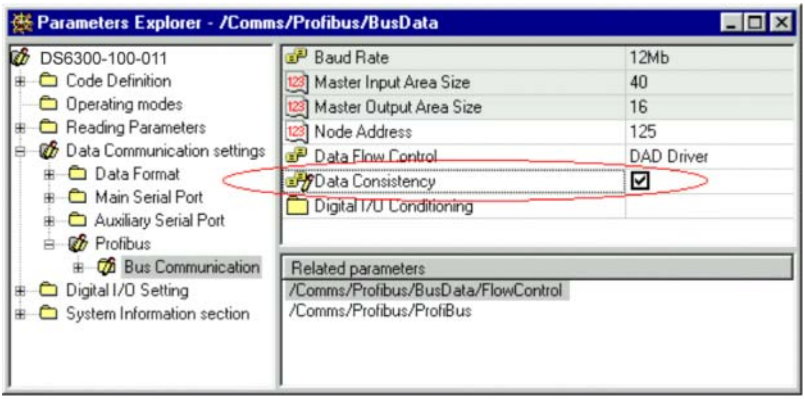
Figure 2.15 Exchange Input/Output Area buffer

The Application Data buffer holds the information starting from byte IN[4] and the fragmentation process is managed by VB3X when barcode messages are longer than (InputAreaSize – 4).

2.4 Data Consistency

When Flow Control (either DAD or DPD Driver) is used, the Data Consistency option is available.

If enabled, the driver implements a specific mechanism to guarantee consistency of both transmitted and received data over the entire size of the exchange area.



Parameters Explorer - /Comms/Profibus/BusData

- DS6300-100-011
  - Code Definition
  - Operating modes
  - Reading Parameters
  - Data Communication settings
    - Data Format
    - Main Serial Port
    - Auxiliary Serial Port
    - Profibus
      - Bus Communication
  - Digital I/O Setting
  - System Information section

Baud Rate	12Mb
Master Input Area Size	40
Master Output Area Size	16
Node Address	125
Data Flow Control	DAD Driver
Data Consistency	☑
Digital I/O Conditioning	

Related parameters

/Comms/Profibus/BusData/FlowControl
/Comms/Profibus/ProfiBus

Figure 2.16 Data consistency

2.4.1 Data Consistency with DAD Driver

As shown in the following figure, the application layer copies the Control Field byte in the last position of the exchange area. In this way, as the Control Field changes from message to message, the whole message is consistent as long as the first and last bytes are matching.

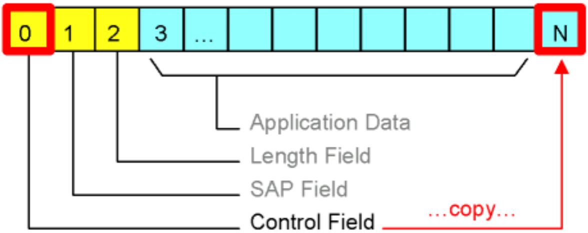


Figure 2.17 Exchange Input/Output Area buffer

From the practical point of view, a complete message in the Input area could be considered consistent as soon as the PLC verifies that IN[0] is equal to IN[InputAreaSize - 1].

The PLC should take care to double OUT[0] in OUT[OutputAreaSize - 1] to let the driver check the consistency.

Due to the new byte used by the DAD driver, barcode messages longer than (InputAreaSize – 4) will be split into pieces through an automatic fragmentation process.

Let's analyse a fragmented and consistent data exchange based on:

- Flow Control = DAD Driver
- Data Consistency = Enable
- Input Area Size = 16
- Output Area Size = 8

- After power up Input and Output areas are generally filled by zero. According to DAD driver implementation, Input area has Control Field = IN[0] = IN[15] = 80Hex and SAP = 00Hex.

Input Area	80h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 80h
Output Area	00h 00h 00h 00h 00h 00h 00h 00h

- PLC must set Control Field = OUT[0] = OUT[7] = 80Hex according to DAD messaging protocol with Data Consistency.

Input Area	80h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 00h 80h
Output Area	80h 00h 00h 00h 00h 00h 00h 80h

- VB3X reads a barcode with content “1234567890abcde1234567890abcde”. Let’s assume standard data formatting with <STX> as header and <CR><LF> as terminators. VB3X transmits first fragment "<STX>1234567890a" only (setting More Bit = 1) then it toggles bit A.

Input Area	89h 00h 0Ch 02h 31h 32h 33h 34h 35h 36h 37h 38h 39h 30h 61h 89h
Output Area	80h 00h 00h 00h 00h 00h 00h 80h

- PLC detects transition of bit A so now it can read first incoming fragment (it copies 12 bytes in its memory from IN[3] on) then toggles bit B as acknowledge.

Input Area	89h 00h 0Ch 02h 31h 32h 33h 34h 35h 36h 37h 38h 39h 30h 61h 89h
Output Area	81h 00h 00h 00h 00h 00h 00h 81h

- VB3X detects transition of bit B so it sends second fragment "bcde12345678“ (still More Bit = 1) and toggles bit A.

Input Area	88h 00h 0Ch 62h 63h 64h 65h 31h 32h 33h 34h 35h 36h 37h 38h 88h
Output Area	81h 00h 00h 00h 00h 00h 00h 81h

- PLC reads second fragment (it copies 12 bytes in its memory from IN[3] on) then toggles bit B as acknowledge.

Input Area	88h 00h 0Ch 62h 63h 64h 65h 31h 32h 33h 34h 35h 36h 37h 38h 88h
Output Area	80h 00h 00h 00h 00h 00h 00h 80h

- VB3X sends last fragment "90abcde<CR><LF>“ (finally More Bit = 0) and toggles bit A.

Input Area	81h 00h 09h 39h 30h 61h 62h 63h 64h 65h 0Dh 0Ah 00h 00h 00h 81h
Output Area	80h 00h 00h 00h 00h 00h 00h 80h

- PLC reads last fragment (it copies 9 bytes in its memory from IN[3] on) and now the reassembling can be completed. Then it toggles bit B as acknowledge.

Input Area	81h 00h 09h 39h 30h 61h 62h 63h 64h 65h 0Dh 0Ah 00h 00h 00h 81h
Output Area	81h 00h 00h 00h 00h 00h 00h 81h

- Whole message has been completely transmitted.

2.4.2 Data Consistency with DPD Driver

Here the application layer still copies the Control Field byte in the last position of the exchange area. Data exchange structure appears slightly different, as shown in following picture:

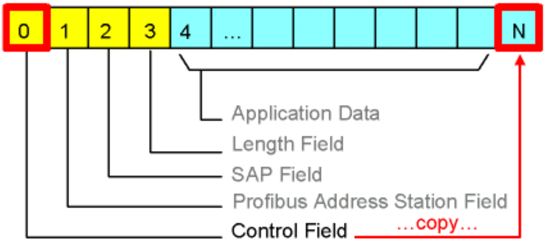


Figure 2.18 Exchange Input/Output Area buffer

If the DPD driver is used, barcode messages longer than (InputAreaSize – 5) will be split into pieces through an automatic fragmentation process.

3 Digital I/O Conditioning

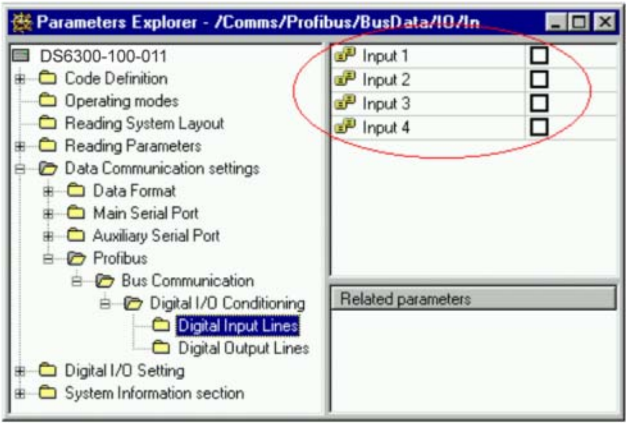
This function offers the possibility to use the scanner I/O as remote peripherals of the Host:

- Digital Output Conditioning allows the PLC to force the level of the three physical VB3X outputs
- Digital Input Conditioning allows the VB3X to notify the status of its four physical inputs to the PLC

Some of the provided benefits are:

- Profibus Master knows the status of each single scanner input, i.e a parcel is coming, an oversized piece of baggage has been detected or a piece of baggage has reached the data Tx line and so on…
- Profibus Master is able to force the scanner in reading mode
- Profibus Master is able to modify the scanner outputs status depending on the received barcode message (sorting application)
- No dedicated sw on the scanner is necessary to manage its I/Os.

VB3X digital I/O conditioning is activated whenever one of the below parameters is checked:



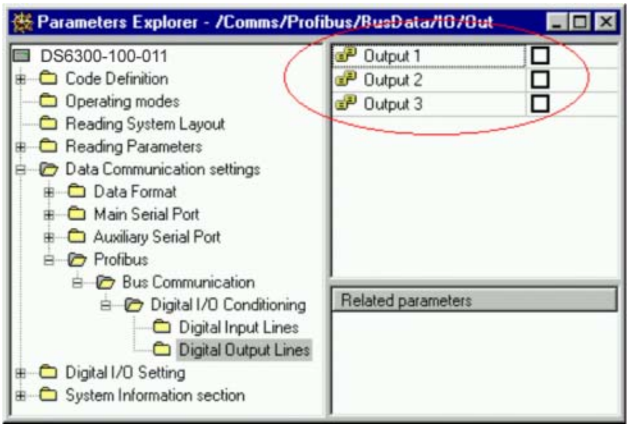
Parameters Explorer - /Comms/Profibus/BusData/IO/In

- DS6300-100-011
  - Code Definition
  - Operating modes
  - Reading System Layout
  - Reading Parameters
  - Data Communication settings
    - Data Format
    - Main Serial Port
    - Auxiliary Serial Port
    - Profibus
      - Bus Communication
        - Digital I/O Conditioning
          - Digital Input Lines
          - Digital Output Lines
  - Digital I/O Setting
  - System Information section

Input 1	□
Input 2	□
Input 3	□
Input 4	□

Related parameters

Figure 3.1 Digital Input lines



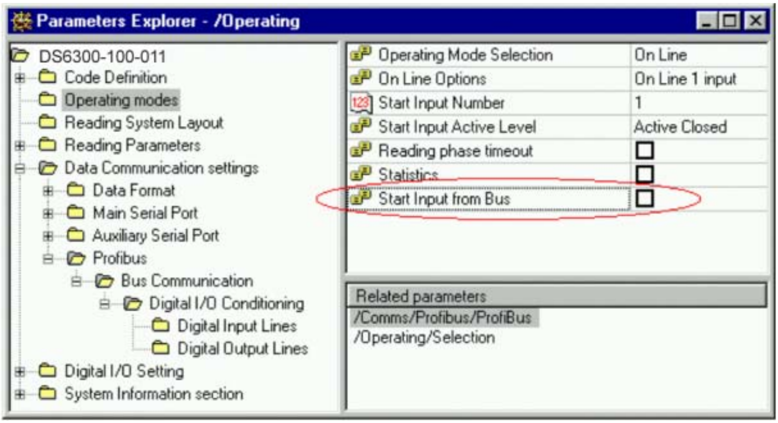
Parameters Explorer - /Comms/Profibus/BusData/IO/Out

- DS6300-100-011
  - Code Definition
  - Operating modes
  - Reading System Layout
  - Reading Parameters
  - Data Communication settings
    - Data Format
    - Main Serial Port
    - Auxiliary Serial Port
    - Profibus
      - Bus Communication
        - Digital I/O Conditioning
          - Digital Input Lines
          - Digital Output Lines
  - Digital I/O Setting
  - System Information section

Output 1	□
Output 2	□
Output 3	□

Related parameters

Figure 3.2 Digital Output lines



Parameters Explorer - /Operating

- DS6300-100-011
  - Code Definition
  - Operating modes
  - Reading System Layout
  - Reading Parameters
  - Data Communication settings
    - Data Format
    - Main Serial Port
    - Auxiliary Serial Port
    - Profibus
      - Bus Communication
        - Digital I/O Conditioning
          - Digital Input Lines
          - Digital Output Lines
  - Digital I/O Setting
  - System Information section

Operating Mode Selection	On Line
On Line Options	On Line 1 input
Start Input Number	1
Start Input Active Level	Active Closed
Reading phase timeout	□
Statistics	□
Start Input from Bus	□

Related parameters

/Comms/Profibus/ProfiBus
/Operating/Selection

Figure 3.3 Operating modes

VB3X digital I/Os are thus mapped over the first byte of both Input and Output exchange areas, as depicted in the next paragraphs.

3.1 Digital Input Conditioning

Note

The I/O Conditioning function works independently from the Flow Control options.

Without any Flow Control driver, the following Input mapping is available:

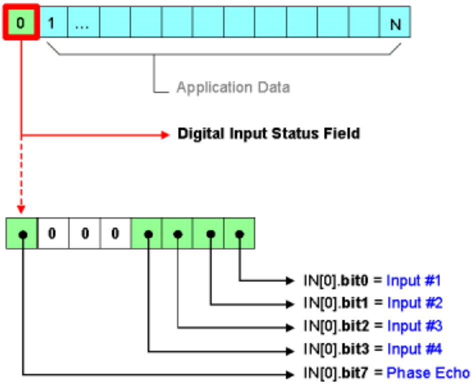


Figure 3.4 Exchange Input Area buffer

With Flow Control drivers, the Input area structure is as follows:

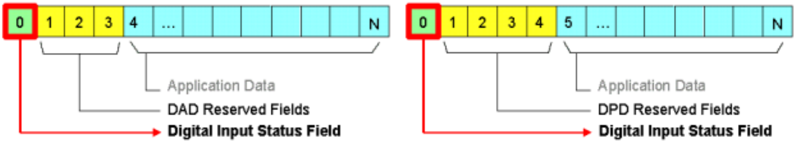
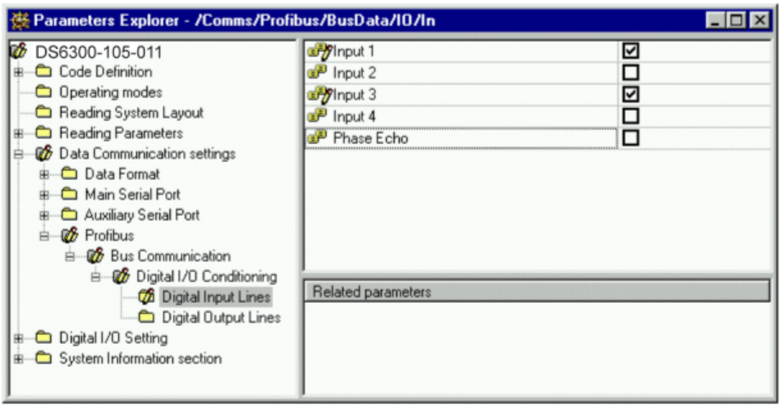


Figure 3.5 Digital Input Conditioning with DAD driver buffer and with DPD driver buffer

The Digital Input Status Field is continuously updated by the VB3X according to the status of the physical inputs of the scanner. Only the selected inputs are mirrored, like Input#1 and Input#3 in the figure below:



Parameters Explorer - /Comms/Profibus/BusData/IO/In

- DS6300-105-011
  - Code Definition
  - Operating modes
  - Reading System Layout
  - Reading Parameters
  - Data Communication settings
    - Data Format
    - Main Serial Port
    - Auxiliary Serial Port
    - Profibus
      - Bus Communication
        - Digital I/O Conditioning
          - Digital Input Lines
          - Digital Output Lines
  - Digital I/O Setting
  - System Information section

Input 1	☑
Input 2	□
Input 3	☑
Input 4	□
Phase Echo	□

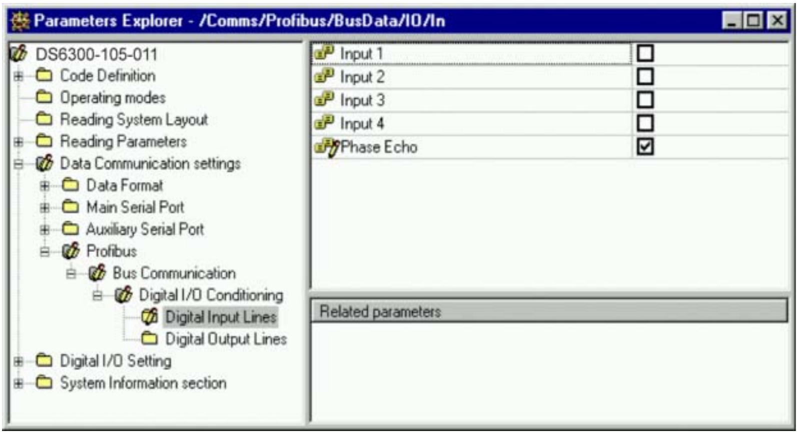
Related parameters

Figure 3.6 Digital Input lines

3.1.1 Phase Echo

The Phase Echo bit indicates the status of the reading phase remotely activated on the VB3X.

This function allows the Master to obtain feedback about the current reading activation via the Profibus network.



Parameters Explorer - /Comms/Profibus/BusData/IO/In

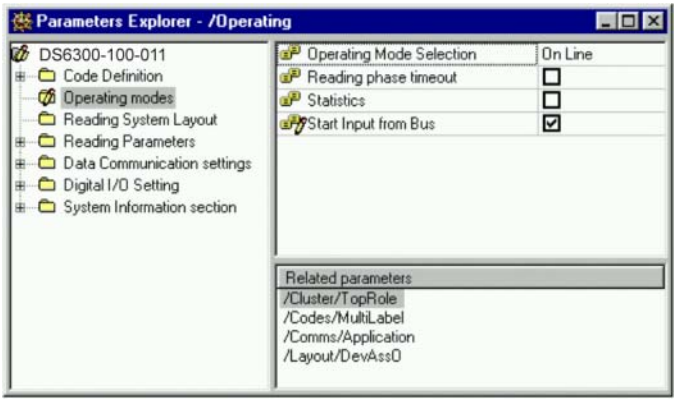
- DS6300-105-011
  - Code Definition
  - Operating modes
  - Reading System Layout
  - Reading Parameters
  - Data Communication settings
    - Data Format
    - Main Serial Port
    - Auxiliary Serial Port
    - Profibus
      - Bus Communication
        - Digital I/O Conditioning
          - Digital Input Lines
          - Digital Output Lines
  - Digital I/O Setting
  - System Information section

Input 1	□
Input 2	□
Input 3	□
Input 4	□
Phase Echo	☑

Related parameters

Figure 3.7 Phase Echo parameterizing

The Phase Echo parameter can be enabled once the Start Input From Bus parameter is selected. See details in the “Reading Phase via Profibus” paragraph.



Parameters Explorer - /Operating

- DS6300-100-011
  - Code Definition
  - Operating modes
  - Reading System Layout
  - Reading Parameters
  - Data Communication settings
  - Digital I/O Setting
  - System Information section

Operating Mode Selection	On Line
Reading phase timeout	□
Statistics	□
Start Input from Bus	☑

Related parameters

/Cluster/TopRole
/Codes/MultiLabel
/Comms/Application
/Layout/DevAss0

Figure 3.8 Start Input from Bus

3.2 Digital Output Conditioning

Note

The I/O Conditioning function works independently from the Flow Control options.

Without any Flow Control driver, the following Output mapping is available:

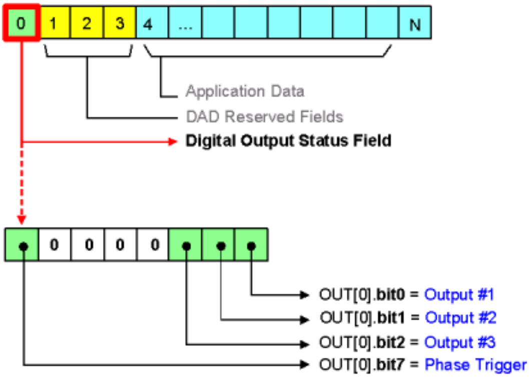


Figure 3.9 Exchange Output Area buffer

With Flow Control drivers, the Output area structure is as follows:

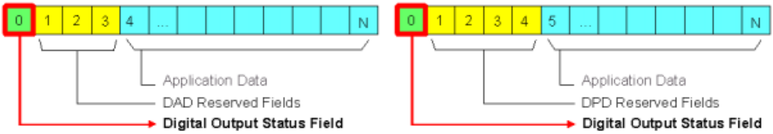


Figure 3.10 Digital Output Conditioning with DAD driver buffer and with DPD driver buffer

The Digital Output Status Field can be set by the PLC at any time, then the VB3X physical outputs will change accordingly. Only the selected outputs are thus controlled.

3.2.1 Phase Trigger

The Phase Trigger bit is used by the Master to force the reading phase of the scanner. Reading activation is provided through the Profibus network so there is no need of external photocells or I/0 signals.

Phase Trigger is active once the Start Input From Bus parameter is selected. See details in the “Reading Phase via Profibus” paragraph.

Note

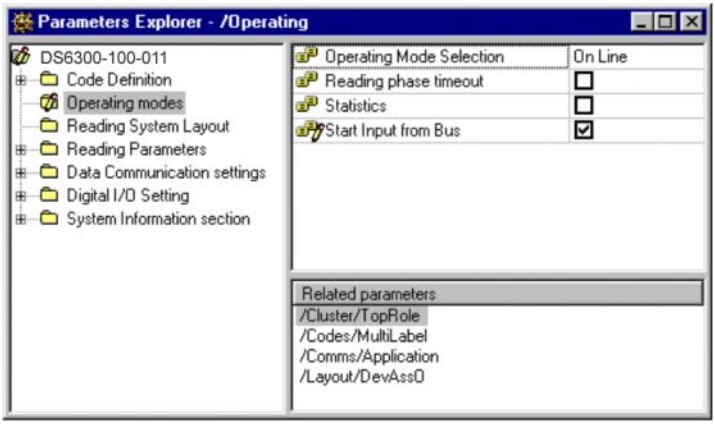
An alternative way to open and close each reading phase via Profibus is letting the VB3X operate in Serial On Line mode and then send the user programmable Serial Start String and Serial Stop String through the PLC.

3.2.2 Reading Phase via Profibus

This implementation requires few steps:

1. Connect the scanner with the configuration tool
2. Check the Start Input from Bus parameter in the Operating Modes folder so the scanner understands that the phase trigger is going to be provided via Profibus

Date of edition: 08/02/2004



Parameters Explorer - /Operating

- DS6300-100-011
  - Code Definition
  - Operating modes
  - Reading System Layout
  - Reading Parameters
  - Data Communication settings
  - Digital I/O Setting
  - System Information section

Operating Mode Selection	On Line
Reading phase timeout	☐
Statistics	☐
Start Input from Bus	☑

Related parameters
/Cluster/TopRole
/Codes/MultiLabel
/Comms/Application
/Layout/DevAss0

Figure 3.11 Start Input from Bus

3. Send the modified configuration to the VB3X permanent memory (EEPROM)

Now the reading phase of the scanner is controlled as follows:

- Reading phase starts as soon as the Master sets OUT[0].bit7
- Reading phase stops as soon as the Master clears OUT[0].bit7

4 Network Configuration

4.1 GSD file

A GSD file is a readable ASCII text file that contains a complete description of the specific device. Basically GSD includes both general info (i.e. vendor and device name, hw/sw releases) and device specific info (Input and Output area size, communication parameters, scanner setup parameters and so on).

Powerful configuration tools (i.e. Siemens SIMATIC Manager) are available to setup a Profibus network. Based on the GSD files, these allow easy configuration of Profibus networks with devices from different manufacturers.

Note

Firstly a GSD file must be installed into the PLC environment in order to let a new device be identified and to work on the Profibus network. Follow the instructions in the “GSD Installation” paragraph.

VB3X is equipped with the following files:

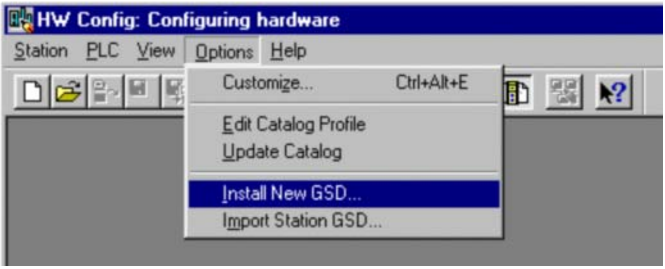
• PF_07F0.GSD	Device description file
• PFVB34.DIB	
• PFVB34DI.DIB	Custom identification icons
• PFVB34SF.DIB	

The GSD file is a certified part of the device and must not be changed manually. This file is also not changed by the configuration tool.

4.2 GSD Installation

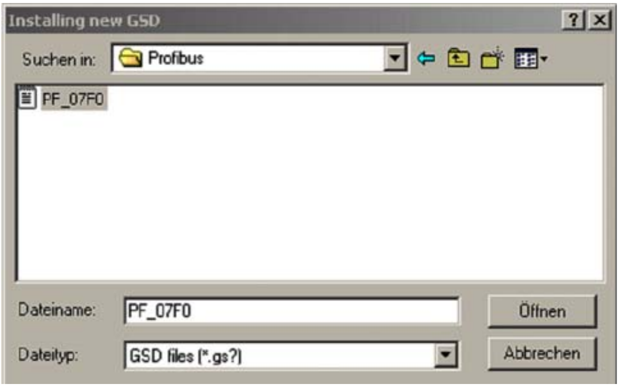
The first action that must be done is to add the VB3X as a new Profibus-DP Slave among the catalogue of suitable devices for the PLC.

1. Install the new GSD file...



2. Find the GSD file…

(GSD and *.DIB files must be in the same directory)

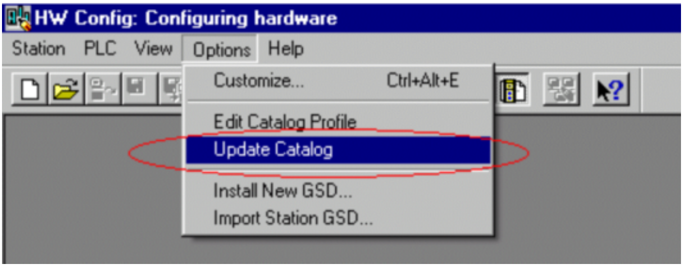


Suchen in: Profibus

Dateiname: PF_07F0

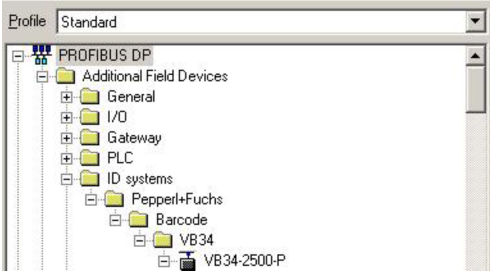
Dateityp: GSD files (*.gs?)

3. Update Catalogue…



4. Find new device…

A new VB3X device will appear in the PLC catalogue under Profibus-DP ➜ Additional Field Devices ➜ ID Systems ➜ Pepperl+Fuchs ➜ Barcode ➜ VB3X folder

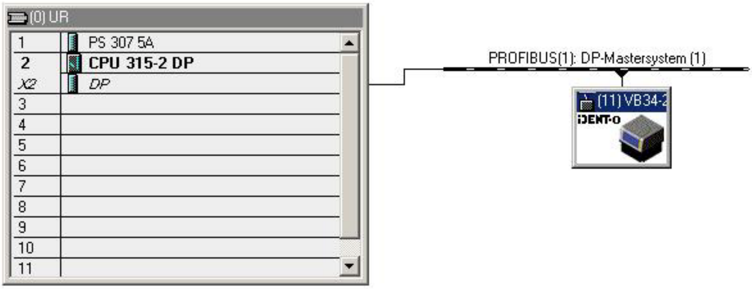


Profile: Standard

- PROFIBUS DP
  - Additional Field Devices
    - General
    - I/O
    - Gateway
    - PLC
    - ID systems
      - Pepperl+Fuchs
        - Barcode
          - VB34
            - VB34-2500-P

Insert the device in the Profibus network…

The easy drag&drop function allows inserting the VB3X device in your own network



(0) UR

1	PS 307 5A
2	**CPU 315-2 DP**
X2	*DP*
3	
4	
5	
6	
7	
8	
9	
10	
11	

Figure 4.1 Inserting the VB3X device in your own network

4.3 Scanner Programming via GSD File

The major benefit of the GSD file is to setup the scanner by means of the Profibus Master without the using the Configuration Tool: it is possible to select the barcodes, the scanner operating mode and many other relevant parameters.

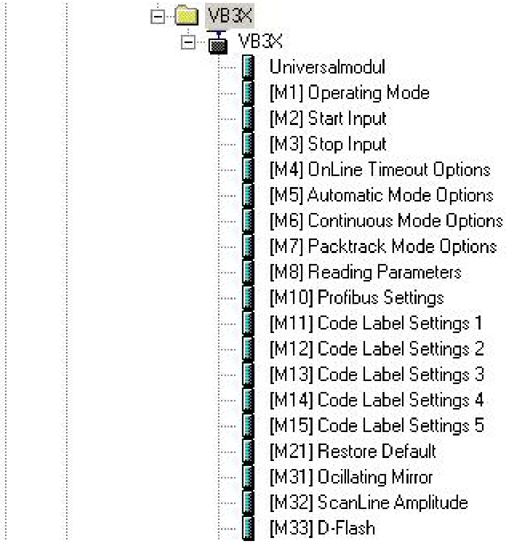
Different scanner parameters are grouped in Project Modules that can be used to create custom configurations. The User should include the desired modules in his own PLC project and select the proper values for each parameter. Once completed, the PLC will automatically program the scanner at each power on and also any time it is re-discovered on the network (reconnected).

Since the scanner configuration is stored in the PLC, scanner replacement operations become extremely easy and quick.

4.3.1 Project Modules

Three types of modules are available:

- [IOMCxx] modules ➜ I/O module with Data Consistency over the whole string
- [IOMxx] modules ➜ I/O module with Data Consistency over bytes/words
- [Mxx] modules ➜ Scanner Parameters module



VB3X
VB3X
Universalmodul
[M1] Operating Mode
[M2] Start Input
[M3] Stop Input
[M4] OnLine Timeout Options
[M5] Automatic Mode Options
[M6] Continuous Mode Options
[M7] Packtrack Mode Options
[M8] Reading Parameters
[M10] Profibus Settings
[M11] Code Label Settings 1
[M12] Code Label Settings 2
[M13] Code Label Settings 3
[M14] Code Label Settings 4
[M15] Code Label Settings 5
[M21] Restore Default
[M31] Ocillating Mirror
[M32] ScanLine Amplitude
[M33] D-Flash

Figure 4.2 Modules overview

At least one I/O module is required to set the dimensions of both the Input and Output exchange areas. If needed, one or more [**IOMCxx**] (or [**IOMxx**]) modules can be combined.

Scanner parameterisation is optional. In this case, the [M1] Operating Mode module must always be present.

It is suggested to use the [M21] Restore Default module first, to allow a congruent configuration. All of the available modules, except the I/O modules, are generally executed in the same order in which they are inserted.

Let’s see the details of each module:

[M1] Operating Mode module

Parameter	Options
Operating Mode	"On Line 1In" "On Line 2In" "On Line 1In/Timeout" "On Line ProfiBus" "On Line ProfiBus/Timeout" "Serial On Line" "Serial On Line/Timeout" "Continuous" "Test""Packtrack" "Automatic"

[M2] Start Input module
[M3] Stop Input module

Parameter	Options
Presence Sensor	"Input 1" "Input 2" "Input 3" "Input 4"
PS Active Level	"Active Closed" "Active Open"

[M4] OnLine Timeout Options module

Parameter	Options
Timeout	"40(ms)" "50(ms)" "70(ms)" "100(ms)" "150(ms)" "200(ms)" "500(ms)" "1000(ms)" "2000(ms)" "3000(ms)" "4000(ms)" "5000(ms)" "7000(ms)" "10000(ms)" "12000(ms)" "15000(ms)"

Date of edition: 08/02/2004

[M5] Automatic Mode Options module
[M6] Continuous Mode Options module

Parameter	Options
Code Filter Depth	"0" ... "10" (in steps of 1) "15" ... "50" (in steps of 5)
Threshold	"10 scans" "20 scans" "50 scans" "80 scans" "100 scans" "150 scans" "200 scans" "250 scans" "300 scans" "350 scans" "400 scans" "500 scans" "750 scans" "1000 scans" "5000 scans" "10000 scans" "15000 scans" "20000 scans" "30000 scans"

[M7] Packtrack Mode Options module

Parameter	Options
Physical Encoder	"Enable" "Disable"
Tx Edge	"Trailing" "Leading"
Max Number Of Packs	"2" … "15"

[M8] Reading Parameters module

Parameter	Options
Reading Mode	"Reconstruction" "Linear"
Overflow Ratio	"0" … "50"
Beam Shutter	"Disable" "Triggered" "Enable"

[M10] Profibus Settings module

Parameter	Options
Flow Control	"Disable" "DAD Driver" "DPD Driver"
Data Consistency	"Enable" "Disable"
Input 1	"Enable" "Disable"
Input 2	"Enable" "Disable"
Input 3	"Enable" "Disable"
Input 4	"Enable" "Disable"
Phase Echo	"Enable" "Disable"
Output 1	"Enable" "Disable"
Output 2	"Enable" "Disable"
Output 3	"Enable" "Disable"

[M11] Code Label Setting 1 … [M15] Code Label Setting 5 modules

Parameter	Options
Code Type	"Code 128" "Interleaved 2 of 5" "Code 39" "Code EAN 128" "EAN-13" "EAN-8" "UPC-A" "UPC-E" "All EAN_UPC" "CODABAR" "Code 93"
Label Length	"Variable" "1" … "48"

[M21] Restore Default module

Parameter	Options
<No Parameters>	<No options>

[M31] Oscillating Mirror Settings module

Parameter	Options
OM Mode	Continuous Fixed
OM Angle	"-2.5" ... "37.5" (in steps of 0.5)
OM Frequency	"0.0" ... "6.0" (in steps of 0.1) "6.5" ... "19.5" (in steps of 0.5)

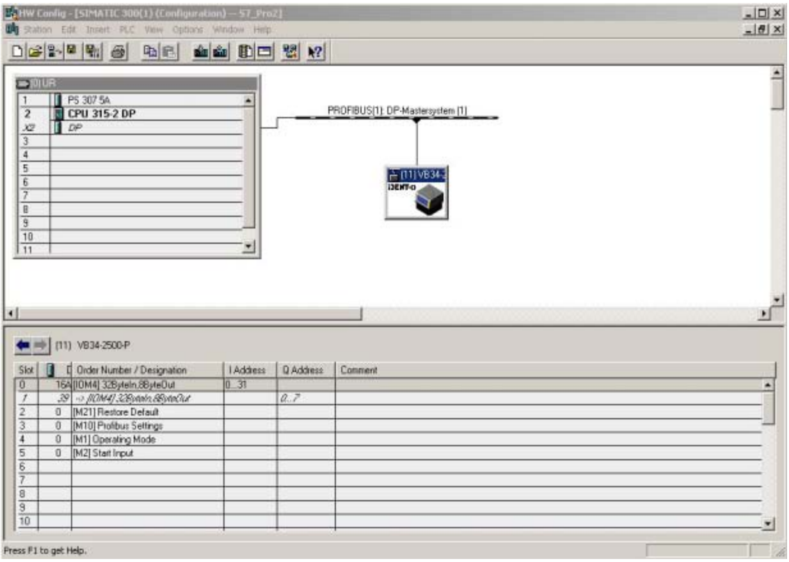
[M32] ScanLine Amplitude Settings module

Parameter	Options
Scan Line Position	"0" ... "255" (in steps of 5)

[M33] D-Flash Settings module

Parameter	Options
Offset Distance	"-255" ... "255" (in steps of 5)
Behaviour	"Nearest" "Background Suppression"
StandBy Distance	"1" ... "8" (in steps of 1)
Distance 1 up to Distance 8	"45" ... "190" (in steps of 5)

A typical scanner programming project is as follows:


Station Edit Insert PLC View Options Window Help
PROFIBUS(1): DP-Mastersystem (1)
(11) VB34-2
IDENT-I

HW Config - [SIMATIC 300(1) (Configuration) -- S7_Pro2]

(0) UR

1	PS 307 5A
2	CPU 315-2 DP
X2	DP
3	
4	
5	
6	
7	
8	
9	
10	
11	

(11) VB34-2500-P

Slot	D	Order Number / Designation	I Address	Q Address	Comment
0	16A	[IOM4] 32ByteIn,8ByteOut	0...31		
1	39	-> [IOM4] 32ByteIn,8ByteOut		0...7	
2	0	[M21] Restore Default			
3	0	[M10] Profibus Settings			
4	0	[M1] Operating Mode			
5	0	[M2] Start Input			
6					
7					
8					
9					
10					

Press F1 to get Help.

Here you have:

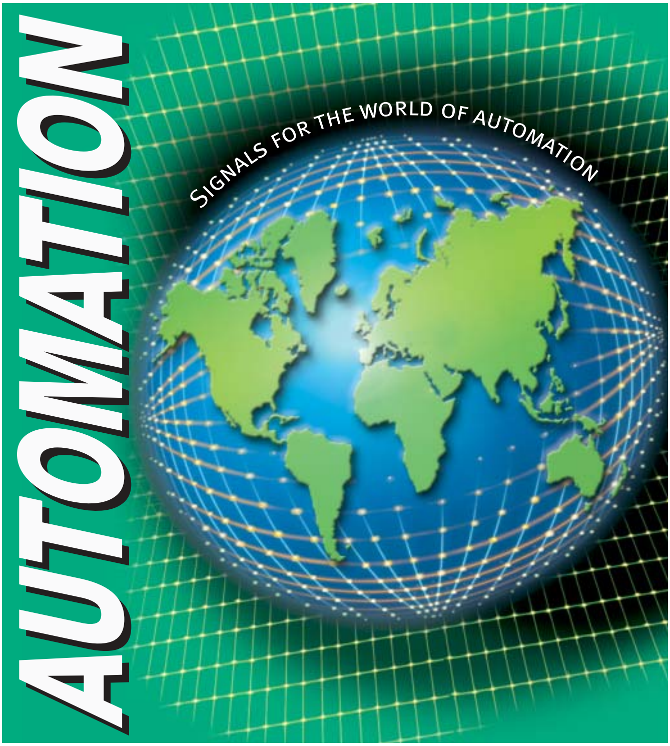
- **[IOMC4] module** to set the Input/Output exchange areas.
- **[M21] module** to restore the scanner configuration to Default.
- **[M10] module** to define possible Flow Control protocol and/or Data Consistency options.
- **[M1] module** to select the scanner operating mode, i.e. OnLine mode.
- **[M2] module** to indicate which digital Input provides the trigger.

See also details of the demo project in the following chapter.

Complete parameter descriptions are available in the Configuration Tool On Line Help.

Notes

Subject to reasonable modifications due to technical advances.



www.pepperl-fuchs.com

Worldwide Headquarters
Pepperl+Fuchs GmbH · Königsberger Allee 87
68307 Mannheim · Germany
Tel. +49 621 776-0 · Fax +49 621 776-1000
e-mail: info@de.pepperl-fuchs.com

USA Headquarters
Pepperl+Fuchs Inc. · 1600 Enterprise Parkway
Twinsburg, Ohio 44087 · Cleveland-USA
Tel. +1 330 4253555 · Fax +1 330 4 25 93 85
e-mail: sales@us.pepperl-fuchs.com

Asia Pacific Headquarters
Pepperl+Fuchs Pte Ltd. · P+F Building
18 Ayer Rajah Crescent · Singapore 139942
Tel. +65 67799091 · Fax +65 68731637
e-mail: sales@sg.pepperl-fuchs.com

PEPPERL+FUCHS
SIGNALS FOR THE WORLD OF AUTOMATION