

# Self-Service Systems

## Cash Media Dispenser

Version 4

### Do you have ...

... any questions or suggestions regarding this manual?

Please contact the following address giving the order number of this manual in your message:

Address:  
WINCOR NIXDORF International GmbH  
Handbuchredaktion BD PSD 43  
33094 Paderborn  
Germany

Fax: +49 (0) 52 51 693-62 09

E-mail: [manuals.hardware@wincor-nixdorf.com](mailto:manuals.hardware@wincor-nixdorf.com)

... any technical questions or problems?

If you have any technical questions or problems please contact your local representative.

# **Cash Media Dispenser**

Version 4

**Software Manual**

**Edition July 2010**

© WINCOR NIXDORF International GmbH 2010

All rights, including rights of translation and rights of reproduction by reprinting, copying or similar methods, even of parts, are reserved.

Any violations give rise to a claim for damages.

All rights, including rights created by patent grants or registration of a utility model or design, are reserved. Delivery subject to availability; right of technical modifications reserved.

All names of hardware and software products mentioned in this manual are trade names and/or trademarks of their respective manufacturers.

This manual was  
produced using paper  
treated with chlorine-free  
bleach.

---

# Contents

<b>CMD-V4 Cash Media Dispenser Version 4</b>	<b>1</b>
References	1
Abbreviations	2
Overview	3
Purpose and scope of this document	3
Introduction	3
Device type	3
Error tolerance	3
Control	3
Firmware	3
Serviceability	4
Compatibility	4
Device description	5
Cassette rack	5
Dispensing unit	5
Stacker	6
Handling “bad” notes	6
Vertical Cash Media Dispenser	7
Output transport	7
Cash output	7
Cash trapping detection	8
Evaluation during cash presentation	9
Reset of a cash trapping detection error	10
Cassettes	11
Versions	11
Compatibility to other WN cassettes	12
Monitoring the available cassettes	13
Cassette change	14
Controller	15
External messages	15
Status display	15
Function button	16
Firmware	17
Firmware modules	17
Bootstrap loader	17
Controller	17
USB – CMD-V4 as USB device	18

Introduction .....	18
Firmware behavior .....	18
Implementation overview .....	19
Device descriptor .....	20
Configuration descriptor .....	21
Interface descriptor .....	21
Endpoint descriptor (endpoint 2 job channel Device > PC) .....	22
Endpoint descriptor (endpoint 1 job channel PC > Device ) .....	23
Endpoint descriptor (endpoint 3 Debug channel Device > PC) .....	23
String descriptor (country code) .....	23
String descriptor (manufacturer string) .....	23
String descriptor (product string) .....	24
Configuration .....	24
Jumper .....	24
Sensor .....	25
Firmware configuration job .....	25
Recover status .....	25
Dual dispense evaluation by the measurement station .....	25
Automatic baud rate switch: 19200 / 56000 baud .....	26
Encryption .....	26
Encryption procedure, general .....	26
DES .....	26
ECB .....	27
CBC .....	28
Triple-DES .....	29
Encryption CMD-V4 .....	31
Commands .....	33
General .....	34
Plausibility check .....	34
Device lock .....	34
DA – Reset .....	35
Command .....	35
Acknowledgement .....	35
Description .....	35
Status Codes .....	39
DAB – Clearing .....	40
Command .....	40
Acknowledgement .....	40
Description .....	41
Status Codes .....	43
Dispensing .....	44
Command .....	44
Acknowledgement .....	45
Description .....	46
Special handling .....	50
Determine reference value .....	54

Command.....	54
Acknowledgement.....	54
Description .....	55
Status Codes.....	56
Shutter commands .....	57
Command.....	57
Acknowledgement.....	57
Description .....	57
Status codes .....	58
DEK – read FW configuration .....	59
Command.....	59
Acknowledgement.....	59
Description .....	59
Status Codes.....	59
DFG – define FW configuration.....	61
Command.....	61
Acknowledgement.....	61
Description .....	61
Status Codes.....	62
Encryption .....	63
Key handling.....	63
Request initial value .....	65
Key test .....	66
DQC – read error stack .....	68
Command.....	68
Acknowledgement.....	68
Description .....	68
Status Codes.....	69
DQE – read error information.....	70
Command.....	70
Acknowledgement.....	70
Description .....	70
Status Codes.....	70
Structure.....	71
DQF – read stacker-EEPROM .....	74
Command.....	74
Acknowledgement.....	74
Description .....	74
Status Codes.....	74
DQI - Describe stacker-EEPROM .....	75
Command.....	75
Acknowledgement.....	75
Description .....	75
Data structure in the stacker-EEPROM .....	75
Status Codes.....	77
DQJ - read manufacturer's and controller identification.....	78

Command.....	78
Acknowledgement.....	78
Description .....	78
Structure of the manufacturer's ID .....	79
Status Codes .....	79
GetAccessCode .....	80
Command.....	80
Acknowledgement.....	80
Description .....	80
Status codes.....	80
SetAccessCode .....	81
Command.....	81
Acknowledgement.....	81
Description .....	81
Status codes.....	81
DYR – read internal statistics .....	83
Command.....	83
Acknowledgement.....	83
Structure of the statistics data .....	84
Description .....	85
Status Codes .....	86
DYS – Resetting internal statistics .....	87
Command.....	87
Acknowledgement.....	87
Description .....	87
Status Codes.....	87
Appendix A: Cross reference .....	88
Command/status .....	88
Booter commands .....	92
Execution times .....	92
Appendix B: Status display codes .....	94
Displays in normal operation.....	94
Contaminated photosensors/sensors.....	98
Photosensors covered during initialization.....	98
Displays when function button is pressed .....	99
Booter outputs to the 7-segment display.....	99
Appendix D: Electrical and mechanical components .....	101
Electrical elements .....	101
Stacker and output transport.....	101
Components of dispensing unit and cassette .....	103
Position of the elements in the CMD_V4 .....	104
Position of elements in VCMD.....	105
Appendix E: Jumper settings on controller.....	106
Jumper field.....	106
Default settings.....	106
Jumper positions .....	107



Appendix F: Transport positions.....	108
Outdoor .....	108
Indoor .....	109
Appendix G: Error Number Table.....	111
Appendix V: DQC – Error stack.....	151
Appendix X: Status areas.....	155
D – Device status .....	155
C – Cassette status .....	165
F – Features .....	170
K – Firmware configuration .....	173
Definitions.....	176
Configuration after CMOS initialization .....	177
Device status.....	177
Features .....	177
FW configuration .....	177
Appendix Y: Special features .....	178
Different behavior of the clamp for DCE and DCP.....	178
DCW and the locking handle.....	178
Detecting shutter manipulation after DCW.....	179



---

# CMD-V4 Cash Media Dispenser Version 4

## References

- [L1] AZM\_NG Software Manual  
PFHT 205  
Edition 2.7 of 2005-06-09
- [L3] Cash dispenser module – ProCash 1000; Software Manual  
Doc. no.: 0170019430  
Edition: 1.3 of 2001-08-23
- [L4] AZM\_NG SW description IBM emulation  
Doc. no.: \_\_\_\_\_  
Edition: 1.0 from 29.01.1998
- [L5] WN Development Guide 01/00 – Project execution  
Doc. no.: 803a  
Edition: 1 of 2000-10-06
- [L6] Wincor Nixdorf Device Descriptor Definition  
Doc. no.: 0170022314  
Edition: 2.2 of 2000-09-18
- [L7] Error stack for self-service components  
Doc. no.: 0170022507  
Edition: 1.1 of 2000-08-28
- [L8] ProCash CRS (Cash Recycling System); Software Manual  
Doc. no.: BSCH 730  
Edition: 3.1 of 2001-07-06
- [L9] CMD-V4; Power Up Test and Self-test  
Doc. no.: \_\_\_\_\_  
Edition: \_\_ of \_\_.\_\_.2002
- [L10] CMD-V4; Analysis Tool – Device Information; Trace; Error Log  
Doc. no.: \_\_\_\_\_  
Edition: \_\_ of \_\_.\_\_.2002
- [L11] Schneider, Bruce:  
Angewandte Kryptographie (applied cryptography)  
Bonn: Addison-Wesley, 1996

## Abbreviations

!=	Not Equal To
\$MOD\$	WN module identifier (SW)
AZM	Dispensing module
AZM_NG	Dispensing module (new generation)
BCS	Block Check Sum
CBC	Cipher Check Sum
CMD-V4	Cash Media Dispenser - Version 4
CRS	Cash Recycling System
DEA	Data Encryption Algorithm
DES	Data Encryption Standard
DK	Printing cassette
ECB	Electronic Code Book
ECM	External counter management
EDA	Express Delivery Application (OS/2 – IBM application)
EDM	Encryption Decryption Mode
FW	Firmware
HOST	Higher-level system
ISO	International Organization for Standardization
IV	Initial Value
KCV	Key Check Value
KEK	Master Key (KEK)
KGK	Key Generating Key
NA	Power failure
OEM	Original Equipment Manufacturer
PLD	Programmable Logic Device
RR box	Reject-/Retract box
SCOP	Secure Cash Out Procedure
SE	Special electronics module
TAN	Transaction number
TDES/ 3DES	Triple-DES
UDS	User Defined Status
ÜMA	Universal Serial Bus
VCMD	Vertical Cash Media Dispenser
EDM	Encryption / decryption module

# Overview

## Purpose and scope of this document

This document provides detailed information about the characteristics and the command set of the of the **CMD-V4**.



This document only describes the characteristics and the commands that are not included in [L 1], [L 2], [L 3], [L 4] or that have been modified.

## Introduction

The main characteristics of the CMD-V4 are:

### Device type

The different device types (indoor/outdoor, rearload/frontload) and the variable number of dispenser modules (1-6) enable each user to choose the device type that best suits his individual needs.

### Error tolerance

Within the CMD-V4, each note is checked after dispensing. Any errors detected, such as double dispenses, are corrected internally.

### Control

The physical interface complies with the RS232C standard. In a deviation from this standard, the system is connected via a 9-pin D-sub plug.

The protocol (data link layer of the basic reference model according to ISO 7498) is based on DIN 66348 part1 control procedure C.

### Firmware

The controller of the CMD-V4 features a loader that can be used to load the correct firmware for the device variant in question.

When the devices are delivered, the current FW is already loaded. If so required, updates are provided in the WN intranet.

## Serviceability

### ERROR STACK

An error stack is managed in a separate memory area on the controller board. There are also suitable commands available for concerted error stack management on the HOST.

### FIRMWARE ID

The FW version number can be read on the controller status display without the need for additional tools.

## Compatibility

Compatibility analysis takes account of both the application and the configuration.

### APPLICATION

In the CMD-V4 standard mode (default setting), no adjustments need be made to existing applications.

### CONFIGURATION

The CMD-V4 configuration is, as with previous versions, set for a specific customer during ATM manufacture.

However, the way in which it is configured differs from the previous procedure. The size of the jumper field has been reduced.

The first jumper (DOOR) is now used for security relevant settings only. It is made with the safe door open.

Connector for (number of dispensing units, shutter available (YES / NO), output direction and printing cassette) replace the remaining jumpers.

A default controls how the CMD-V4 is to behave.

## Device description

This chapter outlines the individual hardware components of the CMD-V4.

More detailed information is available in related documents.

### Cassette rack

A combination of 4-cassette and 1-cassette racks (max. 6) is used to build any configuration required.

Number of racks required for a      5-cassette tower: 1 x 4-cassette + 1 x 1-cassette;  
3-cassette tower: 3 x 1-cassette.



A single 4-cassette rack with corresponding 4-way distributor board but only using 2 dispensing units (positions 1+ 2) is operable if B is jumpered.

### Dispensing unit

The cassette racks accommodate 1-cassette and 2-cassette dispensing units.

Number of dispensing units required for a      5-cassette tower: 2 x 2-cassette + 1 x 1-cassette;  
3-cassette tower: 1 x 2-cassette + 1 x 1-cassette.

The upper dispensing unit houses the measurement station. This checks whether double or overlapping notes have been dispensed.

## Stacker

Notes that are dispensed are transported via the stacker incl. measurement station. All 'good' notes are made available as a bundle in an open clamp via the stacker wheel.

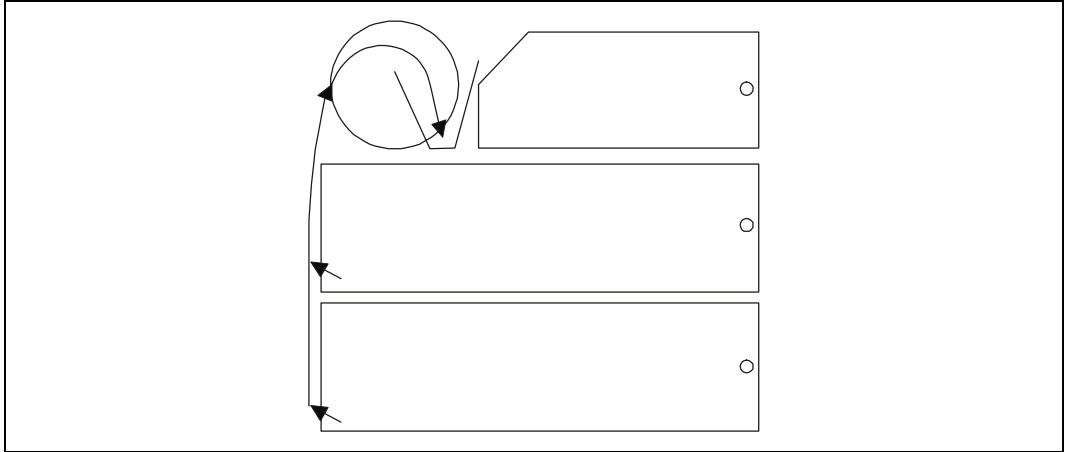


Diagram of the dispense transaction

When machines are installed outside Euroland, an optional single reject deflector can be integrated in the stacker.

## Handling “bad” notes

### **SINGLE REJECT DEFLECTOR AVAILABLE**

Up to two 'bad' notes are initially parked in a separate tray. A replacement note is then dispensed.

After cash-out, the singled out notes are transported to the RR box.

Although the parked notes are not yet in the RR box, the reject counter (RACT) is incremented for each parked note.

### **SINGLE REJECT DEFLECTOR MISSING**

Note dispensing is interrupted until all notes have reached the clamp. A note bundle reject then takes place.

The original dispense job command is repeated.



## Vertical Cash Media Dispenser

This device version has a modified stacker and output transport and no shutter. The single reject deflector is always available. As soon as a “bad” note is detected, it is singled out via the single reject deflector. All “good” notes are made available as a bundle in the stacking compartment via the stacker wheel and can be transported towards the cash output via the output transport.



There is no longer a separate retract tray in the VCMD.

## Output transport

The note bundle is held in the closed clamp. It is transported to its destination via gear tracks. In the VCMD, transport occurs via belt transport. The version is shown in the diagram "Component positions".

## Cash output

The cash is first forwarded to a waiting position. The shutter – if installed – at the end of the transport route is opened. The bundle is then transported to the output position. The note bundle is presented via integrated belt transport. Cash output is complete when the notes have been removed.

Notes which are not removed can be withdrawn and stored unsorted on the retract tray of the RR-box. The number of retracts is counted.

Optionally, a bundle can be retracted and stored securely in the stacking position. If this option is used, however, no further device operations are possible until the bundle is removed by an operator and the software has canceled the lock.

## Cash trapping detection

Depending on the device version, the systems can be equipped with cash trapping detection. This is meant to detect cash trapping devices that are directly attached to the original shutter with a short distance.

Specific shutter electronics are required for cash trapping detection. Data evaluation gives information about a possibly detected attachment or a manipulation. Error handling is expanded/modified.

The behavior of the cash trapping detection can be configured (see firmware configuration: SHUT\_PROTECT\_OPT). Like all firmware configuration options, the configuration becomes operative after the next RESET/CLEARING due to the non-volatile storage. Existence of a cash trapping detection is indicated in the device status <SHUT\_PROTECT>. If no cash trapping detection is available, the device works like a customary CMD-V4.

During initialization of the CMD main firmware (after Power on, RESET, or CLEARING), the function and initialization values are checked when a cash trapping detection is available and working <SHUT\_PROTECT\_OPT>=Y. If the result is negative, CLEARING, RESET or the first status request after Power on are acknowledged with status D ("Out of service") and in device status <SHUT\_PROTECT>=D is set.

Before every dispense transaction, the sensory mechanism for cash trapping detection is checked when one is present and in operation <SHUT\_PROTECT\_OPT>=Y and the shutter is closed. The sensors might have been manipulated during the last transaction. If the result is negative, the command is acknowledged with status E ("Device error") and in device status <SHUT\_PROTECT>=M is set.

For a reset of the error see section "Reset of a cash trapping detection error" (page 10).

## Evaluation during cash presentation

The command DCP (cash presentation) effects that the bundle is transported behind the shutter as before and that the shutter is opened.

Before note output, evaluation for cash trapping detection is started when the detection is activated. If no cash trapping device is detected, the bundle is presented as it has been before. Otherwise the shutter is closed, reopened and another detection process is started.

When the result remains negative after 2 repeats, then:

- the shutter is closed.
- the bundle is transported into the stacker.
- <MON>=P is set.
- the command is acknowledged with status E ("Device error").

The bundle can be transported into the retract cassette by the retract request command. For a reset of the error see section "Reset of a cash trapping detection error" (page 10).

### Status codes

#### E = Device error

Cash trapping device detected	<SHUT_PROTECT>	= K
	SCOD	= 28
	SHERR	= B
	SHUT	= U

Until the next transport command (DC..., e.g. Retract, Open shutter, Close shutter, ...) RESET, CLEARING or Power on remains in device status <SCOD>=28, <SHUT>=U, <SHERR>=B. If the device works without any errors and a cash trapping detection error has not taken place yet, these commands are acknowledged with status A and in device status <SCOD>=00, <SHUT>=C, <SHERR>=N are set, <SHUT\_PROTECT> remains unchanged.

## Reset of a cash trapping detection error

After fault elimination or removal of the cash trapping device, a cash trapping detection error can be reset as follows:

- by DCT (shutter test) command. It is checked whether a cash trapping detection error is still present. If so, the transaction is terminated with the status E (Device error), <SCOD>=28, <SHUT>=U, <SHERR>=B are set and <SHUT\_PROTECT> remains unchanged.
- previous to the next dispense transaction, the system checks whether a cash trapping detection error is still present. If so, the transaction is terminated with the status E (Device error), <SCOD>=00, <SHUT>=C, <SHERR>=N are set and <SHUT\_PROTECT> remains unchanged.

## Cassettes

In addition to the dispenser cassettes, a combined RR box is used in the CMD-V4. Both types also function in AZM\_NG and similar versions.

## Versions

The following versions are available.

	Lever	Lock	Lead seal	Tamper indicator	Ink dye (integrated degradation system)
<b>Standard</b>	X		X	X	
<b>Security I</b>		X	X	X	
<b>Security II</b>		X	X	X	X



The status of the indicator is displayed in a window located at the upper left on the front of the cassette.

The display is normally **green**.

A **blue** display indicates that the cassette has either been inserted in a voltageless dispenser or has been manipulated or opened by force.

In this case, you should check the cassette contents before inserting the cassette.

When you open the cassette, the indicator is reset.

### Dispenser cassette

The cassette contains an “INSERT” which can be adapted to the required banknote size without using any tools. The pressure is provided by a motor-operated pressure carriage. During operation, a “physically empty” cassette is detected.

Each dispenser cassette contains a non-volatile memory which stores the cassette number as well as cassette-specific parameters (e.g. currency, value and release date). The cassette number is preset in the factory. It can be changed by the user if required.

### RR BOX

The combined RR box has a capacity of approximately 400 notes in the reject area and approximately 100 notes in the retract area.



There is no longer a separate retract area in the VCMD.

## PRECONDITIONS

The operator must ensure that the CMD-V4 has the correct data and fill volumes before customer mode is activated!.



Dispensing is not prevented if:

- the RR box is full,
- a logical dispenser cassette is now empty.

## Compatibility to other WN cassettes

Two cases must be considered: Forward and backward compatibility.

### FORWARD COMPATIBILITY

There are no restrictions to the use of cassettes from the AZM\_NG family in the CMD-V4.

In detail these are cassettes with:

- No indicator

Cassettes of this type are not equipped with an indicator.

- Tamper indicator (green/blue)

The indicator has the same meaning as on CMD-V4 cassettes.



The indicator does not change to blue when a cassette that was inserted in a voltageless dispenser is removed.

- Re-insertion lock (green/red)

These cassettes must always be opened before they are re-inserted.

This hampers troubleshooting if it means that an additional person is needed to open the cassettes.

### BACKWARD COMPATIBILITY

In principle, the CMD-V4 cassettes can also be used in the AZM\_NG family.



However, the indicator changes to blue each time a cassette is removed.

The AZM\_NG family does not have the technology needed to operate the optimized tamper indicator on the CMD-V4 cassettes.

## Monitoring the available cassettes

All cassette positions are checked cyclically for changes. As soon as the CMD-V4 detects a change, this is noted (generally in the device status and, for individual cassettes, in the cassette status or in the CMD-V4 capabilities).

### Device status

This documents whether the cassette inventory has changed since the last synchronization command 'Confirm cassettes installed'.

<CEX>=N     No change  
<CEX>=Y     Cassettes available changed

### Cassette status

<nSTA> displays the statuses of a cassette:

<nSTA>= D    The newly inserted cassette is faulty (e.g. stepper motor, EE-PROM or dispensing sensor)  
<nSTA>= E    Cassette is physically empty.  
<nSTA>= M    Cassette removed or incorrectly inserted.  
<nSTA>= P    Cassette is not ready. The banknote-specific features of the cassette must be declared to the measurement station via reference value calculation.  
<nSTA>= R    The cassette is ready. It is filled and the banknote-specific features have been declared to the measurement station. If it has been removed since, the contents may have changed.

## CMD-V4 CHARACTERISTICS

<C\_TYPE> displays the cassette type for each dispenser module.

## Cassette change

After a cassette change, the host and the CMD-V4 must first be synchronized. If this does not take place, dispensing jobs for certain cassette positions may be rejected with **context error**.

Example:

- Synchronization in the HOST, e. g. by service ID card. HOST switches from customer operation to the SERVICE status.
- Execution of cassette change.
- During the installation procedure, each cassette must be pushed into its position until it engages in its working position. The required pressure is then created. The operator is informed of the result via an acoustic signal:

No. of tones	Meaning
1	The correct pressure was created.
3	Action not successful; remove cassette and re-position it.

- Synchronization with the host.  
Check whether cassettes have been changed. If they have, check the new cassette set. If necessary, update the banknote parameters and/or cassette contents in the CMD-V4 and reconcile them with the HOST.

Restart cassette monitoring in the CMD-V4 (confirm cassette inventory).

HOST reverts to customer operation.



## Controller

The controller is responsible for complete control of the CMD-V4. From a rearload viewpoint, it is mounted at the left on the output transport.

The controller receives commands from the HOST, acknowledges them, checks them, and executes them. Afterwards, the command acknowledgment is sent to the HOST.

## External messages

As an option, the CMD-V4 controller can handle 2 external messages.

### Door status

If the CMD-V4 is installed in a safe and the status of the safe door is to be reported to the HOST system, the first external input on the controller board can be used for the message line.

### Safety switch

The second external input on the controller board can be used if operation of the CMD-V4 is to be protected by means of a safety switch.

### LOCKING HANDLE

If the dispenser is not fully inserted into the safe, the FW queries the locking handle and prevents the clamp moving to the stop-over position. This only applies to out-safe versions.

## Status display

The 2-digit status display shows the current status of the CMD-V4.

The last status which occurred is always displayed. Status code "00" is displayed before a command that triggers mechanical movements.

**Appendix B** (page **Fehler! Textmarke nicht definiert.Fehler! Textmarke nicht definiert.**) lists possible displays and appropriate reactions.

## Function button

In certain cases it is advisable to check whether the CMD-V4 is ready to operate without using further tools.

A function key is provided on the controller for this purpose. When the key is pressed, all possible functions are cyclically displayed on the status display. Releasing the key for a particular display activates the selected function.

	Function	Description
0	Normal operation	Return to normal mode (not a function).
1	Reset	The result is displayed on the status display after a 'RESET' has been carried out.
2	Overall test + 'RESET'	The mechanical components of the CMD-V4 are moved one after another.
3	Test dispensing + RESET	One banknote is dispensed from each available cassette and transported to the RR box.  <nACT> is modified; <nNDV> remains unchanged.
4	FW release	The release number is displayed in alternating mode.  Example:  From the firmware ID (read with DIL)  \$MOD\$ 061203 1137 CMD_V4_0.BIN  version number <b>1137</b> is displayed as follows:  <b>11</b> for 1 seconds <b>37</b> for 1 second dark for 1 second  After nine attempts, the device returns to normal mode.
5	Resetting statistics counter	The resettable counters are reset to 0 and the current date of the real time clock is saved.
6	Deactivation of encryption	In case of an open door switch and DES_CLEAR_BUTTON=Y the encryption is deactivated.

# Firmware

## Firmware modules

The FW that controls the CMD-V4 is contained in the following modules:

- Bootstrap loader
- Controller

## Bootstrap loader

Until the controller is completely loaded, the commands are processed by the bootstrap loader.

The following commands are executed by the loader:

- Load firmware

After loading (see [L 1], chapter software transfer) a check verifies that the right component has been loaded.

If this is not the case, the response to the load command is a negative acknowledgment and the CMD-V4 remains in the status “FW missing”.

## Controller

The control program of the CMD-V4 contains the complete command set.

In addition to general commands such as software provision, encryption and commands to support serviceability, the behavior can be set to defined values.

## USB – CMD-V4 as USB device

### Introduction

So far command transfer has been realized via the V24 interface by means of the ISO protocol (ISO7498). Due to the introduction of USB, the former 24V transfer has been replaced by USB whereupon the short package procedure of USB is used for flow control.

The CMD\_V4 is a full-speed USB device.

### Firmware behavior

Jobs are no longer transferred with the ISO protocol via USB. In its place, short package mode is used for flow control. The national USB chip that is used supports a maximum FIFO depth of 64 bytes. If more than 64 bytes are to be transferred to or from the CMD\_V4, the data is split into 64-byte blocks and sent separately. Any packet that contains less than 64 bytes of data is known as a short package. Only when a short package is received is the receive buffer with the receive data returned to the caller.

If the USB cable is connected to the CMD\_V4 controller and enumeration has been executed with the PC host, commands can subsequently only be sent and received via USB.

After disconnecting the USB cable from the CMD\_V4 controller or deactivating the PC host, the system switches back to V24, i.e. USB has higher priority than V.24.

In addition to the normal command channel, there is also a debug channel, which communicates warnings and error messages from the firmware to the PC host. The debug channel is immediately activated during device enumeration (CMD\_V4). The PC application can optionally evaluate the debug channel. The debug channel is not a service channel, i.e. it cannot be used to send data to the device.

## Implementation overview

USB module USBN9603 from National Semiconductor is used on the CMD\_V4. This component has one standard endpoint (EP0) and a maximum of six additional endpoints (EP1 – EP6).

Endpoints / type	Assignment
Endpoint 0 (EP0)	Class/Vendor-Requests
Endpoint 2 (EP2) / Bulk	IN endpoint from Host
Endpoint 1 (EP1) / Bulk	OUT endpoint from Host
Endpoint 3 (EP3) / Bulk	Debug channel IN endpoint from Host
Endpoint 4 (EP4)	Not used
Endpoint 5 (EP5) / Bulk	Not used
Endpoint 6 (EP6) / Bulk	Not used

The host (PC) is always the initiator of a transaction, i.e. the device (CMD\_V4) will always only respond to queries from the host and never launch a transaction itself.

The CMD\_V4 behaves as described in the USB specification [1] for full-speed devices.

## Device descriptor

Offset	Field	Length	Value	Description
0	bLength	1 byte	0x12	Descriptor length
1	bDescriptorType	1 byte	0x01	Device
2	bcdUSB	2 byte	0x0110	USB Spec 1.1
4	bDeviceClass	1 byte	0x00	Each interface in a configuration has its own class
5	bDeviceSubClass	1 byte	0x00	Must be 0
6	bDeviceProtokoll	1 byte	0x00	No protocol
7	bMaxPacketSize0	1 byte	0x08	Maximum packet size for endpoint 0
8	idVendor	2 byte	0xaa7	VendorID from Wincor Nixdorf
9	idProduct	2 byte	0x4100	ProductID for CMD_V4
12	bcdDevice	2 byte	0x1130	FW release (ECO) number e.g. 11.30
14	iManufacturer	1 byte	0x01	String descriptor for manufacturer
15	iProduct	1 byte	0x02	String descriptor for product name
16	iSerialNumber	1 byte	0x00	String descriptor for serial number
17	bNumConfiguration	1 byte	0x01	Number of configurations is 1

## Configuration descriptor

Offset	Field	Length	Value	Description
0	bLength	1 byte	0x09	Descriptor length is 9 bytes
1	bDescriptorType	1 byte	0x02	Configuration descriptor type
2	wTotalLength	2 byte	0x0027	Total length is 39 bytes
4	bNumInterfaces	1 byte	0x01	Number of interfaces is 1
5	bConfigurationValue	1 byte	0x01	Value as argument for SetConfiguration() is 1
6	iConfiguration	1 byte	0x00	No string descriptor available
7	bmAttributes	1 byte	0xC0	Configuration characteristics: D7: occupied(1) D6: Bus-powered D5: No Wakeup D4: occupied(0)
8	bMaxPower	1 byte	0x00	Max. current is 0 mA

## Interface descriptor

Offset	Field	Length	Value	Description
0	bLength	1 byte	0x09	Descriptor length
1	bDescriptorType	1 byte	0x04	Interface descriptor type
2	bInterfaceNumber	1 byte	0x00	Interface number
3	bAlternateSetting	1 byte	0x00	No alternative setting
4	bNumEndpoints	1 byte	0x03	Number of used endpoints
5	bInterfaceClass	1 byte	0xff	Vendor specific class
6	bInterfaceSubClass	1 byte	0x00	Reserved
7	bInterfaceProtocol	1 byte	0x00	Reserved
8	iInterface	1 byte	0x00	No string descriptor for this interface

**Endpoint descriptor (endpoint 2 job channel Device > PC)**

Offset	Field	Length	Value	Description
0	bLength	1 byte	0x07	Descriptor length
1	bDescriptorType	1 byte	0x05	Endpoint descriptor type
2	bEndpointAddress	1 byte	0x82	IN endpoint with number 2
3	bmAttributes	1 byte	0x02	Bulk endpoint
4	wMaxPacketSize	2 byte	0x0040	Max. packet size is 64 bytes
6	bInterval	1 byte	0x00	Polling interval all 0 frames/ MicroFrames



**Endpoint descriptor (endpoint 1 job channel PC > Device )**

Offset	Field	Length	Value	Description
0	bLength	1 byte	0x07	Descriptor length
1	bDescriptorTyp	1 byte	0x05	Endpoint descriptor type
2	bEndpointAddress	1 byte	0x01	OUT endpoint with number 1
3	bmAttributes	1 byte	0x02	Bulk endpoint
4	wMaxPacketSize	2 byte	0x0040	Max. packet size is 64 bytes
6	bInterval	1 byte	0x00	Polling interval all 0 frames/ MicroFrames

**Endpoint descriptor (endpoint 3 Debug channel Device > PC)**

Offset	Field	Length	Value	Description
0	bLength	1 byte	0x07	Descriptor length
1	bDescriptorTyp	1 byte	0x05	Endpoint descriptor type
2	bEndpointAddress	1 byte	0x83	IN endpoint with number 3
3	bmAttributes	1 byte	0x02	Bulk endpoint
4	wMaxPacketSize	2 byte	0x0040	Max. packet size is 64 bytes
6	bInterval	1 byte	0x00	Polling interval all 0 frames/ MicroFrames

**String descriptor (country code)**

Offset	Field	Length	Value	Description
0	bLength	1 byte	0x04	Descriptor length
1	bDescriptorTyp	1 byte	0x03	String descriptor type
2	wLangID	2 byte	0x0409	Country code

**String descriptor (manufacturer string)**

Offset	Field	Length	Value	Description
0	bLength	1 byte	0x04	Descriptor length
1	bDescriptorTyp	1 byte	0x03	String descriptor type
2	bString	28		Wincor Nixdorf

## String descriptor (product string)

Offset	Field	Length	Value	Description
0	bLength	1 byte	0x04	Descriptor length
1	bDescriptorTyp	1 byte	0x03	String descriptor type
2	bString	12		WN Cash Media Dispenser (CMD-V4)

## Configuration

The CMD-V4 is configured via jumpers, sensors, connections and configuration commands. The speed of the V.24 line is adjusted automatically.

## Jumper

Application-specific configuration options on the controller board:

- Observe door contact during encryption.

**Appendix E** (page 106) shows the assignment of the jumper field on the controller board.



The remaining positions on the jumper field are intended for extensions or as an aid to controller replacement. They are therefore not described here in detail.

## Sensor

The **safe door** and **safety switch** external sensors affect FW behavior (when they have been activated via a jumper on the controller).  
In addition, the evaluation of HW module IDs determines the current device version (e.g. frontload/rearload).  
The safe switch detects whether the dispenser is completely inserted into the safe or not. If it is not, the DCW command for out-safe machines (clamp moves upwards out of the safe) is not permitted in the relevant status (acknowledgment status D).

## Firmware configuration job

A default defines how the CMD-V4 is to behave.



The settings made are administered in a power failure-proof memory area which is generally not overwritten during a control software update.

## Recover status

Default:	Deactivated
Possible variants:	Switching on / switching off

## Dual dispense evaluation by the measurement station

Default:	Do not reject 2 GOOD notes
Possible variants:	DEFAULT / always reject dual dispenses

## Automatic baud rate switch: 19200 / 56000 baud

In CMD-V4, the baud rate is detected and switched automatically. Two line speeds (19200/56000 baud) are supported by the booter and the firmware. During each DIN protocol inquiry phase, synchronization takes place for both baud rates with the ENQ character.



The remaining V.24 line parameters (ODD/8/2) cannot be changed.

## Encryption

### Encryption procedure, general

The following section provides an insight into the procedures and techniques that are used for the encryption.

### DES

The Data Encryption Standard (DES) serves as the basic principle for encryption processes. This standard is referred to as Data Encryption Algorithms (DEA) by ANSI and DEA-1 by ISO.

As a block cipher, DES transforms data with a block length of 8 bytes into enciphered data blocks of 8 bytes.

For a detailed explanation of the DES encryption, please refer to [L 11].

Based on a symmetric-key algorithm, DES uses the same key for encryption and decryption. Even though a 56-bit key is used, a key is displayed by 64 bits. The remaining 8 bits are used as parity check bits.

To indicate DES encryption and decryption, **e** for encryption and **d** for decryption are used.

DES operating modes:

Below two ways to use the DES for encryption of longer data blocks are presented: the operation modes ECB and CBC.

These two operation modes are also used in other block ciphers, but are at this point specifically presented relating to DES.

## ECB

The electronic codebook (ECB) mode is the simplest of the encryption modes.

The plaintext is divided into blocks of the same length, 8 bytes in case of DES, and each block is encrypted separately under the same key  $K$ .

**i** The disadvantage of ECB mode is that identical plaintext blocks are always encrypted into identical ciphertext blocks.

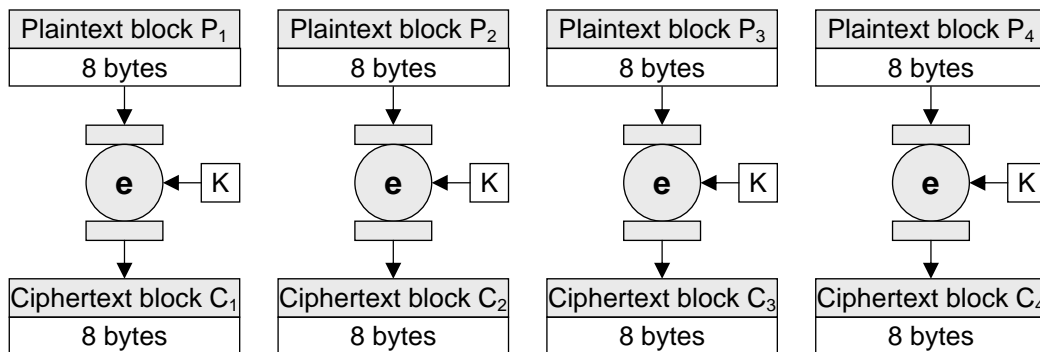


Fig. 1: ECB encryption

**!** Keep in mind that the plaintext that needs to be encrypted is always filled up to a multiple of the block size.

## CBC

The Cipher-block chaining (CBC) mode interconnects the encryption of the blocks by incorporation of the result of the previous encryption into the result of the subsequent one.

Therefore each block of plaintext is XORed with the previous ciphertext block before being encrypted.

The first block is XORed to a start value, the initialization vector (IV) that does not necessarily need to be secret.

**i** Compared to EBC mode, CBC mode effects a protection of the ciphertext over its complete length.

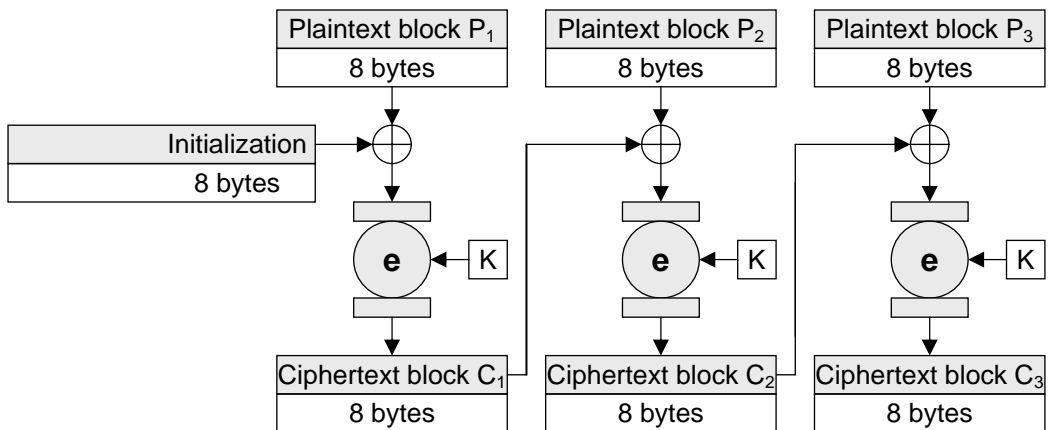


Fig. 2: CBC encryption

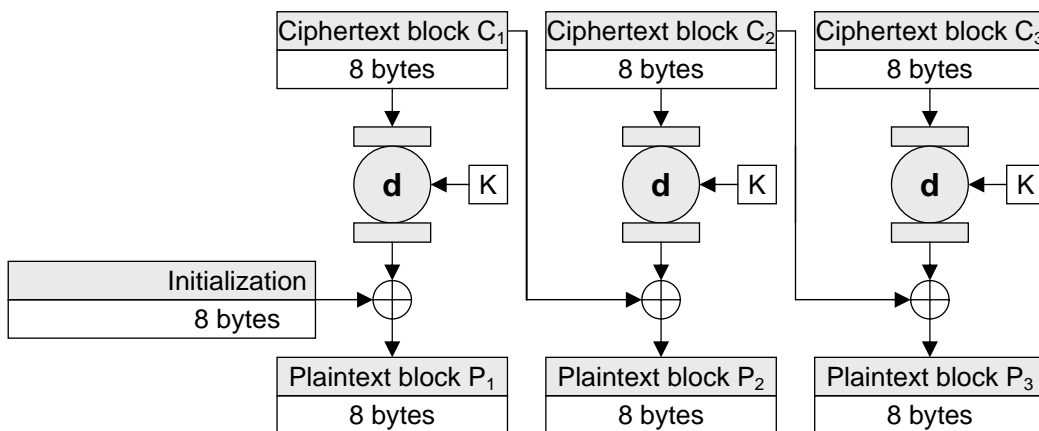
**!** According to the EBC mode, you need to keep in mind that the plaintext that needs to be encrypted is always filled up to a multiple of the block size.

For decryption in CBC the result of a ciphertext decryption is XORed with the previous ciphertext block.

The decryption result of the first block is connected to a start value, the initialization vector (IV) that does not necessarily need to be secret.



For a correct CBC decryption the same IV must be used for both processes, encryption and decryption.



III. 3: CBC decryption

## Triple-DES

Due to the weaknesses of the 8-byte keys of DES, for outstandingly security relevant objects the Triple DES cipher is used. This uses 16-byte keys, also named double-sized keys. Data blocks encrypted with Triple DES have a length of 8 bytes and thus the same length as the ones that are encrypted in DES mode.



16-byte keys always indicate a Triple DES encryption.

If you split a 16-byte key in half, you can regard the two resulting 8-byte halves as two independent keys. The designation of a complete 16-byte key is **KK** and thus it is split into the left half **K<sub>L</sub>** and the right half **K<sub>R</sub>**.

Application of the Triple DES algorithm means that the DES cipher algorithm is applied three times under the subkeys K<sub>L</sub> and K<sub>R</sub>.

The Plaintext block P is successively DES encrypted under K<sub>L</sub>, DES decrypted under K<sub>R</sub> and again DES encrypted under K<sub>L</sub>.

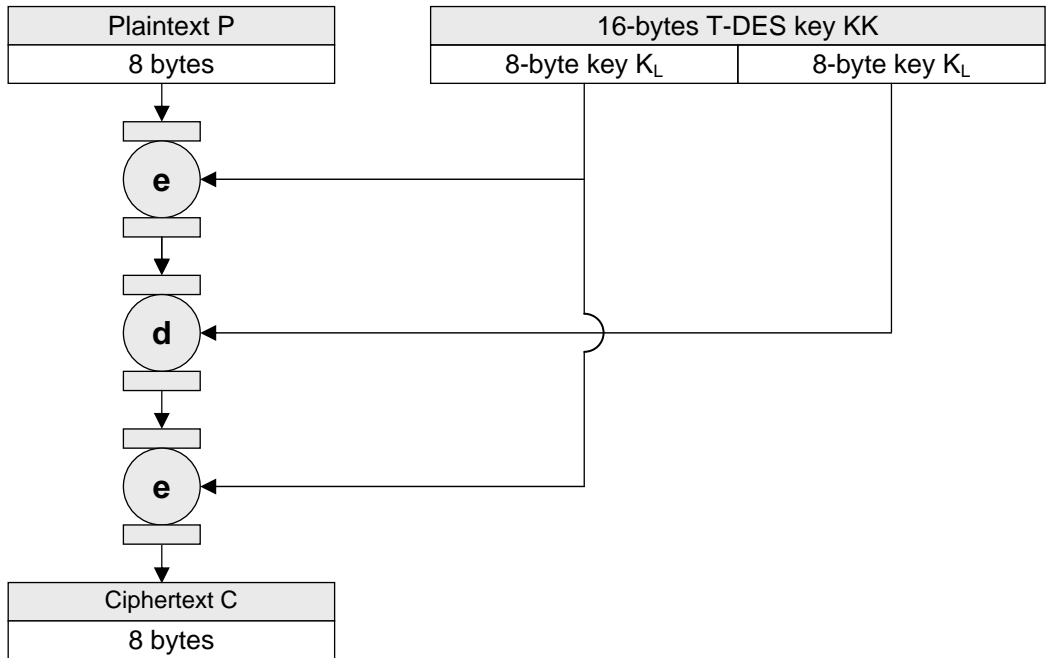


Fig.:4 Triple DES encryption

**i** In case of two identical subkeys, a Triple DES encryption equates to a simple DES encryption.

Accordingly, in Triple DES mode decryption is carried out by decryption under the first subkey, encryption under the right subkey, and another decryption under the left subkey.

As due to the triply applied DES algorithm Triple DES remains a block cipher, the operation modes ECB and CBC are usable as well.



## Encryption CMD-V4

The interface to the CMD-V4 can operate with encryption.

Regarding commands and acknowledges, encryption is limited to the user data.

**For identification, user data is double framed.**

In general, commands for software transfer are not encrypted.

Two different modes are used:

### Key exchange

For key exchanges, the ECB (Electronic codebook) mode is used.

### Data exchange

The encryption algorithm used for data exchange is the CBC (Cipher-block chaining) mode which contains the Triple DES Algorithm.



When encryption is active, user data length is a multiple of 8.

This also applies to commands without user data.

Consequently, it might be necessary to fill up the user data with filler bytes (00 H).

At least 2 filler bytes are needed because prior to encryption a CRC code is calculated over the user data (calculation according to BCS [L1]) and filed in the penultimate and ultimate buffer position.

This checks the synchronization of the used keys.

Initialization is shown in the following illustration.

HOST		CMD-V4
<SEDM>- Request status	=>	
	<=	Transport key TK missing
Generate and pass TK	=>	Save TK when safe door is open and shall be considered for encryption.
Request EDM status	=>	
	<=	Line key LK missing
Generate LK, encrypt it under TK and pass it	=>	Decrypt LK under TK and secure it
Request EDM status	=>	
	<=	Key available
Request initial value	=>	Generate IV, encrypt it under TK
Decrypt IV under TK and secure it	<=	and pass it
<b>encrypted data</b>	=>	
	<=	<b>encrypted data</b>
Key test	=>	
	<=	Check pattern, positive acknowledgement if valid, key error otherwise.

Key synchronization HOST <=> CMD\_V4



On transport of the non-encoded transport key, the jumper “Observe door contact during encryption” on the controller is evaluated. If the jumper is activated, the transport key is only accepted when the safe door is open.

Prior to the activation of line encryption, the initiator must ensure himself that no attempts at manipulation of the data conductor have taken place.

Once encryption is activated, it can only - in case of an activated jumper - be deactivated via command when the safe door is open. When the safe door is open, encryption can also be deactivated by DES\_CLEAR\_BUTTON\_6=Y + pressing button 6.

# Commands

The operating phases of an ATM can be subdivided into customer transaction, supply and disposal, and servicing.

Commands are available in the CMD-V4 for each phase:

## Customer transaction

- Reset
- Dispensing
- Transport
- Cancel
- Status

## Supply and disposal

- Update system data
- Determine reference values
- Encryption
- Software transfer

## Service

- Self-test
- Internal commands

## General

### Plausibility check

Before **each command** is executed, it is first checked for plausibility.

All unknown commands, commands with parameter values outside their valid range and parameter values with illegal data types are rejected with a “SYNTAX” status code.

Commands or parameters which are not legal at present or which cannot be executed in the current hardware configuration are rejected with a “CONTEXT” status code.

The possible status codes and their causes are described in the command acknowledgments.

### Device lock

The CMD-V4 takes two device locks into consideration during processing:

- Safety switch
- Lock due to cash being retained in the stacking position.

As soon as a lock is set, the CMD-V4 rejects any commands involving device operations.

#### Removing locks:

Safety switch (S\_SW flag)

- **Action:** Push CMD-V4 completely into the safe.

Banknote retract (DLOC flag)

- **Action:** Remove cash, then remove lock via command.

If required, the lock flag can be canceled without removing the bundle first.

The command is then acknowledged with a device error, since the transport paths are blocked.

The bundle can then be re-offered or placed in the RR box.

## DA – Reset

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	A	Command specification: RESET	(ASCII)
1	blank	no additional specification	(ASCII)
	00 H	any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	A	Command specification: RESET	(ASCII)
1	blank	no additional specification	(ASCII)
1	a	Status code (-> Device status <SRES>)	(ASCII)
	00 H	any filler bytes	(HEXA)

### Description

A 'RESET' checks the ready status of the CMD-V4; this is only given after an error-free execution of the command!

Principally, a RESET involves the same routines as during Power on, apart from the power up tests. Only the routines during 'Power Up Test' are left out.

After a device reset the CMD-V4 is in a defined state.

Execution is canceled when the first error is detected.

#### Memory reset

The internal markers are reset.

In addition to the general memory area, there are other areas (e.g. all status areas and the control program) that retain their contents even in the event of a power failure, thanks to backup battery capacity.

### Determination of the device type

The following cells are set in the device status of the CMD-V4 using the codes on the controller board:

<b>&lt;TYPE&gt;</b>	<b>Device type</b>	<b>(Front-/ Rearload/ Undefined)</b>
<b>&lt;N_VM&gt;</b>	No. of dispenser modules	(1...6)
<b>&lt;DOOR&gt;</b>	Status of door sensor	
<b>&lt;S_SW&gt;</b>	Status of safety switch	
<b>&lt;SHUT&gt;</b>	Shutter status	
<b>&lt;DO_S&gt;</b>	Observe door contact during encryption	
<b>&lt;TRANSPORT&gt;</b>	Output transport variant	in Safe front/rear, out Safe front/rear

### Checking the cassettes

First type and quantity of the available cassettes are determined. The corresponding cells in the cassette status are updated.

The values from the cassette memory are compared with the internal data. If there is a difference, the corresponding cassette status **<nSTA> = N** is set.

To operate correctly, the device requires a minimum of 1 ready cash-out cassette and the reject/retract cassette.

If this minimum requirement is not met, it is indicated in the device status **<CAS> = N** and the command terminates with 'Out of service'.

### Checking device locks and initialization

First of all, the status of the safety switch is determined (open/closed) and recorded in the device status **<S\_SW>**.

The device is subsequently initialized.

During initialization an attempt is made to standardize the routing disk and to move the clamp to the stacking position and to open it.

Depending on the current status of the cash dispenser, it can result in banknotes being retracted and the shutter closing.

Then the device status is checked to determine whether the system is locked **<DLOC> = Y**.

Processing is terminated with the status 'Out of service' if the device lock is on, if the safety switch is 'open' or if initialization has failed.

Initialization is also carried out if the system is locked <DLOC> = Y, but cash, if available, is not retracted into the RR-box..

### Checking the transport paths

All transport routes are checked. Banknotes being transported are stored in the RR-box. It is differentiated between:

- **Customer money** is stored in the retract compartment of the RR-box.  
The action is documented in the cassette status <RRET> + 1.
- **Bank cash** is removed via bundle reject.

The result of all actions is documented in the device status <SCLE> and <SRES>.

### Checking the printing cassette

'POWER\_ON-/RESET' checks the cassette empty sensor of the 'DK' (printing cassette) for emptiness. If a document is present, a print proof is created. Additionally a control is performed to ensure that the empty sensor does not detect the document again after the time-out period. If the empty sensor detects another document (printer prints next document), up to 5 dispensing procedures with subsequent time-out periods are executed when the empty sensor is covered again. If another document is detected at the cassette empty sensor, processing is aborted with the status 'Out of service'.

### Handling jams

A paper jam is documented in the device status <TER> = J. The area in which the problem has to be solved is stated in <TS> (start and destination).

If the jam occurred during a dispensing command, it is checked whether the jam has been cleared in the corresponding transport.

To do this, at least one test note is taken from the lowest cash-out cassette and transported to the stacking compartment. If this note reaches its destination, it is disposed of as a REJECT in the RR box.

Following a JAM that occurs during dispensing, the transport path is checked with a test banknote during reset.

After 10 unsuccessful RESETs, mechanical movements cease. Instead, the command is acknowledged with 'D'.

This mode is only exited after power off/on, key reset, or a reset via the safety switch. This is to ensure that an engineer/operator has remedied irreparable damage caused by the firmware.

### **Checking the shutter (optional)**

If available, the functionality of the shutter is checked in a 'shutter test'.

The result is documented in the device status **<SHUT>**.

The command is terminated with 'Out of service' if the test cycle is not completed successfully.

### **Checking the photosensors**

All photosensors are checked. The respective result is administrated internally.

### **Checking the cash trapping detection**

During initialization of the CMD main firmware (after Power on, RESET, or CLEARING), the function and initialization values are checked when a cash trapping detection is available and working **<SHUT\_PROTECT\_OPT>=Y**. In case of a negative result, CLEARING, RESET, or the first status request after the Power On are acknowledged with status D ("Out of service") and in device status **<SHUT\_PROTECT>=D** is set.



## Status Codes

### A = OK

No actions performed	<SCLE> = 0 & <n_STA> = R
Documents successfully placed in RR-box	<SCLE> = 1 & <n_STA> = R

### B = Software missing

Control program not available

### D = Out of service

Cassettes cannot be processed	<CAS> = N
Safety switch open	<S_SW> = O
Device lock set	<DLOC> = Y
Transport error	<TER> = J
Operator request	<OR> = Y
Self-test active	<TST> = Y
Shutter error	<SHERR> = B
Printing cassette only	
5 documents in RR-box, additional document available	<n_STA> = V
Transport error	<TER> = J
only when cash trapping detection is activated	
Cash trapping detection faulty and additionally	<SHUT_PROTECT> = D <SHERR> = B, <SHUT> = U, <SCOD> = 28

### E = Device error

Documents from printing cassette cannot be separated	<n_STA> = V
--	-------------

**Appendix A** (page 88) shows the possible status codes for each command in a cross-reference list.

**Appendix X** (page 155) shows the possible device and cassette states for each command in a cross-reference list.

## DAB – Clearing

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	A	Command specification: RESET	(ASCII)
1	B	Additional specification: Clearing	(ASCII)
	00 H	Any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	A	Command specification: RESET	(ASCII)
1	B	Additional specification: Clearing	(ASCII)
1	a	Status code	(ASCII)
	00 H	Any filler bytes	(HEXA)

## Description

### Checking device locks and initialization

First of all, the status of the safety switch is determined (open/closed) and recorded in the device status <S\_SW>.

The device is subsequently initialized.

During initialization an attempt is made to standardize the routing disk and to move the clamp to the stacking position and to open it.

Depending on the current status of the cash dispenser, it can result in banknotes being retracted and the shutter closing.

Then the device status is checked to determine whether the system is locked <DLOC> = Y.

Processing is terminated with the status 'Out of service' if the device lock is on, if the safety switch is 'open', or if initialization has failed.

Initialization is also carried out if the system is locked <DLOC> = Y, but cash, if available, is not retracted into the RR-box..

### Checking the transport paths

All transport routes are checked. Banknotes being transported are stored in the RR-box. It is differentiated between:

- **Customer money** is stored in the retract compartment of the RR-box.  
The action is documented in the cassette status <RRET> + 1.
- **Bank cash** is removed via bundle reject.

The result of all actions is documented in the device status <SCLE> and <SRES>.

### Checking the printing cassette

A document ready for dispensing ('cassette empty sensor' being covered) is temporarily stored in the stacking department. Should the situation arise, upcoming documents (4 at the most) are transported into the stacking department as well.

The documents that collect there are removed via bundle reject.

If another document is detected at the cassette empty sensor, processing is aborted with the status 'Device not ready'.

### Handling jams

A paper jam is documented in the device status <TER> = J. The area in which the problem has to be solved is stated in <TS> (start and destination).

If the jam occurred during a dispensing command, it is checked whether the jam has been cleared in the corresponding transport.

To do this, at least one test note is taken from the lowest cash-out cassette and transported to the stacking compartment. If this note reaches its destination, it is disposed of as a REJECT in the RR box.

### Checking the shutter (optional)

If available, the functionality of the shutter is checked in a 'shutter test'.

The result is documented in the device status <SHUT>.

The command is terminated with 'Out of service' if the test cycle has not been completed successfully.

After 10 unsuccessful RESETs, mechanical movements cease. Instead, the command is acknowledged with 'D'.

This mode is only exited after power off/on, key reset, or a reset via the safety switch. This is to ensure that an engineer/operator has remedied irreparable damage caused by the firmware.

### Checking the photosensors

All photosensors are checked. The respective result is administrated internally.

### Checking the cash trapping detection

During initialization of the CMD main firmware (after Power on, RESET, or CLEARING), the function and initialization values are checked when a cash trapping detection is available and working <SHUT\_PROTECT\_OPT>=Y. In case of a negative result, CLEARING, RESET, or the first status request after the Power On are acknowledged with status D ("Out of service") and in device status <SHUT\_PROTECT>=D is set.

## Status Codes

### A = OK

No actions performed	<SCLE> = 0 & <n_STA> = R
Documents successfully placed in RR-box	<SCLE> = 1 & <n_STA> = R

### B = Software missing

Control program not available

### D = Out of service

Safety switch open	<S_SW> = O
Device lock set	<DLOC> = Y
Transport error	<TER> = J
Operator request	<OR> = Y
Self-test active	<TST> = Y
Shutter error	<SHERR> = B
Printing cassette only	
5 documents in RR-box, additional document available	<n_STA> = V
Transport error	<TER> = J
only when cash trapping detection is activated	
Cash trapping detection faulty	<SHUT_PROTECT> = D
then additionally	<SHERR> = B, <SHUT> = U, <SCOD> = 28

### E = Device error

Documents from printing cassette cannot be separated	<n_STA> = V
--	-------------

**Appendix A** (page 88) shows the possible status codes for each command in a cross-reference list.

**Appendix X** (page 155) shows the possible device and cassette states for each command in a cross-reference list.

## Dispensing

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	B	Command specification: 'Dispensing'	(ASCII)
1	a	Additional specification: S = Standard C = Dispensing with notes in stacker	(ASCII)
		per cassette from which notes are to be dispensed:	(ASCII)
1	d	Cassette location (1- 6)	(decimal in ASCII)
1	,	Separator	(ASCII)
2	dd	Number of notes (01 - 60)	(decimal in ASCII)
1	: or ;	Separator (after last cassette ";" = END)	(ASCII)
	00 H	any filler bytes	(HEXA)

The following agreement applies for numbering the cassette locations:

The top location for cash-cassettes is location number 1. The locations below that are numbered accordingly in ascending order.

## Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	B	Command specification: 'Dispensing'	(ASCII)
1	a	Additional specification: S = Standard C = Dispensing with notes in stacker	(ASCII)
1	a	Status code	(ASCII)
1	d	per cassette processed: (in ascending order of cassette locations) Cassette location	(decimal in ASCII)
1	,	Separator	(ASCII)
2	dd	Number of notes in collecting tray	(decimal in ASCII)
1	,	Separator	(ASCII)
3	ddd	Number of notes in reject cassette	(decimal in ASCII)
1	,	Separator	(ASCII)
1	a	Filling level, status of cassette O = Minimum quantity not yet reached L = Fallen below minimum quantity E = Cassette physically empty M = Too many notes with incorrect dimensions S = Too many multiple dispenses N = Note dispensing not possible	(ASCII)
1	: or ;	Separator (after last cassette ";" = END)	(ASCII)
	00 H	any filler bytes	(HEXA)

## Description

The description is organized into subcommands.

### Checks

The command is rejected if

- its syntax is incorrect.
- a device lock is set.
- the shutter is not closed.
- the device is out of service.
- a cash trapping detection error is set.

### Basic status check

A check is run to determine if:

- banknotes are in the collecting tray (only for additional specification: 'Standard').
- the transport paths are free.
- all requested cassette locations have the status nSTA = R (ready).
- or the printing cassette has the status nSTA = V (ready).



**Note dispensing and collecting**

- Dispensing is not prevented if:
- a logical dispenser cassette is now empty.
  - the reject/ retract cassette is full.

The operator must ensure that the CMD-V4 has the correct data before customer mode is activated!.

**Additional specification: S (standard dispensing)**

Prior to standard dispensing, the dispensing-specific counter readings of all the cassettes (**<nL\_D> = 0**; **<nREJ> = 0**) are standardized.

**Additional specification: C (dispensing with notes in stacker)**

This version can be advantageous if the specified denominations have not been achieved via the standard command and the required rest is to be dispensed via an altered HOST specification.

The required number of notes is dispensed from each cassette. The order specified in the command is complied with.



Up to max. 60 notes can be processed per transaction in the CMD-V4 and VCMD.

This value requires a note quality corresponding to the paper specification and does not apply for all types of notes and application areas.

This restriction must be monitored by the application. Up to max. 5 documents can be dispensed per transaction for the printing cassette.

With each successfully dispensed note the cassette-specific supply counter is decremented **<nACT> - 1**. If the counter is not decremented below the minimum number (0).

The dispensed notes are transported to the measurement station.

All 'good notes' are stored in the stacking department. The VCMD features a rejection of objectionable notes via single reject. Concerning CMD-V4, up to two objectionable notes can be singled out in case the device features a single reject function on the hardware side. Further objectionable notes are collected in the stacking compartment which, in case of the additional specification S, leads to a note bundle reject, an emptying of the reject tray (if available) and subsequently to a restarted dispense transaction. For those notes that have been rejected via single reject, replacement notes are provided via post dispensing.

Dispensing ends when:

- the required number of notes has been achieved.
- a cassette empties during a dispense process.
- a fault which cannot be eliminated occurs during current the dispensing process.

### **Editing the acknowledgment**

For each processed cassette position, the command acknowledgement reports the number of notes that have been transported from the respective cassette into the stacking compartment **<nL\_D>** or respectively into the reject cassette **<nREJ>** to the HOST. Additionally, the filling level is reported. It is evaluated from the reports of the minimum quantity sensor **<nLOW>** and the empty sensor **<nSTA>**.

If a note jam occurs during dispensing, it is probable that acknowledgment parameters are also transferred for cassette locations which have not been requested (trial dispense transaction).

During dispensing with notes in a stacker, the acknowledgment contains the total number of notes which have been retracted since the last standard dispensing.

The transferred number of notes in the acknowledgment corresponds to the values **<nL\_D>** and **<nREJ>** in the cassette status.

The acknowledgment parameters are not transferred if status codes B, C, and I are returned.

With D the acknowledgment parameters are only transferred if dispensing was started and an error occurred.

**Status Codes****B = Software missing**

Control program not available

**C = Syntax**

invalid cassette location no. (unequal 1- 6)

too many dispensing commands (more than 10)

Number of notes &gt; maximum value (60)

Number of notes > maximum value (5) for printing  
cassette**I = Context**

Cassette error	<nSTA>	unequal R
Control command 'Dispensing with notes in stacker' (only for DBS)	<TS>	unequal 12
Number of notes in stacker with notes in stacker > 60		
Shutter blocked	<SHERR>	= B
Shutter is not closed	<SHUT>	= O/U
Manipulation at output transport	<TER>	= M
Cash path not free or	<TF>	= N
Clearing after cash presentation not OK		
Error during dispensing with notes in stacker	<DIS>	= J

**D = Out of service**

Safety switch open	<S_SW>	= O
Device lock set	<DLOC>	= Y
Minimum configuration missing	<CAS>	= N
Single reject switch defective (facing the reject direction)	<SR>	= R
Banknote jam	<TER>	= J
Operator request	<OR>	= Y
Self-test active	<TST>	= Y

**E = Device error**

Cassette is empty	<n_STA>	= E
Too many wrong-sized notes*	<DIS>	= M
Too many multiple dispenses *	<DIS>	= S
Note cannot be dispensed *	<DIS>	= N
Too many bundle rejects	<DIS>	= E
Banknote jam during post dispensing	<DIS>	= J
Cash trapping detection manipulated	<SHUT_P ROTECT>	= M
Cash trapping device detected	<SHUT_P ROTECT>	= K
Cash trapping detection faulty	<SHUT_P ROTECT>	= D

Printing cassette only

Document cannot be dispensed	<n_STA>	= V
Timeout: documents are not available	<n_STA>	= R

\* Within the acknowledgment parameter 'Filling level, status' for every command, this status value is also transferred cassette-specifically.

The cross-reference list in Appendix A shows the possible status codes for each command (see "Appendix A").

## Special handling

### Undefined or invalid notes

If undefined or invalid notes arrive at the collecting tray, a bundle reject occurs with the additional specification S. Thereafter, the original command is repeated.

After 3 consecutive bundle rejects, the command is aborted with the status 'Device error' **<DIS> = E** (too many bundle rejects).

With the additional specification C (dispensing with notes in the stacker) such notes in the stacking compartment result in dispensing being terminated with the acknowledgment E = 'Device error', **<DIS = J>**.

### Multiple dispense

In the CMD-V4 and VCMD all the dispensed notes are measured by a double note detector. Besides detecting the note length and thickness, this gauge can also recognize whether it is just one note or two or more overlapping notes or notes without a gap between them. When it is more than just one note, it is known as a multiple dispense. The multiple-note detection unit can detect multiple dispenses with correct single notes. To ensure that the reject rate at the CMD-V4 and VCMD remains low, it is possible to configure what should happen to valid multiple dispenses. Two methods can be set via SDO\_BAD (see Appendix X – K- Firmware configuration):

**SDO\_BAD = Y:** all the multiple dispenses are singled out via the single reject and a replacement note is dispensed.

**SDO\_BAD = N:** all the multiple dispenses which consist of two valid single notes are not singled out until the required number of notes has been achieved or exceeded.

The cassette-specific counter is corrected when a multiple dispense is detected  
**<nACT> - 1.**

Dispensing aborts with the status 'Device error' **<DIS> = S** (too many multiple dispenses) if more than five multiple dispenses have to be rejected.

### Dispensing of notes not possible

If the CMD-V4 does not succeed in dispensing a note, the dispense transaction is initiated again.

The dispensing attempt is repeated five times at the most. After that, the command is terminated under the status 'device error' **<DIS> = N** (cassette is empty or note bundle is sticky).

The acknowledgment also includes the dispenser-specific counter per cassette (no. of notes in collecting tray / no. of notes in the reject cassette).

With this information it can be decided in the HOST whether the desired amount can be dispensed via a new dispense process or whether the customer transaction must be cancelled.

### Banknote jam

A banknote jam may occur during the dispense process.

If a note jam occurs, power is applied to the transport motor for approx. another 2 s. This ensures that transportable notes reach the stacking compartment.

A trial dispensing process is then performed by dispensing one note from the bottom cassette and transporting it to the collecting tray.

- The note arrives the stacking compartment:

When the note arrives in the stacking compartment, the transport route is cleared. With the additional specification S (standard dispensing), a bundle reject is conducted and the original command repeated. With the additional specification C (dispensing with notes in the stacker) such notes in the collecting tray result in dispensing being terminated with the acknowledgment E = 'Device error', **<DIS = J>**

- The note does not arrive in the stacking compartment:

The procedure is repeated twice at the most. Afterwards, a non removable banknote jam is clearly ascertained. There is no further device operation. **<TER> = J** indicates the lack of service in device status. The command is terminated with 'Out of service', a check is necessary.

The troubleshooting must be completed with 'RESET'. The flag is reset (**<TER>=O**) when this is done successfully.

### Special handling 'Printing cassette'

When several documents shall be dispensed from the printing cassette during one transaction, the AZM awaits a covering of the LKLx with a time-out period of 6 to 7 seconds with the motor running. When the LKLx is covered by the next document printed, this is rejected with a delay of 200 ms.

### Single reject deflector defective

An AZM\_NG recognizes a single reject deflector error if notes do not reach the destination chosen but arrive at a point in the opposite direction.

This circumstance is reported in the device status **<SR> = S / R**. Further treatment depends upon the switch position.

- Transport direction: collecting tray **<SR> = S**

In this case, the AZM\_NG can continue to operate with limited capacity.

As soon as a single reject becomes necessary, the notes dispensed up to that point are lost. A note bundle reject then takes place (including counter increment).

When dispensing with notes in a stacker, the command is acknowledged with E = Device error, **<DIS = J>**.

Standard dispensing is repeated. If the specified number of notes can be presented, the command is positively acknowledged.

- Transport direction: reject cassette **<SR> = R**

In this case, further dispensing is impossible. All notes that have been dispensed up until this point are placed in a bundle (incl. counter increment).

The command is terminated with 'Out of service', a check is necessary. Correction must be completed with 'RESET'. The inhibit flag is reset (**<SR> = O**) when this is done successfully.

### Note dimensions outside the tolerance values.

If the measurement station detects notes whose height does not lie within the prescribed tolerances, the note is probably a counterfeit.

In this case, the cassette-specific counter is corrected **<nACT> - 1**, the note is singled out via the single reject and a replacement note is dispensed.

Dispensing aborts with the status 'Device error' **<DIS> = M** (too many notes with wrong dimensions) if several counterfeit notes are detected during dispensing from a cassette.

### Cash trapping detection check

With cash trapping device available and activated **<SHUT\_PROTECT\_OPT>=Y** and the shutter closed, the cash trapping detection sensors are checked prior to dispensing. The sensors might have been manipulated during the last transaction. If the result is negative, the command is terminated under status E ('Device error') and **<SHUT\_PROTECT>=M**.

If an error **<SHUT\_PROTECT>=K** ("Cash trapping detected") occurs, a check for cash trapping detection is performed, both with the shutter open and closed. If the result is negative, the command is terminated under status E ('Device error') and **<SHUT\_PROTECT>=K**.

For a reset of the error see section "Reset of a cash trapping detection error" (page 10).

The commands (DC..., e.g. Retract, Open shutter, Close shutter, ...), RESET, CLEARING or Power on are, upon error-free function and a pending cash trapping detection error, acknowledged with A and **<SHUT\_PROTECT>** remains unchanged.

## Determine reference value

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	G	Command specification: 'Determine reference value'	(ASCII)
1	blank	No additional specification	(ASCII)
1	d	Cassette location (1- 6)	(decimal in ASCII)
	00 H	any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	G	Command specification: 'Determine reference value'	(ASCII)
1	blank	No additional specification	(ASCII)
1	a	Status code	(ASCII)
1	d	Cassette location	(decimal in ASCII)
1	,	Separator	(ASCII)
2	dd	Number of notes in stacking compartment	(decimal in ASCII)
1	,	Separator	(ASCII)
3	ddd	Number of notes in reject cassette	(decimal in ASCII)
1	,	Separator	(ASCII)
1	a	Filling level, status of cassette O = Minimum quantity not yet reached = Minimum quantity not reached L = Cassette physically empty E = Note dispensing not possible N	(ASCII)
1	: or ;	Separator (after last cassette “,” = END)	(ASCII)
	00 H	any filler bytes	(HEXA)



## Description

The determination of the reference value can be performed for a maximum of 60 different unknown banknotes.

From the specified cassette the currency characteristics, the value of the banknotes and the release date will be read out and transferred to the measurement point. For cassette locations that have not yet been initialized or are faulty <nCUR> = \*\*\*, no reference value determination is performed.

From the current cassette, as many banknotes are dispensed as necessary for the reference value determination.

The process of determining the reference value will be terminated at the latest if the reference value cannot be determined within the maximum possible number of notes.

The command acknowledgement reports the number of notes that have been dispensed.

After termination of the command, all dispensed notes are in the stacking compartment.



After the reference value determination you need to compare the reported number of notes to the number of notes in the stacking compartment to exclude the possibility of multiple dispenses influencing the reference value.

Therefore the notes can:

- be transported to the output.
- taken out of the stacking compartment when the safe is open and the AZM\_NG extracted.

Subsequently, the notes can be relocated in the stacking compartment and be transported into the reject cassette via 'RESET'.

## Status Codes

### B = Software missing

Control program not available

### C = Syntax

invalid cassette location no. (unequal 1- 6)

### I = Context

Cassette status	<nSTA>	unequal R, P
Cassette faulty or not yet initialized	<nCUR>	= "****"
Cash path not free	<TF>	= N

### D = Out of service

Self-test active	<TST>	= Y
Safety switch open	<S_SW>	= O
Device lock set	<DLOC>	= Y
Reject switch	<SR>	= R
Reject cassette not processable	<RSTA>	unequal R
Banknote jam	<TER>	= J

### E = Device error

Note cannot be dispensed	<DIS>	= N
Reference value could not be determined as the max. number of notes has been achieved	<nSTA>	= P
Cassette has become physically empty.	<nSTA>	= E
Cash trapping detection manipulated	<SHUT_P ROTECT>	= M
Cash trapping device detected	<SHUT_P ROTECT>	= K
Cash trapping detection faulty	<SHUT_P ROTECT>	= D

The cross-reference list in Appendix A shows the possible status codes for each command (see "Appendix A").

## Shutter commands

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	C	Command specification: 'TRANSPORT'	(ASCII)
1	a	Additional specification: Shutter commands T = Shutter test	(ASCII)
	00 H	any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	C	Command specification: 'TRANSPORT'	(ASCII)
1	a	Additional specification: Shutter commands	(ASCII)
1	a	Status code	(ASCII)
	00 H	any filler bytes	(HEXA)

### Description

#### Shutter test

The operability of this module is checked by a quick motion of the shutter mechanics. The shutter flap is not opened in this process. The result is documented in the device status **<SHUT>**.

During the shutter test, a test of the cash trapping detection is performed as well when the detection is available and **<SHUT\_PROTECT\_OPT>=Y**. In case of **<SHUT\_PROTECT>=M** the test is performed exclusively when the shutter is open. For **<SHUT\_PROTECT>=K**, it does not matter whether the shutter is open or closed.

If no manipulation or cash trapping device is detected, **<SHUT\_PROTECT>=O** is set in device status.

A positive acknowledgement guarantees that the shutter is connected, the mechanics not blocked, and the shutter flap closed.

Otherwise, the command is acknowledged with 'Device error'.

## Status codes

### B = Software missing

Control program not available

### I = Context

Shutter not available	<SHUT>	= M, D
Cash path not free	<TF>	= N
and shutter not blocked	<SHUT>	unequal B
Cash path not free	<TF>	= N
and no manipulation in transport	<TER>	unequal M

### D = Out of service

Banknote jam	<TER>	= J
Self-test active	<TST>	= Y
Device lock set	<DLOC>	= Y
Safety switch open	<S_SW>	= O

### E = Device error

Manipulation in transport: (close shutter)	<TER>	= M
Shutter does not reach target position	<SHERR>	= B
Cash trapping detection faulty	<SHUT_P ROTECT>	= D
Cash trapping manipulation	<SHUT_P ROTECT>	= M
Cash trapping device detected	<SHUT_P ROTECT>	= K
and in case of SHUT_PROTECT = M / K	<SHUT>	= B
	<SHERR>	= U

The cross-reference list in Appendix A shows the possible status codes for each command (see "Appendix A").

## DEK – read FW configuration

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	E	Command specification: Status	(ASCII)
1	K	Additional specification: Read firmware configuration	(ASCII)
	00 H	Any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	E	Command specification: Status	(ASCII)
1	K	Additional specification: Read firmware configuration	(ASCII)
1	a	Status code	(ASCII)
n	aa..aa	<b>Configuration data</b> See "Appendix X – FW configuration" (page 177)	(ASCII)
	00 H	Any filler bytes	(HEXA)

### Description

This command returns the current firmware configuration.  
The status can be up to 2 Kbytes in size.



The individual elements **cannot** be addressed via the code word. The position of a code word is **not fixed!** The entire area is always transferred.

### Status Codes

#### B = Software missing

Control program not available

**Appendix A** (page 88) shows the possible status codes for each command in a cross-reference list.

**Appendix X** (page 155) shows the possible device and cassette states for each command in a cross-reference list.

## DFG – define FW configuration

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	F	Command specification: System parameter	(ASCII)
1	G	Additional specification: Define firmware configuration	(ASCII)
n	aa..aa	Parameter definition	(ASCII)
1	;	End character	(ASCII)
	00 H	Any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	F	Command specification: System parameter	(ASCII)
1	G	Additional specification: Define firmware configuration	(ASCII)
1	a	Status code	(ASCII)
	00 H	Any filler bytes	(HEXA)

### Description

This command configures firmware execution and features.



Only one code word and its content are allowed to be specified in the command.

A subsequent reset is required to activate the set configuration(s).

“Appendix X - FW configuration” (page 177) contains a list of the code words used in the CMD-V4 incl. a list of value range per code word.

Example: (User data only):

SDO_BAD=Y;	Dual notes are rejected (more information: see page 173)
------------	---

Status Codes

**B = Software missing**

Control program not available

**C = Syntax**

Status cell unknown  
Default value invalid

**I = Context**

**Appendix A** (page 88) shows the possible status codes for each command in a cross-reference list.  
**Appendix X** (page 155) shows the possible device and cassette states for each command in a cross-reference list.



## Encryption

The following commands are available for encryption:

- Key handling.
- Request initial value.
- Key test.
- Firmware configuration DES\_options

## Key handling

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	H	Command specification: 'Key handling'	(ASCII)
1	a	Additional specification: T = transport key L = line key O = deactivating encryption	(ASCII)
1	:	Separator	(ASCII)
n	xx .. xxH	Key: n=8, or 00H in case of additional specification 0 n= 16 in case of 3DES encryption (see DES_OPTION)	(HEXA)

**Acknowledgement**

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	H	Command specification: 'Key handling'	(ASCII)
1	a	Additional specification: T = transport key L = line key O = deactivating encryption	(ASCII)
1	a	Status code	(ASCII)
	00 H	any filler bytes	(HEXA)

**Description**

It is possible to configurate via the controller board whether the door switch is to be evaluated during encryption.

If YES, the corresponding actions are only executed by the SW-VEM of the CMD-V4 when the door switch gives the 'Open'-message!

**Status Codes****B = Software missing**

Control program not available

**I = Context**

Transport key is missing (only for additional specification L)

**F = Safe is not opened.**

Safe door is not opened (only for additional specification T, O)      <DOOR> = M / C

The cross-reference list in Appendix A shows the possible status codes for each command (see "Appendix A").

## Request initial value

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	H	Command specification: 'Key handling'	(ASCII)
1	I	Additional specification: request initial value	(ASCII)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	H	Command specification: 'Key handling'	(ASCII)
1	I	Additional specification: request initial value	(ASCII)
1	a	Status code	(ASCII)
8	xx .. xxH	Initial value	(HEXA)

### Description

In AZM\_NG, the initial value is generated as a random number and sent to the HOST within the acknowledgement.

The initial value is TK encrypted (ECB mode)

### Status Codes

#### **B = Software missing**

Control program not available

#### **I = Context**

LK missing

LK and TK missing

#### **K = Key error**

The cross-reference list in Appendix A shows the possible status codes for each command (see "Appendix A").

## Key test

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	H	Command specification: 'Key handling'	(ASCII)
1	C	Additional specification: Key test	(ASCII)
8	xx .. xxH	Test pattern (00000000)	(HEXA)
	00 H	any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	H	Command specification: 'Key handling'	(ASCII)
1	C	Additional specification: Key test	(ASCII)
1	a	Status code	(ASCII)
	00 H	any filler bytes	(HEXA)

### Description

For a test of the line keys of both partners, a 8-byte pattern (zeros) is encrypted in the HOST and sent to the CMD-V4.

This character string is decrypted and checked by the CMD-V4.

In case of correct line keys, this results in the initial pattern.

If this is not the case, a synchronization with the HOST is compulsory.

**Status Codes****B = Software missing**

Control program not available

**I = Context**

AZM does not operate with encryption

**K = Key error**

Key test negative, initial value must be requested

Wrong test pattern

The cross-reference list in Appendix A shows the possible status codes for each command (see “Appendix A”).

## DQC – read error stack

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Q	Command specification: Internal commands	(ASCII)
1	C	Additional specification: Read error stack	(ASCII)
	00 H	Any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Q	Command specification: Internal commands	(ASCII)
1	C	Additional specification: Read error stack	(ASCII)
1	a	Status code	(ASCII)
n	aa..aa	<b>Data</b> (see Appendix V (page 10))	(ASCII)
	00 H	Any filler bytes	(HEXA)

### Description

The error stack saves information about the most recent 10 errors. If a new error is added, the oldest error is deleted. Information about an error (see "Appendix V (page 10)") is output via the command DQC. Starting with the most recent error, information about the next, oldest error is output with each further request. Once the end of the error stack has been reached (the oldest error), the next command DQC is transferred without data. This displays that the end of the error stack has been reached and that the first error entry is displayed again with the next command. Therefore, 11 DQC commands are required (10x error entry and 1x end entry) to display the entire error stack.

With a deleted error stack or not required entries, the entries only consist of filler bytes and the end separator. The size of an error stack page can be max. 2 Kbytes.

## Status Codes

### **B = Software missing**

Control program not available

**Appendix A** (page 88) shows the possible status codes for each command in a cross-reference list.

**Appendix X** (page 155) shows the possible device and cassette states for each command in a cross-reference list.

## DQE – read error information

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Q	Command specification: Internal commands	(ASCII)
1	E	Additional specification: Read error info	(ASCII)
1	;	End character	(ASCII)
	00 H	any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Q	Command specification: Internal commands	(ASCII)
1	E	Additional specification: Read error info	(ASCII)
1	a	Status code	(ASCII)
n	bb .. bb	Device info (see table)	(BINARY)
	00 H	any filler bytes	(HEXA)

### Description

Returns all the data that are relevant for error analysis.

### Status Codes

#### **B = Software missing**

Control program not available

**Appendix A** (page 88) shows the possible status codes for each command in a cross-reference list.

**Appendix X** (page 155) shows the possible device and cassette states for each command in a cross-reference list.



## Structure

char modId[] = "DIL=\$FRM\$ 010101 1000 CMD_BOOT.BIN: \$MOD\$ 010101 1000 CMD_V4_0.BIN;"
char ded[]="DED="
char len[4] ;
char sSw[1] ;
char dLoc[1] ;
char cas[1] ;
char shErr[1];
char shut[1];
char mon[1] ;
char ter[1] ;
char ts[2] ;
char tf[1] ;
char sr[1] ;
char dis[1] ;
char cex[1] ;
char door[1] ;
char doS[1] ;
char type[1] ;
char nVm[1] ;
char sedm[1] ;
char lCmd[2] ;
char lSta[1] ;
char sCle[1] ;
char sRes[1] ;
char tst[1] ;
char sCod[2];
char or[1];
char transport[1] ;
char cOut[1] ;
char shutProtect[1] ;
char error[16];
char warning[16];
char semicolon[1] ;
char dec[]="DEC="

char len[4];
char rSta[1];
char rAct[4];
char rRet[2];
struct Cassette
{ // the following structure is repeated 8 times (for all cassettes supported)
char sta[1];
char num[7];
char cur[3];
char rel[4];
char val[8];
char len[3];
char tol[2];
char act[4];
char ndv[4];
char low[4];
char ld[2];
char rej[3];
} cassettes[Dispenser ::MAX_CASSETTES];
char hall[]="HALL="
{ // the following structure is repeated 4 times
char handle[2]
char state[2]
}
char semicolon
char hall[]="DPHOTO="
{ // the following structure is repeated n times, up to the semi-colon
char handle[2]
char state[2]
}
char semicolon[1]
char hall[]="PRESS="
{ // the following structure is repeated 8 times
char handle[2]
char pressure[2]
char current[2]

char workPressure[2]
char maxPressure[2]
}
char semicolon[1]
char hall[]="DDU="
char adValue1[1]
char adValue2[1]
char current1[1]
char current2[1]
char semicolon
char photo[]="PHOTO="
{ // the following structure is repeated n times, up to the semi-colon
char name[4]
short iActual
short iWork
short iReference
short changeFactor
char Regulation
char dirtState (MISSING = '0', OK = '1', PROD_WEAK = '2', TKD_WEAK = '3', DIRTY = '4', DEFECT = '5', UNKNOWN = '6')
short value
char state (FREE=0, CLOSED=1, UNDEFINED=2)
}
char semicolon[1]
char dek[]="DEK="
char sdoBad[1]
char maxNotes[2]
char noSr[1]
char semicolon[1]
char tCount[]="TCOUNT="
char transactions[2]
char semicolon[1]
char hws[]="DYTHWS="
char hwsData[]="....."
Data up to semi-colon

## DQF – read stacker-EEPROM

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Q	Command specification: Internal commands	(ASCII)
1	F	Additional specification: read stacker EEPROM	(ASCII)
	00 H	any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Q	Command specification: Internal commands	(ASCII)
1	E	Additional specification: Read error info	(ASCII)
1	a	Status code	(ASCII)
76	bb .. bb	Device info (see table 'Describe stacker-EEPROM')	(BINARY)
	00 H	any filler bytes	(HEXA)

### Description

The data of the stacker EEPROM are displayed.  
This command is only available for reasons of compatibility.

### Status Codes

#### **B = Software missing**

Control program not available

#### **E = Device error**

Stacker-EEPROM unreadable

**Appendix A** (page 88) shows the possible status codes for each command in a cross-reference list.

**Appendix X** (page 155) shows the possible device and cassette states for each command in a cross-reference list.

## DQI - Describe stacker-EEPROM

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Q	Command specification: Internal commands	(ASCII)
1	I	Additional specification: describe stacker EEPROM	(ASCII)
76	bb .. bb	Device info (see table)	(BINARY)
	00 H	any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Q	Command specification: Internal commands	(ASCII)
1	I	Additional specification: Read error info	(ASCII)
1	a	Status code	(ASCII)
	00 H	any filler bytes	(HEXA)

### Description

The data of the stacker EEPROM are written for the whole area.

The individual elements cannot be addressed via code words. The device info data are not checked for validity by the firmware.

### Data structure in the stacker-EEPROM

The EEPROM is organized in 64 \* 16Bit ( word )

Position of the word	Contents	Meaning
0	0x0002	Version number
1	0x5557	Checksum ( = version number ^ 0x5555 )
2 -24	X...X	Reserved for initial start-up (e.g. firmware filename, serial number, Singlereject, gradual correction routing disk, etc.)
25 – 26	XXXX	Reserved for development of measuring and test equipment
27 – 32	,XXXXXXXXXXXX	Part number ( "," plus 11 bytes)
33 – 37	,XXXXXXXXXXXX	BAUZ/revision level ( "," plus 9 bytes)
...		
...		
...		
63		

“,” = separator, “.” = end character, X = Byte

The positions 2 - 24 are reserved for the initial start-up of the device and are therefore initialized with the corresponding data.

The positions 25 - 26 are reserved for the development of measuring and test equipment.

The positions 27 - 37 are reserved for the part number and the revision level.



The different positions may only be overwritten by their owners.  
The positions 0 – 24 are partly evaluated by previous firmware levels of the CMD-V4 (Release prior to 16.00 ).

General example:

, 01750107630,000001976  
separator part number separator revision level

Structure revision level:

For each revision level one bit is set in the memory and the information is returned as ASCII code via the interface.

Identification sign:

26	25	24	23	22	21	20	19	18	17	16	15	14
13	12	11	10	9	8	7	6	5	4	3	2	1

26	25	..	..	..	13	12	11	10	9	8	7	6	5	4	3	2	1	Position of identification sign
0	0	..	..	..	0	0	0	0	1	1	0	0	1	0	1	1	1	Value
0	..				0	1				9				7				Hexadecimal Value
30	..				30	31				39				37				ASCII value

## Status Codes

### **B = Software missing**

Control program not available

### **C = Syntax**

invalid length

### **E = Device error**

Stacker-EEPROM not writeable

**Appendix A** (page 88) shows the possible status codes for each command in a cross-reference list.

**Appendix X** (page 155) shows the possible device and cassette states for each command in a cross-reference list.

## DQJ - read manufacturer's and controller identification

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Q	Command specification: Internal commands	(ASCII)
1	J	Additional specification: read manufacturer's and controller identification	(ASCII)
	00 H	Any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Q	Command specification: Internal commands	(ASCII)
1	J	Additional specification: read manufacturer's and controller identification	(ASCII)
1	a	Status code	(ASCII)
4	a ... a	Manufacturer's ID corresponding to the type plate definitions * = not initialized or checksum error	(ASCII)
26	a ...a	Reserve	(ASCII)
1	,	Separator	(ASCII)
2	aa	Controller ID e.g. 01, 02 or 03 = Intel controller 80 = Reproduction of Innovasic controller	(ASCII)
1	;	End character	(ASCII)
	00 H	Any filler bytes	(HEXA)

### Description

This command returns the manufacturer's and controller identification. The controller ID is read from the controller PCB register STEPID. The manufacturer's ID is stored in the memory contents of the writeable 32 byte area between the addresses E800:7FE0 and



E800:7FFF. The memory contents are to be written according to the manufacturer's ID during manufacturing of the controller board.

## Structure of the manufacturer's ID

Address position	Type	Meaning
0xEFFE0	byte [1]	Checksum: sum ( all 32 bytes ) = 0
0xEFFE1	byte [1]	01 = Version 1: is assigned by Wincor Nixdorf
0xEFFE2 - 0xEFFE5	char [4]	Manufacturer's ID corresponding to the type plate definition. Identification of the production plant by the 2- to 4-digit ID before the serial number. In case of the 2- or 3-digit serial number, the leading digits must be filled with 0. e.g. 0050, 0051 or 0053 = Wincor-Nixdorf Paderborn
0xEFFE6 - 0xEFFF	byte [26]	0... 0 = Reserve filler byte

## Status Codes

### **B = Software missing**

Control program not available

### **C = Syntax**

invalid length

**Appendix A** (page 88) shows the possible status codes for each command in a cross-reference list.

**Appendix X** (page 155) shows the possible device and cassette states for each command in a cross-reference list.

## GetAccessCode

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	S	Command specification: 'SPECIAL'	(ASCII)
1	C	Additional specification: GetAccessCode	(ASCII)
	00 H	any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	S	Command specification: 'SPECIAL'	(ASCII)
1	C	Additional specification: GetAccessCode	(ASCII)
1	a	Status code	(ASCII)
8	xx...xxH	Random number	(HEXA)
	00 H	any filler bytes	(HEXA)

### Description

In the CMD-V4, an 8 byte random number is generated, transferred within the acknowledgement, and **<ACCESS\_CODE>=I** set. The random number must be requested anew prior to each 'SetAccessCode' command. Transmission of the acknowledgement is delayed after a faulty activation procedure. A delay time of 5 seconds is fixed and with every further failed activation procedure the delay time will extend by the fixed preset value.

### Status codes

#### **B = Software missing**

Control program not available

#### **I = Context**

no PC8xxx system

<TRANSPORT>

unequal E, F, G

## SetAccessCode

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	S	Command specification: 'SPECIAL'	(ASCII)
1	D	Additional specification: SetAccessCode	(ASCII)
8	xx...xxH	Access code	(HEXA)
	00 H	any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	S	Command specification: 'SPECIAL'	(ASCII)
1	D	Additional specification: SetAccessCode	(ASCII)
1	a	Status code	(ASCII)
	00 H	any filler bytes	(HEXA)

### Description

The transmitted AccessCode is checked by the firmware. The random number must be requested anew prior to each 'SetAccessCode' command. In case of correct check, **<ACCESS\_CODE>=Y** is set and the number of failed attempts cleared, if not, the **<ACCESS\_CODE>=I** and the number of failed attempts is incremented.

### Status codes

#### B = Software missing

Control program not available

#### I = Context

no PC8xxx system

Random number not requested

<TRANSPORT>

unequal E, F, G

**K = Key error**

Key test negative, a new random number must be requested

## DYR – read internal statistics

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Y	Command specification: Internal commands	(ASCII)
1	R	Additional specification: Read statistics	(ASCII)
	00 H	any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Y	Command specification: Internal commands	(ASCII)
1	R	Additional specification: Read statistics	(ASCII)
1	a	Status code	(ASCII)
n	a	Statistics data (see the following table)	(ASCII)
	00 H	any filler bytes	(HEXA)

## Structure of the statistics data

(persistent = not deletable)

All long values are listed in the sequence LSB to MSB.

Byte position	Cell	Meaning	
0	yearOfBirth	Year persistent statistics commenced	(BCD)
1	monthOfBirth	Month	(BCD)
2	dateOfBirth	Day	(BCD)
3	hourOfBirth	Hour	(BCD)
4	minuteOfBirth	Minute	(BCD)
5	secondOfBirth	Second	(BCD)
6 – 9 (6=lsb, 9=msb)	persistentNoteCounter	persistent note counter at DDU	unsigned long (
10 – 13	PersistentTransactionCounter	persistent counter 'Clamp reaching shutter'	unsigned long (
14	yearOfReset	Year persistent statistics commenced	(BCD)
15	monthOfReset	Month	(BCD)
16	dateOfReset	Day	(BCD)
17	hourOfReset	Hour	(BCD)
18	minuteOfReset	Minute	(BCD)
19	secondOfReset	Second	(BCD)
20 – 23 (20=lsb, 23=msb)	resettableNoteCounter	resettable note counter at DDU	unsigned long (
24 – 27	resettableTransactionCounter	resettable counter 'Clamp reaches shutter'	unsigned long (
28 – 31	bundleRejectCounter[0]	resettable counter 'Cassette 1 causes bundle reject'	unsigned long (
32 – 35	bundleRejectCounter[1]	resettable counter 'Cassette 2 causes bundle reject'	unsigned long (
36 – 39	bundleRejectCounter[2]	resettable counter 'Cassette 3 causes bundle reject'	unsigned long (
40 – 43	bundleRejectCounter[3]	resettable counter 'Cassette 4 causes bundle reject'	unsigned long (
44 – 47	bundleRejectCounter[4]	resettable counter 'Cassette 5 causes bundle reject'	unsigned long (

Byte position	Cell	Meaning	
48 – 51	bundleReject Counter[5]	resettable counter 'Cassette 6 causes bundle reject'	unsigned long (
52 – 55	bundleReject Counter[6]	resettable counter 'Cassette 7 causes bundle reject'	unsigned long (
56 – 59	bundleReject Counter[7]	resettable counter 'Cassette 8 causes bundle reject'	unsigned long (
60 – 63	SingleReject Counter[0]	resettable counter 'Cassette 1 causes single reject'	unsigned long (
64 – 67	SingleReject Counter[1]	resettable counter 'Cassette 2 causes single reject'	unsigned long (
68 – 71	SingleReject Counter[2]	resettable counter 'Cassette 3 causes single reject'	unsigned long (
72 – 75	SingleReject Counter[3]	resettable counter 'Cassette 4 causes single reject'	unsigned long (
76 – 79	SingleReject Counter[4]	resettable counter 'Cassette 5 causes single reject'	unsigned long (
80 – 83	SingleReject Counter[5]	resettable counter 'Cassette 6 causes single reject'	unsigned long (
84 – 87	SingleReject Counter[6]	resettable counter 'Cassette 7 causes single reject'	unsigned long (
88 – 91	SingleReject Counter[7]	resettable counter 'Cassette 8 causes single reject'	unsigned long (
92 – 93	jamLocks	resettable counter 'Flag detected'	unsigned short (

## Description

The current note counters (32 bit wide) are delivered.

All cells are stored in the controller's static, write-protected RAM (the so-called CMOS). The cells are re-created if the CMOS is found to be invalid (via checksum + version).

This is the case:

- In first use (production)
- When the battery jumper has been removed for long periods ( > 5min )
- After initiating a CMOS reset by jumper 'CL CMOS'
- After deleting the CMOS per DXRAL command
- If the CMOS is corrupt (e.g. electrical faults, firmware error, etc.)

Following contents are stored:

- 2 transaction counters (1\*permanent, 1\*resettable):  
If the clamp reaches the shutter without an error (Hall sensor 4), the counter is incremented by 1.
- 2 note counters (1\*permanent, 1\*resettable):  
With every note recognized by the thickness measurement station, the counter is incremented by 1.  
  
When the note is detected to be double, the counter is incremented by 2.
- 16 Reject counter (all resettable):  
There is one counter for bundle rejects and one for single rejects at each cassette location, i.e. 8 cassettes \* 2 counters = 16 entries  
If a dispense transaction has to be canceled and a bundle reject takes place, the bundle reject counter of the cassette that is currently in use (which is normally responsible for cancellation) is incremented by 1.  
  
The single-reject counter of the cassette where the note is from is incremented by 1 for each single-rejected note.  
When the note is detected to be double, the counter is incremented by 2.
- 2 \* date/time:  
One date/time on the real-time clock is maintained for the permanent counters and for the resettable counters.

All counters are 32 bit wide, hence they can count  $2^{32}$  events. (Exception 'jamLocks')  
All resettable counters are reset to 0 with key function 5.  
At the same time, the date/time for these counters is updated.  
For performance reasons, counter data is currently saved with a slight delay, so that power failures at unsuitable moments may corrupt the counters.  
The date ID notes when the counters were deleted.



Double notes count as 2 notes.

## Status Codes

### **B = Software missing**

Control program not available

**Appendix A** (page 88) shows the possible status codes for each command in a cross-reference list.

**Appendix X** (page 155) shows the possible device and cassette states for each command in a cross-reference list.



## DYS – Resetting internal statistics

### Command

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Y	Command specification: Internal commands	(ASCII)
1	S	Additional specification: Reset statistics	(ASCII)
	00 H	any filler bytes	(HEXA)

### Acknowledgement

Length	Contents	Meaning	
1	D	Device ID	(ASCII)
1	Y	Command specification: Internal commands	(ASCII)
1	S	Additional specification: Reset statistics	(ASCII)
1	a	Status code	(ASCII)
	00 H	any filler bytes	(HEXA)

### Description

This command resets the resettable counters and the date/time for these counters is simultaneously updated for these counters. See description “DYR - read internal statistics”.

### Status Codes

#### **B = Software missing**

Control program not available

## Appendix A: Cross reference

### Command/status

CMD	Reference	Meaning	A / Q – OK	B – SW missing	C – Syntax	D / T – Out of service	E / U – Device Error	F – Safe not open	G – Time out	H – Withdrawal	I – Context	K – Key-Error	S – SCOP-Error
Dabblank	Page: 35	Reset / Power up	X	X	X	X						X	
DAB	Page: 40	Clearing	X	X	X	X						X	
DBS	Page: <b>Fehler! Textmar ke nicht definiert. Fehler! Textmar ke nicht definiert.</b>	Dispense	X	X	X	X	X				X	X	
DBC	Page: <b>Fehler! Textmar ke nicht definiert. Fehler! Textmar ke nicht definiert.</b>	DispenseAdditional	X	X	X	X	X				X	X	
DCA	[L 1]	Transport: Wait for cash removal	X	X	X	X	X		X		X	X	
DCB	[L 1]	Transport: Bundle reject	X	X	X	X	X				X	X	
DCC	[L 1]	Transport: Close shutter	X	X	X	X	X				X	X	

CMD	Reference	Meaning	A / Q – OK	B – SW missing	C – Syntax	D / T – Out of service	E / U – Device Error	F – Safe not open	G – Time out	H – Withdrawal	I – Context	K – Key-Error	S – SCOP-Error
DCE	[L 1]	Transport: Output cash	X	X	X							X	
DCO	[L 1]	Transport: Open shutter	X	X	X	X	X				X	X	
DCP	[L 1]	Transport: Offer cash	X	X	X	X	X		X		X	X	
DCQ	[L 1]	Transport: Cash retract without storage	X	X	X	X	X			X	X	X	
DCR	[L 1]	Transport: Cash retract with storage	X	X	X	X	X			X	X	X	
DCT	Page: 57	Transport: Shutter – Test		X	X							X	

DCW	[L 1]	Transport: Target = Stop Over	X	X	X	X	X				X	X	
DDblank	[L 1]	Cancel	X	X	X						X	X	
DEC	[L 1]	Cassette status	X	X	X							X	
DED	[L 1]	Device status	X	X	X							X	
DEF	[L 3]	Features	X	X	X							X	
DEK	Page: 57	Read firmware configuration	X	X									
DES	[L 2]	Read SCOP status	X	X									
DFA	[L 1]	Confirm number of notes	X	X	X						X	X	
DFB	[L 1]	Confirm no. of notes (without errors)	X	X	X						X	X	
DFC	[L 1]	Confirm existing cassettes	X	X	X						X	X	

CMD	Reference	Meaning	A / Q – OK	B – SW missing	C – Syntax	D / T – Out of service	E / U – Device Error	F – Safe not open	G – Time out	H – Withdrawal	I – Context	K – Key-Error	S – SCOP-Error
DFD	[L 1]	Note parameter II	X	X	X						X	X	
DFG	Page: 61	Write firmware configuration	X	X	X						X	X	
DFI	[L 1]	Cassette ID (Standard)	X	X	X						X	X	
DFL	[L 1]	Set minimum quantity	X	X	X						X	X	
DFN	[L 1]	Number of notes	X	X	X						X	X	
DFO	[L 1]	Number of notes (without check)	X	X	X						X	X	
DFP	[L 1]	Note parameter I	X	X	X						X	X	
DFR	[L 1]	Reset lock flag	X	X	X	X					X	X	
DFT	[L 1]	Delete retract counter	X	X	X						X	X	
DFX	[L 1]	Write value into cassette	X	X	X						X	X	
DFY	[L 1]	Read value from cassette	X	X	X						X	X	
DFZ	[L 1]	'Power on' - RESET without PUT	X	X	X						X	X	
DGblank	Page: 54	Determine reference value	X	X	X	X	X				X	X	
DHC	Page: 66	Key test	X	X	X						X	X	
DHI	Page: 65	Request initial value	X	X	X						X	X	
DHL	Page: 63	Load line code	X	X	X						X	X	
DHO	Page: 63	Switch encryption off	X	X	X			X			X	X	
DHT	Page: 63	Load transport key	X	X	X			X				X	
DHZ	[L 2]	Transparent ChipCard access	X	X	X		X				X		

CMD	Reference	Meaning	A / Q – OK	B – SW missing	C – Syntax	D / T – Out of service	E / U – Device Error	F – Safe not open	G – Time out	H – Withdrawal	I – Context	K – Key-Error	S – SCOP-Error
DIC	[L 1]	Software transfer: Activate booter	X		X								
DID	[L 1]	Software transfer: Data	X	X	X						X		
DIH	[L 2]	Software transfer: Firmware 'HASH'	X		X		X						X
DIL	[L 1]	Software transfer: Read firmware ID	X		X						X		
DQA	[L 1]	Set the real-time clock	X	X	X								
DQB	[L 1]	Read the real-time clock	X	X	X								
DQC	Page: 68	Read error stack	X	X	X								
DQE	Page: Fehler! Textmarke nicht definiert. Fehler! Textmarke nicht definiert.	Internal commands: Read error information	X	X	X								
DQF	Page: 74	Internal commands: read EE-PROM	X	X	X		X						
DQI	Page: 75	Internal commands: Write EEPROM	X	X	X		X						
DQJ	Page: 78	Internal commands: Read out manufacturer/controller ID	X	X	X		X						
DQR	[L 8]	Internal commands: Read Recovery Info	X	X	X								

<b>CMD</b>	<b>Reference</b>	<b>Meaning</b>	<b>A / Q – OK</b>	<b>B – SW missing</b>	<b>C – Syntax</b>	<b>D / T – Out of service</b>	<b>E / U – Device Error</b>	<b>F – Safe not open</b>	<b>G – Time out</b>	<b>H – Withdrawal</b>	<b>I – Context</b>	<b>K – Key-Error</b>	<b>S – SCOP-Error</b>
DSC	Page: 80	Read access code	X	X	X						X		
DSD	Page: 81	Set access code	X	X	X						X	X	
Dxa	[L 1]	Self-test commands	X	X	X						X		
Dya	[L 1]	Test commands	X	X	X						X		
DYR	Page: 80	Test commands: Read internal statistics	X	X	X								
DYS	Page: 87	Test commands: Reset internal statistics	X	X	X								

## Booter commands

The following commands are processed in the booter:

Dix	Software transfer
DEF	Read properties

## Execution times

All times stated below do not include the ISO wait of max. 3 \* 8 seconds.

<b>CMD</b>	<b>Meaning</b>	<b>304.8 mm (12")</b>
Dablank	RESET	60 s
DAB	Clearing	60 s

CMD	Meaning	304.8 mm (12")	
DBC	Additional dispense	60 s	1 note from 1st cass.
DBS	Dispensing: standard	60 s	1 note from 1st cass.
DCA	Transport: Wait for cash removal	15 s	+ waiting time
DCB	Transport: Bundle reject	20 s	
DCC	Transport: Close shutter	20 s	
DCE	Transport: Output cash	20 s	+ waiting time
DCO	Transport: Open shutter	20 s	
DCP	Transport: Offer cash	20 s	
DCQ	Transport: Cash retract without storage	25 s	
DCR	Transport: Cash retract with storage	25 s	
DCT	Transport: Shutter test	20 s	
DCW	Transport: Target = Stop Over	20 s	
DDblank	Cancel	15 s	
DEC	Cassette status	15 s	
DED	Device status	15 s	
DEF	Features	15 s	
DEK	Read firmware configuration	15 s	
DES	Read SCOP status	15 s	
DFA	Confirm number of notes	15 s	
DFB	Confirm no. of notes (without errors)	15 s	
DFC	Confirm existing cassettes	15 s	
DFD	Note parameter II	15 s	
DFG	Write firmware configuration	15 s	
DFI	Cassette ID (Standard)	15 s	
DFL	Set minimum quantity	15 s	
DFN	Number of notes	15 s	
DFO	Number of notes (without check)	15 s	
DFP	Note parameter I	15 s	

<b>CMD</b>	<b>Meaning</b>	<b>304.8 mm (12")</b>
DFR	Reset lock flag	15 s
DFT	Delete retract counter	15 s
DFX	Write value into cassette	15 s
DFY	Read value from cassette	15 s
DFZ	'Power on' - RESET without PUT	15 s
DGblank	Determine reference value	60 s
DHC	Key test	15 s
DHI	Request initial value	15 s
DHL	Load line code	15 s
DHO	Switch encryption off	15 s
DHT	Load transport key	15 s
DHZ	Transparent ChipCard access	15 s
DIC	Software transfer: Activate booter	15 s
DID	Software transfer: Data	15 s
DIH	Software transfer: Firmware 'HASH'	180 s
DIL	Software transfer: Read firmware ID	240 s
DQA	Set the real-time clock	15 s
DQB	Read the real-time clock	15 s
DQC	Internal commands: Read error stack	15 s
DQE	Internal commands: Read error information	15 s
DQF	Internal commands: Read EE-PROM	15 s
DQI	Internal commands: Write EEPROM	15 s
DQJ	Internal commands: Read out manufacturer and controller ID	15 s
DQR	Internal commands: Read Recovery information	15 s
DSC	Read Access code	60 s
DSD	Set access code	15 s



<b>CMD</b>	<b>Meaning</b>	<b>304.8 mm (12")</b>
Dxa	Self-test commands	xx s
Dya	Internal commands	xx s
DYR	Internal commands: Read internal statistics	15 s
DYS	Internal commands: Reset internal statistics	15 s

## Appendix B: Status display codes

### Displays in normal operation

<b>No.</b>	<b>Meaning</b>	<b>Reaction</b>
off	No power supply	Switching the device on
- -	Boot up (after RESET or electronic self-test)	
bu	burn; program controller	
C1	compare; check controller	
IO	Controller OK	
00	No error	
01	Software faulty	If possible, read hardware status, switch device off/on, read hardware status (field stop info), download if necessary
05	Communication problem with ChipCard controller	Check/replace lines to the ChipCard controller, ChipCard controller
06	Communication problem with ChipCard / or wrong ChipCard inserted (ChipCard number not '226-0PR')	Check/replace ChipCard on the controller
09	Locking handle open	Push dispenser completely into the safe, if necessary adjust switch
10	Controller defective or	replace controller

No.	Meaning	Reaction
10	Battery empty or battery jumper not connected: only with self-test commands for battery testing	Mount battery jumper or replace battery
11	Software is missing (download)	Perform download
12	Safety switch open	Insert CMD-V4
13	Device lock present	Remove notes from CMD-V4; then per SW: <b>RESET lock flag</b>
14	Minimum configuration of CMD-V4 not available (RR-box + at least 1 cash-out cassette)	Insert cassettes; if necessary, determine reference values or cassette defective
15	Cassette board has invalid coding, is not inserted or defective	Replace cassette distributor board, check cables
16	Output transport coding invalid	Perform coding on output transport cable connector.
17	Belt drive of clamp defective <b>M3</b> or jam	Press the function button on the CMD controller until '01' appears on the display. Call service personnel if the error is still displayed after the RESET.
18	Banknote jam during dispensing	Clear paper jam; then perform a RESET.
19	Clamp transport defective/blocked  <b>VCMD:</b> Note jam in output transport	Press the function button on the CMD controller until '01' appears on the display. Call service personnel if the error is still displayed after the RESET.
20	Single reject switch defective/blocked <b>MA2/1, MA2/2, SM1</b>	Remove block; then RESET; if nec. replace stacker or controller
21	Faulty measuring station <b>DDU</b>	Press the function button on the CMD controller until '01' appears on the display. Call service personnel if the error is still displayed after the RESET.
22	Photosensor amplifier faulty or photosensor initialization not successful	Clear cash paths; then perform a RESET or replace controller.

No.	Meaning	Reaction
23	Routing disk defective/blocked <b>SM2/SM3</b>  <b>VCMD:</b> Note bundle reject switch defective/blocked <b>SM6</b>	Press the function button on the CMD controller until '01' appears on the display. Call service personnel if the error is still displayed after the RESET.
24	Reject/reject drive defective/blocked <b>MA6</b>	If necessary, replace cassette, output transport or controller.
25	Dispensing motor <b>DCM1</b> faulty	Replace output transport or controller.
26	Stacker wheel drive defective/blocked <b>SM9</b>  <b>VCMD:</b> Comb-type lifting bar defective/blocked <b>SM5</b> Switching of dispensing/transport defective/blocked <b>M3</b>	Remove block; then RESET; if nec. replace stacker or controller
28	Shutter error	Eliminate error; then perform a RESET.
29	Cash-out photosensor covered (manipulation)	Eliminate error; then perform a RESET.
3x	Too many problems during dispensing from cassette x (x = 1 – 6)	Check cash paths; then perform a RESET; if nec. dispense test note(s)
4x	note dispensing from cassette x not possible (x = 1 - 6); possibly faulty note contact pressure with new cassette	Check the contents of cassette x; RESET
5x	too many bad notes from cassette x (x = 1 - 6)	Check the contents of cassette x; RESET
6x	Cassette x defective (x = 1 - 6)	Write error (cassette EEPROM) Replace cassette x, RESET; or contact pressure cannot be built up, insert CMD-V4.

## Contaminated photosensors/sensors

No.	Meaning	Designation
7x	Dispensing sensor of cassette x (x = 1 - 6) or pressure sensor defective	PSDx
8x	Empty sensor of cassette x (x = 1 - 6)	PSEx
90	Stacker wheel input	PS 1
91	Tray monitoring	PS 18
93	'Tray empty' (single reject) <b>VCMD:</b> Monitoring of entrance reject box	PS 2 PS21
95	Removal photosensor <b>VCMD:</b>	PS 27 PS 26
9A	Bundle rear edge control, shutter sensor	PS 28

**Reaction:** Clean the relevant photosensor / measurement station.  
Then check the CMD-V4's operability by performing a RESET.

Appendix D – Electrical & mechanical components describes the positions of the components:

**Outdoor** (page 104)

**Indoor** (page 104)

## Photosensors covered during initialization

No.	No.	Meaning	Designation
<b>T</b>	<b>T + 1</b>	<b>Blink alternately, repeated several times</b>	
L	.Ex	Empty sensor of cassette x (x = 1 - 2)	PSEx
L	Ax	Dispensing sensor of cassette x (x = 1 - 2)	PSDx
L	x	Photosensor x (see contaminated Photosensors/Sensors)	PS x

## Displays when function button is pressed

No.	Meaning
0	Normal operation
1	Reset
2	Overall test + 'RESET'
3	Trial dispensing process + RESET (can also be configured without trial dispensing process)
4	Display firmware release number
5	Reset statistics (the internal resettable counters are set to 0 and the date stored)
6	In case of an open door switch and DES_CLEAR_BUTTON=Y the encryption is deactivated.

When the function key is pressed, the available function range is output to the status display in a loop.

See also chapter "Function pushbuttons" (page 16)

## Booter outputs to the 7-segment display

T		T + 1		in temporal alternation
High	Low	High	Low	
-	-			Booter indicates power-up and test of loaded FW
c	l			Indicates CMOS deletion via jumper
b	E			Battery empty or battery jumper not connected
i	o			Display following correct CMOS clearing
1	1			Booter operating without error
.1	1			Alternate booter operating without error
.1	1	1	1	FW download (decimal points blinking)
1	.1	1	1	FW download (decimal points blinking)
Error without unconditional break				
1	1	0	1	Booter CMOS invalid (inactive in booter releases from 15.11.02)
1	1	0	2	SCOP-CMOS invalid (inactive in booter releases from 15.11.02)
1	1	0	3	Stacker EEPROM incorrectly written
1	1	0	4	FW length or CRC not OK
1	1	0	5	SCOP-Line error

1	1	0	6	AVT error
1	1	0	7	SCOP-FW not Valid
1	1	0	8	FW name in stacker EEPROM not OK
1	1	0	9	Error writing or deleting flash
1	1	1	2	Program memory cells lose content
1	1	1	3	SCOP jumper changed without reboot
Error with unconditional break				
0	1			Booter defective( interrupt or NMI)
0	1	-	2	Writing to wrong CMOS area
1	0	b	t	Booter defective (Flash error)
1	0	E	E	No stacker EEPROM connected
8	8	8	8	Undefined interrupt occurred
n		N		NMI

## Appendix D: Electrical and mechanical components

### Electrical elements

#### Stacker and output transport

Component	Designation	Location
D	Permanent magnets (3 units) – without sensors	Clamp
DPS 1 / 3	<b>VCMD:</b> Dispensing/transport switching sensor	Stacker input
DPS 2	Stacker wheel position query	Stacker wheel input
DPS 4	Sensor: position of retract compartment <b>VCMD:</b> Dispensing/transport switching	RR box Stacker wheel input
DPS 5	Sensor: home position	Routing disk drive, left
DPS 6	Sensor: home position	Routing disk drive, right
DPS 7	Home position of sliding surface	Single reject
DPS 8	Sensor: Position	Routing disk drive, left
DPS 9	Sensor: Position	Routing disk drive, right
DPS 8 / 9	<b>VCMD:</b> Sensor of comb-type lifting bar	Above reject box
DPS 10	Hybrid photosensor	Shutter
DPS 11	Hybrid photosensor	Shutter
DPS 14	Home position of switch	Single reject
DPS 15	Switch position: single reject	Single reject
HS 1	(Hall sensor) home position	Clamp positioning
HS 2	(Hall sensor) Initial RR position (behind the routing disk)	Clamp positioning
HS 3	(Hall sensor) Reversal in FL direction (if required.)	Clamp positioning
HS 4	(Hall sensor), 'Stop over' position and clamp final position	Clamp positioning
PS 1	Stacker wheel input	Stacker wheel input
PS 2	'Tray empty' check	Single reject
PS 18	Tray monitoring/ input monitoring 'clamp empty'	Swivel drive for clamp flap/RR box
PS 21	<b>VCMD:</b> Monitoring of entrance reject box	RR box

Component	Designation	Location
PS 26	<b>VCMD:</b> Removal photosensor	Output transport
PS 27	Removal photosensor	Clamp positioning
PS 28	Bundle rear edge control	Clamp positioning
DCM 1	Main drive motor	Stacker wheel input
DCM 2	Traction motor + planetary gears	Clamp
DCM 3	Belt drive + planetary gears <b>VCMD:</b> Dispensing/transport switching	Clamp Stacker wheel input
DCM 7	DC motor (outdoor)	Shutter
MA 2	Switch control magnet	Single reject
MA 6	Magnet for retract compartment	Reject box
SW 1	RR box inserted	Reject box
SW 2	Safety switch <b>VCMD:</b> Safety switch stacker cover	Slide rack Cover plate in front of the bundle reject switch
SW 3	Locking switch <b>VCMD:</b> second function key	Above reject box at the control panel on the chassis
SW 4	<b>VCMD:</b> Safety switch	Slide rack
SM 1	Pressure on sliding surface for emptying	Single reject
SM 2	Routing disk motor	Routing disk drive, left
SM 3	Routing disk motor	Routing disk drive, right
SM 5	<b>VCMD:</b> Comb-type lifting bar	Stacker wheel input
SM 6	<b>VCMD:</b> Bundle reject deflector	Above reject box
SM 7	Stepper motor (indoor)	Shutter
SM 9	Stacker wheel stepper motor	Stacker wheel input
CG 1	Timing disk	on main motor DCM1

Note: The switch SW2 switches off the power elements for CMD-V4 and VCMD via the electronics. This is monitored by the firmware and displayed via indicator 12. As soon as the switch has been switched back the original position, the power elements are switched on and the firmware triggers a reset.

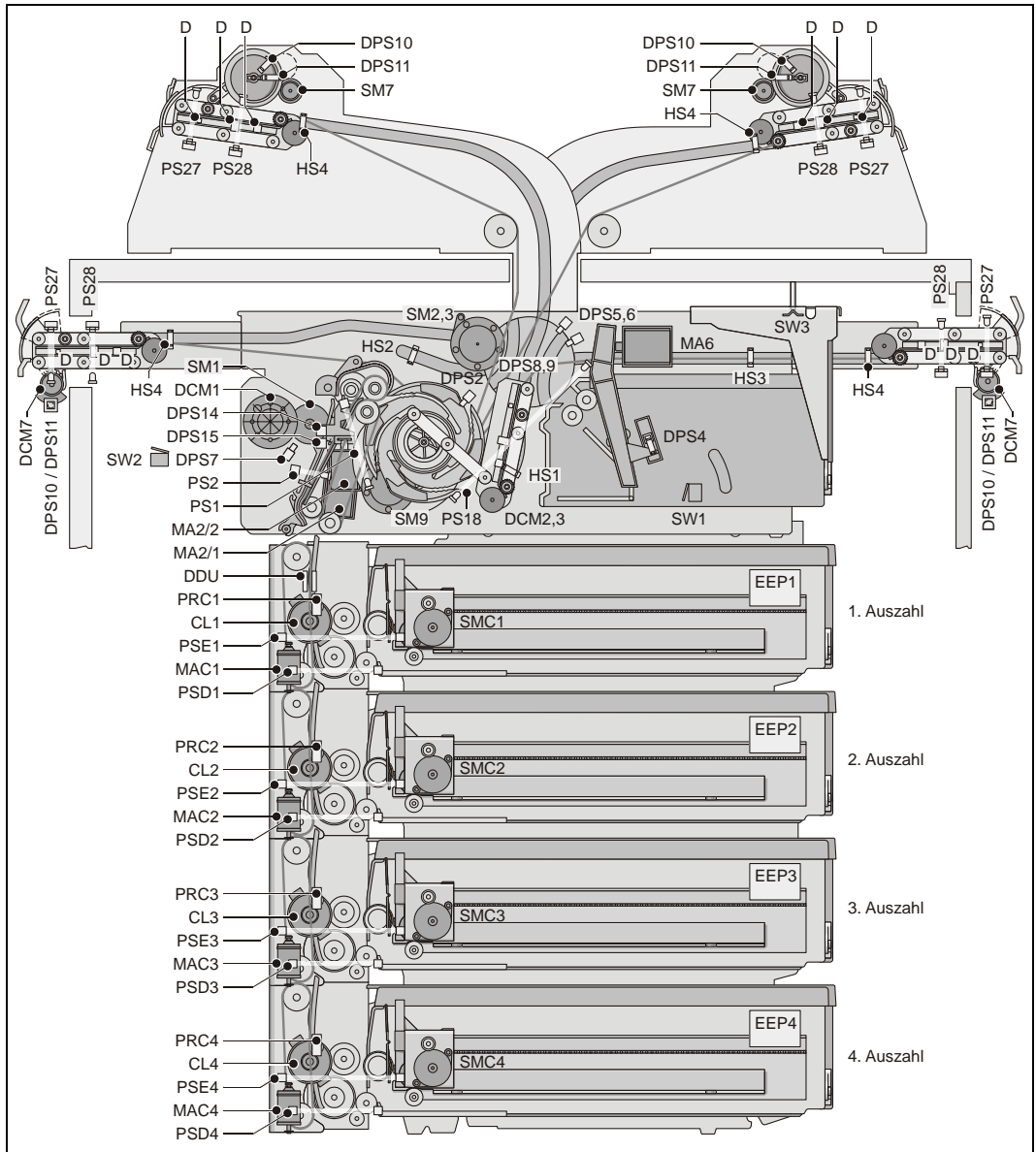
Switch SW4 at the VCMD switches off the main motor DCM1. This is not monitored by the firmware.



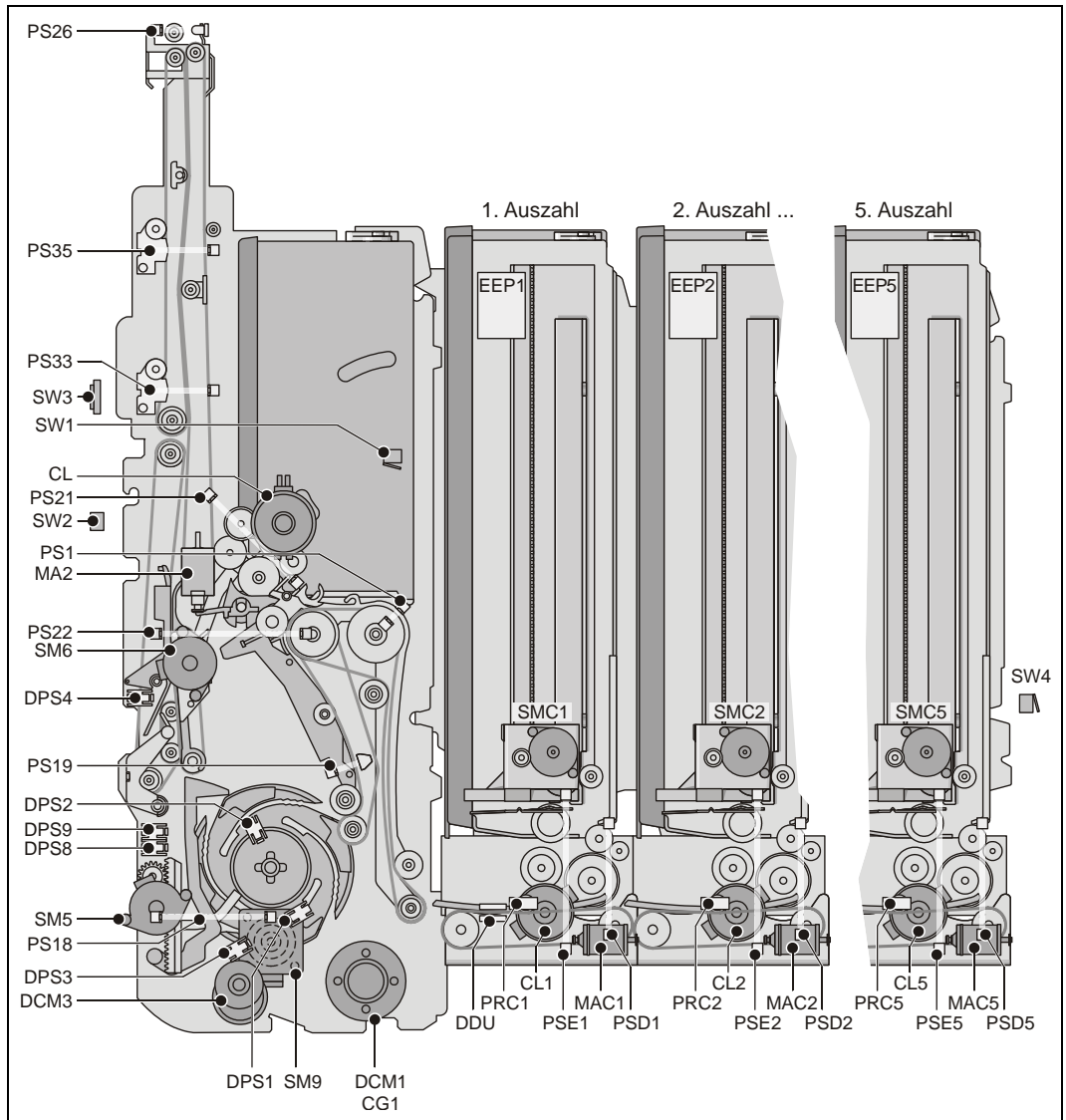
**Components of dispensing unit and cassette**

Component	Designation	Location
MACx	Retaining magnet	Dispensing unit; cassette x (x = 1 – 8)
CLx	Dispensing clutch	Dispensing unit; cassette x (x = 1 – 8)
PSDx	Dispensing sensor	Dispensing unit; cassette x (x = 1 – 8)
PSEx	Empty sensor	Dispensing unit; cassette x (x = 1 – 8)
DDU	Length measurement, measuring station	at top dispensing unit
SMCx	Stepper motor for pressure	in cassette x (x = 1 – 8)
PRCx	Pressure sensor	Dispensing unit; cassette x (x = 1 – 8)
EEPx	EEPROM	in cassette x (x = 1 – 8)

## Position of the elements in the CMD\_V4



## Position of elements in VCMD



## Appendix E: Jumper settings on controller

### Jumper field

Designation	Jumper connected
DOOR	Observe door contact during encryption
SCOP	Activate SCOP handling (with HW extension installed)
Jump A	Standard line parameter (19200, 8, 2, ODD) ; Trigger RESET (subsequently removing jumper again)
Jump B	Operate 4-cassette rack as 2-cassette rack (the two lower dispensing units may not be installed when a 4-way distributor board is used)
Jump C	Reserve
Jump D	Reserve
CI CMOS	Delete CMOS; trigger RESET (subsequently removing jumper again)
NEN	NEN signal is generated by the CMD-V4 (open: SE controls power-saving mode)
Battery	CMOS-support battery active
Jump PLD	Reserve input to Programmable Logic Device

Door / SCOP should always only be selected with one jumper.  
If this is not the case (both jumpers connected), SCOP handling has priority over standard encryption!

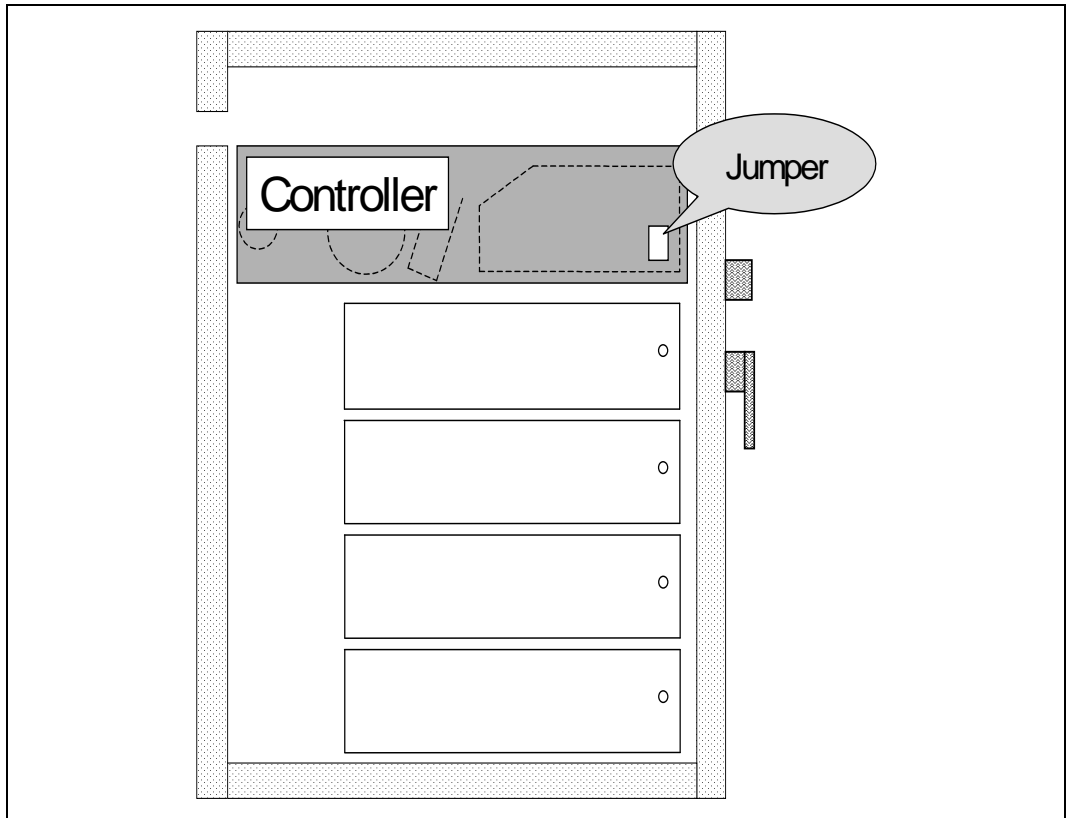


The dispense amount is a component of the SCOP procedure. When the procedure is activated, the system automatically switches to the controller's standard command set.

### Default settings

Jumper connected
NEN
Battery
Jumper connected on one side (on one PIN only)
DOOR (option for activating DOOR or SCOP)

## Jumper positions

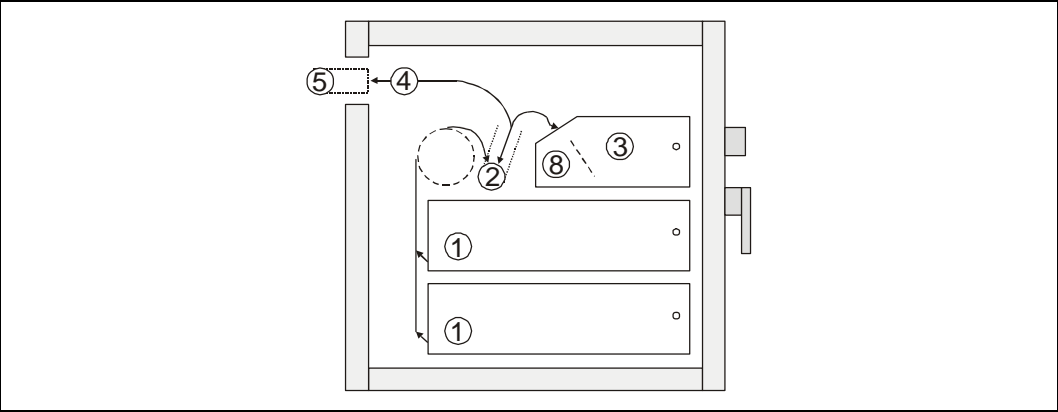


The jumper positions are labeled.

# Appendix F: Transport positions

## Outdoor

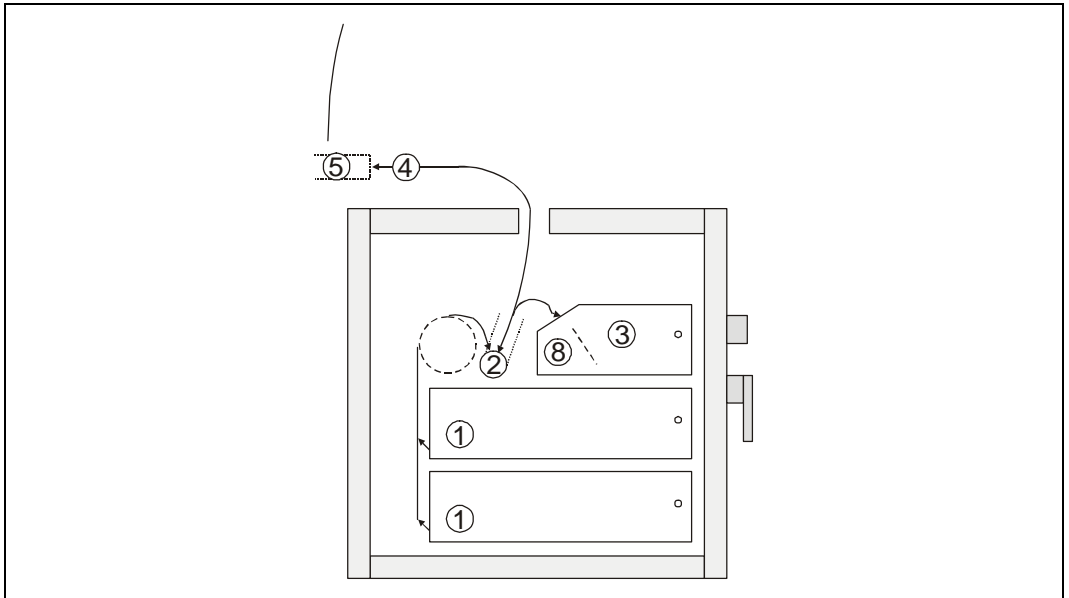
### Rearload



- |                          |                        |
|--------------------------|------------------------|
| 1 Cassette               | 4 Stop over (cash-out) |
| 2 collecting compartment | 5 Removal position     |
| 3 Reject                 | 8 Retract              |

## Indoor

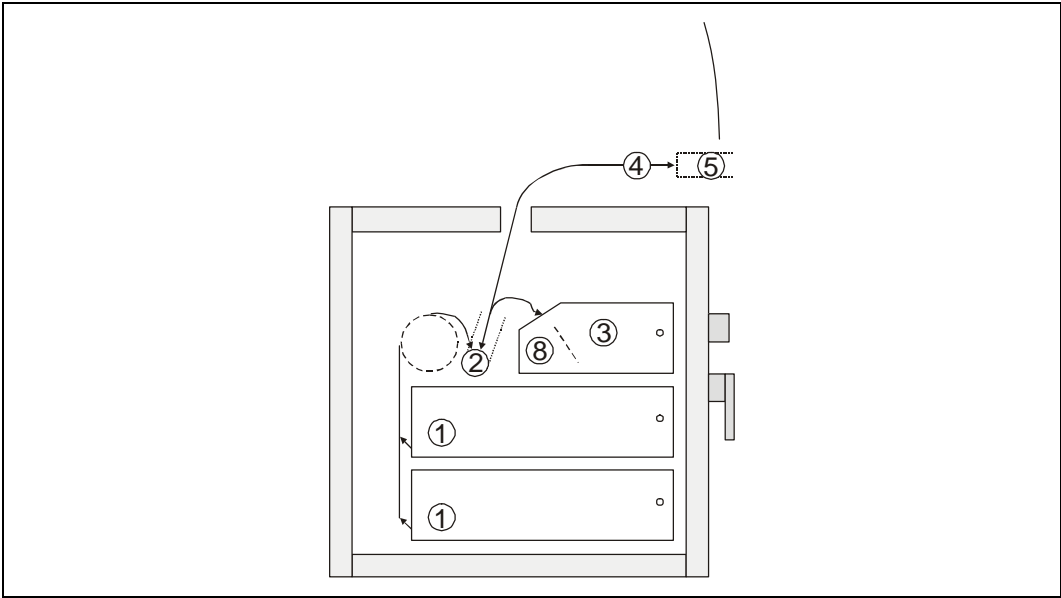
### Rearload



- 1 Cassette
- 2 collecting compartment
- 3 Reject

- 4 Stop over (cash-out)
- 5 Removal position
- 6 Retract

Frontload



- |                          |                        |
|--------------------------|------------------------|
| 1 Cassette               | 4 Stop over (cash-out) |
| 2 collecting compartment | 5 Removal position     |
| 3 Reject                 | 6 Retract              |



## Appendix G: Error Number Table

The internal 64-bit error number is returned in 128-bit (16-byte) ASCII format.

The first 4 bytes represent the class which the error signaling object belongs to, the bytes 5 - 8 the object that triggered the error, bytes 9 - 12 represent the class-specific error code, and bytes 13 - 16 an optional error value for further error ascertainment.

Each error is thus described uniquely and can be located in the following tables. There is not a detailed description of the optional error values since numerous values can only be interpreted by the developer. The description of the error code occasionally refers to possible optional values.

Code	Class
0x0000XXXXXXXXXX	NO_CLASS
Code	Component
0x00000000XXXXXXXX	NO_ID
0x00000060XXXXXXXX	IO_TABLE
0x00004010XXXXXXXX	KONTRON
0x00004050XXXXXXXX	NOTE_BUNDLE

Code	Class	
0x0001XXXXXXXXXXXX	COMPONENT	
Code	ERROR	Meaning
0x0001XXXX0010XXXX	NOT_INITIALIZED	checkAllObjects() found a component, that is disabled. Special conditions prevented the components from being usable.
0x0001XXXX0020XXXX	DOUBLE_ID	checkAllObjects() found a component-id, that is already in use. This a programmer's error.

Code	Class
0x0008XXXXXXXXXX	STATISTIC
Code	Component
0x00084030XXXXXXXX	STATISTIC

Code	Class
0x0010XXXXXXXXXX	AD_CONVERTER

Code	Component
0x00101400XXXXXXXX	AD_CONVERTER

Code	Class
0x0020XXXXXXXXXXXX	AT89S53
Code	Component
0x00201300XXXXXXXX	DDU_CONTROLLER_HW
0x00201870XXXXXXXX	STACKER_CONTROLLER_HW

Code	ERROR	Meaning
0x0020XXXX0010XXXX	ERASE_FLASH_ERROR	flash rom could not be erased, value is 0
0x0020XXXX0020XXXX	BURN_FW_ERROR	error during firmware burning, value is 0
0x0020XXXX0030XXXX	VERIFY_BURNED_FW_ERROR	error during verifying the firmware, value is 0

Code	Class
0x0030XXXXXXXXXXXX	AT90S2313
Code	Component
0x00301800XXXXXXXX	MAIN_MOTOR_HW

Code	ERROR	Meaning
0x0030XXXX0010XXXX	CONTROLLER_OPEN_ERROR	flash rom could not be accessed
0x0030XXXX0020XXXX	BURN_FW_ERROR	error during firmware burning
0x0030XXXX0030XXXX	VERIFY_BURNED_FW_ERROR	error during verifying the firmware
0x0030XXXX0040XXXX	CONTROLLER_SEARCH_ERROR	controller could not be found

Code	Class
0x0035XXXXXXXXXXXX	MAX7301

Code	Class
0x0037XXXXXXXXXXXX	TLE6208
Code	Component
0x00370102XXXXXXXX	TLE6208_LED

Code	Class
0x0040XXXXXXXXXX	BEEPER
Code	Component
0x00400B10XXXXXXXX	BEEPER

Code	Class
0x0050XXXXXXXXXX	BOOT_CMOS

Code	Class
0x005AXXXXXXXXXX	BUNDLE_REJECT_FORK
Code	Component
0x005A18C0XXXXXXXX	BUNDLE_REJECT_FORK

Code	ERROR	Meaning
0x005AXXXX0001XXXX	INITIALIZATION_FAILED	serious errors occurred during the initialization, position not reached (photo-sensor, motor defect, not connected or hw damaged)
0x005AXXXX0002XXXX	DRIVE_TO_REJECT_FAILED	it was not possible to move the bundle reject fork to the reject position (photo-sensor, motor defect or hw damaged)
0x005AXXXX0003XXXX	DRIVE_TO_OUTPUT_FAILED	it was not possible to move the bundle reject fork to the output position (photo-sensor, motor defect or hw damaged)
0x005AXXXX0004XXXX	MOVE_FAILED	the fork could not move to a desired position (photo-sensor, motor defect or hw damaged)
0x005AXXXX0005XXXX	MOTOR_FAILURE	the step motor of the bundle reject fork could not be started (hw damaged or not connected)
0x005AXXXX0006XXXX	SELFTEST_FAILED	the self test failed, fork could be blocked, motor defect or hw damaged
0x005AXXXX0007XXXX	REJECT_TO_EDGE_JITTER	the step-differences during the self test between two reject-to-edge measurements were greater than MAX_STEP_JITTER
0x005AXXXX0008XXXX	OUTPUT_AREA_JITTER	the step-differences during the self test between two output area measurements were greater than MAX_STEP_JITTER
0x005AXXXX0009XXXX	REJECT_AREA_JITTER	the step-differences during the self test between two reject area measurements were greater than MAX_STEP_JITTER

Code	Class
0x0060XXXXXXXXXX	BUTTON
Code	Component
0x00600B00XXXXXXXX	BUTTON

Code	Class
0x0070XXXXXXXXXXXX	CASSETTE
Code	Component
0x00701710XXXXXXXX	CASSETTE_1
0x00701720XXXXXXXX	CASSETTE_2
0x00701730XXXXXXXX	CASSETTE_3
0x00701740XXXXXXXX	CASSETTE_4
0x00701750XXXXXXXX	CASSETTE_5
0x00701760XXXXXXXX	CASSETTE_6
0x00701770XXXXXXXX	CASSETTE_7
0x00701780XXXXXXXX	CASSETTE_8

Code	ERROR	Meaning
0x0070XXXX0010XXXX	OUT_CONTEXT_ERROR	dispense is refused due to invalid state (EMPTY or MISSING stored in value) State
0x0070XXXX0030XXXX	DOUBLE_NOTES	too many multiple notes, value holds number of dispense-retries
0x0070XXXX0040XXXX	INVALID_NOTES	too many invalid notes, value holds number of dispense-retries
0x0070XXXX0050XXXX	EMPTY_ERROR	cassette become empty during dispense, value is 0
0x0070XXXX0060XXXX	EMPTY_ERROR_INT	same circumstances like EMPTY_ERROR
0x0070XXXX0065XXXX	NO_INT_RESPONSE	no response from interrupt, internal FW-error value is 0
0x0070XXXX0070XXXX	NO_DISPENSE_ERROR	no note could be dispensed, value holds number of dispense-retries
0x0070XXXX0090XXXX	NOTE_STUCK_ERROR	note did not leave the dispense-sensor within a certain time, so it looks like a jam, value is 0
0x0070XXXX00A0XXXX	BAD_NOTE_ERROR	a very short note must be invalid, value is 0
0x0070XXXX00B0XXXX	IS_EMPTY_ERROR	photo-sensor error in empty-check, internal FW-error, value holds bool isEmpty()=0/1
0x0070XXXX00C0XXXX	CHECK_EEPROM_ERROR	error when reading eeprom-data occurred, value holds number of read-cycles
0x0070XXXX00D0XXXX	PRINT_TIMEOUT	the receipt to be printed was not available in time
0x0070XXXX00E0XXXX	TRANSPORT_LOCK	dispense is stopped by a transport-lock ( unexpected note PS19 or PS27 )
0x0070XXXX00F0XXXX	SAFETY_SWITCH_DURING_DISPENSE	open safety-switch detected during dispense
0x0070XXXX0100XXXX	EEPROM_DEFECT	eeprom defect

Code	Class
0x0075XXXXXXXXXX	CASSETTE_EEPROM
Code	Component
0x00750E10XXXXXXXX	EEPROM_CASSETTE_1
0x00750E20XXXXXXXX	EEPROM_CASSETTE_2
0x00750E30XXXXXXXX	EEPROM_CASSETTE_3
0x00750E40XXXXXXXX	EEPROM_CASSETTE_4
0x00750E50XXXXXXXX	EEPROM_CASSETTE_5
0x00750E60XXXXXXXX	EEPROM_CASSETTE_6
0x00750E70XXXXXXXX	EEPROM_CASSETTE_7
0x00750E80XXXXXXXX	EEPROM_CASSETTE_8

Code	ERROR	Meaning
0x0075XXXX0010XXXX	READ_VERSION_ERROR	read error during reading the version number
0x0075XXXX0020XXXX	CHECKSUM_ERROR	error during checking the checksum
0x0075XXXX0030XXXX	WRONG_VERSION	version number is invalid
0x0075XXXX0040XXXX	WRITE_VERSION_ERROR	write error during writing the version number
0x0075XXXX0050XXXX	EEPROM_ERASED	the eeprom has been erased

Code	Class	
0x <b>0080</b> XXXXXXXXXXXX	CIRCLE_FORK	
Code	Component	
0x0080 <b>1920</b> XXXXXXX	CIRCLE_FORK	
Code	ERROR	Meaning
0x0080XXXX <b>0010</b> XXXX	NO_INITIAL_EDGE	an initial edge was not found. Value is always 0
0x0080XXXX <b>0020</b> XXXX	NO_SEPARATE_EDGE	a separate edge was not found. Value holds stepperIdx (0/1)
0x0080XXXX <b>0030</b> XXXX	NO_FINE_TUNE_EDGE	the edge for fine tuning was not found. Value holds stepperIdx (0/1)
0x0080XXXX <b>0040</b> XXXX	NO_FINE_TUNE_EDGE_DARK_LIGHT	a dark/light change was not found. Value holds stepperIdx (0/1)
0x0080XXXX <b>0050</b> XXXX	AUTO_CORRECT_NOT_POSSIBLE	an automatic correction of the position is not possible. Value holds the position
0x0080XXXX <b>0060</b> XXXX	CHECK_POSITION_ERROR	a check of the position is not possible. Value holds the position
0x0080XXXX <b>0070</b> XXXX	INIT_NOT_POSSIBLE	initialization was unsuccessful

Code	Class	
0x0090XXXXXXXXXXXX	CLAMP	
Code	Component	
0x00901930XXXXXXX	CLAMP	
Code	ERROR	Meaning
0x0090XXXX0010XXXX	UNEXPECTED_SENSOR	an unexpected sensor was found. Value holds the unexpected position
0x0090XXXX0020XXXX	NO_SENSOR_OR_JAM	next sensor not found in time, jam possible. Value holds the target position
0x0090XXXX0030XXXX	NO_2 <sup>ND</sup> _EDGE	the second edge was not found in time. Value holds the target position
0x0090XXXX0040XXXX	NO_3 <sup>RD</sup> _EDGE	the third edge was not found in time. Value holds the target position
0x0090XXXX0050XXXX	UNDEFINIED_SENSOR_EDGE_STATE	undefined sensor edge state
0x0090XXXX0060XXXX	STACKER_RECOVERY_ERROR	clamp cannot unlock after recovery
0x0090XXXX0070XXXX	ADD_TIME_REACHED	max. additional time for unlock clamp reached
0x0090XXXX0100XXXX	SB_CLAMP_START_ERROR	cannot find the clamp in backward direction, because of an error while starting the clamp
0x0090XXXX0110XXXX	SB_UNDEFINIED_SENSOR_EDGE_STAT E	undefined sensor edge state by backward-searching for the clamp
0x0090XXXX0200XXXX	SF_CLAMP_START_ERROR	cannot find the clamp in forward direction, because of an error while starting the clamp
0x0090XXXX0210XXXX	SF_UNDEFINIED_SENSOR_EDGE_STAT E	undefined sensor edge state by forward-searching for the clamp

Code	Class
0x00A0XXXXXXXXXX	CLUTCH
Code	Component
0x00A00590XXXXXXXX	CLUTCH_REJECT_BOX
0x00A00C10XXXXXXXX	CLUTCH_CASSETTE_1
0x00A00C20XXXXXXXX	CLUTCH_CASSETTE_2
0x00A00C30XXXXXXXX	CLUTCH_CASSETTE_3
0x00A00C40XXXXXXXX	CLUTCH_CASSETTE_4
0x00A00C50XXXXXXXX	CLUTCH_CASSETTE_5
0x00A00C60XXXXXXXX	CLUTCH_CASSETTE_6
0x00A00C70XXXXXXXX	CLUTCH_CASSETTE_7
0x00A00C80XXXXXXXX	CLUTCH_CASSETTE_8

Code	Class
0x00B0XXXXXXXXXX	CMD

Code	Class	
0x00C0XXXXXXXXXXXX	CMOS	
Code	ERROR	Meaning
0x00C0XXXX0010XXXX	CMOS_RESET	content invalid or new version

Code	Class	
0x00E0XXXXXXXXXXXX	COLLECTOR	
Code	Component	
0x00E01960XXXXXXXX	COLLECTOR	
Code	ERROR	Meaning
0x00E0XXXX0001XXXX	DISABLED	component not in use due to hw-error
0x00E0XXXX0010XXXX	INVALID_NOTE	invalid note in stacker
0x00E0XXXX0020XXXX	START_COLLECT_ERROR	most probably a belt motor error
0x00E0XXXX0021XXXX	STOP_COLLECT_ERROR	most probably a belt motor error
0x00E0XXXX0030XXXX	TRANSPORT_IN_BOX_LIGHT_JAM	photo-sensor did not become covered in reject/retract
0x00E0XXXX0031XXXX	TRANSPORT_IN_BOX_DARK_JAM	photo-sensor did not become free in reject/retract
0x00E0XXXX0033XXXX	UNEXPECTED_NOTE_IN_REJECT_RETRACT	an unexpected note appeared in the dispense-transport during reject/retract
0x00E0XXXX0035XXXX	CLEAR_SINGLE_REJECT_DARK_JAM	photo-sensor did not become free in clearing singlereject tray
0x00E0XXXX0036XXXX	CLEAR_SINGLE_REJECT_LIGHT_JAM	photo-sensor did not become covered in clearing singlereject tray
0x00E0XXXX0040XXXX	SWITCH_RR_BOX_ERROR	magnet did not switch, value is desired direction
0x00E0XXXX0045XXXX	SWITCH_RR_BOX_REJECT_POS_ERROR	magnet did not switch tray in reject position
0x00E0XXXX0050XXXX	TRANSPORT_IN_BOX_ERROR	transport in box failed, e.g. belt motor error
0x00E0XXXX0060XXXX	FORK_ERROR	fork not off in clearing of singlereject tray
0x00E0XXXX0062XXXX	FORK_NOT_OFF	fork is not off, although switched off
0x00E0XXXX0064XXXX	FORK_NOT_ON	fork is not on, although switched on
0x00E0XXXX0080XXXX	CLEAR_SINGLE_REJECT_ERROR	error in clearing singlereject tray, e.g. motor failure
0x00E0XXXX0090XXXX	CLAMP_EMPTY_ERROR	internal fw-error, call of isClampEmpty() failed, photo-sensors not accessible
0x00E0XXXX00A0XXXX	COLLECTOR_INIT_ERROR	the initialization failed, the stacker or the reject box might be damaged

Code	Class
0x00E2XXXXXXXXXXXX	COLLECTOR_EEPROM
Code	Component
0x00E20E90XXXXXXXX	COLLECTOR_EEPROM

Code	ERROR	Meaning
0x00E2XXXX0010XXXX	READ_VERSION_ERROR	read error during reading the version number
0x00E2XXXX0020XXXX	CHECKSUM_ERROR	error during checking the checksum
0x00E2XXXX0030XXXX	WRONG_VERSION	version number is invalid
0x00E2XXXX0040XXXX	WRITE_VERSION_ERROR	write error during writing the version number
0x00E2XXXX0050XXXX	EEPROM_ERASED	the eeprom has been erased
0x00E2XXXX0060XXXX	INIT_ERROR	write error during init sequence

Code	Class
0x00E8XXXXXXXXXXXX	COLLECTOR_HDM
Code	Component
0x00E81968XXXXXXXX	COLLECTOR_HDM

Code	ERROR	Meaning
0x00E8XXXX0001XXXX	DISABLED	component not in use due to hw-error
0x00E8XXXX0002XXXX	PHOTO_SENSORS_FAILED	Photo sensor task failed, no photo sensor access/release (hardware broken?)
0x00E8XXXX0010XXXX	INVALID_NOTE	invalid note in stacker
0x00E8XXXX0011XXXX	NOTE_JAM	Bundle could not be moved into reject box, possible note or photo sensor jam
0x00E8XXXX0020XXXX	START_COLLECT_ERROR	most probably a belt motor error
0x00E8XXXX0021XXXX	STOP_COLLECT_ERROR	most probably a belt motor error
0x00E8XXXX0025XXXX	MAIN_MOTOR_FAILED	Main motor could not be started, hardware broken?
0x00E8XXXX0030XXXX	TRANSPORT_IN_BOX_LIGHT_JAM	photo-sensor did not become covered in reject/retract
0x00E8XXXX0031XXXX	TRANSPORT_IN_BOX_DARK_JAM	photo-sensor did not become free in reject/retract
0x00E8XXXX0033XXXX	UNEXPECTED_NOTE_IN_REJECT_RETRACT	an unexpected note appeared in the dispense-transport during reject/retract
0x00E8XXXX0035XXXX	CLEAR_SINGLE_REJECT_DARK_JAM	photo-sensor did not become free in clearing singlereject tray
0x00E8XXXX0036XXXX	CLEAR_SINGLE_REJECT_LIGHT_JAM	photo-sensor did not become covered in clearing singlereject tray
0x00E8XXXX0040XXXX	SWITCH_RR_BOX_ERROR	magnet did not switch, value is desired direction
0x00E8XXXX0050XXXX	TRANSPORT_IN_BOX_ERROR	transport in box failed, e.g. belt motor error
0x00E8XXXX0060XXXX	FORK_ERROR	fork not off in clearing of singlereject tray
0x00E8XXXX0062XXXX	FORK_NOT_OFF	fork is not off, although switched off
0x00E8XXXX0064XXXX	FORK_NOT_ON	fork is not on, although switched on
0x00E8XXXX0080XXXX	CLEAR_SINGLE_REJECT_ERROR	error in clearing singlereject tray, e.g. motor failure
0x00E8XXXX0090XXXX	CLAMP_EMPTY_ERROR	internal fw-error, call of isClampEmpty() failed, photo-sensors not accessible



0x00E8XXXX00A0XXXX	COLLECTOR_INIT_ERROR	the initialization failed, the stacker or the reject box might be damaged
0x00E8XXXX00B0XXXX	CONTROLGEAR_DISPENSE_POS_NOT_REACHED	control-gear cannot reach dispense-position

Code	Class
0x00EAXXXXXXXXXXXX	COLLECTOR_SLIM_CASH
Code	Component
0x00EA196AXXXXXXXX	COLLECTOR_SLIM_CASH

Code	ERROR	Meaning
0x00EAXXXX0001XXXX	DISABLED	component not in use due to hw-error
0x00EAXXXX0002XXXX	PHOTO_SENSORS_FAILED	Photo sensor task failed, no photo sensor access/release (hardware broken?)
0x00EAXXXX0010XXXX	INVALID_NOTE	Invalid or unexpected note in output direction
0x00EAXXXX0011XXXX	NOTE_JAM	Note could not move to target position, possible note or photo sensor jam
0x00EAXXXX0020XXXX	START_COLLECT_ERROR	not used
0x00EAXXXX0021XXXX	STOP_COLLECT_ERROR	not used
0x00EAXXXX0025XXXX	MAIN_MOTOR_FAILED	Main motor could not be started, hardware broken?
0x00EAXXXX0060XXXX	FORK_ERROR	not used
0x00EAXXXX0062XXXX	FORK_NOT_OFF	fork is not off, although switched off
0x00EAXXXX0064XXXX	FORK_NOT_ON	fork is not on, although switched on
0x00EAXXXX00A0XXXX	COLLECTOR_INIT_ERROR	the initialization failed, the stacker or the reject box might be damaged

Code	Class
0x00F0XXXXXXXXXXXX	CODEC_IBM
Code	Component
0x00F00310XXXXXXXX	CODEC_IBM

Code	Class
0x00F8XXXXXXXXXXXX	CODEC_NCR
Code	Component
0x00F80350XXXXXXXX	CODEC_NCR

Code	Class
0x0100XXXXXXXXXX	CODEC_WINCOR
Code	Component
0x01000300XXXXXXX	CODEC_WINCOR

Code	Class
0x0110XXXXXXXXXX	CODEC_SERVICE
Code	Component
0x01100330XXXXXXX	CODEC_SERVICE

Code	Class
0x0120XXXXXXXXXX	CODEC_SELF_TEST (old CODEC_TEST)
0x0128XXXXXXXXXX	(old) CODEC_SELF_TEST
Code	Component
0x01200340XXXXXXX	CODEC_SELFTEST
0x01280340XXXXXXX	CODEC_SELFTEST

Code	ERROR	Meaning
0x0120XXXX0000XXXX	NO_ERROR	no error
0x0120XXXX0001XXXX	SELFTEST_COMMAND_UNKNOWN	command decode error in class CodecWincor
0x0120XXXX0002XXXX	COMMAND_RESULT_UNEQUAL	hardware task result differs form selftest result
0x0120XXXX0003XXXX	OPEN_SECURITY_SWITCH	security switch is open
0x0120XXXX0004XXXX	RTC_TIME_INVALID	result of time is zero (rtos.getTimeSince(startTime))
0x0120XXXX0010XXXX	PHOTO_SENSOR_INIT_FAILED	photo sensor init failed
0x0120XXXX0011XXXX	CMOS_BATTERY_EMPTY	cmos battery is empty or jumper not installed
0x0120XXXX0020XXXX	CMOS_CLEAR_BASE	free, not used
0x0120XXXX0021XXXX	CMOS_CLEAR_FAILED	cmos write/read failed
0x0120XXXX0022XXXX	NO_RESET_AFTER_CMOS_CLEAR	after cmos clear reset is the only executable command
0x0120XXXX0040XXXX	RR_BOX_MAGNET_BASE	free, not used
0x0120XXXX0041XXXX	RR_BOX_MAGNET_NOT_OFF	RR_BOX magnet (MA6) switched off but not in reject (GL4) position
0x0120XXXX0042XXXX	RR_BOX_MAGNET_NOT_ON	RR_BOX magnet (MA6) switched on but not in retract (GL4) position
0x0120XXXX0043XXXX	RR_BOX_MAGNET_ON_TIMEOUT	RR_BOX magnet (MA6) switched on time out of range
0x0120XXXX0044XXXX	RR_BOX_MAGNET_OFF_TIMEOUT	RR_BOX magnet (MA6) switched off time out of range
0x0120XXXX0060XXXX	SINGLE_REJECT_MAGNET_BASE	free, not used

0x0120XXXX0061XXXX	SINGLE_REJECT_MAGNET_NOT_OFF	single reject magnet (MA2) switched off but not in initial (GL14) position
0x0120XXXX0062XXXX	SINGLE_REJECT_MAGNET_NOT_ON	single reject magnet (MA2) switched on but not reject (GL14) position
0x0120XXXX0063XXXX	SINGLE_REJECT_MAGNET_ON_TIMEOUT	single reject magnet (MA2) switched on time out of range
0x0120XXXX0064XXXX	SINGLE_REJECT_MAGNET_OFF_TIMEOUT	single reject magnet (MA2) switched off time out of range
0x0120XXXX0065XXXX	SINGLE_REJECT_NO_HW_SWITCH_OFF	single reject magnet (MA2) not switched off by hardware
0x0120XXXX0066XXXX	SINGLE_REJECT_HW_SWITCH_TIME_ERR	single reject magnet (MA2) switched off by hardware but not in time limit
0x0120XXXX0100XXXX	CLAMP_BELT_MOTOR_BASE	free, not used
0x0120XXXX0101XXXX	CLAMP_BELT_MOTOR_STATE	clamp belt motor (M3) state received while running
0x0120XXXX0200XXXX	MAIN_MOTOR_BASE	free, not used
0x0120XXXX0201XXXX	MAIN_MOTOR_SPEED_SLOW_STATE	main motor (M1) state at speed slow check
0x0120XXXX0202XXXX	MAIN_MOTOR_SPEED_UP_SLOW_FAILED	main motor (M1) speed up at slow speed failed
0x0120XXXX0203XXXX	MAIN_MOTOR_SPEED_SLOW_TO_SLOW	main motor (M1) speed check speed slow to slow
0x0120XXXX0204XXXX	MAIN_MOTOR_SPEED_SLOW_TO_FAST	main motor (M1) speed check speed slow to fast
0x0120XXXX0205XXXX	MAIN_MOTOR_SPEED_SLOW_JITTER	main motor (M1) speed check speed slow jitter
0x0120XXXX0206XXXX	MAIN_MOTOR_SPEED_SLOWDOWN_SLOW_FAILED	main motor (M1) speed slow down at slow speed failed
0x0120XXXX0207XXXX	MAIN_MOTOR_SPEED_MEDIUM_STATE	main motor (M1) state at speed medium check
0x0120XXXX0208XXXX	MAIN_MOTOR_SPEED_UP_MEDIUM_FAILED	main motor (M1) speed up at medium speed failed
0x0120XXXX0209XXXX	MAIN_MOTOR_SPEED_MEDIUM_TO_SLOW	main motor (M1) speed check speed medium to slow
0x0120XXXX0210XXXX	MAIN_MOTOR_SPEED_MEDIUM_TO_FAST	main motor (M1) speed check speed medium to fast
0x0120XXXX0211XXXX	MAIN_MOTOR_SPEED_MEDIUM_JITTER	main motor (M1) speed check speed medium jitter
0x0120XXXX0212XXXX	MAIN_MOTOR_SPEED_SLOWDOWN_MEDIUM_FAILED	main motor (M1) speed slow down at medium speed failed
0x0120XXXX0213XXXX	MAIN_MOTOR_SPEED_FAST_STATE	main motor (M1) state at speed fast check
0x0120XXXX0214XXXX	MAIN_MOTOR_SPEED_UP_FAST_FAILED	main motor (M1) speed up at fast speed failed
0x0120XXXX0215XXXX	MAIN_MOTOR_SPEED_FAST_TO_SLOW	main motor (M1) speed check speed fast to slow
0x0120XXXX0216XXXX	MAIN_MOTOR_SPEED_FAST_TO_FAST	main motor (M1) speed check speed fast to fast
0x0120XXXX0217XXXX	MAIN_MOTOR_SPEED_FAST_JITTER	main motor (M1) speed check speed fast jitter
0x0120XXXX0218XXXX	MAIN_MOTOR_SPEED_SLOWDOWN_FAST_FAILED	main motor (M1) speed slow down at fast speed failed
0x0120XXXX0219XXXX	MAIN_MOTOR_TIMER_NO_CLOCK	main motor (M1) clock missing at timer input
0x0120XXXX0220XXXX	MAIN_MOTOR_TIMER_NOT_AVAILABLE	timer semaphore not available for main motor (M1) clock measurement
0x0120XXXX0300XXXX	STACKER_WHEEL_BASE	free, not used

0x0120XXXX0301XXXX	STACKER_WHEEL_COMMAND_HOLD_FAILED	stackerwheel command hold failed
0x0120XXXX0302XXXX	STACKER_WHEEL_COMMAND_INIT_FAILED	stackerwheel command init failed
0x0120XXXX0303XXXX	STACKER_WHEEL_COMMAND_STEP_FAILED	stackerwheel command step failed
0x0120XXXX0304XXXX	STACKER_WHEEL_STATE	stackerwheel state received
0x0120XXXX0305XXXX	STACKER_WHEEL_INIT_GL2_NO_DARK_LIGHT_CHANGE	stackerwheel (SM9) init without sensor (GL2) dark/light change
0x0120XXXX0306XXXX	STACKER_WHEEL_INIT_OVERDRIVES_GL2	stackerwheel (SM9) init overdrives sensor (GL2)
0x0120XXXX0307XXXX	STACKER_WHEEL_STEP_GL2_NO_DARK_LIGHT_CHANGE	stackerwheel (SM9) step without sensor (GL2) dark/light change
0x0120XXXX0308XXXX	STACKER_WHEEL_STEP_OVERDRIVES_GL2	stackerwheel (SM9) step overdrives sensor (GL2)
0x0120XXXX0309XXXX	STACKER_WHEEL_STEP_TIME_FAILED	stackerwheel (SM9) step time out of range
0x0120XXXX0400XXXX	CIRCLE_FORK_BASE	free, not used
0x0120XXXX0401XXXX	CIRCLE_FORK_INIT_FAILED	circle fork init failed
0x0120XXXX0402XXXX	CIRCLE_FORK_GL8_ERROR	GL8 not LIGHT after circle fork init
0x0120XXXX0403XXXX	CIRCLE_FORK_GL9_ERROR	GL9 not LIGHT after circle fork init
0x0120XXXX0404XXXX	CIRCLE_FORK_CLAMP_OPEN_FAILED	circle fork position not reached
0x0120XXXX0405XXXX	CIRCLE_FORK_REJECT_FAILED	circlefork error
0x0120XXXX0406XXXX	CIRCLE_FORK_IN_SAFE_REAR_FAILED	circlefork error
0x0120XXXX0407XXXX	CIRCLE_FORK_CATCH_FAILED	circlefork error
0x0120XXXX0408XXXX	CIRCLE_FORK_OUT_SAFE_FAILED	circlefork error
0x0120XXXX0409XXXX	CIRCLE_FORK_IN_SAFE_FRONT_FAILED	circlefork error
0x0120XXXX0410XXXX	CIRCLE_FORK_DEAD_END_FAILED	circlefork error
0x0120XXXX0411XXXX	CIRCLE_FORK_CLAMP_CLOSED_FAILED	circlefork error
0x0120XXXX0412XXXX	CIRCLE_FORK_MECHANIK_TEST_FAILED	circlefork error
0x0120XXXX0500XXXX	SHUTTER_BASE	free, not used
0x0120XXXX0501XXXX	SHUTTER_NOT_INSTALLED	shutter (SM7/M7) not installed or type unknown
0x0120XXXX0502XXXX	SHUTTER_CLOSE_FAILED	shutter (SM7/M7) close failed
0x0120XXXX0503XXXX	SHUTTER_OPEN_FAILED	shutter (SM7/M7) open failed
0x0120XXXX0504XXXX	SHUTTER_OFFER_FAILED	shutter (SM7/M7) offer failed
0x0120XXXX0600XXXX	SINGLE_REJECT_BOX_BASE	free, not used
0x0120XXXX0601XXXX	SINGLE_REJECT_BOX_NOT_INSTALLED	single reject box (SM1) not installed
0x0120XXXX0602XXXX	SINGLE_REJECT_BOX_MONEY	single reject box (SM1) contains money
0x0120XXXX0603XXXX	SINGLE_REJECT_BOX_TIMEOUT	single reject box (SM1) init timeout
0x0120XXXX0604XXXX	SINGLE_REJECT_BOX_STATE	single reject box (SM1) state received
0x0120XXXX0605XXXX	SINGLE_REJECT_BOX_INIT_FAILED	single reject box (SM1) init failed
0x0120XXXX0606XXXX	SINGLE_REJECT_BOX_STANDART_	single reject box (SM1) standard position

	FAILED	error
0x0120XXXX0607XXXX	SINGLE_REJECT_BOX_RESERVE_FAILED	single reject box (SM1) reserve position error
0x0120XXXX0608XXXX	SINGLE_REJECT_BOX_EMPTY_FAILED	single reject box (SM1) empty position error
0x0120XXXX0700XXXX	OUTPUT_TRANSPORT_BASE	free, not used
0x0120XXXX0701XXXX	NO_OUTPUT_TRANSPORT_INSTALLED	no output transport installed (RD 0x86 sensor 3)
0x0120XXXX0702XXXX	OUTPUT_TRANSPORT_WAGEN_1_SWITCH_OPEN	for out safe transport, switch Wagen_1 must be closed
0x0120XXXX0703XXXX	OUTPUT_TRANSPORT_HALL_SENSOR_ERROR	more then on hall sensor (HL1..4) is active
0x0120XXXX0704XXXX	OUTPUT_TRANSPORT_CLAMP_NOT_EMPTY_LS18	clamp not empty in position stacker
0x0120XXXX0705XXXX	OUTPUT_TRANSPORT_CLAMP_NOT_EMPTY_LS27_LS28	clamp not empty in position shutter
0x0120XXXX0706XXXX	OUTPUT_TRANSPORT_INIT_FAILED	output transport init failed
0x0120XXXX0707XXXX	OUTPUT_TRANSPORT_CLAMP_OPEN_FAILED	output transport clamp open failed
0x0120XXXX0708XXXX	OUTPUT_TRANSPORT_CLAMP_CLOSE_FAILED	output transport clamp close failed
0x0120XXXX0709XXXX	OUTPUT_TRANSPORT_DRIVE_TO_STACKER_FAILED	output transport drive to stacker failed
0x0120XXXX0710XXXX	OUTPUT_TRANSPORT_DRIVE_TO_SHUTTER_FAILED	output transport drive to shutter failed
0x0120XXXX0711XXXX	OUTPUT_TRANSPORT_DRIVE_TO_REJECT_FAILED	output transport drive to reject failed

Code	Class	
0x012AXXXXXXXXXXXX	CONTROL_GEAR	
Code	Component	
0x012A18A0XXXXXXXX	CONTROL_GEAR	
Code	ERROR	Meaning
0x012AXXXX0001XXXX	INITIALIZATION_FAILED	serious errors occurred during the initialization, position not reached (photo-sensors or motor defect or not connected or hw damaged)
0x012AXXXX0002XXXX	SELF_TEST_FAILED	serious errors occurred during the initialization, position not reached (photo-sensors or motor defect or not connected or hw damaged)
0x012AXXXX0004XXXX	POSITION_NOT_REACHED	device jammed during movement to target, value holds target position
0x012AXXXX0006XXXX	POSITION_NOT_LEFT	device jammed = did not leave position, value holds source position
0x012AXXXX000AXXXX	MOTOR_FAILURE	motor did not start
0x012AXXXX000CXXXX	MOVE_TOO_SLOW	device too slow in self test

Code	Class	
0x0130XXXXXXXXXX	DES	
Code	Component	
0x01300070XXXXXXXX	DES	
Code	ERROR	Meaning
0x0130XXXX0010XXXX	NO_ODD_TRANSPORT_KEY	transport key is not odd
0x0130XXXX0020XXXX	LOAD_TRANSPORT_KEY_FAILED	loading failed
0x0130XXXX0030XXXX	NO_ODD_LINE_KEY	line key is not odd
0x0130XXXX0040XXXX	LOAD_LINE_KEY_FAILED	loading failed

Code	Class
0x0140XXXXXXXXXX	DIGITAL_PHOTO_SENSOR

Code	Class	
0x0150XXXXXXXXXX	DISPENSER	
Code	Component	
0x01501990XXXXXXXX	DISPENSER	
Code	ERROR	Meaning
0x0150XXXX0011XXXX	CHECK_TRANSPORT_ERROR	the transport check failed, value is 0
0x0150XXXX0020XXXX	DISPENSE_FAILED	the dispense command failed, value is 0
0x0150XXXX0030XXXX	REFERENCE_FAILED	the reference command failed, value is 0
0x0150XXXX0040XXXX	CASH_OUT_NOT_AVAILABLE	cash out is not available, value is 0
0x0150XXXX0050XXXX	SAFETY_SWITCH_OPENED	the safety switch is open, value is 0
0x0150XXXX0051XXXX	DEVICE_IS_LOCKED	the device is locked (retract & quit), value is 0
0x0150XXXX0060XXXX	PRESSURE_NOT_OK	the pressure in a cassette could not be updated, value holds the cassette number
0x0150XXXX0064XXXX	RETRACT_NOTES_TO_STACKER_FAILED	in case of a retract or bundle-reject from shutter the notes could not be retracted from the offer or shutter position into the stacker position
0x0150XXXX0065XXXX	RETRACT_AND_QUIT_NOTES_FAILED	in case of a retract & quit from shutter the notes could not be retracted from the offer or shutter position into the stacker position
0x0150XXXX0080XXXX	RETRACT_FAILED	the retract failed, the notes could be transported to the retract tray, but could not be deposited in it
0x0150XXXX0090XXXX	REJECT_FAILED	the reject failed, the notes could be transported to the retract tray, but could not be deposited in it
0x0150XXXX00A0XXXX	DRIVE_TO_BOX_FAILED	the clamp could not be driven from the stacker to the box
0x0150XXXX00A8XXXX	DRIVE_TO_STACKER_FAILED	the clamp could not be driven to the

		stacker
0x0150XXXX00B0XXXX	CLAMP_OPEN_FAILED	the clamp could not be opened
0x0150XXXX00C0XXXX	CLEAR_AFTER_NOTE_REMOVAL_FAILED	missing exception text
0x0150XXXX00D0XXXX	CLOSE_OFFER_FAILED	not used
0x0150XXXX00F0XXXX	SHUTTER_NOT_EMPTY	not used
0x0150XXXX0100XXXX	CLAMP_IN_SHUTTER_EMPTY	after the transport to the shutter, the clamp is unexpected empty
0x0150XXXX0110XXXX	STACKER_NOT_EMPTY	unexpected notes in shutter after reject/retract
0x0150XXXX0120XXXX	DRIVE_TO_SHUTTER_FAILED	moving money to shutter failed
0x0150XXXX0130XXXX	SHUTTER_MANIPULATION_DETECTED	missing exception text

Code	Class	
0x0160XXXXXXXXXXXX	DISPENSE_TRANSPORT	
Code	Component	
0x01601980XXXXXXXX	DISPENSE_TRANSPORT	
Code	ERROR	Meaning
0x0160XXXX0020XXXX	SCAN_TIMEOUT	time for dispense reached, value holds current cassette number
0x0160XXXX0022XXXX	SCAN_TIMEOUT_ADDITIONAL	timeout for additional dispense reached, value holds current cassette number
0x0160XXXX0024XXXX	REFERENCE_TIMEOUT	timeout for reference reached, value is 0
0x0160XXXX0026XXXX	CHECK_TIMEOUT	timeout for check transport reached, value is 0 or cassette number
0x0160XXXX0040XXXX	DISPENSE_FAILED	dispense interrupted because an external event, value holds the current cassette number
0x0160XXXX0042XXXX	DISPENSE_ERROR	dispense failed because an error appears, value holds the current cassette number
0x0160XXXX0044XXXX	EMPTY_CASSETTE_DISPENSE	dispense failed because cassette went empty.
0x0160XXXX0046XXXX	EMPTY_CASSETTE_REFERENCE	reference failed because cassette went empty
0x0160XXXX0050XXXX	REFERENCE_FAILED	reference failed, value holds the cassette number
0x0160XXXX0052XXXX	REFERENCE_ERROR	reference interrupted because an error appears, value holds the cassette number
0x0160XXXX0060XXXX	CHECK_ERROR	check transport failed in case of a motor error, value is 0
0x0160XXXX0070XXXX	INVALID_CASSETTE_CODE	undefined cassette code detected, value holds undefined cassette code
0x0150XXXX0080XXXX	CASSETTE_EXCHANGE_ERROR	main motor error during cassette exchanged, value is 0
0x0150XXXX0090XXXX	MAIN_MOTOR_ERROR	main motor error during dispense, value holds last hardware state of the mainMotor
0x0150XXXX00A0XXXX	MAIN_MOTOR_STOPPED	main motor stopped during dispense problems
0x0150XXXX00B0XXXX	TRANSPORT_LOCK	transport lock during dispense, value holds current cassette number

Code	Class
0x0170XXXXXXXXXX	DMA188
Code	Component
0x01700F00XXXXXXXX	DMA0
0x01700F10XXXXXXXX	DMA1

Code	Class	
0x0180XXXXXXXXXX	DOUBLE_DETECTION_CONTROLLER	
Code	Component	
0x01801310XXXXXX	DDU_CONTROLLER	
Code	ERROR	Meaning
0x0180XXXX0010XXXX	WRITE_ERROR	write error detected
0x0180XXXX0020XXXX	READ_ERROR	read error detected
0x0180XXXX0030XXXX	NO_ACK_HIGH_LOW	controller did not response in time
0x0180XXXX0040XXXX	NO_ACK_LOW_FOR_MEASURING	controller not ready for measuring
0x0180XXXX0050XXXX	NO_ACK_HIGH_FOR_MEASURING	controller did not finish measuring

Code	Class
0x0190XXXXXXXXXXXX	DOUBLE_DETECTION_UNIT
Code	Component
0x01901940XXXXXXXX	DOUBLE_DETECTION_UNIT

Code	ERROR	Meaning
0x0190XXXX0010XXXX	GET_DATA_ERROR	error when receiving data from controller
0x0190XXXX0020XXXX	INIT_FAILED	current regulation failed
0x0190XXXX0030XXXX	START_NOTE_UNKOWN	note not referenced yet
0x0190XXXX0032XXXX	START_FAILED	start not possible
0x0190XXXX0040XXXX	HIGH_CURRENT_DIFFERENCE	current difference between left and right channel too high

Code	Class
0x01A0XXXXXXXXXX	DRIVER_USB
Code	Component
0x01A00200XXXXXXXX	DRIVER_USB



Code	Class
0x01B0XXXXXXXXXX	DRIVER_RS232
Code	Component
0x01B00230XXXXXXXX	PC_RS232
0x01B00240XXXXXXXX	MUX_RS232

Code	Class
0x01B2XXXXXXXXXX	USB_N9603
Code	Component
0x01B20160XXXXXXXX	USB_N9603

Code	Class
0x01C0XXXXXXXXXX	TL16C552
Code	Component
0x01C00130XXXXXXXX	UART1
0x01C00140XXXXXXXX	UART2
0x01C00150XXXXXXXX	TL16552_P

Code	Class
0x01D0XXXXXXXXXX	EEPROM

Code	Class
0x01E0XXXXXXXXXX	ERROR_HANDLER
Code	Component
0x01E00030XXXXXXXX	ERROR_HANDLER

Code	Class	
0x01F0XXXXXXXXXX	FORK	
Code	Component	
0x01F00B30XXXXXXXX	SINGLE_REJECT_FORK	
Code	ERROR	Meaning
0x01F0XXXX0010XXXX	ON_FAILED	switching on failed, value is 0
0x01F0XXXX0011XXXX	ON_TOO_SLOW	switching on was too slow, value is switch time in ms

0x01F0XXXX0020XXXX	OFF_FAILED	switching off failed, value is 0
0x01F0XXXX0021XXXX	OFF_TOO_SLOW	switching off was too slow, value is switch time in ms
0x01F0XXXX0022XXXX	BLOCKED	fork is in wrong position (checked in scanning)
0x01F0XXXX0030XXXX	NOT_ON	the state of a fork is off, without it has been switched off, value is 0
0x01F0XXXX0040XXXX	NOT_OFF	the state of a fork is on, without it has been switched on, value is 0
0x01F0XXXX0050XXXX	ON_NOT_FREE	fork could not switched on, because fork is not free, value is 0
0x01F0XXXX0060XXXX	OFF_NOT_FREE	fork could not switched off, because fork is not free, value is 0

Code	Class
0x0200XXXXXXXXXXXX	HALL_SENSOR
Code	Component
0x02001200XXXXXXXX	HALL_SENSOR_STACKER
0x02001210XXXXXXXX	HALL_SENSOR_SACKHOLE
0x02001220XXXXXXXX	HALL_SENSOR_DEAD_END
0x02001230XXXXXXXX	HALL_SENSOR_SHUTTER

Code	Class
0x0208XXXXXXXXXXXX	HT1381
Code	Component
0x02080100XXXXXXXX	REAL_TIME_CLOCK

Code	Class	
0x0208XXXXXXXXXXXX	HT1381	
Code	Component	
0x02080100XXXXXXXX	REAL_TIME_CLOCK	
Code	ERROR	Meaning
0x0208XXXX0010XXXX	CLOCK_NOT_SET	the clock is not set yet, value is 0

Code	Class
0x0210XXXXXXXXXXXX	INTERRUPT_HANDLER
Code	Component
0x02100020XXXXXXXX	INTERRUPT_CONTROLLER

Code	Class
0x0220XXXXXXXXXX	IO_TABLE
Code	Component
0x02200060XXXXXXXX	IO_TABLE

Code	Class
0x0228XXXXXXXXXX	LED
Code	Component
0x022816F0XXXXXXXX	LED_RIGHT
0x022816F2XXXXXXXX	LED_LEFT

Code	Class
0x022AXXXXXXXXXXXX	LIFTING_COMB
Code	Component
0x022A18B0XXXXXXXX	LIFTING_COMB

Code	ERROR	Meaning
0x022AXXXX0001XXXX	INITIALIZATION_FAILED	serious errors occurred during the initialization, position not reached (photo-sensors or motor defect or not connected or hw damaged)
0x022AXXXX0002XXXX	SELF_TEST_FAILED	serious errors occurred during the initialization, position not reached (photo-sensors or motor defect or not connected or hw damaged)
0x022AXXXX0004XXXX	POSITION_NOT_REACHED	device jammed during movement to target, sensor-cabling defect or not plugged, value holds target position
0x022AXXXX0006XXXX	POSITION_NOT_LEFT	device jammed, sensor-cabling defect or not plugged = device did not leave current position, value holds source position
0x022AXXXX000AXXXX	MOTOR_FAILURE	motor did not start, check motor, cabling and if power is available (24V)
0x022AXXXX0010XXXX	MOVE_TOO_SLOW	a warning, if the time for a movement between the two positions is too low. Maybe the device is soiled

Code	Class
0x0230XXXXXXXXXX	PHOTO_SENSOR
Code	Component
0x02300400XXXXXXXX	SENSOR_STACKER_WHEEL
0x02300410XXXXXXXX	SENSOR_RETRACT_REJECT

0x02300 <b>420</b> XXXXXXXXX	SENSOR_CATCH_LEFT
0x02300 <b>430</b> XXXXXXXXX	SENSOR_END_LEFT
0x02300 <b>440</b> XXXXXXXXX	SENSOR_CATCH_RIGHT
0x02300 <b>450</b> XXXXXXXXX	SENSOR_END_RIGHT
0x02300 <b>460</b> XXXXXXXXX	SENSOR_SHUTTER_OPEN_LOCKED
0x02300 <b>470</b> XXXXXXXXX	SENSOR_SHUTTER_CLOSED
0x02300 <b>480</b> XXXXXXXXX	SENSOR_MAGNET_TO_STACKER
0x02300 <b>490</b> XXXXXXXXX	SENSOR_MAGNET_TO_REJECT
0x02300 <b>4A0</b> XXXXXXXXX	SENSOR_SINGLE_REJECT_TRAY_POSITION
0x02300 <b>510</b> XXXXXXXXX	SENSOR_LIFTING_COMB_COLLECT
0x02300 <b>520</b> XXXXXXXXX	SENSOR_LIFTING_COMB_TRANSPORT
0x02300 <b>530</b> XXXXXXXXX	SENSOR_CONTROL_GEAR_DISPENSE_TRANSPORT
0x02300 <b>540</b> XXXXXXXXX	SENSOR_CONTROL_GEAR_OUTPUT_TRANSPORT
0x02300 <b>710</b> XXXXXXXXX	SENSOR_EMPTY_CASSETTE_1
0x02300 <b>720</b> XXXXXXXXX	SENSOR_EMPTY_CASSETTE_2
0x02300 <b>730</b> XXXXXXXXX	SENSOR_EMPTY_CASSETTE_3
0x02300 <b>740</b> XXXXXXXXX	SENSOR_EMPTY_CASSETTE_4
0x02300 <b>750</b> XXXXXXXXX	SENSOR_EMPTY_CASSETTE_5
0x02300 <b>760</b> XXXXXXXXX	SENSOR_EMPTY_CASSETTE_6
0x02300 <b>770</b> XXXXXXXXX	SENSOR_EMPTY_CASSETTE_7
0x02300 <b>780</b> XXXXXXXXX	SENSOR_EMPTY_CASSETTE_8
0x02300 <b>805</b> XXXXXXXXX	SENSOR_TRANSPORT_2
0x02300 <b>825</b> XXXXXXXXX	SENSOR_TRANSPORT_22
0x02300 <b>830</b> XXXXXXXXX	SENSOR_TRANSPORT_26
0x02300 <b>835</b> XXXXXXXXX	SENSOR_TRANSPORT_27
0x02300 <b>840</b> XXXXXXXXX	SENSOR_TRANSPORT_28
0x02300 <b>845</b> XXXXXXXXX	SENSOR_TRANSPORT_33
0x02300 <b>850</b> XXXXXXXXX	SENSOR_TRANSPORT_35

Code	Class	
0x0 <b>240</b> XXXXXXXXXXXXX	MAGNET_WITH_SENSOR	
Code	Component	
0x02400 <b>910</b> XXXXXXXXX	MAGNET_SINGLE_REJECT	
0x02400 <b>B20</b> XXXXXXXXX	MAGNET_RR_BOX	
Code	ERROR	Meaning
0x0240XXXX <b>0010</b> XXXX	TEST_FAILED	the test of the magnet failed, value holds the state of the magnet.

Code	Class
0x0250XXXXXXXXXX	MAGNET
Code	Component
0x02500900XXXXXXXX	MAGNET_FOR_SINGLE_REJECT
0x02500920XXXXXXXX	MAGNET_REJECT_RETRACT
0x02500A10XXXXXXXX	MAGNET_CASSETTE_1
0x02500A20XXXXXXXX	MAGNET_CASSETTE_2
0x02500A30XXXXXXXX	MAGNET_CASSETTE_3
0x02500A40XXXXXXXX	MAGNET_CASSETTE_4
0x02500A50XXXXXXXX	MAGNET_CASSETTE_5
0x02500A60XXXXXXXX	MAGNET_CASSETTE_6
0x02500A70XXXXXXXX	MAGNET_CASSETTE_7
0x02500A80XXXXXXXX	MAGNET_CASSETTE_8
0x02500A90XXXXXXXX	ALL_CASSETTE_MAGNETS

Code	Class
0x0260XXXXXXXXXX	MAIN_MOTOR
Code	Component
0x02601810XXXXXXXX	MAIN_MOTOR

Code	ERROR	Meaning
0x0260XXXX0010XXXX	START_TIMEOUT	timeout during motor startup
0x0260XXXX0015XXXX	START_ERROR	start failed, value holds response from controller @see MotorController::Response
0x0260XXXX0020XXXX	STOP_TIMEOUT	timeout during motor slowdown
0x0260XXXX0025XXXX	STOP_ERROR	stop failed, value holds response from controller @see MotorController::Response
0x0260XXXX0030XXXX	OFF_TIMEOUT	timeout during motor off
0x0260XXXX0050XXXX	HW_FAILURE	main motor error occur

Code	Class
0x0270XXXXXXXXXX	MOTOR_CONTROLLER
Code	Component
0x02701820XXXXXXXX	CLAMP_DRIVE_MOTOR
0x02701830XXXXXXXX	CLAMP_BELT_MOTOR
0x02701840XXXXXXXX	SHUTTER_DC_MOTOR
0x02701850XXXXXXXX	SHUTTER_STEPPER_MOTOR
0x02701860XXXXXXXX	SINGLE_REJECT_MOTOR
0x027018A2XXXXXXXX	CONTROL_GEAR_MOTOR

Code	ERROR	Meaning
0x0270XXXX0010XXXX	MOTOR_CONTROLLER_START_NO_OK	error during motor start
0x0270XXXX0020XXXX	MOTOR_CONTROLLER_START_STATE_NO_OK	motor state does not allow to start the motor
0x0270XXXX0030XXXX	MOTOR_CONTROLLER_START_SERIAL_ERROR	serial error during motor start
0x0270XXXX0040XXXX	MOTOR_CONTROLLER_STOP_NO_OK	error during motor stop
0x0270XXXX0050XXXX	MOTOR_CONTROLLER_STOP_SERIAL_ERROR	serial error during motor stop
0x0270XXXX0060XXXX	MOTOR_CONTROLLER_START_LOGICAL_ERROR	logical error during motor start
0x0270XXXX0070XXXX	MOTOR_CONTROLLER_SLOWDOWN_NO_OK	error during motor slowdown
0x0270XXXX0080XXXX	MOTOR_CONTROLLER_SLOWDOWN_SERIAL_ERROR	serial error during motor slowdown
0x0270XXXX0090XXXX	MOTOR_CONTROLLER_OFF_NO_OK	error during motor off
0x0270XXXX0100XXXX	MOTOR_CONTROLLER_OFF_SERIAL_ERROR	serial error during motor off
0x0270XXXX0110XXXX	MOTOR_CONTROLLER_GETHWSTATE_SERIAL_ERROR	serial error during getting hardware state
0x0270XXXX0120XXXX	MOTOR_CONTROLLER_SENDBYTE_SERIAL_ERROR	serial error during sending a byte

Code	Class
0x0278XXXXXXXXXXXX	NMC9346
Code	Component
0x02780E90XXXXXXXX	COLLECTOR_EEPROM

Code	ERROR	Meaning
0x0278XXXX0010XXXX	WRITE_ERROR	write to eeprom failed, value is 0, eeprom could be defect, not connected
0x0278XXXX0020XXXX	READ_ERROR	read from eeprom failed, value is 0, eeprom could be defect, not connected

Code	Class
0x0279XXXXXXXXXX	NMC93468
Code	Component
0x02790D30XXXXXXXX	NMC_EEPROM_1
0x02790D40XXXXXXXX	NMC_EEPROM_2
0x02790D50XXXXXXXX	NMC_EEPROM_3
0x02790D60XXXXXXXX	NMC_EEPROM_4
0x02790D70XXXXXXXX	NMC_EEPROM_5
0x02790D80XXXXXXXX	NMC_EEPROM_6
0x02790D90XXXXXXXX	NMC_EEPROM_7
0x02790DA0XXXXXXXX	NMC_EEPROM_8

Code	ERROR	Meaning
0x0279XXXX0010XXXX	WRITE_ERROR	write to eeprom failed, value is 0, eeprom could be defect, not connected
0x0279XXXX0020XXXX	READ_ERROR	read from eeprom failed, value is 0, eeprom could be defect, not connected

Code	Class
0x0280XXXXXXXXXX	NOTE

Code	Class	
0x0290XXXXXXXXXXXX	OUTPUT_TRANSPORT_CMD_V4	
Code	Component	
0x02901970XXXXXXXX	OUTPUT_TRANSPORT	
Code	ERROR	Meaning
0x0290XXXX0010XXXX	INIT_FROM_UNDEFINED_FAILED	search for the lost clamp failed, value is 0
0x0290XXXX0020XXXX	INIT_CLAMP_ERR1	search for the lost clamp failed, value is 0
0x0290XXXX0030XXXX	INIT_CLAMP_ERR2	search for the lost clamp failed, value is 0
0x0290XXXX0040XXXX	INIT_CLAMP_ERR3	search for the lost clamp failed, value is 0
0x0290XXXX0050XXXX	INIT_CLAMP_ERR4	search for the lost clamp failed, value is 0
0x0290XXXX0060XXXX	INIT_CLAMP_ERR5	search for the lost clamp failed, value is 0
0x0290XXXX0070XXXX	INIT_CLAMP_ERR6	search for the lost clamp failed, value is 0
0x0290XXXX0080XXXX	INIT_CLAMP_ERR7	search for the lost clamp failed, value is 0
0x0290XXXX0090XXXX	INIT_CLAMP_ERR8	search for the lost clamp failed, value is 0
0x0290XXXX00A0XXXX	INIT_CLAMP_ERR9	search for the lost clamp failed, value is 0

0x0290XXXX00B0XXXX	INIT_CIRCLE_FORK_ERR1	circlefork initialization or movement with a lost clamp failed
0x0290XXXX00C0XXXX	INIT_CIRCLE_FORK_ERR2	circlefork initialization or movement with a lost clamp failed
0x0290XXXX00D0XXXX	INIT_CIRCLE_FORK_ERR3	circlefork initialization or movement with a lost clamp failed
0x0290XXXX00F0XXXX	LEAVE_REJECT_POSITION_MOVE_ERR1	circlefork could not move correctly when leaving the reject-position
0x0290XXXX0100XXXX	LEAVE_REJECT_POSITION_MOVE_ERR2	circlefork could not move correctly when leaving the reject-position
0x0290XXXX0110XXXX	LEAVE_REJECT_POSITION_MOVE_ERR3	circlefork could not move correctly when leaving the reject-position
0x0290XXXX0120XXXX	LEAVE_REJECT_POSITION_INIT_ERR	circlefork could not initialize when leaving the reject-position
0x0290XXXX0130XXXX	MOVE_TO_REJECT_POSITION_INIT_ERR1	circlefork could not initialize when driving into reject-position
0x0290XXXX0135XXXX	MOVE_TO_REJECT_POSITION_INIT_ERR2	clamp cannot leave sackhole-position
0x0290XXXX0140XXXX	MOVE_TO_REJECT_POSITION_INIT_ERR3	circlefork could not initialize when driving into reject-position
0x0290XXXX0150XXXX	DRIVE_CLAMP_TO_SENSOR_RECOVER_ERR	driving to sensor failed, clamp is lost between sensors, retry also failed
0x0290XXXX0160XXXX	HANDLE_CLAMP_BETWEEN_SENSORS_INIT_ERR	when driving to sensor, a circlefork error occurred and the circlefork init failed
0x0290XXXX0170XXXX	DRIVE_TO_PROBLEM	driving to sensor failed, a retry is initiated, value is target position (@see ClampTarget)
0x0290XXXX0175XXXX	DRIVE_TO_ERROR	all retries failed when driving to a sensor, value is target position (@see ClampTarget)
0x0290XXXX0178XXXX	DRIVE_TO_EXCEPTION	an exception
0x0290XXXX0180XXXX	OPEN_ERROR	circlefork could not reach the open position when opening the clamp, value is 0
0x0290XXXX0190XXXX	NOT_IN_STACKER	after opening the clamp, the hall-sensor for the stacker-position does not report stacker position
0x0290XXXX01A0XXXX	CLOSE_ERROR	after closing the clamp, the hall-sensor for the stacker-position does not report stacker position
0x0290XXXX01B0XXXX	CLOSE_MOVE_ERROR	circlefork movement for closing the clamp failed
0x0290XXXX01C0XXXX	TRANSPORT_EJECT_DARK_JAM	both photosensors in shutter remain dark for more than 2s when driving money out of clamp (eject) = JAM !
0x0290XXXX01D0XXXX	TRANSPORT_OFFER_DARK_JAM	rear photosensor in shutter remains dark for more than 2s when offering money = JAM !
0x0290XXXX01E0XXXX	TRANSPORT_RETRACT_LIGHT_JAM	rear photosensor in shutter did not become dark within 300ms when retracting the offered money = JAM !
0x0290XXXX01F0XXXX	RECOVER_ERROR	when trying to drive clamp back to start position (in retry), the circlefork initialization failed
0x0290XXXX0200XXXX	CLAMP_NOT_FOUND	in start-up, when driving from reject-position into sack-hole, the clamp does not appear at sack-hole sensor
0x0290XXXX0210XXXX	CIRCLE_FORK_INIT_ERR	in start-up, when driving from reject-



		position into sack-hole, the circlefork initialization failed
0x0290XXXX0220XXXX	TYPE_UNKOWN	the detected type (via cable-coding) is unknown
0x0290XXXX0230XXXX	TRANSPORT_FAILED	belt-movement error, motor or photo-sensor error
0x0290XXXX0240XXXX	PHOTO_SENSOR_ERROR1	photo-sensors could not be switched on or off (internal FW-error), value is 0
0x0290XXXX0250XXXX	PHOTO_SENSOR_ERROR2	photo-sensors could not be switched on or off (internal FW-error), value is 0
0x0290XXXX0260XXXX	TRANSPORT_NOT_IN_SAFE	system is out of the safe, safety switch is open
0x0290XXXX0270XXXX	CLAMP_OPEN_ERROR	clamp cannot unlock, recovery error
0x0290XXXX0280XXXX	CLAMP_CLOSE_ERROR	clamp cannot close, recovery error
0x0290XXXX0290XXXX	CLAMP_TO_HL3_ERROR	cannot move clamp to HL3, recovery error
0x0290XXXX02A0XXXX	CLAMP_TO_HL1_ERROR	cannot move clamp to HL1, recovery error
0x0290XXXX02B0XXXX	CLAMP_TO_HL3_WARNING	driving to sensor HL3 failed, a retry is initiated, value is target position (@see ClampTarget)
0x0290XXXX02C0XXXX	CLAMP_TO_HL1_WARNING	driving to sensor HL1 failed, a retry is initiated, value is target position (@see ClampTarget)

Code	Class
0x0291XXXXXXXXXXXX	OUTPUT_TRANSPORT_HDM
Code	Component
0x02911970XXXXXXXX	OUTPUT_TRANSPORT

Code	ERROR	Meaning
0x0291XXXX0010XXXX	INIT_FAILED	the initialization failed, subcomponents like the lifting-comb, the bundle-reject-fork or the control-gear may have caused the error
0x0291XXXX0020XXXX	DRIVE_TO_STACKER_FAILED	driving the note bundle into the stacker position failed, bundle-reject-fork, control-gear, lifting-comb or photo sensors may have caused the error
0x0291XXXX0030XXXX	DRIVE_TO_SHUTTER_FAILED	driving the note bundle into the shutter position failed, bundle-reject-fork, control-gear, lifting-comb or photo sensors may have caused the error
0x0291XXXX0040XXXX	DRIVE_TO_REJECT_FAILED	driving the note bundle into the reject position failed, bundle-reject-fork, control-gear, lifting-comb or photo sensors may have caused the error
0x0291XXXX0041XXXX	OFFERING_FAILED	driving the note bundle into the offer position failed, main motor, photo sensors or stucked notes may have caused the error
0x0291XXXX0042XXXX	EJECTING_FAILED	ejecting the note bundle failed main motor, photo sensors or stucked notes may have caused the error
0x0291XXXX0043XXXX	REJECTING_FAILED	rejecting offered notes failed, photo sensors or stucked notes may have caused

		the error
0x0291XXXX0050XXXX	PHOTO_SENSORS_FAILED	Photo sensor task failed, no photo sensor access/release (hardware broken?)
0x0291XXXX0060XXXX	MAIN_MOTOR_FAILED	Main motor could not be started, hardware broken?
0x0291XXXX0070XXXX	NOTES_JAM	Unexpected note at photo sensor
0x0291XXXX0080XXXX	LIGHT_JAM	Sensor light timeout, note has not reached the desired sensor in time
0x0291XXXX0090XXXX	DARK_JAM	Sensor dark timeout, note has not left the desired sensor in time
0x0291XXXX00A0XXXX	OFFER_NOTES_FAILED	offering the notes failed, shutter could not open or note bundle stuck in transport
0x0291XXXX00B0XXXX	EJECT_NOTES_FAILED	ejecting the notes failed, shutter could not open or note bundle stuck in transport
0x0291XXXX00C0XXXX	CLOSE_OFFER_FAILED	an error occurred while trying to backtrack the note bundle or closing the shutter
0x0291XXXX00D0XXXX	TRANSPORT_OFFER_LIGHT_JAM	bundle could not be moved to the shutter, bundle is stuck in the transport
0x0291XXXX00E0XXXX	TRANSPORT_EJECT_DARK_JAM	bundle could not be ejected, bundle covers the shutter sensor too long
0x0291XXXX00F0XXXX	TRANSPORT_REJECT_DARK_JAM	bundle could not be rejected, bundle covers the shutter sensor too long
0x0291XXXX0100XXXX	TRANSPORT_FAILED	motor or photo sensors failed while transporting the bundle
0x0291XXXX0110XXXX	CONTROL_GEAR_INIT_ERROR	control gear init error
0x0291XXXX0120XXXX	CONTROL_GEAR_OUTPUT_POS_NOT_REACHED	control-gear cannot reach output-position
0x0291XXXX0130XXXX	CONTROL_GEAR_DISPENSE_POS_NOT_REACHED	control-gear cannot reach dispense-position

Code	Class
0x02A0XXXXXXXXXXXX	PRESSURE_REGULATOR
Code	Component
0x02A01510XXXXXXXX	PRESSURE_REGULATOR_CASSETTE_1
0x02A01520XXXXXXXX	PRESSURE_REGULATOR_CASSETTE_2
0x02A01530XXXXXXXX	PRESSURE_REGULATOR_CASSETTE_3
0x02A01540XXXXXXXX	PRESSURE_REGULATOR_CASSETTE_4
0x02A01550XXXXXXXX	PRESSURE_REGULATOR_CASSETTE_5
0x02A01560XXXXXXXX	PRESSURE_REGULATOR_CASSETTE_6
0x02A01570XXXXXXXX	PRESSURE_REGULATOR_CASSETTE_7
0x02A01580XXXXXXXX	PRESSURE_REGULATOR_CASSETTE_8

Code	ERROR	Meaning
0x02A0XXXX0010XXXX	AD_CONVERTER_LOCKED_IN_UPDATE	ad-converter was locked during updating, so the method returns the last status value, value holds the user who locks the converter
0x02A0XXXX0011XXXX	AD_CONVERTER_LOCKED_IN_GET_PSSURE	ad-converter was locked during getting pressure, so the method returns 0 as pressure value, value holds the user who

		locks the converter
0x02A0XXXX0012XXXX	SELFTEST_ERROR	self test error
0x02A0XXXX0020XXXX	SELECTED_RESOURCE_LOCKED	photosensortask was locked during getting pressure, so the method returns 0 as pressure value, value holds getHandle()
0x02A0XXXX0030XXXX	INVALID_STATE	state-machine of update() used with invalid status, value holds the invalid status
0x02A0XXXX0040XXXX	FIND_CURRENT_FAILED	initializing of the pressure by current incrementing failed, value holds pressure value
0x02A0XXXX0041XXXX	INIT_WITHOUT_CASSETTE_FAILED	initializing of cmos data failed, value holds pressure value
0x02A0XXXX0042XXXX	INIT_WITH_CASSETTE_FAILED	initializing of cmos data with inserted cassette failed, value holds pressure value
0x02A0XXXX0050XXXX	PRESSURE_REGULATION_FAILED	pressure regulation failed, value is 0
0x02A0XXXX0070XXXX	MAX_PRESSURE_TOO_LOW	the maximum pressure reaches not the old work-pressure, value holds the highest pressure value, which was found
0x02A0XXXX0080XXXX	PRESSURE_NOT_FOUND	the necessary pressure could not be reached, value holds the target pressure

Code	Class
0x02B0XXXXXXXXXXXX	PROTOCOL_BYTE

Code	Class
0x02B8XXXXXXXXXXXX	PROTOCOL_NCR_IBM / PROTOCOL_HW
Code	Component
0x02200290XXXXXXXX	PROTOCOL_NCR_IBM

Code	Class
0x02C0XXXXXXXXXXXX	PROTOCOL_CONSOLE
Code	Component
0x02C00280XXXXXXXX	PROTOCOL_WIN32

Code	Class
0x02D0XXXXXXXXXXXX	PROTOCOL_ISO
Code	Component
0x02D00250XXXXXXXX	PROTOCOL_ISO_PC
0x02D00260XXXXXXXX	PROTOCOL_ISO_SCOP

Code	Class	
0x02E0XXXXXXXXXX	REAL_TIME_CLOCK	
Code	Component	
0x02E00100XXXXXXXX	REAL_TIME_CLOCK	
Code	ERROR	Meaning
0x02E0XXXX0010XXXX	CLOCK_NOT_SET	the clock is not set yet, value is 0

Code	Class	
0x02F0XXXXXXXXXX	REJECT_RETRACT_BOX	
Code	Component	
0x02F01620XXXXXXXX	REJECT_RETRACT_BOX	
Code	ERROR	Meaning
0x02F0XXXX0010XXXX	SWITCH_TRAY_FAILED	switch between reject/retract tray via magnet failed, the desired direction is stored in value @see Tray Either the magnet or the photo-sensor is defective or the mechanic is not working properly (e.g. blocked)
0x02F0XXXX0020XXXX	SELFTEST_ERROR	Selftest of the reject-/retract-tray failed

Code	Class
0x02F5XXXXXXXXXX	REJECT_BOX
Code	Component
0x02F51625XXXXXXXX	REJECT_BOX

Code	Class
0x02F8XXXXXXXXXX	ROLLER

Code	Class
0x0300XXXXXXXXXX	RTOS
Code	Component
0x03000040XXXXXXXX	RTOS

Code	Class
0x0310XXXXXXXXXX	SCAN_PHOTO_SENSOR

Code	Component
0x03100610XXXXXXXX	SENSOR_DISPENSE_CASSETTE_1
0x03100620XXXXXXXX	SENSOR_DISPENSE_CASSETTE_2
0x03100630XXXXXXXX	SENSOR_DISPENSE_CASSETTE_3
0x03100640XXXXXXXX	SENSOR_DISPENSE_CASSETTE_4
0x03100650XXXXXXXX	SENSOR_DISPENSE_CASSETTE_5
0x03100660XXXXXXXX	SENSOR_DISPENSE_CASSETTE_6
0x03100670XXXXXXXX	SENSOR_DISPENSE_CASSETTE_7
0x03100680XXXXXXXX	SENSOR_DISPENSE_CASSETTE_8
0x03100800XXXXXXXX	SENSOR_TRANSPORT_1
0x03100810XXXXXXXX	SENSOR_TRANSPORT_18
0x03100815XXXXXXXX	SENSOR_TRANSPORT_19
0x03100820XXXXXXXX	SENSOR_TRANSPORT_21
0x03100835XXXXXXXX	SENSOR_TRANSPORT_27
0x03101940XXXXXXXX	DOUBLE_DETECTION_UNIT

Code	ERROR	Meaning
0x0310XXXX0010XXXX	LIGHT_JAM	a note did not reach the sensor in a given time, value holds
0x0310XXXX0020XXXX	DARK_JAM	a note did not leave the sensor in a given time, value holds the current motor-clock or timestamp
0x0310XXXX0030XXXX	UNEXPECTED_NOTE	an unexpected note was seen at the sensor, value holds the current motor-clock or timestamp
0x0310XXXX0040XXXX	NOTE_EARLY	a note did reach the sensor too early, value holds the expected note arrival in motor-clocks or timestamps
0x0310XXXX0050XXXX	NOTE_LATE	a note did reach the sensor too late, value holds the expected note arrival in motor-clocks or timestamps
0x0310XXXX0100XXXX	NULL_POINTER	the current note-pointer is 0 (internal firmware-error), value is 0
0x0310XXXX0101XXXX	NEW_FAILED	no more memory for a new note (internal firmware-error), value is 0

Code	Class
0x0320XXXXXXXXXXXX	SCOP
Code	Component
0x03204000XXXXXXXX	SCOP

Code	Class
0x0330XXXXXXXXXXXX	Shutter
Code	Component
0x03301910XXXXXXXX	Shutter

Code	ERROR	Meaning
0x0330XXXX0010XXXX	TEST_ERROR	the shutter test failed, value holds the shutter position
0x0330XXXX0020XXXX	SELFTTEST_ERROR	the shutter self test failed
0x0330XXXX0030XXXX	OPEN_ERROR	the opening of the shutter failed, value holds the shutter position
0x0330XXXX0040XXXX	CLOSE_ERROR	the closing of the shutter failed, value holds the shutter position
0x0330XXXX0050XXXX	OFFER_ERROR	the offering of the money failed, value holds the shutter position
0x0330XXXX0060XXXX	TIMEOUT_WHILE_TESTING	timeout during testing the shutter, value holds shutter position
0x0330XXXX0070XXXX	NOT_INSTALLED_OR_UNKOWN	type detect error, the shutter is not installed or unknown, value is 0
0x0330XXXX0080XXXX	TIMEOUT_WHILE_LEAVING_POSITION	the shutter left a position and found the target position not in time. value holds the position of the shutter
0x0330XXXX0081XXXX	CLOSE_BLOCKED	just a warning that the shutter was blocked during closing, if shutter is always blocked after the retries, it comes to an CLOSE_ERROR. Value holds number of retries

Code	Class	
0x0331XXXXXXXXXXXX	CMD_V4_IN_SAFE_SHUTTER	
Code	Component	
0x03311910XXXXXXXX	Shutter	
Code	ERROR	Meaning
0x0331XXXX0010XXXX	TEST_ERROR	the shutter test failed, value holds the shutter position
0x0331XXXX0011XXXX	SELFTTEST_ERROR	the shutter self test failed
0x0331XXXX0020XXXX	OPEN_ERROR	the opening of the shutter failed, value holds the shutter position
0x0331XXXX0030XXXX	CLOSE_ERROR	the closing of the shutter failed, value holds the shutter position
0x0331XXXX0040XXXX	OFFER_ERROR	the offering of the money failed, value holds the shutter position
0x0331XXXX0050XXXX	TIMEOUT_WHILE_TESTING	timeout during testing the shutter, value holds shutter position
0x0331XXXX0060XXXX	NOT_INSTALLED_OR_UNKOWN	type detect error, the shutter is not installed or unknown, value is 0
0x0331XXXX0070XXXX	TIMEOUT_WHILE_LEAVING_POSITION	the shutter left a position and found the target position not in time. value holds the position of the shutter
0x0331XXXX0080XXXX	CLOSE_BLOCKED	just a warning that the shutter was blocked during closing, if shutter is always blocked after the retries, it comes to an CLOSE_ERROR. Value holds number of retries
0x0331XXXX0081XXXX	OPEN_BLOCKED	just a warning that the shutter was blocked during opening, if shutter is always blocked after the retries, it comes to an OPEN_ERROR. Value holds number of retries
0x0331XXXX0082XXXX	POSITION_ERROR	unknown shutter position, possible broken sensors

Code	Class	
0x0332XXXXXXXXXX	CMD_V4_OUT_SAFE_SHUTTER	
Code	Component	
0x03321910XXXXXXXX	Shutter	
Code	ERROR	Meaning
0x0332XXXX0010XXXX	TEST_ERROR	the shutter test failed, value holds the shutter position
0x0332XXXX0011XXXX	SELFTEST_ERROR	the shutter self test failed
0x0332XXXX0020XXXX	OPEN_ERROR	the opening of the shutter failed, value holds the shutter position
0x0332XXXX0030XXXX	CLOSE_ERROR	the closing of the shutter failed, value holds the shutter position
0x0332XXXX0040XXXX	OFFER_ERROR	the offering of the money failed, value holds the shutter position
0x0332XXXX0050XXXX	TIMEOUT_WHILE_TESTING	timeout during testing the shutter, value holds shutter position
0x0332XXXX0060XXXX	NOT_INSTALLED_OR_UNKOWN	type detect error, the shutter is not installed or unknown, value is 0
0x0332XXXX0070XXXX	TIMEOUT_WHILE_LEAVING_POSITION	the shutter left a position and found the target position not in time. value holds the position of the shutter
0x0332XXXX0080XXXX	CLOSE_BLOCKED	just a warning that the shutter was blocked during closing, if shutter is always blocked after the retries, it comes to an CLOSE_ERROR. Value holds number of retries
0x0332XXXX0081XXXX	OPEN_BLOCKED	just a warning that the shutter was blocked during opening, if shutter is always blocked after the retries, it comes to an OPEN_ERROR. Value holds number of retries
0x0332XXXX0082XXXX	POSITION_ERROR	unknown shutter position, possible broken sensors

Code	Class	
0x0333XXXXXXXXXXXX	HDM_SHUTTER	
Code	Component	
0x03331910XXXXXXXX	Shutter	
Code	ERROR	Meaning
0x0333XXXX0010XXXX	TEST_ERROR	the shutter test failed, value holds the shutter position
0x0333XXXX0011XXXX	SELFTEST_ERROR	the shutter self test failed
0x0333XXXX0020XXXX	OPEN_ERROR	the opening of the shutter failed, value holds the shutter position
0x0333XXXX0030XXXX	CLOSE_ERROR	the closing of the shutter failed, value holds the shutter position
0x0333XXXX0040XXXX	OFFER_ERROR	the offering of the money failed, value holds the shutter position
0x0333XXXX0050XXXX	TIMEOUT_WHILE_TESTING	timeout during testing the shutter, value holds shutter position

0x0333XXXX0060XXXX	NOT_INSTALLED_OR_UNKOWN	type detect error, the shutter is not installed or unknown, value is 0
0x0333XXXX0070XXXX	TIMEOUT_WHILE_LEAVING_POSITION	the shutter left a position and found the target position not in time. value holds the position of the shutter
0x0333XXXX0080XXXX	CLOSE_BLOCKED	just a warning that the shutter was blocked during closing, if shutter is always blocked after the retries, it comes to an CLOSE_ERROR. Value holds number of retries
0x0333XXXX0081XXXX	OPEN_BLOCKED	just a warning that the shutter was blocked during opening, if shutter is always blocked after the retries, it comes to an OPEN_ERROR. Value holds number of retries
0x0333XXXX0082XXXX	POSITION_ERROR	unknown shutter position, possible broken sensors

Code	Class
0x0334XXXXXXXXXXXX	SHUTTER_PROTECT_PC8xxx
Code	Component
0x033419A0XXXXXXXX	SHUTTER_PROTECT

Code	ERROR	Meaning
0x0334XXXX0010XXXX	PROTECT_COM_ERROR	the communication to the shutter protect failed (communication defect or not connected)
0x0334XXXX0015XXXX	PROTECT_INIT_FAILD	the shutter protect initialization failed (shutter protect hw not initialization or hw damaged)
0x0334XXXX0020XXXX	PROTECT_MEASUREMEND_WRONG	shutter protect measurement wrong
0x0334XXXX0025XXXX	PROTECT_MANIPULATION	shutter protect detected manipulation
0x0334XXXX0030XXXX	PROTECT_WRONG	the shutter protect detect front building

Code	Class
0x0335XXXXXXXXXXXX	SHUTTER_PROTECT_IR_LS
Code	Component
0x033519A0XXXXXXXX	SHUTTER_PROTECT

Code	ERROR	Meaning
0x0335XXXX0010XXXX	PROTECT_COM_ERROR	the communication to the shutter protect failed (communication defect or not connected)
0x0335XXXX0015XXXX	PROTECT_INIT_FAILD	the shutter protect initialization failed (shutter protect hw not initialization or hw damaged)
0x0335XXXX0020XXXX	PROTECT_MEASUREMEND_WRONG	shutter protect measurement wrong
0x0335XXXX0025XXXX	PROTECT_MANIPULATION	shutter protect detected manipulation
0x0335XXXX0030XXXX	PROTECT_WRONG	the shutter protect detect front building



Code	Class	
0x0340XXXXXXXXXX	SINGLE_REJECT_TRAY	
Code	Component	
0x03401900XXXXXXXX	SINGLE_REJECT_TRAY	
Code	ERROR	Meaning
0x0340XXXX0010XXXX	INIT_FAILED	the init of the single reject tray failed, value is 0
0x0340XXXX0020XXXX	IS_EMPTY_ERROR	isEmpty() throws exception, value is 0
0x0340XXXX0030XXXX	POSITION_ERROR	unknown target position

Code	Class	
0x0350XXXXXXXXXX	SINGLE_REJECT_MOTOR	
Code	Component	
0x03501860XXXXXXXX	SINGLE_REJECT_TRAY	
Code	ERROR	Meaning
0x0350XXXX0010XXXX	INIT_FAILED	initial sequence failed
0x0350XXXX0020XXXX	DRIVE_TO_STANDARD_TRAY_FAILED	driving to standard tray failed
0x0350XXXX0030XXXX	DRIVE_TO_RESERVE_TRAY_FAILED	driving to reserve tray failed
0x0350XXXX0040XXXX	DRIVE_TO_EMPTY_POSITION_FAILED	driving to empty position failed
0x0350XXXX0050XXXX	WAIT_FOR_COMPLETION_FAILED	timeout during waiting for completion

Code	Class	
0x0360XXXXXXXXXX	STACKER_WHEEL	
Code	Component	
0x03601880XXXXXXXX	STACKER_WHEEL	
Code	ERROR	Meaning
0x0360XXXX0010XXXX	STEP_TIMEOUT	timeout during set execution
0x0360XXXX0020XXXX	STEP_MOTOR_ERROR	motor error while driving a step
0x0360XXXX0030XXXX	SCAN_STEP_FAILED	scan step execution failed
0x0360XXXX0040XXXX	INIT_FAILED	initial sequence failed
0x0360XXXX0050XXXX	NULL_POINTER	null pointer exception

Code	Class
0x0370XXXXXXXXXX	STATE_MACHINE
Code	Component
0x03704020XXXXXXXX	STATE_MACHINE

Code	ERROR	Meaning
0x0370XXXX0010XXXX	STACKER_NOT_FREE	unexpected notes in stacker
0x0370XXXX0020XXXX	CLAMP_IS_EMPTY	general error
0x0370XXXX0022XXXX	CLAMP_NOT_EMPTY	general error
0x0370XXXX0030XXXX	SHUTTER_NOT_EMPTY	general error
0x0370XXXX0040XXXX	SHUTTER_CLOSED	general error
0x0370XXXX0042XXXX	SHUTTER_NOT_OPEN	general error
0x0370XXXX0044XXXX	SHUTTER_NOT_CLOSED	general error
0x0370XXXX0050XXXX	CLAMP_OPEN_FAILED	general error
0x0370XXXX0060XXXX	PRESENT_FAILED	general error
0x0370XXXX0070XXXX	REJECT_RETRACT_FAILED	general error
0x0370XXXX0072XXXX	REJECT_FAILED	general error
0x0370XXXX0074XXXX	RETRACT_FAILED	general error
0x0370XXXX0080XXXX	CLEAR_SINGLE_REJECT_FAILED	general error
0x0370XXXX0090XXXX	DEVICE_LOCK_ERROR	general error
0x0370XXXX0100XXXX	DRIVE_TO_SHUTTER_FAILED	general error
0x0370XXXX0102XXXX	DRIVE_TO_STACKER_FAILED	general error
0x0370XXXX0104XXXX	DRIVE_TO_REJECT_FAILED	general error
0x0370XXXX0200XXXX	TEST_DISPENSE_FAILED	general error

Code	Class
0x0372XXXXXXXXXXXX	CLEARING_MACHINE
Code	Component
0x03722002XXXXXXXX	CLEARING_MACHINE

Code	ERROR	Meaning
0x0372XXXX0010XXXX	START_CLEARING_FAILED	due to open security switch
0x0372XXXX0015XXXX	POWER_NOT_ENABLED	power is not enabled
0x0372XXXX0020XXXX	CLEAR_INT0_BOX_ERROR	context: see previous errors for true reason
0x0372XXXX0030XXXX	CLEAR_OUTPUT_TRANSPORT_ERROR	context: see previous errors for true reason
0x0372XXXX0040XXXX	DEVICE_LOCKED	please reset device lock
0x0372XXXX0050XXXX	CLEAR_SINGLE_REJECT_ERROR	context: see previous errors for true reason
0x0372XXXX0060XXXX	CHECK_TRANSPORT_ERROR	context: see previous errors for true reason
0x0372XXXX0070XXXX	REJECT_FAILED	context: see previous errors for true reason
0x0372XXXX0080XXXX	RETRACT_FAILED	context: see previous errors for true reason
0x0372XXXX0090XXXX	DRIVE_TO_REJECT_FAILED	context: see previous errors for true reason

0x0372XXXX00A0XXXX	CLAMP_OPEN_FAILED	clamp could not open (stepper, photo-sensor or clamp defect)
0x0372XXXX00B0XXXX	DRIVE_TO_STACKER_FAILED	context: see previous errors for true reason
0x0372XXXX00C0XXXX	DISPENSER_INIT_FAILED	hardware failure in dispenser

Code	Class	
0x0374XXXXXXXXXXXX	STANDARD_MACHINE	
Code	Component	
0x03742001XXXXXXXXXX	STANDARD_MACHINE	
Code	ERROR	Meaning
0x0374XXXX0001XXXX	INVALID_COMMAND	command not allowed in this state
0x0374XXXX0010XXXX	DISPENSE_FAILED	context: see previous errors for true reason
0x0374XXXX0011XXXX	SINGLE_REJECT_FORK_BLOCKED	single reject fork is blocked in reject position
0x0374XXXX0020XXXX	OPEN_SHUTTER_FAILED	shutter error
0x0374XXXX0022XXXX	CLOSE_SHUTTER_FAILED	shutter error
0x0374XXXX0024XXXX	TEST_SHUTTER_FAILED	shutter error
0x0374XXXX0030XXXX	CLAMP_IN_SHUTTER_EMPTY	no notes in clamp
0x0374XXXX0040XXXX	STACKER_NOT_EMPTY	unexpected notes in stacker
0x0374XXXX0050XXXX	DRIVE_TO_SHUTTER_FAILED	context: see previous errors for true reason
0x0374XXXX0052XXXX	DRIVE_TO_STACKER_FAILED	context: see previous errors for true reason
0x0374XXXX0054XXXX	DRIVE_TO_REJECT_FAILED	context: see previous errors for true reason
0x0374XXXX0060XXXX	PRESENT_FAILED	context: see previous errors for true reason
0x0374XXXX0061XXXX	PRESENT_FROM_STACKER_FAILED	context: see previous errors for true reason
0x0374XXXX0062XXXX	EJECT_FAILED	context: see previous errors for true reason
0x0374XXXX0070XXXX	CLOSE_OFFER_FAILED	context: see previous errors for true reason
0x0374XXXX0080XXXX	CLAMP_OPEN_FAILED	context: see previous errors for true reason
0x0374XXXX0090XXXX	CLAMP_NOT_EMPTY	clamp should be empty
0x0374XXXX00A0XXXX	SHUTTER_NOT_EMPTY	shutter should be empty
0x0374XXXX00B0XXXX	RETRACT_FAILED	context: see previous errors for true reason
0x0374XXXX00C0XXXX	REJECT_FAILED	context: see previous errors for true reason

Code	Class
0x0376XXXXXXXXXXXX	ERROR_MACHINE

Code	ERROR	Meaning
0x0376XXXX0010XXXX	RESET_FAILED	reset failed

Code	Class	
0x0378XXXXXXXXXXXX	CONTROL_MACHINE	
Code	Component	
0x03782000XXXXXXXX	CONTROL_MACHINE	
Code	ERROR	Meaning
0x0378XXXX0010XXXX	TOO_MANY_FAILED_RESETS	failed resets are limited to 10
0x0378XXXX0020XXXX	UNCAUGHT_EXCEPTION	an unexpected exception occurred

Code	Class
0x0380XXXXXXXXXXXX	STATUS_CODE_DISPLAY
Code	Component
0x03800D20XXXXXXXX	STATUS_CODE

Code	Class
0x0390XXXXXXXXXXXX	STEPPER

Code	Class	
0x0392XXXXXXXXXXXX	TIMER_STEPPER	
Code	Component	
0x03921010XXXXXXXX	STEPPER_CASSETTE_1	
0x03921020XXXXXXXX	STEPPER_CASSETTE_2	
0x03921030XXXXXXXX	STEPPER_CASSETTE_3	
0x03921040XXXXXXXX	STEPPER_CASSETTE_4	
0x03921050XXXXXXXX	STEPPER_CASSETTE_5	
0x03921060XXXXXXXX	STEPPER_CASSETTE_6	
0x03921070XXXXXXXX	STEPPER_CASSETTE_7	
0x03921080XXXXXXXX	STEPPER_CASSETTE_8	
0x03921100XXXXXXXX	STEPPER_23	
0x039218B2XXXXXXXX	STEPPER_5	
0x039218C2XXXXXXXX	STEPPER_6	
Code	ERROR	Meaning
0x0392XXXX0010XXXX	DMA0_LOCKED	dma channel was locked
0x0392XXXX0020XXXX	START_FAILED	reasons can be: DMA0_LOCKED, invalid parameters (e.g. 0)

Code	Class
0x03A0XXXXXXXXXX	SWITCH
Code	Component
0x03A01600XXXXXXXX	REJECT_RETRACT_BOX_SWITCH
0x03A01610XXXXXXXX	IN_SAFE_SWITCH
0x03A01630XXXXXXXX	DOOR_SWITCH
0x03A01640XXXXXXXX	DOOR_ENABLE_SWITCH
0x03A01650XXXXXXXX	SAFETY_SWITCH
0x03A01660XXXXXXXX	SAFETY_SWITCH_FLAG
0x03A01670XXXXXXXX	JUMPER_DOOR
0x03A01680XXXXXXXX	JUMPER_SCOP
0x03A01690XXXXXXXX	JUMPER_CLEAR_CMOS
0x03A01698XXXXXXXX	JUMPER_NEN
0x03A016A0XXXXXXXX	JUMPER_A
0x03A016B0XXXXXXXX	JUMPER_B
0x03A016C0XXXXXXXX	JUMPER_C
0x03A016D0XXXXXXXX	JUMPER_D

Code	Class
0x03B0XXXXXXXXXXXX	SYSTEM
Code	Component
0x03B00010XXXXXXXX	SYSTEM_V4
Code	ERROR
0x03B0XXXX0020XXXX	RESPONSE_NOT_SEND

Code	Class
0x03C0XXXXXXXXXX	TASK

Code	Class
0x03C2XXXXXXXXXXXX	HARDWARE_TASK
Code	Component
0x03C23030XXXXXXXX	HARDWARE_TASK

Code	ERROR	Meaning
0x03C2XXXX0010XXXX	MACHINE_STATE_INVALID	unknown state
0x03C2XXXX0020XXXX	DEVICE_STATE_INVALID	unknown value
0x03C2XXXX0030XXXX	COMP_STATE_INVALID	invalid cassette state
0x03C2XXXX0040XXXX	Manipulation	Manipulation

0x03C2XXXX0100XXXX	PHOTO_SENSOR_ERROR	general error
0x03C2XXXX0110XXXX	NO_CASSETTES	general error
0x03C2XXXX0120XXXX	NO_OUTPUT_TRANSPORT	invalid code on output-transport connector (or not connected)
0x03C2XXXX0130XXXX	HARDWARE_DEFECT	hardware defect
0x03C2XXXX0140XXXX	ERROR_RR_MAGNET	magnet MA6 ( reject/retract-tray ) blocked or damaged
0x03C2XXXX0141XXXX	PLD_VERSION_ERROR	Invalid PLD Version
0x03C2XXXX0150XXXX	ERROR_SR_MAGNET	magnet MA2 ( singlereject ) blocked or damaged
0x03C2XXXX0200XXXX	MAILBOX_CREATE_ERROR	general error
0x03C2XXXX0210XXXX	RESPONSE_NOT_SEND	internal error
0x03C2XXXX0300XXXX	UNCAUGHT_EXCEPTION	internal error

Code	Class	
0x03C4XXXXXXXXXXXX	PHOTO_SENSOR_TASK	
Code	Component	
0x03C43000XXXXXXXX	PHOTO_SENSOR_TASK	
Code	ERROR	Meaning
0x03C4XXXX0010XXXX	START_SENSORS_NO_OK	hw-failure
0x03C4XXXX0020XXXX	INIT_SENSORS_NO_OK	init failed, hw-failure
0x03C4XXXX0030XXXX	ON_NO_OK	internal error
0x03C4XXXX0040XXXX	ON_CASSNR_NO_OK	internal error
0x03C4XXXX0050XXXX	RELEASE_NO_OK	internal error
0x03C4XXXX0060XXXX	OFF_NO_OK	internal error
0x03C4XXXX0070XXXX	SUSPEND_NO_OK	internal error
0x03C4XXXX0080XXXX	RESUME_NO_OK	internal error
0x03C4XXXX0090XXXX	ADJUST_NO_OK	internal error
0x03C4XXXX0100XXXX	ADAPT_NO_OK	internal error
0x03C4XXXX0110XXXX	SEND_RECEIVE_STATE	internal error
0x03C4XXXX0120XXXX	SEND_RECEIVE_SENDER_NO_OK	internal error
0x03C4XXXX0130XXXX	SEND_RECEIVE_NO_RESPONSE	internal error
0x03C4XXXX0140XXXX	INTERNAL_ERROR	internal error
0x03C4XXXX0150XXXX	COVERED_PHOTOSENSOR	internal error
0x03C4XXXX0160XXXX	PHOTOSENSOR_HW_FAILED	internal error

Code	Class
0x03C6XXXXXXXXXXXX	COMMAND_TASK

Code	Component
0x03C63020XXXXXXXX	COMMAND_TASK

Code	Class
0x03C8XXXXXXXXXXXX	LINE_TASK
Code	Component
0x03C83010XXXXXXXX	LINE_TASK

Code	Class
0x03CAXXXXXXXXXXXX	SCAN_TASK
Code	Component
0x03CA3040XXXXXXXX	SCAN_TASK

Code	Class
0x03CBXXXXXXXXXXXX	CONTROLLER_TASK
Code	Component
0x03CB3050XXXXXXXX	CONTROLLER_TASK

Code	ERROR	Meaning
0x03CBXXXX0010XXXX	SEND_TO_DEVICE_ERROR	sending the data to the remote device failed, the device is out of operation or the communication is damaged
0x03CBXXXX0020XXXX	RECEIVE_FROM_DEVICE_ERROR	receiving data from the remote device failed, the device is out of operation or the communication is damaged

Code	Class
0x03CCXXXXXXXXXXXX	SELF_TEST
Code	Component
0x03CC4040XXXXXXXX	SELF_TEST

Code	ERROR	Meaning
0x03CCXXXX0001XXXX	BOARD_TEST_FAILED	controller self test failed
0x03CCXXXX0002XXXX	RAM_TEST_FAILED	RAM self test failed
0x03CCXXXX0010XXXX	BCD_TEST_FAILED	Bcd self test failed
0x03CCXXXX0020XXXX	INTERRUPT_CONTROLLER_TEST_FAILED	interrupt controller self test failed
0x03CCXXXX0030XXXX	THREAD_TEST_FAILED	thread self test failed
0x03CCXXXX0040XXXX	SEMAPHORE_TEST_FAILED	Semaphore self test failed

Code	Class	
0x03D0XXXXXXXXXX	TIMER188	
Code	Component	
0x03D00110XXXXXXXX	TIMER0	
0x03D00111XXXXXXXX	TIMER1	
0x03D00112XXXXXXXX	TIMER2	
Code	ERROR	Meaning
0x03D0XXXX0001XXXX	TIMER_NOT_FREE	timer start failed, because already in use, value is mode (@see Mode)

Code	Class
0x03E0XXXXXXXXXX	TRACE
Code	Component
0x03E00050XXXXXXXX	TRACE

Code	Class
0x03F0XXXXXXXXXX	USB

Code	Class
0x0400XXXXXXXXXX	SEVEN_SEGMENT_DISPLAY
Code	Component
0x04000D00XXXXXXXX	DISPLAY_LOW
0x04000D10XXXXXXXX	DISPLAY_HIGH

Code	Class
0x0410XXXXXXXXXX	LOGICAL_USB
Code	Component
0x04100210XXXXXXXX	PC_USB
0x04100220XXXXXXXX	DEBUG_USB

Code	Class	
0x0500XXXXXXXXXX	POWER_MANAGER	
Code	Component	
0x05000001XXXXXXXX	POWER_MANAGER	
Code	ERROR	Meaning
0x0500XXXX0010XXXX	IO_LOG	set if it is necessary, after critical errors, to log the io-flags



## Appendix V: DQC – Error stack

### **Before firmware release 16.20:**

The error stack contains the device status, cassette status and photosensor values for each command that was acknowledged with status 'E' or 'D'. For reasons of storage space and performance, only the useful data is saved and transmitted; code words, separators and similar characters that do not contain any data are not.

The error stack holds maximum of 10 entries, with the oldest being overwritten as soon as it is full.

char year[2];
char dot;
char month[2];
char dot2 ;
char date[2] ;
char comma ;
char hour[2];
char doubleDot;
char minute[2];
char doubleDot2;
char second[2];
char comma1;
char len[4];
char sSw[1];
char dLoc[1];
char cas[1];
char shErr[1];
char shut[1];
char mon[1];
char ter[1] ;
char ts[2] ;
char tf[1] ;
char sr[1] ;
char dis[1] ;
char cex[1] ;
char door[1];
char doS[1];

char type[1];
char nVm[1];
char sedm[1];
char lCmd[2];
char lSta[1];
char sCle[1];
char sRes[1];
char tst[1];
char sCod[2];
char or[1];
char transport[1];
char cOut[1];
char error[16];
char warning[16];
char len[4];
char rSta[1];
char rAct[4];
char rRet[2];
struct Cassette
{ // the following structure is repeated 8 times (for all cassettes supported)
char sta[1];
char num[7];
char cur[3];
char rel[4];
char val[8];
char len[3];
char tol[2];
char act[4];
char ndv[4];
char low[4];
char ld[2];
char rej[3];
} cassettes[Dispenser ::MAX_CASSETTES];
char photoSensorData[40]

**From firmware release 16.20 and higher:**

The error stack contains the date, time, firmware ID, device status, and the first 10 error and warning codes (each 16 Byte) for each command answered with the acknowledgment status 'E' or 'D'. For reasons of storage space and performance, only the useful data is saved and transmitted; code words and most separators are not. The error stack holds maximum of 10 entries, with the oldest being overwritten as soon as it is full.

Since the error stack has been modified during firmware development, but older error entries should not be lost during a FW update, the size of a total error stack entry is maybe larger than the actual amount of reference data. This is recognized by the filler bytes (Hex 0x00) at the end of the data.

**Structure**

char	year[2]
char	dot
char	month[2]
char	dot2
char	day[2]
char	comma
char	hour[2]
char	doubleDot
char	minute[2]
char	doubleDot
char	second[2]
char	comma
char	Firmware ID[61] (Booter and FW ID as under DIL)
char	comma
char	len[4]
char	sSw[1]
char	dLoc[1]
char	cas[1]
char	shErr[1]
char	shut[1]
char	mon[1]
char	ter[1]
char	ts[2]
char	tf[1]

char sr[1]
char dis[1]
char cex[1]
char door[1]
char doS[1]
char type[1]
char nVm[1]
char sedm[1]
char lCmd[2]
char lSta[1]
char sCle[1]
char sRes[1]
char tst[1]
char sCod[2]
char or[1]
char transport[1]
char cOut[1]
char shutProtect[1]
char comma
char error0[16]
char doubleDot
: :
char error8[16]
char doubleDot
char error9[16]
char comma
char warning0[16]
char doubleDot
: :
char warning8[16]
char doubleDot
char warning9[16]

## Appendix X: Status areas

All statuses are described via code words and value. Code words are 2-10 characters (bytes) long. The upper limit in AZM-NG was 8.

### D – Device status

Code word	Entry	Meaning	
LEN=	dddd,	Total length (incl. length specification)	(decimal in ASCII)
S_SW=	a,	Safety switch C = switched O = open	(ASCII)
DLOC=	a,	Device lock after note retract N = no Y = yes, set	(ASCII)
CAS=	a,	Cassettes can be processed N = no Y = yes	(ASCII)
SHERR=	a,	Error on shutter B = blocked or cash trapping detection error N = no error	(ASCII)
SHUT=	a,	Shutter status C = closed M = missing O = open U = undefined or cash trapping detection error	(ASCII)
MON=	a,	Cash output status N = Cash not accessible to customer P = Cash accessible to customer R = Retract W = Cash removed	(ASCII)
TER=	a,	Transport error J = JAM M = Manipulation at output sensor O = OK	(ASCII)

Code word	Entry	Meaning	
TS=	aa,	Transport status (byte legend for transports: 1. Byte is the start position and the 2nd byte the target position) 0 = No notes existing 1 = Cassette 2 = Stacker 3 = Reject 4 = Stop over (output direction) 5 = Output position 8 = Retract 9 = Undefined position	(ASCII)
TF=	a,	Transport – free (transport status of the last device command) N = Cash paths were not free Y = Free cash paths and shutter closed	(ASCII)
SR=	a,	Status of single reject switch O = OK R = Defective (transport direction reject) S = Defective (transport direction collecting tray)	(ASCII)
DIS=	a,	Device error during dispensing E = Too many bundle rejects J = Repeat of 'Additional dispense' not possible M = Too many notes with incorrect dimensions N = Note dispensing not possible O = OK S = To many multiple dispenses	(ASCII)
CEX=	a,	Cassette inventory changed N = no Y = yes	(ASCII)
DOOR=	a,	Door switch status C = closed M = missing O = open	(ASCII)
DO_S=	a,	Observe door contact during encryption N = no Y = yes	(ASCII)

Code word	Entry	Meaning	
TYPE=	a,	Device type U = Undefined F = Frontload R = Rearload	(ASCII)
N_VM=	a,	Number of dispenser modules 1 - 6	(decimal in ASCII)
SEDM=	a,	SW EDM I = IV request necessary L = Line code missing O = Encryption active T = Transport code missing	(ASCII)
LCMD=	aa,	Specification and additional specification of the last command	(ASCII)
LSTA=	a,	Status code of the last command	(ASCII)
SCLE=	a,	Clearing Status (after RESET / Power up) 0 = Transport paths were free 1 = No cash accessible to customer; Reject OK 2 = Cash was accessible to customer; Retract OK 3 = Cash was accessible to customer. Meanwhile they were removed. 4 = Cash was not accessible to customer. An error occurred during clearing. (See SRES) 5 = Cash was accessible to customer. An error occurred during clearing. (See SRES)	(ASCII)
SRES=	a,	Status code after RESET / Clearing / Power up	(ASCII)
TST=	a,	Self-test was active N = no Y = yes	(ASCII)
SCOD=	aa,	Current status display	(ASCII)
OR=	a,	Operator Request N = no W = Warning Y = yes	(ASCII)

Code word	Entry	Meaning	
TRANSPORT =	a,	Output transport configuration A = in safe front B = in safe rear C = out safe rear D = out safe front E = out safe rear, PC8xxx - system F = out safe front, PC8xxx - system G = in safe rear, PC8xxx - system H = not yet realized for the CMD family U = ProCash 5100 (VCMD) * = unknown coding	(ASCII)
C_OUT=	a,	CashOut available C = Minimum configuration missing (RR box + cash-out cassette) J = JAM (note jam) L = device lock M = Main motor DCM1 defective P = Start-up phase S = Safety switch T = Clamp transport defective Y = CashOut possible	(ASCII)
SHUT_PROT ECT=	a,	Cash trapping detection O = OK, no error K = Cash trapping device detected M = Cash trapping device manipulation (e.g. color spraying) D = Cash trapping detection defective (e.g. no communication, not initialized, faultily initialized, implausibly initialized) U = Cash trapping detection not present (see shutter plug for coding)	(ASCII)
ERROR=	dddddd dddddd dddd,	Error number (digit 1 – 4 = class, 5 – 8 = object, 9 – 12 = error code, 13 – 16 = error value)	(hexadecimal in ASCII)
WARNING=	dddddd dddddd ,	Warning number (cf. interpretation of ERROR above)	(hexadecimal in ASCII)
	;	End character after last parameter	(ASCII)

The contents of the two cells are only changed by commands that initiate device operations. (Reset, dispense transaction, ...)



**Definitions**

- LEN** Contains the total length of the status range (incl. length specification).
- S\_SW** Status of the safety switch (is ascertained again for every device command with a specification A, B, C and G).  
C = Closed; device in operating position  
O = Open; device not in operating position
- DLOC** Device lock. Is set after DCQ (cash retained but not stored). Device lock enabled as soon as the clamp has reached the stacking position after retract (output position for the operator). If the clamp is not in the stacking position during a reset/clearing with a set device lock, it is moved to this position and opened. Any available cash can then be removed. A retract in the RR-box is not carried out  
N = Lock not set  
Y = Lock set, Reset per DFR command (RESET lock flag).
- CAS** General status of the cassettes  
N = Cassettes cannot be processed  
Y = Cassettes can be processed  
At least 2 ready cassettes are needed to operate the CMD-V4: the RR-box as well as the cash-out cassette. Operability is not given if this minimum requirement is not fulfilled. The flag is set accordingly.
- SHERR** Shutter error  
B = Blocked or cash trapping detection error  
N = No error  
This cell indicates whether or not a shutter error has occurred. If an error has occurred, the <SHUT> status can be evaluated. A cash trapping detection error is additionally registered in this cell and the status can be evaluated in <SHUT\_PROTECT>.
- SHUT** Shutter status  
C = Closed and locked  
M = Missing  
O = Open  
U = Undefined or cash trapping detection error

The current status after the last transport command as well as after RESET / Clearing is managed here. The cell is additionally set following a cash trapping detection error.

**MON**      Status of note output

N = No cash accessible to customer

P = Cash accessible to customer

R = Retract

W = Cash has been removed

The status switches to 'N' at the beginning of a dispensing. Switch to 'P' when the shutter is opened, i.e. it is no longer closed and locked. The flag is updated accordingly after note output, retract or RESET / CLEARING. A change to 'R' is made if the cash is no longer accessible to the customer, i.e. if the shutter is closed and the output sensor is free.

**TER**      Transport error

J = Banknote jam

M = Manipulation

O = OK; no error

The flag is set to 'J' (jam) if the CMD-V4 detects a note jam or if the stacking compartment is blocked. This is used together with the following flag 'TS' to determine where the cash is currently situated. Status 'M' indicates an output sensor that was probably covered by a manipulation.

**TS**      Transport status

This flag is divided into two parts. The CMD-V4 manages the flag in case of transports, dispensings, and clearing runs as follows:

Prior to the transport, the start position is stated in the 1st byte and in the 2nd byte the target position. If there is a note jam during transport <TER>=J, evaluation of the two bytes of the transport status helps identifying the area where the jam has occurred.

Possible positions:

0 = No notes available

1 = Cassette

2 = Stacker

3 = Reject compartment

4 = Stop over (output)

5 = Removal position

8 = Retract compartment

9 = Undefined position

**Appendix F** (page Fehler! Textmarke nicht definiert.Fehler! Textmarke nicht definiert.) indicates the possible transport positions.

**TF**      Transport free

This is the status value for the cash paths after the last dispensing, reference value, reset flag, RESET or clearing command, or after power-on.

N = Cash paths were not free

Y = Cash paths were free and shutter closed

#### **SR** Status of single reject deflector

O = OK

R = effect (Transportrichtung Reject)

S = Faulty (collecting tray transport direction)

This flag is set if the reject switch is faulty. Handling of the error in the CMD-V4 depends on the switch position:

Transport direction: collecting tray

This case concerns a status.

The CMD-V4 can **continue to operate with limited capacity**.

As soon as a single reject becomes necessary, the notes dispensed up to that point are lost. A note bundle reject then takes place.

Transport direction: Reject

In this case, further dispensing is impossible. The device can no longer be operated.

All notes dispensed so far are stored in the reject escrow (including counter increment). Troubleshooting must be completed with RESET. The flag is reset (<SR>=O) when this is done successfully.

#### **DIS** Device error during dispensing

E = Too many bundle rejects

J = Repeat of an 'Additional dispense' not possible. During the process:

- A note jam was eliminated with a test note.

- Single reject could not be performed (single reject deflector missing or direction facing stacker).

M = Too many wrong-sized notes

N = Note cannot be dispensed

O = No error

S = Too many double dispenses

The CMD-V4 sets this flag to 'O' before the 1st note is being dispensed.

The event that has caused the device error is registered if necessary.

#### **CEX** Cassettes available changed

N = no

Y = yes

The flag is set by the CMD-V4 if it detects that the number of cassettes available has changed.

This is reset by means of a command sent by the HOST.

**DOOR** Door sensor status

C = Closed

M = Missing

O = Open

The status of the door sensor is newly determined with each command to the CMD-V4.

CMD-V4 itself considers the flag when handling encryption commands, provided **<DO\_S> = Y** has been configured.

**DO\_S** CMD-V4 behavior at the start/end of encryption

N = Do not consider door sensor

Y = Consider door sensor

The flag is read out and updated from the hardware settings during system startup and during RESET.

**TYPE** Device type

The current device version is mapped to a defined ID:

R = Rearload

F = Frontload

U = undefined

**N\_VM** Number of separator modules 1 - 6 (contains the configuration presetting)

**SEDM** Status of software EDM

I = Initial value must be requested

L = Line key LK missing

O = Encryption active

T = Transport key TK missing

If the line is not encrypted, the status is 'T / L / I'.

**LCMD** Specification + additional specification of the last command

**LSTA** Status code of the last command



The contents of the two cells are only changed by commands that initiate device operations. (Reset, dispense transaction, ...)

**SCLE** Clearing status after RESET / Clearing / POWER UP

- 0 = Transport paths were free
- 1 = No cash accessible to customer; Reject successful.
- 2 = Cash was accessible to customer; retract was successful.
- 3 = Cash was accessible to customer; it has been withdrawn in the meantime.
- 4 = Cash was not accessible to customer; an error occurred during clearing (s. SRES).
- 5 = Cash was accessible to customer; an error occurred during clearing (s. SRES).

**SRES** Status code after RESET / Clearing / POWER UP see **Appendix A** (page 88)

SRES and SCLE belong together. What was done is specified in SCLE; the result is documented in SRES.

**TST** Self-test was active, Reset necessary.

- N = no
- Y = yes

**SCOD** Operator status display

When the safe is closed, the operator can be provided with this information, e.g. on the operator panel. For contents and meaning see **Appendix B** (page **Fehler! Textmarke nicht definiert.Fehler! Textmarke nicht definiert.**).

**OR** Operator request

- N = no
- W = Warning (dirty photosensors)
- Y = A status has occurred that requires operator intervention. Further information can be found in the individual status areas.

Error causes with no explicit code word in the device status e.g. photosensors, faulty motors, are identified on the operator status display. See **Appendix B** (page **Fehler! Textmarke nicht definiert.Fehler! Textmarke nicht definiert.**).

**TRANSPORT** Output transport variant

- A = in safe front
- B = in Safe rear (also known to many as outdoor rearload)

- C = out Safe rear (also known to many as indoor rearload)
- D = out safe front (also known to many people as 'indoor - frontload')
- E = out safe rear (für PC8xxx - systems)
- F = out safe front (for PC8xxx - systems)
- G = in safe rear (for PC8xxx - systems)
- H = (VCMD with horizontal output) does not yet exist
- U = ProCash 5100 (VCMD)
- \* = unknown coding

The flag is recognized by the connector coding and updated at system startup and RESET.

#### **C\_OUT** CashOut available

This cell allows global checking to determine whether the CashOut functionality is available or not.

- C = Minimum configuration not available (RR box + cash-out cassette)
- J = JAM (note jam)
- L = Device lock
- M = Main motor DCM1 faulty
- P = CMD-V4 booting
- S = Safety switch
- T = Clamp transport defective
- Y = CashOut possible



The status can only be changed if:

- CashOut was previously possible.
- The last of several reasons documented is no longer relevant.

#### **SHUT\_P** Cash trapping detection available **ROTECT**

- O = OK, no error
- K = Cash trapping device detected
- M = Cash trapping device manipulation (e.g. color spraying)
- D = Cash trapping detection defective (e.g. no communication, not initialized, faultily initialized, implausibly initialized)
- U = Cash trapping detection not present (see shutter plug for coding)

For reset of the K, M status after error recovery or removal of the cash trapping device see section "Reset of a cash trapping detection error" (page 10).

**ERROR** Error number

The four-digit error number details the cause of the current error. For the cause and remedial measures: see **Appendix G – Error Number Table** (page 111)

**WARNING** Warning number

The four-digit warning number details the cause of the current erroneous function. For the cause and remedial measures: see **Appendix G – Error Number Table** (page 111)

**C – Cassette status**

Code word	Entry	Meaning	
LEN=	dddd,	Total length (incl. length specification)	(decimal in ASCII)
RSTA=	a,	Status of RR box M = Cassette missing or inserted incorrectly N = Cassette newly added R = Cassette ready	(ASCII)
RACT=	dddd,	Number of notes rejected	(decimal in ASCII)
RRET=	dd,	Number of retracts	(decimal in ASCII)
nSTA=	a,	Status of cash-out cassette n (n = 1 - 6) D = Cassette EEPROM read error E = Cassette physically empty M = Cassette missing or not correctly inserted N = Cassette reinserted P = Parameter not known in measurement station R = Cassette ready V = Document ready for being pulled off (printing cass.)	(ASCII)
NNUM=	dddddd d,	Cassette ID	(decimal in ASCII)
nCUR=	aaa,	Currency code	(ASCII)
nREL=	dddd,	Release date (YYMM)	(decimal in ASCII)
nVAL=	dddddd dd,	Value of note	(decimal in ASCII)

Code word	Entry	Meaning	
nLEN=	ddd,	Width of note	(decimal in ASCII)
nTOL=	dd,	Cut tolerance of note	(decimal in ASCII)
nACT=	dddd,	Current number of notes	(decimal in ASCII)
nNDV=	dddd,	Number of notes in the device	(decimal in ASCII)
nLOW=	dddd,	Specification of minimum quantity	(decimal in ASCII)
nL_D=	dd,	Last dispense: no. of notes	(decimal in ASCII)
nREJ=	ddd,	Last dispense: number of rejected notes	(decimal in ASCII)
SRACT=	dd	current number of notes being parked in the single reject compartment	
	;	End character after last parameter	(ASCII)

Data from the dispenser modules is acknowledged (n = 1 to n = <N\_VM>).



Dispenses are not hindered with a logically empty cash-out cassette or a full RR box.

The operator must ensure that the CMD-V4 has the correct data before customer mode is activated!.



## Definitions



The highlighted status of a cell or completely highlighted status words are only returned if standard mode is activated!

**LEN**        Contains the total length of the status range (incl. length specification).

**RSTA**        Status of RR box

M =    Cassette missing or incorrectly inserted

N =    Cassette re-inserted

R =    Cassette ready

The cell is updated by the CMD-V4 during RESET / Clearing and dispensing.

**RACT**        REJECT counter

The number of singled out notes in the device is documented in **RACT**. The number usually corresponds to the total of all nREJ values that were determined during dispensing commands. Differences to the sum of nREJ values can arise due to trial dispensing processes or test notes.

Notes which were identified by the measurement station as being double or too long increment the counter by one. Undefined banknotes (banknotes released without banknote dispensing) are not counted. Notes which are still in the transport path following a jam are counted. The counter is defined via a SW call and remains unchanged, both during a power failure and when the cassettes are changed.

**RRET**        Retract counter

The number of cash storage procedures is documented in **RRET**. The value is increased before the beginning of a retract. This counter is reset with the 'Clear retract counter' command (DFT).

One range exists per cash-out cassette ( $n = 1 - 6$ ):

**nSTA**      Cassette status

D = Read error cassette EEPROM

E = Cassette is physically empty

M = Cassette missing or incorrectly inserted

N = Cassette re-inserted

P = Parameters unknown in measurement station

R = Cassette ready

V = printing cassette, document available (printing cassette only)

The cell is updated cyclically by the CMD-V4. Dispensing commands only include cash-out cassettes with the status:

– 'R' Cassette is ready

– 'V' Document ready for dispensing from printing cassette.

Status **D** can only be quit by removing the cassette.

**nNUM**      Cassette ID; it can be set for specific customers using the software.

The 7-digit cassette ID is read from a non-volatile memory in the dispenser cassette and stored when the system is started or when 'RESET' takes place.

When the cassettes are delivered, the default setting for the cassette ID is the serial number of the cassette.

The **printing cassette** does not have a non-volatile memory. The cassette ID is therefore initialized with the default value (standard: – – – – –), when the command 'Confirm HW configuration' is processed.

**nCUR**      Currency code

**nREL**      Release date

**nVAL**      Value of note

These 3 parameters define the note exactly: They are read from a non-volatile memory in the dispenser cassette and stored when the system is started or when RESET takes place. This parameters can be updated via software.

**nCUR** is the 3-digit currency characteristic to ISO 4217 (codes for representation of currencies and funds).

Coding „★★★“ indicates a cassette that has not yet been initialized or a defective cassette position.

**nREL** contains an additional 4-digit ID (YYMM of the output). This allows the CMD-V4 to process notes with the same nominal value but different note properties.

**nVAL** determines the value of the note.

**nLEN**      Width of note in transport direction

**nTOL**      Cut tolerance in transport direction

Both values are 0 if the Note parameter I command was used and no reference value has yet been determined.

**nACT**      Current number of notes

**nNDV**      Number of notes in the device

The two counters provide information about the position of the notes. They are defined via a SW call and remain unchanged, both during a power failure and when the cassettes are changed.

**nACT** is decremented with every note that is dispensed.

**nNDV** is updated when a note is output. After the shutter has been moved from its 'SHUT' position, the counter is incremented:  $\langle nNDV \rangle - \langle nL\_D \rangle$  for  $n = 1 - 6$

**nLOW**      Minimum quantity specification

A minimum quantity can be defined for each cassette per software (default = 0, no monitoring).

Cassettes with mechanical filling level detector have priority over the software setting.

The position of the dispensed notes and the filling level status of each cassette are transferred in the acknowledgment of the dispensing commands.

L = Reached / fallen below minimum quantity

O = OK; minimum quantity not yet reached

E = Cassette physically empty, if  $\langle nSTA \rangle = E$

**nL\_D**      Number of notes (2-digit)

**nREJ**      Number of rejected notes (3-digit)

**SRACT**    Single reject compartment counter

The number of the singled out notes in **SRACT** presently parked in the single reject compartment is documented. Notes that should not be paid out are transported to the single reject compartment. Notes which were identified by the measurement station as being double or too long increment the counter by one.

The compartment is emptied after note output, note bundle reject, retract, clearing, or power OFF/ON and the counter reset.

## F – Features



The highlighted features cannot be configured via the SW!

Code word	Entry	Meaning	
BARC=	a,	Special cassette commands available N = no Y = yes	(ASCII)
C_TYPE=	a..a,	Cassette type One entry exists for each dispensing module (number as in <N_VM>) ★ = Cassette type not yet initialized D = Printing cassette L = Mechanical fill level sensor active M = Mechanical fill level sensor S = Standard cassette	(ASCII)
PRINT=	a,	Printing cassette extension is part of the firmware N = no Y = yes	(ASCII)
DO_BAD=	a,	Double dispense evaluation configurable N = no Y = yes	(ASCII)
ERRS=	a,	'ERROR info' command is part of the firmware N = no Y = yes	(ASCII)
SINGLE_REJECT=	a,	Availability of single reject (installed and functioning) N = no Y = yes	(ASCII)
RTC=	a,	Real-time clock available N = no Y = yes	(ASCII)

Code word	Entry	Meaning	
SCOP=	a,	SCOP is part of the firmware and available N = no Y = yes	(ASCII)
BAUD=	ddd/ ddd,	possible baud rates (V.24 line) 192/560 = 19200/56000 baud	(decimal in ASCII)
BAUD_CHANGE=	a,	V.24 line parameters configurable N = no A = automatic baud rate switch	(ASCII)
BUILD=	a..a,	Internal version number	(ASCII)
MAX_NOTES=	dd,	Maximum number of notes per bundle 60 for CMD_V4	(decimal in ASCII)
FAMILY=	a,	C = CMD-V4 family	(ASCII)
CONFIG_RETRACT_COUNT	a,	Counting of retracts configurable N = no Y = yes	
CONFIG_BUTTON_3	a,	Trial dispensing process via button function 3 can be disabled via configuration N = no Y = yes	
CHECK_SHUTTER_ATTACK	a,	Shutter monitoring after DCW with automatic clamp retraction configurable N = No Y = Yes	
DES_OPT=	a,	Special encryption procedures available N = no, only DES procedure Y = yes, further procedures available	(ASCII)

Code word	Entry	Meaning	
ACCESS_CODE=	a,	Activation procedure is part of the firmware and commands activated N = not supported; standard in the CMD_V4 I = Activation procedure available. Only limited command utilization possible, all other commands are rejected showing context error. The limited commands are: *IL, DABlack, DAB, DEx, DHx, DSC, and DSD. (with PC8xxx – transports) Y = Activation procedure available, all commands can be used (with PC8xxx – transports)	(ASCII)
SHUT_PROTECT_O PT=	a,	Cash trapping detection is part of the firmware N = is not supported Y = yes, is supported	
	;	End character after last parameter	(ASCII)



For a cassette with mechanical filling level detector, the correct filling level (cassette type) is not displayed until the banknote pressure has built up. For example, when inserting cassettes, the indicator M or L under C\_TYPE is only reliable when the cassette status of the respective cassette changes from M (Missing) to N (New).

## K – Firmware configuration

Code word	Entry	Meaning	
SDO_BAD=	a,	Dual dispense evaluation N = Do not reject 2 GOOD notes Y = Always reject dual dispenses	(ASCII)
MAX_NOTES=	dd,	Maximum possible amount of notes per transaction (20 - 60)	(ASCII)
COUNT_MULTIPLE_NOTES=	a,	Y= Count all notes that can be identified double and threefold (this is not 100 % of all dual and triple combinations!) shall be counted double and threefold (in nACT and RACT) N = Only count valid dual dispenses as double	(ASCII)
RESTRICT_CASSETTE_NOTIFICATION=	a,	N = Logging on cassettes (DFN,DFO,DFA,DFB) also permitted if TF=N Y = as before	(ASCII)
LOW_IDLE_PRESSURE=	a,	Y = In idle position, move the pressure carriage back slightly (for polymer notes) N = previous behavior	(ASCII)
JAM_LOCK_ENABLED=	a,	Reset suppression after banknote jam N = Resets are always permitted. Y = no reset on detected note jam. A reset is only carried out after button function 1 or the safety switch has been actuated.	

Code word	Entry	Meaning	
RETRACT_COUNT_IN_BOX =	a,	Y = A retract is counted in RRET if the notes are stored in the retract box/tray. N = A retract is counted in RRET if the shutter is closed after retracting the notes from the offer cash position.	
BUTTON_3_TESTDISPENSE =	a,	Y = Button function 3 results in a trial dispensing process with subsequent rest. N = Button function 3 only carries out a reset, analog to button function 1.	
CHECK_FOR_SHUTTER_ATTACK=	a,	Y = The shutter is monitored after successful DCW; automatic retract of the clamp as soon as the shutter has left the 'Shut' status. See also Appendix Y: Detection of a shutter manipulation after DCW N = as before	
DES_OPTION =	d,	Encryption procedure Switching-over only possible with the encryption disabled. 0 = DES-encryption, standard 1 = 3DES encryption	(ASCII)
DES_CLEAR_BUTTON_6=	a,	N = Encryption cannot be disabled, standard Y = Encryption can be disabled at open door switch and depressed pushbutton 6.	(ASCII)



Code word	Entry	Meaning	
OFFER_CASH_POS =	d,	Change position when offering the bundle after the DCO command. 0 = Bundle is transported out farthestmost, standard : 7 = Bundle is transported out the shortest possible fashion	(ASCII)
OFFER_SHUTTER_POS =	d,	Shutter position when offering the bundle. 0 = Standard 1 = Shutter is placed onto the bundle the least : 9 = Shutter is placed onto the bundle the strongest	(ASCII)
OPEN_SHUTTER_REDUCE =	a,	Position of the shutter plate at shutter OPEN N = standard Y = Shutter plate is opened some mm less wide	(ASCII)
SHUT_PROTECT_OPT	a,	Cash trapping detection N = switched off Y = switched on, standard	(ASCII)
SHUT_ALARM_RANGE	d,	Distance of alarm threshold to shutter plate 0 = Standard 1 = least distance : 9 = widest distance	(ASCII)
;	a	End character	(ASCII)

## Definitions

**SHUT\_PROTECT\_  
OPT**

Cash trapping detection

Existence of a trapping device detection is not checked under this configuration.

N = trapping device detection is disabled

Y = trapping device detection is enabled, standard

For a reset of the error see section “Reset of a cash trapping detection error” (page 10).

**SHUT\_ALARM\_RA  
NGE**

Distance of alarm threshold to shutter plate

0 = Standard (for outdoor shutter approx. 50 mm, for indoor shutter approx. 35 mm distance) (see shutter plug for coding)

1 = alarm threshold approx. 25 mm distance

:

9 = alarm threshold approx. 70 mm distance

## Configuration after CMOS initialization

### Device status

N_VM=n,	Number of dispenser modules n = Number of types detected from <C_TYPE>
---------	---

### Features

C_TYPE=★..★,	Cassette type of cassettes, per recognized dispenser unit ★ = cassette type not yet initialized D = printing cassette
--------------	---

### FW configuration

SDO_BAD=N	Double dispense evaluation of the measurement station: '2 GOOD' notes are not disqualified.
MAX_NOTES=60	Maximum number of notes per transaction
COUNT_MULTIPLE_NOTES=N	Compatible behavior
RESTRICT_CASSETTE_NOTIFICATION=Y	Compatible behavior
LOW_IDLE_PRESSURE=N	Compatible behavior
JAM_LOCK_ENABLED=N	Compatible behavior
RETRACT_COUNT_IN_BOX=N	Compatible behavior
BUTTON_3_TESTDISPENSE=Y	Compatible behavior
CHECK_FOR_SHUTTER_ATTACK=N	Compatible behavior
DES_OPTION=N	Compatible behavior
DES_CLEAR_BUTTON_6=N	Compatible behavior
OFFER_CASH_POS=0	Compatible behavior
OFFER_SHUTTER_POS=0	Compatible behavior
OPEN_SHUTTER_REDUCE=0	Compatible behavior
SHUT_PROTECT_OPT=Y	Cash trapping detection enabled
SHUT_ALARM_RANGE=0	Distance of alarm threshold to shutter plate

## Appendix Y: Special features

### Different behavior of the clamp for DCE and DCP

There is a distinctive difference in the process control for the commands DCE (output cash) and DCP (cash presentation). With the command DCP the clamp is moved back slightly before the shutter opens. The clamp retracts to its final position after the shutter has opened to then present the notes. With the command DCE, the clamp remains in the end position while the shutter is being opened and subsequently outputs the notes.

### DCW and the locking handle

The CMD-V4 has a locking switch (SW 3) that is positioned above the reject/retract cassette. This switch indicates if the separator has not been correctly inserted into the safe -> status display "09". This is necessary because the clamp's transport route must be correctly placed on devices with their cash output outside of the safe. Otherwise, transport of the clamp from the stacker to the shutter can cause jams.

For this reason, the device type and the status of the locking handle are checked with command DCW (transport target = Stop Over). If it is a device with cash output outside the safe and if the separator has been inserted incorrectly, the command DCW is acknowledged with "D" – Out of service.

## Detecting shutter manipulation after DCW

If the firmware property CHECK\_SHUTTER\_ATTACK = Y has been set, it will be possible to enable an additional security feature via configuration.

The shutter status is monitored by the firmware after a successful DCW command (banknote transport to parking position). If the shutter's position sensors should detect another position than "closed", then:

1. the clamp is moved from the shutter position to the stacker position so as to prevent any unauthorized access to notes
2. The following device statuses are set until the next reset (DA\_):
  - SHUT = U (regardless of whether or not the shutter closes again)
  - TER = M
  - TS = 42
  - SCOD = 29
3. The following device statuses remain unchanged:
  - TF = N
  - MON = N
4. Following commands are still permitted at this status:
  - DAblank (reset)
  - DEx (status commands)
  - DHx (key handling)
  - DIx (software supply except DIC)
  - DQx (error statistics, realtime clock, EEPROM)
  - DYx (test commands)

The device can only be brought back to a ready state following a successful reset (DAblank).



---

# Notes

Published by  
WINCOR NIXDORF International GmbH  
D-33094 Paderborn

Order No.:

**01750070034 C**

2

