BLACK BOX®

ACX048 ACXC8 ACXC32 ACXC64 ACXC48F16
ACX080 ACXC8F ACXC32F ACXC64F ACXC48F32
ACX160 ACXC16 ACXC48 ACXC80
ACX288 ACXC16F ACXC48F ACXC80F

DKM FX and DKM FX Compact

Application Programming Interface (API) Manual

Customer Support Information

Order toll-free in the U.S.: Call 877-877-BBOX (outside U.S. call 724-746-5500) FREE technical support 24 hours a day, 7 days a week: Call 724-746-5500 or fax 724-746-0746 • Mailing address: Black Box Corporation, 1000 Park Drive, Lawrence, PA 15055-1018 • Web site: www.blackbox.com • E-mail: info@blackbox.com

Trademarks Used in this Manual

Black Box and the Double Diamond logo are registered trademarks of BB Technologies, Inc.

TOSLINK is a registered trademark of Kabushiki Kaisha Toshiba dba Toshiba Corporation.

Any other trademarks mentioned in this manual are acknowledged to be the property of the trademark owners.

FEDERAL COMMUNICATIONS COMMISSION AND INDUSTRY CANADA RADIO FREQUENCY INTERFERENCE STATEMENTS

This equipment generates, uses, and can radiate radio-frequency energy, and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio communication. It has been tested and found to comply with the limits for a Class A computing device in accordance with the specifications in Subpart B of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference when the equipment is operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user at his own expense will be required to take whatever measures may be necessary to correct the interference.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This digital apparatus does not exceed the Class A limits for radio noise emission from digital apparatus set out in the Radio Interference Regulation of Industry Canada.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe A prescrites dans le Règlement sur le brouillage radioélectrique publié par Industrie Canada.

Normas Oficiales Mexicanas (NOM) Electrical Safety Statement INSTRUCCIONES DE SEGURIDAD

1. Todas las instrucciones de seguridad y operación deberán ser leídas antes de que el aparato eléctrico sea operado.
2. Las instrucciones de seguridad y operación deberán ser guardadas para referencia futura.
3. Todas las advertencias en el aparato eléctrico y en sus instrucciones de operación deben ser respetadas.
4. Todas las instrucciones de operación y uso deben ser seguidas.

4. Todas las instrucciones de operación y uso deben ser seguidas.

5. El aparato eléctrico no deberá ser usado cerca del agua—por ejemplo, cerca de la tina de baño, lavabo, sótano mojado o cerca de una alberca, etc.

6. El aparato eléctrico debe ser usado únicamente con carritos o pedestales que sean recomendados por el fabricante.

7. El aparato eléctrico debe ser montado a la pared o al techo sólo como sea recomendado por el fabricante.

8. Servicio—El usuario no debe intentar dar servicio al equipo eléctrico más allá lo descrito en las instrucciones de operación. Todo otro servicio deberá ser referido a personal de servicio calificado.

9. El aparato eléctrico debe ser situado de tal manera que su posición no interfiera su uso. La colocación del aparato eléctrico sobre una cama, sofá, alfombra o superficie similar puede bloquea la ventilación, no se debe colocar en libreros o gabinetes que impidan el flujo de aire por los orificios de ventilación.

10. El equipo eléctrico deber ser situado fuera del alcance de fuentes de calor como radiadores, registros de calor, estufas u otros aparatos (incluyendo amplificadores) que producen calor.

11. El aparato eléctrico deberá ser connectado a una fuente de poder sólo del tipo descrito en el instructivo de operación, o como se indique en el aparato.

12. Precaución debe ser tomada de tal manera que la tierra fisica y la polarización del equipo no sea eliminada.

13. Los cables de la fuente de poder deben ser guiados de tal manera que no sean pisados ni pellizcados por objetos colocados sobre o contra ellos, poniendo particular atención a los contactos y receptáculos donde salen del aparato.

14. El equipo eléctrico debe ser limpiado únicamente de acuerdo a las recomendaciones del fabricante.

15. En caso de existir, una antena externa deberá ser localizada lejos de las lineas de energia.

16. El cable de corriente deberá ser desconectado del cuando el equipo no sea usado por un largo periodo de tiempo.

17. Cuidado debe ser tomado de tal manera que objectos liquidos no sean derramados sobre la cubierta u orificios de ventilación.

18. Servicio por personal calificado deberá ser provisto cuando:

    A: El cable de poder o el contacto ha sido dañado; u

    B: Objectos han caído o líquido ha sido derramado dentro del aparato; o

    C: El aparato ha sido expuesto a la lluvia; o

    D: El aparato parece no operar normalmente o muestra un cambio en su desempeño; o

    E: El aparato ha sido tirado o su cubierta ha sido dañada.

Table of Contents

1. About This Manual

1.1 Scope

This manual describes how to install your DKM FX API, how to operate it, and how to perform troubleshooting.

1.2 Validity

This manual is valid for all devices listed on the front page. The product code is printed on the base of the devices.

1.3 Cautions and Notes

The following conventions are used in this manual:

WARNING or CAUTION: This indicates an important operating instruction that should be followed to avoid any potential damage to hardware or property, loss of data, or personal injury.

NOTE: This indicates important information to help you make the best use of this product.

2. Safety Instructions

For reliable and safe long-term operation of your DKM FX, follow these guidelines:

Installation

- Only use this device according to this User Manual. Otherwise, the safety can be affected.
- Only use in dry, indoor environments.
- The DKM FX and the power supply units can get warm. Don't put them in an enclosed space without any airflow.
- Do not obscure ventilation holes.
- Only use power supplies originally supplied with the product or manufacturer-approved replacements. Do not use a power supply if it appears to be defective or has a damaged case.
- Connect all power supplies to grounded outlets. In each case, make sure that the ground connection is maintained from the outlet socket to the power supply's AC power input.
- Do not connect the link interface to any other equipment, particularly network or telecommunications equipment.
- Only connect devices to the serial interface that are sufficiently secured against short circuits and false voltages at the serial interface.
- To disconnect the switch from the power supply, remove the power cord cables of all power supply units that are in use or set the power switch (if available) to the OFF position.
- Take any required ESD precautions.
- To disconnect the device completely from the electric circuit, remove all power cables.

Repair

- Do not attempt to open or repair a power supply unit.
- Do not attempt to open or repair the DKM FX or DKM Compact. There are no user-serviceable parts inside.
- Contact Black Box Technical Support at 724-746-5500 or info@blackbox.com if there is a fault.

3. Overview

3.1 Description

The DKM FX API is used to control the matrix externally by serial commands via serial (RS-232) or network (TCP/IP) connection.

The DKM FX API provides the full scope of switching functionality. It does not support the configuration of a DKM FX system.

3.2 Access Options

You have the following options to access the DKM FX for external serial control:

Table 3-1. Access options.

Access option	Symbol
Serial interface	
TCP/IP Interface	

NOTE: Both serial interface and TCP/IP interface use the same commands for the operation of the DKM FX matrix.

3.3 System Overview

A DKM FX matrix consists of a DKM FX matrix, and, for KVM applications, of one or more CPU units/CON units. The matrix is connected to the CPU units/CON units by interconnect cables, or connected directly to the video devices where used as a video matrix.

The CPU units are connected directly to the sources (computer, CPU) by the provided cables.

Monitor(s), keyboard, and mouse are connected to the CON units.

The communication between the DKM FX matrix and the CPU units/CON units takes place over the respective interconnect cables.

Figure 3-1 shows the system overview. Table 3-2 describes its components.

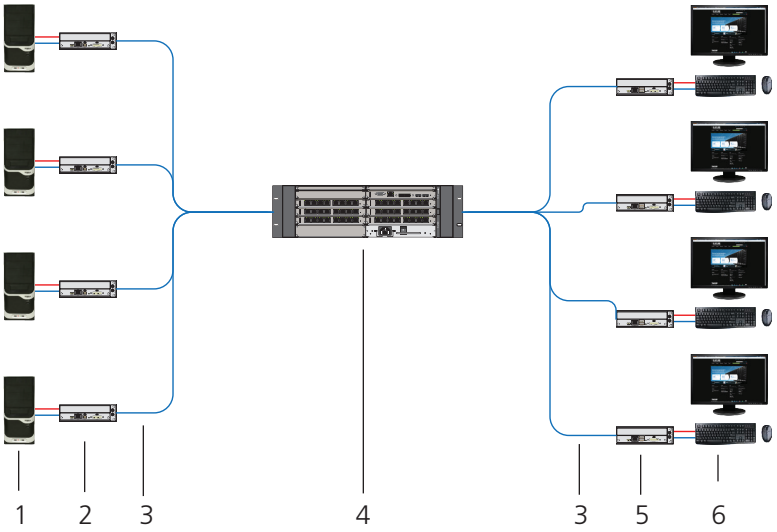


Figure 3-1. System Overview.

Table 3-2. System components.

Number	Component
1	Source (computer, CPU)
2	CPU units
3	Interconnect cable
4	DKM FX matrix
5	CON units
6	Console (monitor, keyboard, mouse)

3.4 System Overview External Control

A DKM FX matrix can be connected to an external serial control via the CPU board and its connectors.

The CPU board provides the possibility for both serial and TCP/IP connections.

The serial connection to an external serial control is established by using a serial cable with DB9 connectors or a DB9-to-RJ-45 adapter cable (DKM FX Compact).

The TCP/IP connection is established by using a CATx network cable.

Figure 3-2 shows the system overview, external control. Table 3-3 describes its components.

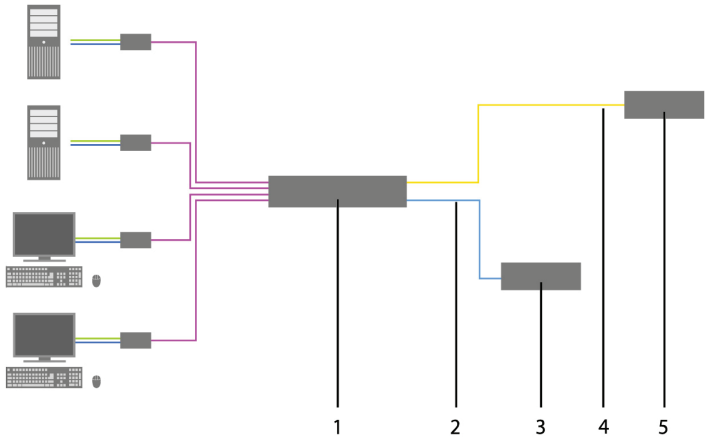


Figure 3-2. System overview (exemplary).

Table 3-3. System components.

Number	Component
1	DKM FX matrix
2	Serial connection cable (DB9 or DB9-to-RJ-45 adapter cable)
3	External serial control (RS-232, Option 1)
4	Network connection cable (CATx)
5	External serial control (TCP/IP, Option 2)

3.5 Product Range

Table 3-4. Available products.

Number	Description
DKM FX and DKM FXC-API	
ACX-API	DKM FX matrix application programming interface (API)

3.6 Device Views

In Sections 3.6.1 through 3.6.4, Figures 2-2 through 2-5 illustrate the ServSwitch DKM FX chassis models. Tables 3-5 through 3-8 describe their components.

NOTE: The following images of the chassis are fully populated with I/O cards and are intended to be example diagrams. The chassis do not come with any I/O cards. You need to purchase the I/O cards separately.

3.6.1 ServSwitch DKM FX 48-Port (ACX048)

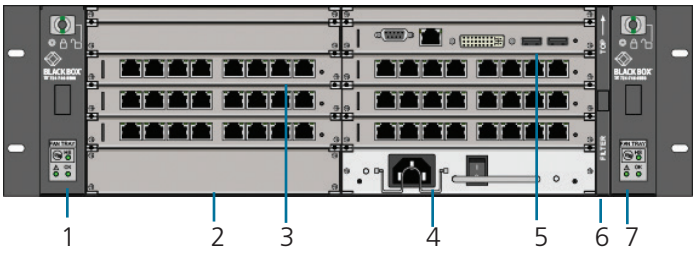


Figure 3-3. Front view, ACX048.

Table 3-5. ACX048 components.

Number	Component
1	Slot for fan tray
2	Slot for Power Supply Unit 1
3	Slot for I/O Boards 1–6
4	Slot for Power Supply Unit 2
5	Slot for CPU board
6	Slot for air filter
7	Slot for Fan Tray 2

3.6.2 ServSwitch DKM FX 80-Port (ACX080)

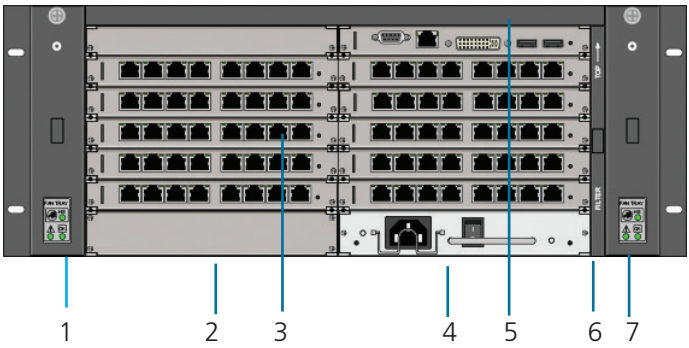


Figure 3-4. Front view, ACX080.

Table 3-6. ACX080 components.

Number	Component
1	Slot for fan tray
2	Slot for Power Supply Unit 1
3	Slot for I/O Boards 1–10
4	Slot for Power Supply Unit 2
5	Slot for CPU board
6	Slot for air filter
7	Slot for Fan Tray 2

3.6.3 ServSwitch DKM FX 160-Port (ACX160)

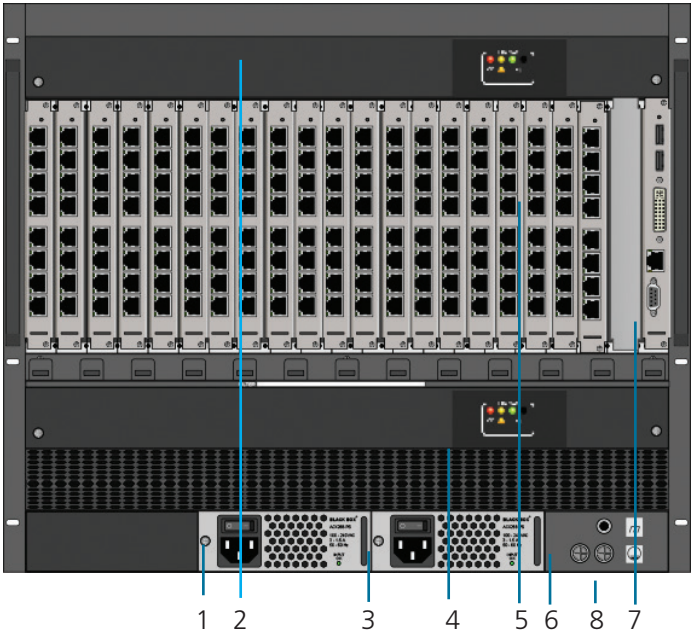


Figure 3-5. Front view, ACX160.

Table 3-7. ACX160 components.

Number	Component
1	Slot for Power Supply Unit 1
2	Slot for Fan Tray 1
3	Slot for Power Supply Unit 2
4	Slot for Fan Tray 2
5	Slot for I/O Boards 1–36
6	Slot for Power Supply Unit 3
7	Slot for CPU board
8	Grounding

3.6.4 ServSwitch DKM FX 288-Port (ACX288)

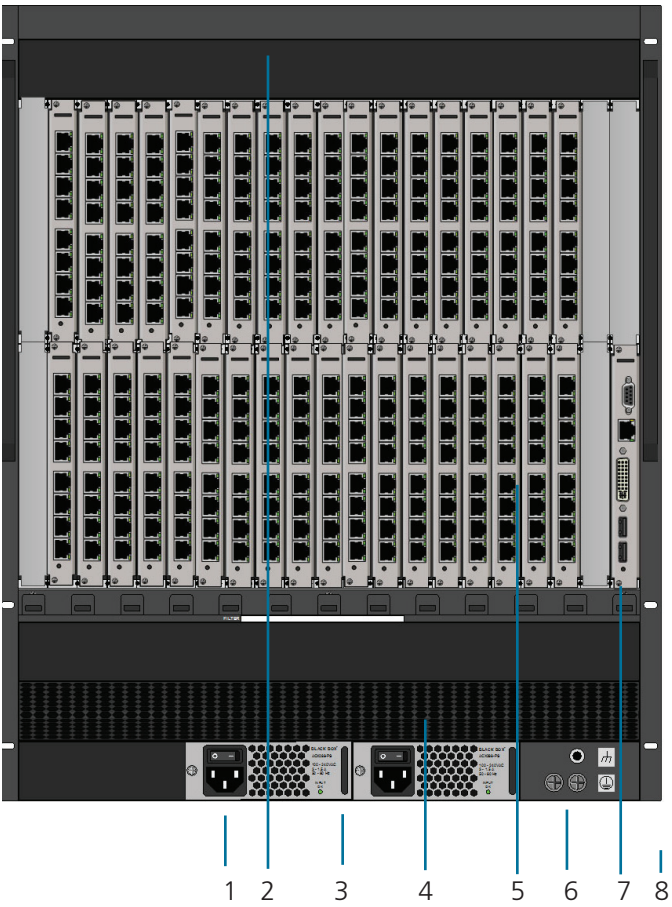


Figure 3-6. Front view, ACX288.

Table 3-8. ACX288 components.

Number	Component
1	Slot for Power Supply Unit 1
2	Slot for Fan Tray 1
3	Slot for Power Supply Unit 2
4	Slot for Fan Tray 2
5	Slot for I/O Boards 1–36
6	Slot for Power Supply Unit 3
7	Slot for CPU board
8	Grounding

NOTE: The ServSwitch DKM FX Compact models are described next.

3.6.5 ServSwitch DKM FX Compact 8-Port, CATx (ACXC8)

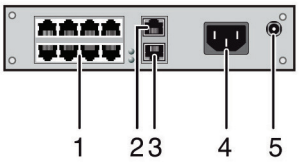


Figure 3-7. Front view, ACXC8.

Table 3-9. ACXC8 components.

Number	Component
1	I/O ports 1–8 (CATx)
2	Serial connection (RJ-45)
3	Network connection (RJ-45)
4	Connect to power supply
5	Connect to a 5-VDC power supply (redundancy, optional)

3.6.6 ServSwitch DKM FX Compact 8-Port, SFP (ACXC8F)

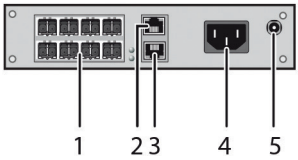


Figure 3-8. Front view, ACXC8F.

Table 3-10. ACXC8F components.

Number	Component
1	I/O ports 1–8 (SFP)
2	Serial connection (RJ-45)
3	Network connection (RJ-45)
4	Connect to power supply
5	Connect to a 5-VDC power supply (redundancy, optional)

3.6.7 ServSwitch DKM FX Compact 16-Port (ACXC16)

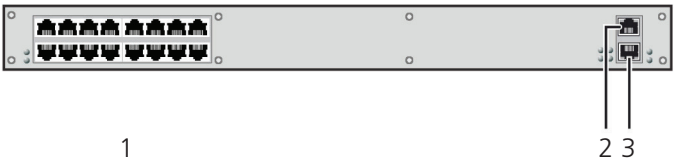


Figure 3-9. Front view, ACXC16.

Table 3-11. ACXC16 components.

Number	Component
1	I/O ports 1–16 (CATx)
2	Serial connection (RJ-45)
3	Network connection (RJ-45)

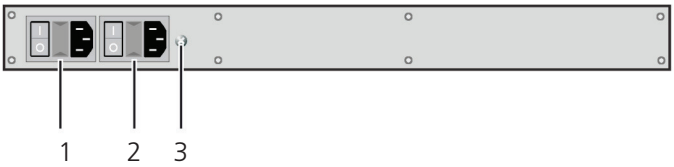


Figure 3-10. Back view, ACXC16.

Table 3-12. ACXC16 back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.6.8 ServSwitch DKM FX Compact 16-Port Fiber (ACXC16F)

Figure 3-11. Front view, ACXC16F.

Table 3-13. ACXC16F components.

Number	Component
1	I/O ports 1–16 (SFP)
2	Serial connection (RJ-45)
3	Network connection (RJ-45)

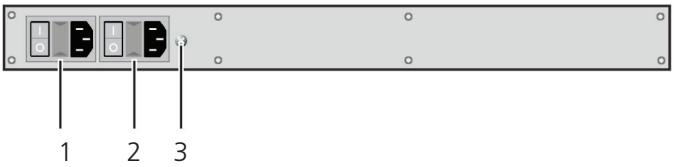


Figure 3-12. Back view, ACXC16F.

Table 3-14. ACXC16F back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.6.9 ServSwitch DKM FX Compact 32-Port (ACXC32)

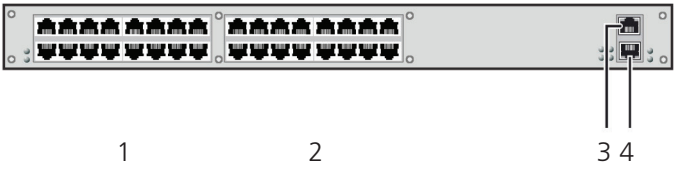


Figure 3-13. Front view, ACXC32.

Table 3-15. ACXC32 components.

Number	Component
1	I/O ports 1–16 (CATx)
2	I/O ports 17–32 (CATx)
3	Serial connection (RJ-45)
4	Network connection (RJ-45)

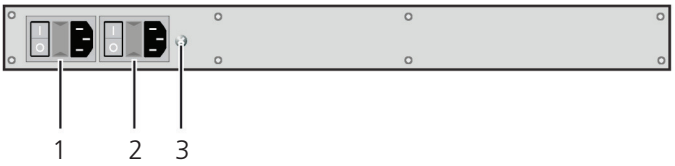


Figure 3-14. Back view, ACXC32.

Table 3-16. ACXC32 back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.6.10 ServSwitch DKM FX Compact 32-Port Fiber (ACXC32F)

Figure 3-15. Front view, ACXC32F.

Table 3-17. ACXC32F components.

Number	Component
1	I/O ports 1–16 (SFP)
2	I/O ports 17–32 (SFP)
3	Serial connection (RJ-45)
4	Network connection (RJ-45)

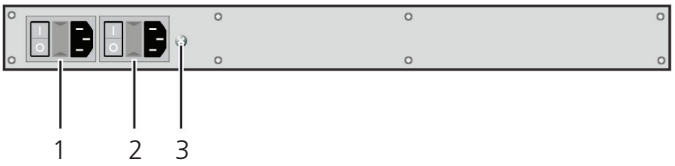


Figure 3-16. Back view, ACXC32F.

Table 3-18. ACXC32F back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.6.11 ServSwitch DKM FX Compact 48-Port (ACXC48)

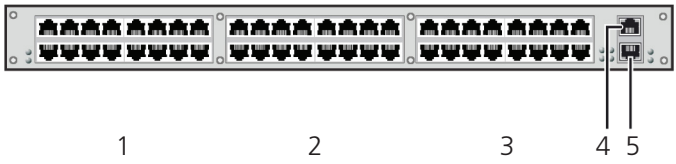


Figure 3-17. Front view, ACXC48.

Table 3-19. ACXC48 front-panel components.

Number	Component
1	I/O ports 1–16 (CATx)
2	I/O ports 17–32 (CATx)
3	I/O ports 33–48 (CATx)
4	Serial connection (RJ-45)
5	Network connection (RJ-45)

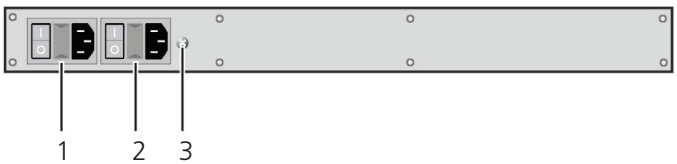


Figure 3-18. Back view, ACXC48.

Table 3-20. ACXC48 back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.6.12 ServSwitch DKM FX Compact 48-Port Fiber (ACXC48F)

Figure 3-19. Front view, ACXC48F.

Table 3-21. ACXC48F front-panel components.

Number	Component
1	I/O ports 1–16 (SFP)
2	I/O ports 17–32 (SFP)
3	I/O ports 33–48 (SFP)
4	Serial connection (RJ-45)
5	Network connection (RJ-45)

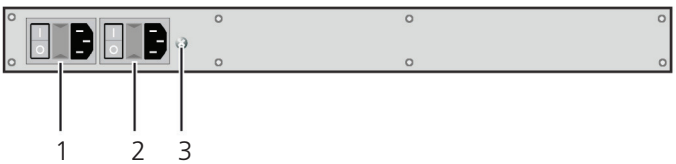


Figure 3-20. Back view, ACXC48F.

Table 3-22. ACXC48 back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.6.13 ServSwitch DKM FX Compact 64-Port

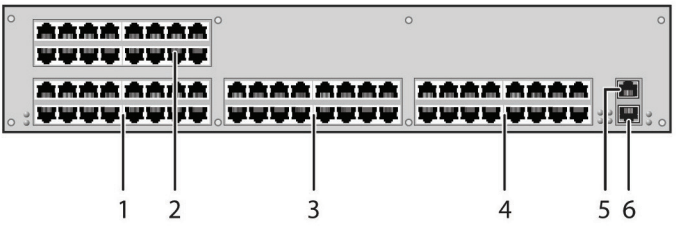


Figure 3-21. Front view, ACXC64.

Table 3-23. ACXC64 components.

Number	Component
1	I/O ports 1–16 (CATx)
2	I/O ports 49–64 (CATx)
3	I/O ports 17–32 (CATx)
4	I/O ports 33–48 (CATx)
5	Serial connection (RJ-45)
6	Network connection (RJ-45)

Figure 3-22. Back view, ACXC64.

Table 3-24. ACXC64 back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.6.14 ServSwitch DKM FX Compact 64-Port Fiber (ACXC64F)

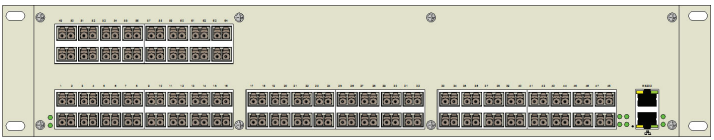


Figure 3-23. Front view, ACXC64F.

Table 3-25. ACXC64F components.

Number	Component
1	I/O ports 1–16 (SFP)
2	I/O ports 49–64 (SFP)
3	I/O ports 17–32 (SFP)
4	I/O ports 33–48 (SFP)
5	Serial connection (RJ-45)
6	Network connection (RJ-45)

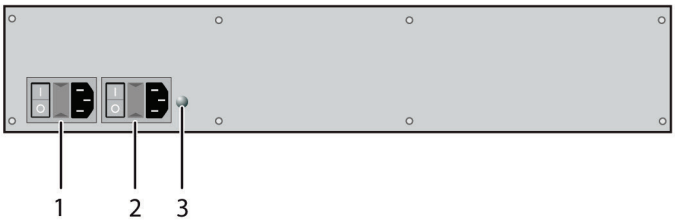


Figure 3-24. Back view, ACXC64F.

Table 3-26. ACXC64F back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.6.15 ServSwitch DKM FX Compact 80-Port (ACXC80)

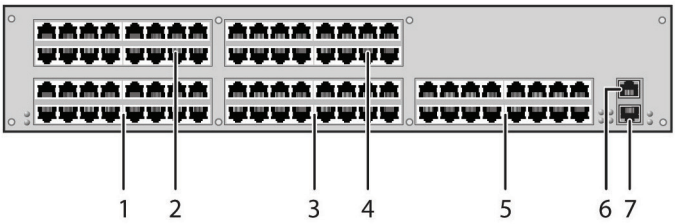


Figure 3-25. Front view, ACXC80.

Table 3-27. ACXC80 components.

Number	Component
1	I/O ports 1–16 (CATx)
2	I/O ports 49–64 (CATx)
3	I/O ports 17–32 (CATx)
4	I/O ports 65–80 (CATx)
5	I/O ports 33–48 (CATx)
6	Serial connection (RJ-45)
7	Network connection (RJ-45)

Figure 3-26. Back view, ACXC80.

Table 3-28. ACXC80 back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.6.16 ServSwitch DKM FX Compact 80-Port Fiber (ACXC80F)

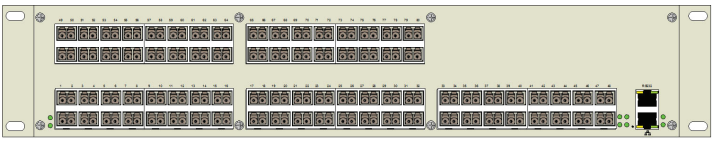


Figure 3-27. Front view, ACXC80F.

Table 3-29. ACXC80F components.

Number	Component
1	I/O ports 1–16 (SFP)
2	I/O ports 49–64 (SFP)
3	I/O ports 17–32 (SFP)
4	I/O ports 65–80 (SFP)
5	I/O ports 33–48 (SFP)
6	Serial connection (RJ-45)
7	Network connection (RJ-45)

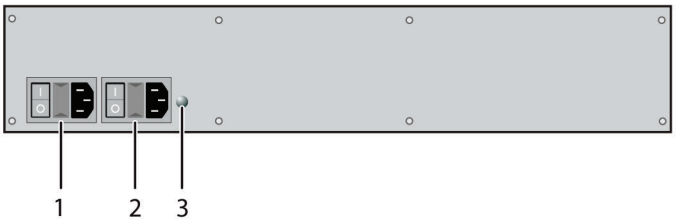


Figure 3-28. Back view, ACXC80.

Table 3-30. ACXC80 back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.6.17 ServSwitch DKM FX Compact 48 CATx Ports, 16 Fiber Ports (ACXC48F16)

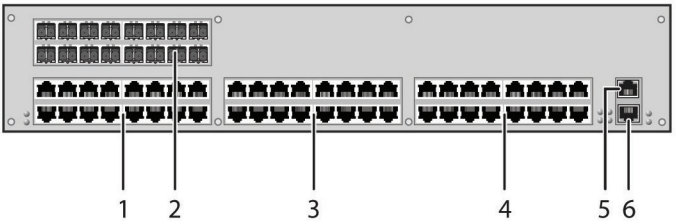


Figure 3-29. Front view, ACXC48F16.

Table 3-31. ACXC48F16 components.

Number	Component
1	I/O ports 1–16 (CATx)
2	I/O ports 49–64 (SFP)
3	I/O ports 17–32 (CATx)
4	I/O ports 33–48 (CATx)
5	Serial connection (RJ-45)
6	Network connection (RJ-45)

Figure 3-30. Back view, ACXC48F16.

Table 3-32. ACXC48F16 back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.6.18 ServSwitch DKM FX Compact 48 CATx Ports, 32 Fiber Ports (ACXC48F32)

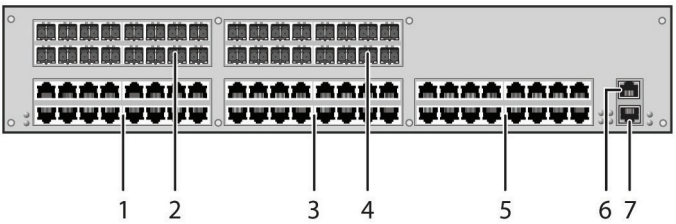


Figure 3-31. Front view, ACXC48F32.

Table 3-33. ACXC48F32 components.

Number	Component
1	I/O ports 1–16 (CATx)
2	I/O ports 49–64 (SFP)
3	I/O ports 17–32 (CATx)
4	I/O ports 65–80 (SFP)
5	I/O ports 33–48 (CATx)
6	Serial connection (RJ-45)
7	Network connection (RJ-45)

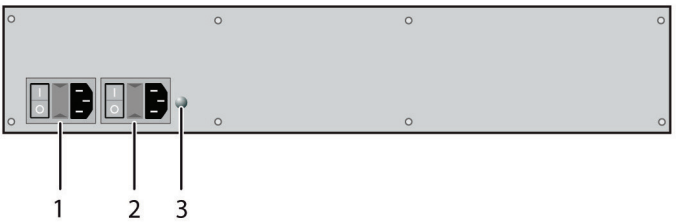


Figure 3-32. Back view, ACXC48F32.

Table 3-34. ACXC48F32 back-panel components.

Number	Component
1	Power supply unit 1
2	Power supply unit 2
3	Grounding

3.7 Status LEDs

The ServSwitch DKM FX and DKM FX Compact LED indicators on the CPU board are shown in Figure 3-33, and described in Table 3-35.

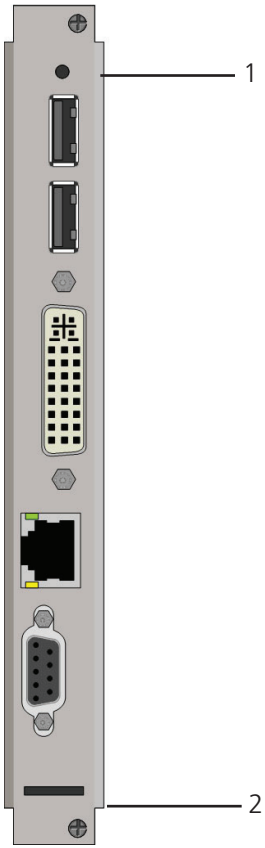


Figure 3-33. CPU board, front view.

Table 3-35. Status LEDs on the CPU board.

Number	LED	Status	Description
1	Status 1	White	CPU board is in registration process
		Blue flashing	Registration at the matrix is started
		Red flashing	Registration is in progess
		Green flashing	Operating condition
		Green	CPU board de-registered
2	TCP/IP Status 1	Red	Operating condition
		Off	No connection
3	TCP/IP Status 2	Green flashing	Active data traffic
		Off	No active data traffic
4	Status 2	White	CPU board is in registration process
		Red flashing	Registration at the matrix is started
		Off	Operating condition

NOTE: Because of variations in the LED type, "white" may also appear as light purple or light blue.

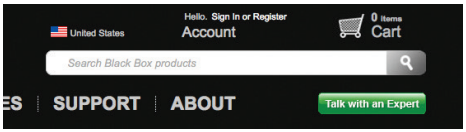
4. Installation

4.1 How to Get Documentation

This API manual can be downloaded from the Black Box Web site.

To download from the Web site:

1. Go to www.blackbox.com
2. Enter the part number in the search box:



3. Click on the "Resources" tab on the product page, and select the document you wish to download.

If you have questions, contact Black Box Technical Support at 724-746-5500 or info@blackbox.com.

4.2 System Setup

NOTE: If you are a first-time user, we recommend that you set up the system in the same room as a test setup. This will allow you to identify and solve any cabling problems, and experiment with your system more conveniently.

Setup of the external control

1. Install the CPU and I/O boards.
2. Connect the keyboard, mouse, and monitor to the CPU board of the DKM FX.
3. Connect the matrix to the power supply.
4. Open OSD via hotkey and log in with administrator rights in the main menu.
5. Configure initially as requested.
6. Connect the external control either via RS-232 or TCP/IP to the matrix.

5. Configuration

5.1 General Remarks

The DKM FX API provides all commands that are necessary to switch the DKM FX matrix.

5.2 DKM FX Configuration

To operate the DKM FX or DKM FX Compact matrix, it has to be configured appropriately. In the following section, all relevant chapters from the DKM FX and DKM FX Compact manual (main manual) are described. For a detailed explanation, refer to the the main manual.

5.2.1 System Data

The DKM FX API relevant system configuration is set in this menu.

You can access the menu via OSD or Java:

Figure 5-1. OSD and Java icons.

You can select between the following DKM FX API relevant settings:

Table 5-1. API relevant settings.

Field	Selection	Description
Enable COM Echo	activated	Send all performed switching commands in the matrix as an echo via serial interface. *NOTE: This function should be enabled when using a media control via serial interface.*
	deactivated	Function not active (default).
Enable LAN Echo	activated	Send all performed switching commands in the matrix as an echo via LAN interface. *NOTE: This function should be enabled when using a media control via TCP/IP connection.*
	deactivated	Function not active (default).

OSD

Select Configuration > System in the main menu.

NOTE: The serial interface can be blocked while the OSD is open.

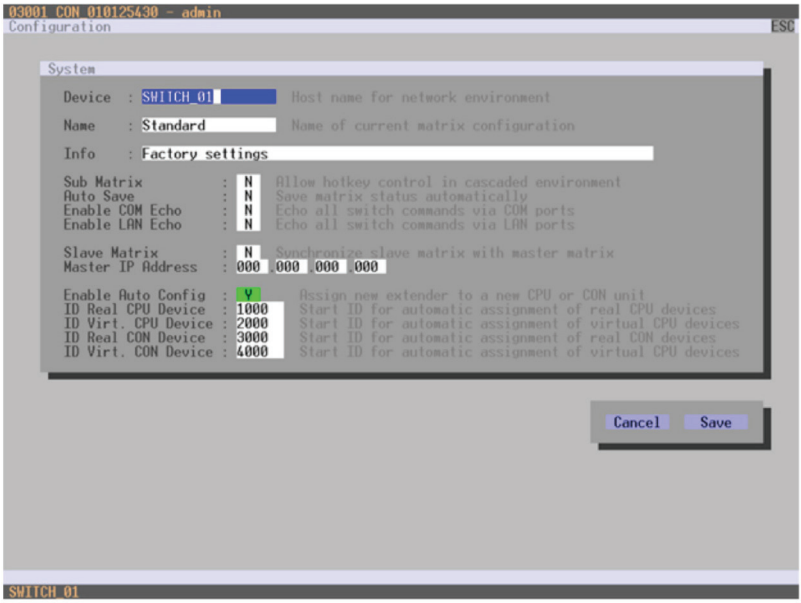


Figure 5-2. Menu Configuration—System.

You can select between the following buttons:

Table 5-2. Button functions.

Number	Component
Cancel	Reject changes
Save	Save changes

Java Tool

Select System > System Data in the main menu.

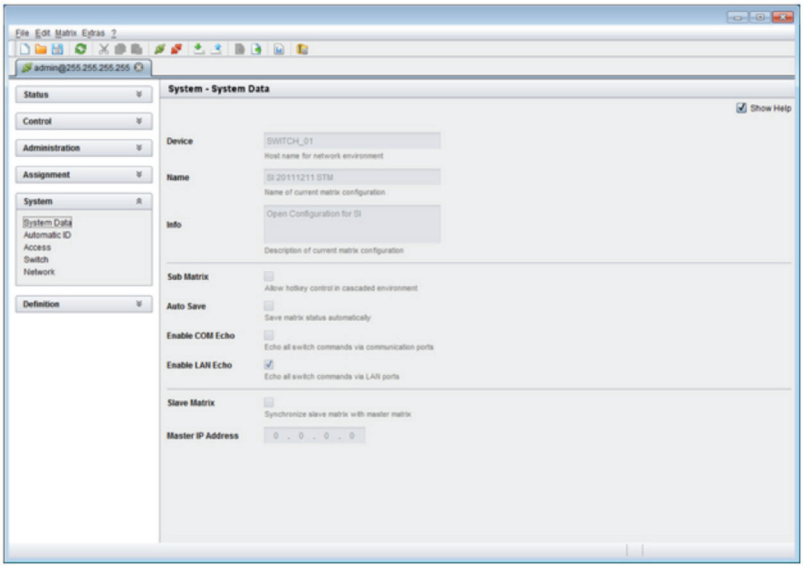


Figure 5-3. Menu System—System Data.

5.2.2 Network

The DKM FX API relevant network configuration is set in this menu.

You can access the menu via OSD or Java:

Figure 5-4. OSD and Java icons.

You can select between the following DKM FX API relevant settings:

Table 5-3. API relevant settings.

Field	Selection	Description
DHCP	activated	The network settings are automatically supplied by a DNS server (default).
	deactivated	Function not active.
IP address	Byte	Input of the IP address in the form “192.168.1.1”, if DHCP is not active.
Subnet mask	Byte	Input of the subnet mask in the form “255.255.255.0”, if DHCP is not active.
Gateway	Byte	Input of the IP address in the form “192.168.1.1”, if DHCP is not active.
Tech Support	activated	LAN interface at the DKM FX activated for access via Java tool (TCP/IP port 5555).
	deactivated	Function not active.
FTP server	activated	FTP server for transmission of configuration files activated.
	deactivated	Function not active.

NOTE: To activate the modified network parameters, restart the API.

CAUTION: Consult your system administrator before modifying the network parameters. Otherwise, unexpected network results and failures can occur.

OSD

Select Configuration > Network in the main menu.

NOTE: The serial interface can be blocked while the OSD is open.

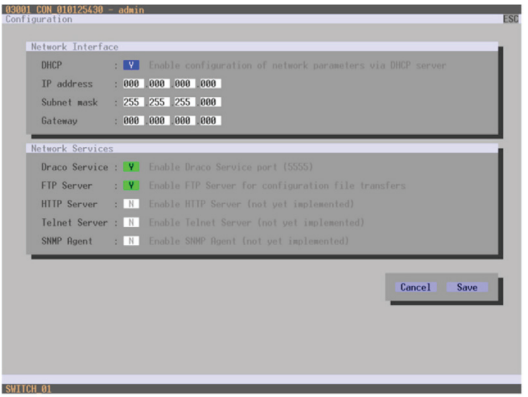


Figure 5-5. Menu Configuration—Network.

You can select between the following buttons:

Table 5-4. Button functions.

Number	Component
Cancel	Reject changes
Save	Save changes

Java Tool

Select System > Network in the task area.

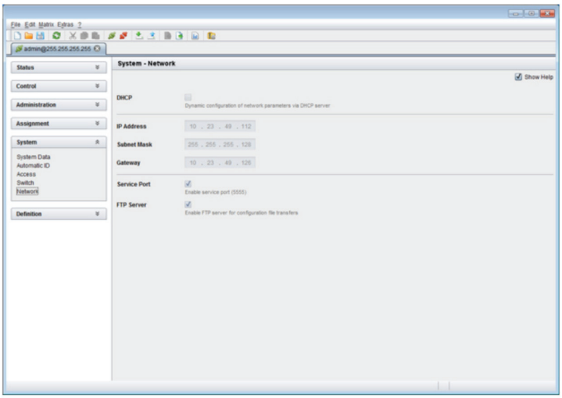


Figure 5-6. Menu System – Network

5.3 Communication Setup

TCP/IP socket connection

To control the DKM FX via TCP/IP socket connection, the Service has to be activated. See Chapter 2.6.4, Network Status and Chapter 4.4.5, Network in the DKM FX and DKM FX Compact manual for more information.

Java code example

```
// Create socket connection
Socket socket = new Socket("192.168.100.108", 5555);
final InputStream is = socket.getInputStream();

// Switch off all ports, Command: ESC [ A
final OutputStream os = socket.getOutputStream();
os.write(0x1B); // ESC
os.write(0x5B); // [
os.write(0x41); // A
os.flush();

if (is.read() == 0x06) {
// acknowledged
}

is.close();
os.close();
socket.close();
```

Serial connection

To establish the serial communication to the DKM FX, set the format for serial data transmission to the following parameters:

115.2K, 8, 1, NO

(115.2 KBAUD, 8 data bits, 1 stop bit, no parity)

5.4 Telegram Structure

5.4.1 Request

ESC <Server identification><Command> [<Size>, <Data>]

[ ] = Optional elements

5.4.2 Response

<ACK>, [<ECHO>]

or

ESC <Server identification><Command><Size><Data>

[ ] = Optional elements

<ACK> Acknowledge

<NAK> Negative Acknowledge

<ECHO> reports the matrix sequences solicited by a command and thus the new switching status of the matrix. The echo can be used to update user applications and to operate several matrices in parallel. See Chapter 4.4.1 System Data in the DKM FX manual, to get more Information about Echo Mode.

NOTE: Use the <ECHO> reports to verify that the switch commands have been executed as requested. Update the external switch status according to the <ECHO> reports rather than according to your commands.

5.5 Constraints

- Maximum buffer size for data transfer is 8192 bytes.
- 12 sockets for TCP/IP communication over Port 5555 are available. Ensure that there will be at least one socket left for the communication with the Java tool.
- Wait for a response before sending another request to the matrix.

6. Operation

The DKM FX API can be addressed in two different ways:

1. Telegrams via serial connection.

2. Telegrams via TCP/IP socket connection.

Both connections accept the same telegrams. Available telegrams support system requests, switch commands, and assignments.

6.1 System Requests

Get System Time

Request
Telegram
ESC (S

Description
Get system time

Example
Get system time. `0x1B 0x28 0x53`

Response
Telegram
`ESC ) S Size Seconds Minutes Hours Day Date Month Year`

Description	
Return system time	
Seconds	Seconds (0–59)
Minutes	Minutes (0–59)
Hours	Hours (0–23)
Day	Days (1–7, Monday = 1)
Date	Date (1–31)
Month	Month (1–12)
Year	Year (+2000)

Example
Return system time: Saturday 15:27:48 28.01.2012 `0x1B 0x29 0x53 0x0C 0x00 0x48 0x27 0x15 0x06 0x28 0x01 0x12`

6.2 Switch Commands

6.2.1 Switch Off All Ports

Request

Telegram
`ESC [ A`

Description
Switch off all ports

Example
Switch off all ports `0x1B 0x5B 0x41`

Response

Telegram
`<ACK> [<ECHO>] or <NAK>` `[ ] = Optional elements`

6.2.2 Get CPU Device Connected to CON Device

Request

Telegram
`ESC [ H Size ConId`

Description
Get CPU device (input) connected to CON device (output) ConId: ID of CON device

Example
Get CPU device connected to CON device `(ConId = 3017)` `0x1B 0x5B 0x48 0x07 0x00 0xC9 0x0B`

Response

Telegram
`ESC ] H Size ConId CpuId`

Description	
Return CPU device (input) connected to CON device (output).	
ConId	ID of CON device
CpuId	ID of CPU device

Example
Return CPU device (`CpuId = 1012`) connected to CON device (`ConId = 3017`) `0x1B 0x5D 0x48 0x09 0x00 0xC9 0x0B 0xF3 0x03` or <NAK>

6.2.3 Set CPU Device Connection to CON Device

Request

Telegram
`ESC [ I Size ConId CpuId`

Description
Set CPU device connection (input) to CON device (output). Input data of CPU device (Video, USB, Audio, …) will be transmitted to CON device. ConId ID of CON device CpuId ID of CPU device

Example
Set CPU device (`CpuId = 1012`) connection to CON device (`ConId = 3017`) `0x1B 0x5B 0x49 0x09 0x00 0xC9 0x0B 0xF4 0x03`

Response

`<ACK> [<ECHO>] or <NAK>`
 	
`[ ] = Optional elements`

6.2.4 Get CPU Devices Connected to CON Devices

Request

Telegram

```
ESC [ J Size ConCnt ConId[1] … ConId[ConCnt]
```

Description

Get CPU devices (input) connected to CON device (output).
For `ConCnt = 0`, all CON devices will be returned.

ConCnt	Number of CON devices
ConId[ ]	List of ConID

Example

Return CPU devices connected to CON devices.

```
(ConId = 3017, 3028, 3040)

0x1B 0x5B 0x4A 0x0D 0x00 0x03 0x00 0xC9 0x0B 0xD4 0x0B
0xE0 0x0B
```

Response

```
ESC ] J Size ConCnt <ConId, CpuId>[1] …

<ConId, CpuId>[ConCnt]
```

Description

Get CPU devices (input) connected to CON devices (output).

Returns a list of pairs of ConId, CpuId.

ConCnt	Number of CON devices
<ConId, CpuId>[ ]	List of IDs (ConID, CpuID)

Example

Get CPU devices connected to CON devices.

```
CpuId[1] = 1012, ConId[1] = 3017;
CpuId[2] = 1013, ConId[2] = 3028;
CpuId[3] = 1020, ConId[3] = 3040;
0x1B 0x5D 0x4A 0x13 0x00 0x03 0x00 0xC9 0x0B 0xF4 0x03
0xD4 0x0B 0xF5 0x03 0x0E 0x0B 0xFC 0x03
```

6.2.5 Set Connections of CPU Devices to CON Devices

Request

Telegram

```
ESC [ K Size ConCnt <ConId, CpuId>[1] …

<ConId, CpuId>[ConCnt]
```

Description

Set connections of CPU devices (input) to CON devices (output).

Data of CPU (video, USB, audio, …) will be transmitted to CON device.

ConCnt	Number of CON devices
<ConId, CpuId>[ ]	List of IDs (ConID, CpuID)

Example

Example: Set connections of CPU devices to CON devices.

```
ConId[1] = 3017, CpuId[1] = 1012;

ConId[2] = 3028, CpuId[2] = 3013;

ConId[3] = 3040, CpuId[3] = 1020;

0x1B 0x5B 0x4B 0x13 0x00 0x03 0x00 0xC9 0x0B 0xF4 0x03
0xD4 0x0B 0xF5 0x03 0x0E 0x0B 0xFC 0x03
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

6.2.6 Get CON Device Connected to CPU Device

Request

Telegram
`ESC [ L Size CpuId`

Description	
Get CON device (input) connected to CPU device (output).	
CpuId	ID of CPU device

Example
Get CON device connected to CPU device (`CpuId = 1012`).
`0x1B 0x5B 0x4C 0x07 0x00 0xF4 0x03`

Response

`ESC ] L Size CpuId ConId`

Description	
Return CON device (input) connected to CPU device (output).	
CpuId	ID of CPU device
ConId	ID of CON device

Example
Return CON device (ConId = 3017) connected to CPU device (`CpuId = 1012`).
`0x1B 0x5D 0x4C 0x09 0x00 0xF4 0x03 0xC9 0x0B`

or <NAK>

6.2.7 Set CON Device Connection to CPU Device

Request

Telegram
`ESC [ M Size CpuId ConId`

Description	
Set CON device (input) connection to CPU device (output). Input data of CON device (USB, Audio) will be transmitted to CPU device.	
CpuId	ID of CPU device
ConId	ID of CON device

Example
Set CON device (`ConId = 3017`) connection to CPU device (`CpuId = 1012`) `0x1B 0x5B 0x4D 0x09 0x00 0xF4 0x03 0xC9 0x0B`

Response

`<ACK> [<ECHO>] or <NAK>` `[ ] = Optional elements`

6.2.8 Get CON Devices Connected to CPU Devices

Request

Telegram

```
ESC [ N Size CpuCnt CpuId[1] … CpuId[CpuCnt]
```

Description

Get CON devices (input) connected to CPU devices (output).

For `CpuCnt = 0`, all CPU devices will be returned.

CpuCnt	Number of CPU devices
CpuId[ ]	List of CpuIDs

Example

Get CON devices connected to CPU devices (`CpuId = 1012, 1013, 1020`)

```
0x1B 0x5B 0x4E 0x0D 0x00 0x03 0x00 0xF4 0x03 0xF5 0x03
0xFC 0x03
```

Response

Telegram

```
ESC ] N Size CpuCnt <CpuId, ConId>[1] …

<CpuId, ConId>[CpuCnt]
```

Description

Return CON devices (input) connected to CPU devices (output).

Return a list of pairs of CpuId, ConId.

CpuCnt	Number of CPU devices
<CpuId, ConId>[ ]	List of IDs (CpuID, ConID)

Example

Return CON devices connected to CPU devices.

```
CpuId[1] = 1012, ConId[1] = 3017;
CpuId[2] = 1013, ConId[2] = 3028;
CpuId[3] = 1020, ConId[3] = 3040;

0x1B 0x5D 0x4E 0x13 0x00 0x03 0x00 0xF4 0x03 0xC9 0x0B
0xF5 0x03 0xD4 0x0B 0xFC 0x03 0x0E 0x0B
```

6.2.9 Set Connection of CON Devices to CPU Devices

Request

Telegram

```
ESC [ O Size CpuCnt <CpuId, ConId>[1] …
<CpuId, ConId>[CpuCnt]
```

Description

Set connection CON devices (input) to CPU devices (output).

Data of CON device (USB, audio) will be transmitted to CPU device.

CpuCnt	Number of CPU devices
<CpuId, ConId>[ ]	List of IDs (CpuID, ConID)

Example

Set connection of CON devices to CPU devices.

```
CpuId[1] = 1012, ConId[1] = 3017;
CpuId[2] = 1013, ConId[2] = 3028;
CpuId[3] = 1020, ConId[3] = 3040;
```

```
0x1B 0x5B 0x4F 0x13 0x00 0x03 0x00 0xF4 0x03 0xC9 0x0B
0xF5 0x03 0xD4 0x0B 0xFC 0x03 0x0E 0x0B
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

6.2.10 Set CON Device Connection to CPU Device (Single Bidirectional Connection – KVM Full Access)

Request

Telegram

```
ESC [ P Size CpuId ConId
```

Description

Set CON device (input) connection to CPU device (output) and CPU device (input) connection to CON device (output).
Data of CON device (USB, audio, …) will be transmitted to CPU device.
Data of CPU device video, USB, audio, …) will be transmitted to CON device.

CpuId	ID of CPU device
ConId	ID of CON device

Example

Set CON device (`ConID = 3017`) connection to CPU device (`CpuId = 1012`).

```
0x1B 0x5B 0x50 0x09 0x00 0xF4 0x03 0xC9 0x0B
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

6.2.11 Set Connection of CON Devices to CPU Devices (Multiple Bidirectional Connections—Full Access)

Request

Telegram

```
ESC [ Q Size Cnt <CpuId, ConId>[1] …

<CpuId, ConId>[Cnt]
```

Description

Set connection of CON devices (input) to CPU devices (output) and CPU devices (input) to CON devices (output).
Data of CON device (USB, audio, …) will be transmitted to CPU device. Data of CPU device (video, USB, audio, …) will be transmitted to CON device.

Cnt	Size of list
<CpuId, ConId>[ ]	List of IDs (CpuID, ConID)

Example

Set connection of CON devices to CPU devices (bidirectional).

```
CpuId[1] = 1012, ConId[1] = 3017;
CpuId[2] = 1013, ConId[2] = 3028;
CpuId[3] = 1020, ConId[3] = 3040;

0x1B 0x5B 0x51 0x13 0x00 0x03 0x00 0xF4 0x03 0xC9 0x0B
0xF5 0x03 0xD4 0x0B 0xFC 0x03 0x0E 0x0B
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

6.2.12 Get All Connections

Request

Telegram

```
ESC [ R
```

Description

Get all CPU device–CON device connections.

Example

Get all CPU device – CON device connections.

```
0x1B 0x5B 0x52
```

Response

Telegram

```
ESC ] R Size CpuCnt ConCnt <CpuId, ConId>[1] …
<CpuId, ConId>[ CpuCnt] <ConId, CpuId>[1] …
<ConId, CpuId>[ ConCnt]
```

Description

Return all CPU device–CON device connections in pairs.

For each defined CPU device, the ConId of the connected CON device will be added, or 0 if the CPU device is disconnected.

For each defined CON device, the CpuId of the connected CPU device will be added, or 0 if the CON device is disconnected.

CpuCnt	Number of CPU devices
ConCnt	Number of CON devices
<CpuId, ConId>[ ]	List of IDs (CpuID, ConID)
<CpuId, ConId>[ ]	List of IDs (ConID, CpuID)

Example

Return all CPU device–CON device connections in pairs.

```
CpuId[1] = 1012, ConId[1] = 3017;
CpuId[2] = 1013, ConId[2] = 3028;
CpuId[3] = 1020, ConId[3] = 3040;
ConId[1] = 3017, CpuId[1] = 1012;
ConId[2] = 3028, CpuId[2] = 0;
```

```
0x1B 0x5D 0x52 0x15 0x00 0x03 0x00 0x02 0x00 0xF4 0x03
0xC9 0x0B 0xF5 0x03 0xD4 0x0B 0xFC 0x03 0x0E 0x0B 0xC9
0x0B 0xF4 0x03 0xD4 0x0B 0x00 0x00
```

6.2.13 Set Connection for All CON Devices and CPU Devices

Request

Telegram

```
ESC [ S Size CpuCnt ConCnt <CpuId, ConId>[1] …
<CpuId, ConId>[ CpuCnt] <ConId, CpuId>[1] …
<ConId, CpuId>[ ConCnt]
```

Description

Set a connection for all defined CON devices and CPU devices.

For each defined CPU device, add the ConId, or 0 if the CPU device is disconnected.

For each defined CON device, add the CpuId, or 0 if the CON device is disconnected.

CpuCnt	Number of CPU devices
ConCnt	Number of CON devices
<CpuId, ConId>[ ]	List of IDs (CpuID, ConID)
<CpuId, ConId>[ ]	List of IDs (ConID, CpuID)

Example

Set a connection for all defined CON devices and CPU devices.

```
CpuId[1] = 1012, ConId[1] = 3017;
CpuId[2] = 1013, ConId[2] = 3028;
CpuId[3] = 1020, ConId[3] = 3040;
ConId[1] = 3017, CpuId[1] = 1012;
ConId[2] = 3028, CpuId[2] = 0;
```

```
0x1B 0x5B 0x53 0x15 0x00 0x03 0x00 0x02 0x00 0xF4 0x03
0xC9 0x0B 0xF5 0x03 0xD4 0x0B 0xFC 0x03 0x0E 0x0B 0xC9
0x0B 0xF4 0x03 0xD4 0x0B 0x00 0x00
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

6.2.14 Set Extended Connection

Request

Telegram

```
ESC [ b Size CpuId ConId Mode
```

Description

Set CON device (input) connection to CPU device (output) and CPU device (input) connection to CON device (output).
Data of CON device (USB, audio, …) is transmitted to a CPU device.
Data of CPU device (video, USB, audio, …) is transmitted to a CON device.

CpuId	ID of CPU device
ConId	ID of CON device
Mode	Connection Mode (0 = full access, 1 = video only, 2 = private mode)

Example

Set CON device connection to CPU device and CPU device connection to CON device.

```
CpuId = 1012, ConId = 3017, Mode = private mode
0x1B 0x5B 0x62 0x0B 0x00 0xF4 0x03 0xC9 0x0B 0x02 0x00
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

6.2.15 Get CPU List

Request

Telegram
`ESC [ g Size First`

Description
Get list of all CPU devices (output) including ID and name First: Index of CPU device from which the list scan will start

Example
Get all CPUs `0x1B 0x5B 0x67 0x07 0x00 0x00 0x00`

Response

Telegram
`ESC ] g Size Count Next List [1] … List [Count]`

Description
Count: Number of items in the CPU list Next: Index of the next CPU, if the list of CPU devices exceeds the telegram size. Contains 0 if there are no more CPU devices

Example

Return list of CPUs
`0x1B 0x5D 0x67 0x21 0x00 0x01 0x00 0x00 0x00 0xE8 0x03`
`0x00 0x00 0x43 0x50 0x55 0x5F 0x56 0x69 0x64 0x65 0x6F`
`0x31 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00`

6.2.16 Get CON List

Request

Telegram
`ESC [ h Size First`

Description
Get list of all CON devices (input) including ID and nameFirst: Index of CON device from which the list scan will start

Example
Get all CONs `0x1B 0x5B 0x68 0x07 0x00 0x00 0x00`

Response

Telegram
`ESC ] h Size Count Next List [1] … List [Count]`

Description
Count: Number of items in the CPU list Next: Index of the next CPU, if the list of CPU devices exceeds the telegram size. Contains 0 if there are no more CPU devices

Example

Return list of CONs
`0x1B 0x5D 0x68 0x21 0x00 0x01 0x00 0x00 0x00 0xB8 0x0B`
`0x00 0x00 0x43 0x4F 0x4E 0x5F 0x56 0x69 0x64 0x65 0x6F`
`0x31 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00`

6.2.17 Get User List

Request

Telegram
`ESC [ i Size First`

Description
Get list of all users First: Index of the user from whom the list scan will start

Example
Get all users `0x1B 0x5B 0x69 0x07 0x00 0x00 0x00`

Response

Telegram
`ESC ] i Size Count Next List [1] … List [Count]`

Description
Count: Number of items in the user list Next: Index of the next user, if the list of users exceeds the telegram size. Contains 0 if there are no more users.

Example
Return list of users 0x1B 0x5D 0x69 0x21 0x00 0x01 0x00 0x00 0x00 0x01 0x00 0x00 0x00 0x61 0x64 0x6D 0x69 0x6E 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00

6.3 Assignments

6.3.1 Get Virtual CON Device

Request

Telegram
`ESC [ T Size RConId`

Description	
Get virtual CON device of a real CON device.	
RConId	ID of real CON

Example
Get virtual CON device of a real CON device (`RConId = 3017`).
`0x1B 0x5B 0x54 0x07 0x00 0xC9 0x0B`

Response

Telegram
`ESC ] T Size RConId VConId`

Description	
Return virtual CON device of a real CON device.	
RConId	ID of real CON device
VConId	ID of virtual CON device

Example
Return virtual CON device (`VConId = 4034`) of a real CON device (`RConId = 3017`).
`0x1B 0x5B 0x54 0x09 0x00 0xC9 0x0B 0xC2 0x0F`

or <NAK>

6.3.2 Set Virtual CON Device to a Real CON Device

Request

Telegram
`ESC [ U Size RConId VConId`

Description
Set virtual CON device to a real CON device. RConId ID of real CON device VConId ID of virtual CON device

Example
Set virtual CON device (`VConId = 4034`) to a real CON device (`RConId = 3017`). `0x1B 0x5B 0x55 0x09 0x00 0xC9 0x0B 0xC2 0x0F`

Response

`<ACK> [<ECHO>] or <NAK>`
 	
`[ ] = Optional elements`

6.3.3 Get Real CPU Device

Request

Telegram
`ESC [ V Size VcpuId`

Description
Get real CPU device of a virtual CPU device. VCpuId ID of virtual CPU device

Example
Get real CPU device of a virtual CPU device (`VCpuId = 2018`). `0x1B 0x5B 0x56 0x07 0x00 0xE2 0x07`

Response

Telegram
`ESC ] V Size VCpuId RCpuId`

Description
Return real CPU device of a virtual CPU device. VCpuId ID of virtual CPU device RCpuId ID of real CPU device

Example
Return real CPU device (`RCpuId = 1012`) of a virtual CPU device (`VCpuId = 2018`). `0x1B 0x5D 0x56 0x09 0x00 0xE2 0x07 0xF4 0x03`

or <NAK>

6.3.4 Set Real CPU Device to a Virtual CPU Device

Request

Telegram

```
ESC [ W Size VCpuId RCpuId
```

Description

Set real CPU device to a virtual CPU device.

VCpuId	ID of virtual CPU device
RCpuId	ID of real CPU device

Example

Example

Set real CPU device (`RCpuId = 1012`) to a virtual CPU device (`VCpuId = 2018`).

```
0x1B 0x5B 0x57 0x09 0x00 0xE2 0x07 0xF4 0x03
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

6.3.5 Get Virtual CON Devices

Request

Telegram
`ESC [ X Size ConCnt RConId[1] … RConId[ConCnt]`

Description
Get virtual CON devices of a real CON devices. For `ConCnt = 0`, all real CON devices with assignments to virtual CON devices will be returned. ConCnt — Number of CON devices RConId[ ] — List of IDs of real CON devices

Example
Get virtual CON devices of a real CON devices (`RConId = 3017, 3028, 3040`). `0x1B 0x5B 0x58 0x0D 0x00 0x03 0x00 0xC9 0x0B 0xD4 0x0B 0xE0 0x0B`

Response

Telegram

`ESC ] X Size ConCnt <RConId, VConId>[1] …`
 	
`<RConId, VConId>[ConCnt]`

Description
Return virtual CON devices of real CON devices as pairs. ConCnt — Number of CON devices <RConId, VConId>[ ] — List of Pairs of RConId and VConId

Example

Return virtual CON devices of real CON devices as pairs.
`RConID[1] = 3017, VConID[1] = 4034;`
`RConID[2] = 3028, VConID[2] = 4042;`
`RConID[3] = 3040, VConID[3] = 4045;`
`0x1B 0x5D 0x58 0x13 0x00 0xC9 0x0B 0xC2 0x0F 0xD4 0x0B 0xCA 0x0F 0xE0 0x0B 0xCD 0x0F`

6.3.6 Set Virtual CON Devices to Real CON Devices

Request

Telegram
`ESC [ Y Size ConCnt <RConId, VConId>[1] …` `<RConId, VConId>[ConCnt]`

Description	
Set virtual CON devices to real CON devices.	
ConCnt	Number of CON devices
<RConId, VConId>[ ]	List of Pairs of RConId and VConId

Example

Set virtual CON devices to real CON devices.
`RConId[1] = 3017, VConId[1] = 4034;`
`RConId[2] = 3028, VConId[2] = 4042;`
`RConId[3] = 3040, VConId[3] = 4045;`
`0x1B 0x5B 0x59 0x13 0x00 0xC9 0x0B 0xC2 0x0F 0xD4 0x0B 0xCA 0x0F 0xE0 0x0B 0xCD 0x0F`

Response

`<ACK> [<ECHO>] or <NAK>` `[ ] = Optional elements`

6.3.7 Get Real CPU Devices

Request

Telegram
`ESC [ Z Size CpuCnt VCpuId[1] … VCpuId[CpuCnt]`

Description	
Get real CPU devices of virtual CPU devices.	
For `CpuCnt = 0`, all virtual CPU devices with assignments to virtual CPU devices will be returned.	
CpuCnt	Number of CPU device
VCpuId[ ]	List of IDs of virtual CPU devices

Example
Get real CPU devices of virtual CPU devices (`VCpuId = 2018, 2030, 2035`).
`0x1B 0x5B 0x5A 0x0D 0x00 0x03 0x00 0xE2 0x07 0xEE 0x07 0xF3 0x07`

Response

Telegram
`ESC ] Z Size CpuCnt <VCpuId, RCpuId>[1] …`
`<VCpuId, RCpuId>[CpuCnt]`

Description	
Return real CPU devices of virtual CPU devices as pairs.	
CpuCnt	Number of CPU devices
<VCpuId, RCpuId>[ ]	List of pairs of VCpuId and RCpuId

Example

Return real CPU devices of virtual CPU devices as pairs.

```
VCpuId[1] = 2018, RCpuId[1] = 1012;

VCpuId[2] = 2030, RCpuId[2] = 1013;

VCpuId[3] = 2035, RCpuId[3] = 1020;

0x1B 0x5D 0x5A 0x13 0x00 0x03 0x00 0xE2 0x07 0xF4 0x03
0xEE 0x07 0xF5 0x03 0xF3 0x07 0xFC 0x03
```

6.3.8 Set Real CPU Devices

Request

Telegram

```
ESC [ a Size CpuCnt <VCpuId, RCpuId>[1] …

<VCpuId, RCpuId>[CpuCnt]
```

Description

Set real CPU devices to virtual CPU devices.

CpuCnt	Number of CPU devices
<VCpuId, RCpuId>[ ]	List of Pairs of VCpuId and RCpuId

Example

Set real CPU devices to virtual CPU devices.

```
VCpuId[1] = 2018, RCpuId[1] = 1012;
VCpuId[2] = 2030, RCpuId[2] = 1013;
VCpuId[3] = 2035, RCpuId[3] = 1020;

0x1B 0x5B 0x61 0x13 0x00 0x03 0x00 0xE2 0x07 0xF4 0x03
0xEE 0x07 0xF5 0x03 0xF3 0x07 0xFC 0x03
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

6.4 Best Practice

This chapter provides an overview of the most commonly used switching commands and how they can be operated by using proven code examples of the external serial control.

6.4.1 Full Access (Establishing a KVM Connection)

Set CON device (`ConID = 3017`) connection to CPU device (`CpuId = 1012`):

```
0x1B 0x5B 0x50 0x09 0x00 0xF4 0x03 0xC9 0x0B
```

Disconnect:

```
0x1B 0x5B 0x50 0x09 0x00 0xF4 0x03 0x00 0x00
```

6.4.2 Video Access (Establishing a Video Only Connection)

Set CPU device (`CpuId = 1012`) connection to CON device (`ConId = 3017`):

```
0x1B 0x5B 0x49 0x09 0x00 0xC9 0x0B 0xF4 0x03
```

Disconnect:

```
0x1B 0x5B 0x49 0x09 0x00 0xC9 0x0B 0x00 0x00
```

6.4.3 Private Access (Establishing an Exclusive KVM Session)

Set CON device connection to CPU device and CPU device connection to CON device, CpuId = 1012 and ConId = 3017:

```
0x1B 0x5B 0x62 0x0B 0x00 0xF4 0x03 0xC9 0x0B 0x02 0x00
```

Disconnect:

```
0x1B 0x5B 0x62 0x0B 0x00 0xF4 0x03 0x00 0x00 0x02 0x00
```

6.4.4 USB 2.0 Access (Establishing a USB 2.0 Data Connection)

To set a USB 2.0 connection based on devices that only consist of USB 2.0 standalone extenders, you have to split the required bidirectional switching into two steps:

1. Set CPU device (CpuId = 1012) connection to CON device (`ConId = 3017`):
   `0x1B 0x5B 0x49 0x09 0x00 0xC9 0x0B 0xF4 0x03`
2. Set CON device (`ConId = 3017`) connection to CPU device (`CpuId = 1012`):
   `0x1B 0x5B 0x4D 0x09 0x00 0xF4 0x03 0xC9 0x0B`

Switching from a device within an existing connection to another device requires closing the current connection at first. The disconnect has to be performed in two steps:

1. Disconnect CPU device (`CpuId = 1012`) from CON device (`ConId = 3017`):
   `0x1B 0x5B 0x49 0x09 0x00 0xC9 0x0B 0x00 0x00`
2. Disconnect CON device (`ConId = 3017`) from CPU device (`CpuId = 1012`):
   `0x1B 0x5B 0x4D 0x09 0x00 0xF4 0x03 0x00 0x00`

NOTE: After disconnecting the existing connection, a switching break of 1-2 seconds is strongly recommended until the next switching operation should be executed.

7. Specifications

The DKM FX can be controlled via RS-232 serial interface or TCP/IP socket (Port 5555).

Table 7-1. Telegram structure.

Type	Bytes	Description
Control character	1	Always: ESC (0x1B)
Server identification	1	Identification of service
Command	1	A special command
Size	2	Optional, if telegram size is greater than 3
Data	n	Optional, n bytes of data

Byte Order: Little Endian

Example: 1012 —> 0xF4 0x03 (*not* 0x03 0xF4)

(Special) characters: ACK 0x06
NAK 0x15

Request

```
ESC <Server identification><Command> [<Size>, <Data>]
[    ] = Optional elements
```

Response

```
<ACK> , [<ECHO>]

or

ESC <Server identification><Command><Size><Data>

[    ] = Optional elements
```

<ECHO> reports the matrix sequences solicited by a command and thus the new switching status of the matrix. The echo can be used to update user applications and to operate several matrices in parallel.

Table 7-2. Sequence of a data communication.

DKM FX Matrix	Control CPU
—	Sending a command
Acquiring a command, processing a command, blocking of further commands	—
a) Errors occurred: <NAK> b) No errors <ACK> c) Optional: <ECHO> d) Optional: Reply telegram with data	—
—	a) Repeat telegram b) Next command c) Receive and process the reply telegram

NOTE: The serial interface can be blocked while the OSD is open.

7.1 System Requests

Get System Time

Request

Telegram

`ESC (S`

General Description

Get system time

Table 7-3. Get system time.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
(	1	Server identification	0x28
S	1	Command	0x53

Example

Example: Get system time

```
0x1B 0x28 0x53
```

Response

Telegram

```
ESC ) S Size Seconds Minutes Hours Day Date Month Year
```

General Description

Return system time

Table 7-4. Get system time.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
(	1	Server identification	0x28
S	1	Command	0x53
Size	2	Total length of telegram (12 bytes)	0x0C 0x00
Seconds	1	Seconds (0–59)	0x00 - 0x59
Minutes	1	Minutes (0–59)	0x00 - 0x59
Hours	1	Hours (0–23)	0x00 - 0x23
Day	1	Day (1–7, Monday = 1)	0x01 - 0x07
Date	1	Date (1–31)	0x01 - 0x31
Month	1	Month (1–12)	0x01 - 0x12
Year	1	Year (+2000)	e. g. 2012 = 0x12

Example

Return system time: Saturday 15:27:48 28.01.2012

```
0x1B 0x29 0x53 0x0C 0x00 0x48 0x27 0x15 0x06 0x28 0x01
0x12
```

NOTE: The system is encoded in the BCD format.

7.2 Switch Commands

7.2.1 Switch Off All Ports

Request

Telegram

```
ESC [ A
```

General Description

Switch off all ports

Table 7-5. Switch off all ports.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
(	1	Server identification	0x5B
S	1	Command	0x41

Example

Switch off all ports

```
0x1B 0x5B 0x41
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

7.2.2 Get CPU Device Connected to CON Device

Request

Telegram

```
ESC [ H Size ConId
```

General Description

Get CPU device (input) connected to CON device (output).

Table 7-6. Get system time.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
(	1	Server identification	0x28
S	1	Command	0x53
Size	2	Total length of telegram (7 bytes)	0x07 0x00
ConID	2	ID of Con Device	e.g. 3017 = 0xC9 0x08

Example

Get CPU device connected to CON device (`ConId` = 3017).

```
0x1B 0x5B 0x48 0x07 0x00 0xC9 0x0B
```

Response

Telegram

```
ESC ] H Size ConId CpuId
```

General Description

General description: Return CPU device (input) connected to CON device (output)

Table 7-7. Return CPU device.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
(	1	Server identification	0x5D
H	1	Command	0x48
Size	2	Total length of telegram (9 bytes)	0x09 0x00
ConID	2	ID of Con device	e.g. 3017 = 0xC9 0x0B
CpuID	2	ID of CPU device	e.g. 1025 = 0x16 0x04

Example

Return CPU device (`CpuId = 1012`) connected to CON device (`ConId = 3017`).

```
0x1B 0x5D 0x48 0x09 0x00 0xC9 0x0B 0xF3 0x03
```

7.2.3 Set CPU Device Connection to CON Device

Request

Telegram

```
ESC [ I Size ConId CpuId
```

General Description

Set CPU device connection (input) to CON device (output).Input data of CPU device (video, USB, audio, …) will be transmitted to CON device.

Table 7-8. Set CPU device connection to CON device.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
I	1	Command	0x49
Size	2	Total length of telegram (9 bytes)	0x09 0x00
ConID	2	ID of Con Device	e.g. 3017 = 0xC9 0x0B
CpuId	2	ID of CPU device	e.g, 1012 = 0xF4 0x03

Example

Set CPU device (CpuId = 1012) connection to CON device (`ConId = 3017`).

```
0x1B 0x5B 0x49 0x09 0x00 0xC9 0x0B 0xF4 0x03
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

7.2.4 Get CPU Devices Connected to CON Devices

Request

Telegram

```
ESC [ J Size ConCnt ConId[1] … ConId[ConCnt]
```

General Description

Get CPU devices (input) connected to CON device (output).

`For ConCnt = 0`, all CON devices will be returned.

Table 7-9. Get CPU devices connected to CON device.

Type	Bytes	Description	Hex Coding
`ESC`	1	Control character	0x1B
`[`	1	Server identification	0X5B
`J`	1	Command	0x4A
Size	2	Total length of telegram (7 Bytes + data)	e.g. for `ConCnt = 3` 0x0D 0x00
ConCnt	2	Number of CON devices	e.g. 3 = 0x03 0x00
ConId	2	ID of CON device	e.g, 3017 = 0xC9 0x0B

Example

Return CPU devices connected to CON devices.

`(ConId = 3017, 3028, 3040)`

```
0x1B 0x5B 0x4A 0x0D 0x00 0x03 0x00 0xC9 0x0B 0xD4 0x0B
0xE0 0x0B
```

Response

Telegramn

```
ESC ] J Size ConCnt <ConId, CpuId>[1] …

<ConId, CpuId>[ConCnt]
```

General Description

Get CPU devices (input) connected to CON devices (output).Returns a list of pairs of ConId, CpuId.

Table 7-10. Get CPU devices (input) connected to CON devices (output).

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
]	1	Server identification	0X5D
J	1	Command	0x4A
Size	2	Total length of telegram (7 Bytes + data)	e.g. for ConCnt = 3 0x13 0x00
ConCnt	2	Number of CON devices	e.g. 3 = 0x03 0x00
ConId	2	ID of CON device	e.g. 3017 = 0xC9 0x0B
CpuId	2	ID of CPU device	e.g. 1012 = 0xF4 0x03

Example

Get CPU devices connected to CON devices.

```
CpuId[1] = 1012, ConId[1] = 3017;
CpuId[2] = 1013, ConId[2] = 3028;
CpuId[3] = 1020, ConId[3] = 3040;

0x1B 0x5D 0x4A 0x13 0x00 0x03 0x00 0xC9 0x0B 0xF4 0x03

0xD4 0x0B 0xF5 0x03 0x0E 0x0B 0xFC 0x03
```

7.2.5 Set Connections of CPU Devices to CON Devices

Request

Telegram

```
ESC [ K Size ConCnt <ConId, CpuId>[1] …
<ConId, CpuId>[ConCnt]
```

General Description

Set connections of CPU devices (input) to CON devices (output). Data of CPU (video, USB, audio, …) will be transmitted to CON device.

Table 7-11. Set connections of CPU devices to CON devices.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5D
K	1	Command	0x4A
Size	2	Total length of telegram (7 Bytes + data)	e.g. for ConCnt = 3 0x13 0x00
ConCnt	2	Number of CON devices	e.g. 3 = 0x03 0x00
ConId	2	ID of CON device	e.g. 3017 = 0xC9 0x0B
CpuId	2	ID of CPU device	e.g. 1012 = 0xF4 0x03

Example

Set connections of CPU devices to CON devices.

```
ConId[1] = 3017, CpuId[1] = 1012;
ConId[2] = 3028, CpuId[2] = 3013;
ConId[3] = 3040, CpuId[3] = 1020;

0x1B 0x5B 0x4B 0x13 0x00 0x03 0x00 0xC9 0x0B 0xF4 0x03
0xD4 0x0B 0xF5 0x03 0x0E 0x0B 0xFC 0x03
```

Response

```
<ACK> [<ECHO>] or <NAK>

[   ] = Optional elements
```

7.2.6 Get CON Device Connected to CPU Device

Request

Telegram

```
ESC [ L Size CpuId
```

General Description

Get CON device (input) connected to CPU device (output).

Table 7-12. Get CON device connected to CPU device.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
L	1	Command	0x4C
Size	2	Total length of telegram (7 bytes)	0x07 0x00
CpuId	2	ID of CPU device	e.g. 1012 = 0xF4 0x03

Example

Get CON device connected to CPU device (`CpuId = 1012`).

```
0x1B 0x5B 0x4C 0x07 0x00 0xF4 0x03
```

Response

Telegram

```
ESC ] L Size CpuId ConId
```

General Description

Return CON device (input) connected to CPU device (output).

Table 7-13. Return CON device (input) connected to CPU device (output).

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
]	1	Server identification	0X5D
L	1	Command	0x4C
Size	2	Total length of telegram (9 Bytes)	0x09 0x00
CpuId	2	ID of CPU device	e.g. 1012 = 0xF4 0x03
ConId	2	ID of CON device	e.g. 3017 - 0xC9 0x0B

Example

Return CON device (`ConId = 3017`) connected to CPU device (`CpuId = 1012`).

```
0x1B 0x5D 0x4C 0x09 0x00 0xF4 0x03 0xC9 0x0B
```

or <NAK>

7.2.7 Set CON Device Connection to CPU Device

Request

Telegram

```
ESC [ M Size CpuId ConId
```

General Description

Set CON device (input) connection to CPU device (output). Input data of CON device (USB, audio) will be transmitted to CPU device.

Table 7-14. Set CON device (input) connected to CPU device (output).

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
M	1	Command	0x4D
Size	2	Total length of telegram (9 bytes)	0x09 0x00
CpuId	2	ID of CPU device	e.g. 1012 = 0xF4 0x03
ConId	2	ID of CON device	e.g. 3017 - 0xC9 0x0B

Example

Set CON device (`ConId = 3017`) connection to CPU device (`CpuId = 1012`).

`0x1B 0x5B 0x4D 0x09 0x00 0xF4 0x03 0xC9 0x0B`

Response

`<ACK> [<ECHO>] or <NAK>`

`[ ] = Optional elements`

7.2.8 Get CON Devices Connected to CPU Devices

Request

Telegram

`ESC [ N Size CpuCnt CpuId[1] … CpuId[CpuCnt]`

General Description

Get CON devices (input) connected to CPU devices (output).

For `CpuCnt = 0`, all CPU devices will be returned.

Table 7-15. Get CON devices (input) connected to CPU devices (output).

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
N	1	Command	0x4E
Size	2	Total length of telegram (7 Bytes + data)	e.g. for `CpuCnt = 3` 0x0D 0x00
CpuCnt	2	ID of CPU device	e.g. 3 = 0x03 0x00
CpuId	2	ID of CON device	e. g. 1012 = 0xF4 0x03

Example

Get CON devices connected to CPU devices.

`CpuId = 1012, 1013, 1020)`

`0x1B 0x5B 0x4E 0x0D 0x00 0x03 0x00 0xF4 0x03 0xF5 0x03 0xFC 0x03`

Response

Telegram

```
ESC ] N Size CpuCnt <CpuId, ConId>[1] …

<CpuId, ConId>[CpuCnt]
```

General Description

Return CON devices (input) connected to CPU devices (output). Returns a list of pairs of CpuId, ConId.

Table 7-16. Return CON devices (input) connected to CPU devices (output).

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
]	1	Server identification	0X5B
N	1	Command	0x4E
Size	2	Total length of telegram (7 Bytes + data)	e.g. for CpuCnt = 3 0x13 0x00
CpuCnt	2	Number of CPU devices	e.g. 3 = 0x03 0x00
CpuId	2	ID of CPU device	e. g. 1012 = 0xF4 0x03
ConId	2	ID of CON device	e.g. 3017 = 0xC9 0x0B

Example

Return CON devices connected to CPU devices.

```
CpuId[1] = 1012, ConId[1] = 3017;

CpuId[2] = 1013, ConId[2] = 3028;

CpuId[3] = 1020, ConId[3] = 3040;

0x1B 0x5D 0x4E 0x13 0x00 0x03 0x00 0xF4 0x03 0xC9 0x0B
0xF5 0x03 0xD4 0x0B 0xFC 0x03 0x0E 0x0B
```

7.2.9 Set Connection of CON Devices to CPU Devices

Request

Telegram

```
ESC [ O Size CpuCnt <CpuId, ConId>[1] …
<CpuId, ConId>[CpuCnt]
```

General Description

Set connection CON devices (input) to CPU devices (output). Data of CON device (USB, Audio) will be transmitted to CPU device.

Table 7-17. Set connection CON devices (input) to CPU devices (output).

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
O	1	Command	0x4F
Size	2	Total length of telegram (7 Bytes + data)	e.g. for CpuCnt = 3 0x13 0x00
CpuCnt	2	Number of CPU devices	e.g. 3 = 0x03 0x00
CpuId	2	ID of CPU device	e. g. 1012 = 0xF4 0x03
ConId	2	ID of CON device	e.g. 3017 = 0xC9 0x0B

Example

Set connection of CON devices to CPU devices.

```
CpuId[1] = 1012, ConId[1] = 3017;
CpuId[2] = 1013, ConId[2] = 3028;
CpuId[3] = 1020, ConId[3] = 3040;
0x1B 0x5B 0x4F 0x13 0x00 0x03 0x00 0xF4 0x03 0xC9 0x0B
0xF5 0x03 0xD4 0x0B 0xFC 0x03 0x0E 0x0B
```

Response

```
<ACK> [<ECHO>] or <NAK>
[   ] = Optional elements
```

7.2.10 Set CON Device Connection to CPU Device (Bidirectional)

Request

Telegram

```
ESC [ P Size CpuId ConId
```

General Description

Set CON device (input) connection to CPU device (output) and CPU device (input) connection to CON device (output).

Data of CON device (USB, audio, …) will be transmitted to CPU device.

Data of CPU device (video, USB, audio, …) will be transmitted to CON device.

Table 7-18. Set CON device (input) connection to CPU device (output) and CPU device (input) connection to CON device (output).

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
P	1	Command	0x50
Size	2	Total length of telegram (9 bytes)	0x09 0x00
CpuId	2	ID of CPU device	e. g. 1012 = 0xF4 0x03
ConId	2	ID of CON device	e.g. 3017 = 0xC9 0x0B

Example

Set CON device (`ConID = 3017`) connection to CPU device (`CpuId = 1012`).

```
0x1B 0x5B 0x50 0x09 0x00 0xF4 0x03 0xC9 0x0B
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

7.2.11 Set Connection of CON Devices to CPU Devices (Bidirectional)

Request

Telegram

```
ESC [ Q Size Cnt <CpuId, ConId>[1] …
<CpuId, ConId>[Cnt]
```

General Description

Set connection of CON devices (input) to CPU devices (output) and CPU devices (input) to CON devices (output).

Data of CON device (USB, audio, …) will be transmitted to CPU device. Data of CPU device (video, USB, audio, …) will be transmitted to CON device.

Table 7-19. Set connection of CON devices to CPU devices (bidirectional).

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
Q	1	Command	0x51
Size	2	Total length of telegram (7 bytes + data)	0x09 0x00
Cnt	2	Size of list	e.g. for Cnt = 3 0x13 0x00
CpuId	2	ID of CPU device	e. g. 1012 = 0xF4 0x03
ConId	2	ID of CON device	e.g. 3017 = 0xC9 0x0B

Example

Connect CONs with CPUs and CPUs with CONs.
CpuId[1] = 1012, ConId[1] = 3017;
CpuId[2] = 1013, ConId[2] = 3028;
CpuId[3] = 1020, ConId[3] = 3040;

```
0x1B 0x5B 0x51 0x13 0x00 0x03 0x00 0xF4 0x03 0xC9 0x0B
0xF5 0x03 0xD4 0x0B 0xFC 0x03 0x0E 0x0B
```

Response

```
<ACK> [<ECHO>] or <NAK>
[    ] = Optional elements
```

7.2.12 Get All Connections for CON and CPU Devices

Request

Telegram

```
ESC [ R
```

General Description

Get all CPU device–CON device connections.

Table 7-20. Get all CPU device–CON device connections.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
R	1	Command	0x52

Example

Get all CPU device–CON device connections

```
0x1B 0x5B 0x52
```

Response

Telegram

```
ESC ] R Size CpuCnt ConCnt <CpuId, ConId>[1] …
<CpuId, ConId>[ CpuCnt] <ConId, CpuId>[1] …
<ConId, CpuId>[ ConCnt]
```

General Description

Return all CPU device–CON device connections in pairs.

For each defined CPU device, the ConId of the connected CON device will be added, or 0 if the CPU device is disconnected.

For each defined CON device, the CpuId of the connected CPU device will be added, or 0 if the CON device is disconnected.

Table 7-21. Return all CPU device–CON device connections in pairs.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
]	1	Server identification	0X5D
R	1	Command	0x52
Size	2	Total length of telegram (9 bytes + data)	e.g. for CpuCnt = 3 ConCnt = 2 0x15 0x00
CpuCnt	2	Number of CPU device	e.g. 3 = 0x03 0x00
ConCnt	2	Number of CON device	e.g. 2 = 0x02 0x00
CpuId	2	ID of CPU device	e.g. 1012 = 0xF4 0x03
ConId	2	ID of CON device	e.g. 3017 = 0xC9 0x0B

Example

Return all CPU device – CON device connections in pairs.

```
CpuId[1] = 1012, ConId[1] = 3017;

CpuId[2] = 1013, ConId[2] = 3028;

CpuId[3] = 1020, ConId[3] = 3040;

ConId[1] = 3017, CpuId[1] = 1012;

ConId[2] = 3028, CpuId[2] = 0;

0x1B 0x5D 0x52 0x15 0x00 0x03 0x00 0x02 0x00 0xF4 0x03
0xC9 0x0B 0xF5 0x03 0xD4 0x0B 0xFC 0x03 0x0E 0x0B 0xC9
0x0B 0xF4 0x03 0xD4 0x0B 0x00 0x00
```

7.2.13 Set Connection for All CON Devices and CPU Devices

Request

Telegram

```
ESC [ S Size CpuCnt ConCnt <CpuId, ConId>[1] …
<CpuId, ConId>[ CpuCnt] <ConId, CpuId>[1] …
<ConId, CpuId>[ ConCnt]
```

General Description

Set a connection for all defined CON devices and CPU devices.

For each defined CPU device add the ConId, or 0 if the CPU device is disconnected.

For each defined CON device add the CpuId, or 0 if the CON device is disconnected.

Table 7-22. Set a connection for all defined CON devices and CPU devices.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
S	1	Command	0x53
Size	2	Total length of telegram (9 bytes + data)	e.g. for CpuCnt = 3 ConCnt = 2 0x15 0x00
CpuCnt	2	Number of CPUs	e.g. 3 = 0x03 0x00
ConCnt	2	Number of CONs	e.g. 2 = 0x02 0x00
CpuId	2	ID of CPU device	e.g. 1012 = 0xF4 0x03
ConId	2	ID of CON device	e.g. 3017 = 0xC9 0x0B

Example

Set a connection for all defined CON devices and CPU devices.
`CpuId[1] = 1012, ConId[1] = 3017;`
`CpuId[2] = 1013, ConId[2] = 3028;`
`CpuId[3] = 1020, ConId[3] = 3040;`
`ConId[1] = 3017, CpuId[1] = 1012;`
`ConId[2] = 3028, CpuId[2] = 0;`
`0x1B 0x5B 0x53 0x15 0x00 0x03 0x00 0x02 0x00 0xF4 0x03 0xC9 0x0B 0xF5 0x03 0xD4 0x0B 0xFC 0x03 0x0E 0x0B 0xC9 0x0B 0xF4 0x03 0xD4 0x0B 0x00 0x00`

Response
`<ACK> [<ECHO>] or <NAK>` `[ ] = Optional elements`

7.2.14 Set Extended Connection

Request

Telegram

```
ESC [ b Size CpuId ConId Mode
```

General Description

Set CON device (input) connection to CPU device (output) and CPU device (input) connection to CON device (output).

Data of CON device (USB, audio, …) is transmitted to a CPU device.

Data of CPU device (video, USB, audio, …) is transmitted to a CON device.

Table 7-23. Set a connection for all defined CON devices and CPU devices.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
b	1	Command	0x62
Size	2	Total length of telegram	0x0B 0x00
CpuId	2	ID of CPU device	e.g. 1012 = 0xF4 0x03
ConId	2	ID of CON device	e.g. 3017 = 0xC9 0x0B
Mode	2	Connection mode (0 = full access, 1 = video only, 2 = private mode)	0 = 0x00 0x00 1 = 0x01 0x00 2 = 0x02 0x00

Example

Set CON device connection to CPU device and CPU device connection to CON device.

```
CpuId = 1012, ConId = 3017, Mode = private mode

0x1B 0x5B 0x62 0x0B 0x00 0xF4 0x03 0xC9 0x0B 0x02 0x00
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

7.2.15 Get CPU List

Request

Telegram

```
ESC [ g Size First
```

General Description

Get list of all CPU devices (output) including ID and name.
First: Index of CPU device from which the list scan will start.

Table 7-24. Get list of all CPU devices (output) .

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
g	1	Command	0x67
Size	2	Total length of telegram (7 bytes)	0x07 0x00
First	2	Index of first CPU	e.g. 1000 = 0xE9 0x03 0 (all) = 0x00 0x00

Example

Get all CPUs

```
0x1B 0x5B 0x67 0x07 0x00 0x00 0x00
```

Response

Telegram

```
ESC ] g Size Count Next List [1] … List [Count]
```

General Description

Count: Number of items in the CPU list
Next: Index of the next CPU, if the list of CPU devices exceeds the telegram size. Contains 0 if there are no more CPU devices.

Table 7-25. Get list of all CPU devices (output) response.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
]	1	Server identification	0X5D
g	1	Command	0x67
Size	2	Total length of telegram	e.g. 33 = 0x21 0x00
Count	2	Number of CPUs	e.g. 1 = 0x01 0x00
Next	2	ID of first CPU in next list	e.g. 0 = 0x00 0x00 (no further CPU)
Id	4	ID of CPU device	e.g. 1000 = 0xE8 0x03 0x00 0x00
Name	20	Name of CPU	e.g. CPU_Video1 = 0x43 0x50 0x55 0x5F 0x56 0x69 0x64 0x65 0x5F 0x31

Example

Return list of CPUs

```
0x1B 0x5D 0x67 0x21 0x00 0x01 0x00 0x00 0x00 0xE8 0x03
0x00 0x00 0x43 0x50 0x55 0x5F 0x56 0x69 0x64 0x65 0x6F
0x31 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

7.2.16 Get CON List

Request

Telegram

```
ESC [ h Size First
```

General Description

Get list of all CON devices (input) including ID and name.
First: Index of CON device from which the list scan will start.

Table 7-26. Get CON List.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
h	1	Command	0x68
Size	2	Total length of telegram (7 bytes)	0x07 0x00
First	2	Index of first CPU	e.g. 3000 = 0xB8 0x0B 0 (all) = 0x00 0x00

Example

Get all CPUs.

```
0x1B 0x5B 0x68 0x07 0x00 0x00 0x00
```

Response

Telegram

```
ESC ] h Size Count Next List [1] … List [Count]
```

General Description

Count: Number of items in the CON list.
Next: Index of the next CON, if the list of CON devices exceeds the telegram size. Contains 0 if there are no more CON devices.

Table 7-27. Get CON List response.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
]	1	Server identification	0X5D
g	1	Command	0x68
Size	2	Total length of telegram	e.g. 33 = 0x21 0x00
Count	2	Number of CONs	e.g. 1 = 0x01 0x00
Next	2	ID of first CON in next list	e.g. 0 = 0x00 0x00 (no further CON)
Id	4	ID of CON device	e.g. 3000 = 0xB8 0x0B 0x00 0x00
Name	20	Name of CON	e.g. CON_Video1 = 0x43 0x4F 0x4E 0x5F 0x56 0x69 0x64 0x65 0x5F 0x31

Example

Return list of CONs

```
0x1B 0x5D 0x68 0x21 0x00 0x01 0x00 0x00 0x00 0xB8 0x0B
0x00 0x00 0x43 0x4F 0x4E 0x5F 0x56 0x69 0x64 0x65 0x6F
0x31 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

7.2.17 Get User List

Request

Telegram

```
ESC [ i Size First
```

General Description

Get list of all users.
First: Index of the user from whom the list scan will start.

Table 7-28. Get User List.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
i	1	Command	0x69
Size	2	Total length of telegram (7 bytes)	0x07 0x00
First	2	Index of first user	e.g. 1 = 0x01 0x00 0 (all) = 0x00 0x00

Example

Get all users.

```
0x1B 0x5B 0x69 0x07 0x00 0x00 0x00
```

Response

Telegram

```
ESC ] i Size Count Next List [1] … List [Count]
```

General Description

Count: Number of items in the user list.
Next: Index of the next user, if the list of users exceeds the telegram size.
Contains 0 if there are no more users.

Table 7-29. Get User List response.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
]	1	Server identification	0X5D
i	1	Command	0x69
Size	2	Total length of telegram	e.g. 33 = 0x21 0x00
Count	2	Number of users	e.g. 1 = 0x01 0x00
Next	2	ID of first user in next list	e.g. 0 = 0x00 0x00 (no further user)
Id	4	ID of user	e.g. 1 - 0x01 0x00 0x00 0x00
Name	20	Name of user	e.g. admin = 0x61 0x64 0x6D 0x69 0x6E

Example

Return list of users

```
0x1B 0x5D 0x69 0x21 0x00 0x01 0x00 0x00 0x00 0x01 0x00
0x00 0x00 0x61 0x64 0x6D 0x69 0x6E 0x00 0x00 0x00 0x00
0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00
```

7.3 Assignments

7.3.1 Get Virtual CON Device

Request

Telegram

```
ESC [ T Size RConId
```

General Description

Get virtual CON device of a real CON device.

Table 7-24. Get virtual CON device of a real CON device.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
T	1	Command	0x54
Size	2	Total length of telegram (7 bytes)	0x07 0x00
RConId	2	ID of real CON device	e.g. 3017 = 0xC9 0x0B

Example

Get virtual CON device of a real CON device (`RConId` = 3017).

```
0x1B 0x5B 0x54 0x07 0x00 0xC9 0x0B
```

Response

Telegram

```
ESC ] T Size RConId VConId
```

General Description

Return virtual CON device of a real CON device.

Table 7-25. Return virtual CON device of a real CON device.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
]	1	Server identification	0X5D
T	1	Command	0x54
Size	2	Total length of telegram (9 bytes)	0x09 0x00
RConId	2	ID of real CON device	e.g. 3017 = 0xC9 0x0B
VConId	2	ID of virtual CON device	e.g. 4034 = 0xC2 0x0F

Example

Return virtual CON device (`VConId = 4034`) of a real CON device (`RConId = 3017`).

0x1B 0x5B 0x54 0x09 0x00 0xC9 0x0B 0xC2 0x0F

or <NAK>

7.3.2 Set Virtual CON Device to a Real CON Device

Request

Telegram

```
ESC [ U Size RConId VConId
```

General Description

Set virtual CON device to a real CON device.

Table 7-26. Set virtual CON device to a real CON device.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
U	1	Command	0x55
Size	2	Total length of telegram (9 bytes)	0x09 0x00
RConId	2	ID of real CON device	e.g. 3017 = 0xC9 0x0B
VConId	2	ID of virtual CON device	e.g. 4034 = 0xC2 0x0F

Example

Set virtual CON device (`VConId = 4034`) to a real CON device (`RConId = 3017`).

```
0x1B 0x5B 0x48 0x09 0x00 0xC9 0x0B 0xC2 0x0F
```

Response

```
<ACK> [<ECHO>] or <NAK>

[   ] = Optional elements
```

7.3.3 Get Real CPU Device

Request

Telegram

```
ESC [ V Size VCpuId
```

General Description

Get real CPU device of a virtual CPU device.

Table 7-27. Get real CPU device of a virtual CPU device.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
V	1	Command	0x56
Size	2	Total length of telegram (7 bytes)	0x07 0x00
VCpuId	2	ID of virtual CPU device	e.g. 2018 = 0xE2 0x07

Example

Get real CPU device of a virtual CPU device (`VCpuId = 2018`).

```
0x1B 0x5B 0x56 0x07 0x00 0xE2 0x07
```

Response

Telegram

```
ESC ] V Size VCpuId RCpuId
```

General Description

Return real CPU device of a virtual CPU device.

Table 7-28. Get real CPU device of a virtual CPU device.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
]	1	Server identification	0X5D
V	1	Command	0x56
Size	2	Total length of telegram (9 bytes)	0x09 0x00
VCpuId	2	ID of virtual CPU device	e.g. 2018 = 0xE2 0x07
RCpuId	2	ID of real CPU device	e.g. 1012 = 0xF4 0x03

Example

Return real CPU device (`RCpuId = 1012`) of a virtual CPU device (`VCpuId = 2018`).

```
0x1B 0x5D 0x56 0x09 0x00 0xE2 0x07 0xF4 0x03
```

or <NAK>

7.3.4 Set Real CPU to a Virtual CPU

Request

Telegram

```
ESC [ W Size VCpuId RCpuId
```

General Description

Set real CPU device to a virtual CPU device.

Table 7-29. Set real CPU device to a virtual CPU device.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
W	1	Command	0x57
Size	2	Total length of telegram (9 bytes)	0x09 0x00
VCpuId	2	ID of virtual CPU device	e.g. 2018 = 0xE2 0x07
RCpuId	2	ID of real CPU device	e.g. 1012 = 0xF4 0x03

Example

Set real CPU device (`RCpuId = 1012`) to a virtual CPU device (`VCpuId = 2018`).

```
0x1B 0x5B 0x57 0x09 0x00 0xE2 0x07 0xF4 0x03
```

Response

```
<ACK> [<ECHO>] or <NAK>

[    ] = Optional elements
```

7.3.5 Get Virtual CON Devices

Request

Telegram

```
ESC [ X Size ConCnt RConId[1] … RConId[ConCnt]
```

General Description

Get virtual CON devices of a real CON device.

For `ConCnt = 0`, all real CON devices with assignments to virtual CON devices will be returned.

Table 7-30. Get virtual CON devices of a real CON device.

Type	Bytes	Description	Hex Coding
`ESC`	1	Control character	0x1B
`[`	1	Server identification	0X5B
`X`	1	Command	0x58
Size	2	Total length of telegram (7 bytes) + data	e.g. for `ConCnt = 3` 0x0D 0x00
ConId	2	Number of CON device	e.g. 3 = 0x03 0x00
RConId	2	ID of real CON device	e.g. 3017 = 0xC9 0x0B

Example

Return virtual CON devices of a real CON devices as pairs

`(RConId = 3017, 3028, 3040).`

```
0x1B 0x5B 0x58 0x0D 0x00 0x03 0x00 0xC9 0x0B 0xD4 0x0B
0xE0 0x0B
```

Response

Telegram

```
ESC ] X Size ConCnt <RConId, VConId>[1] …
<RConId, VConId>[ConCnt]
```

General Description

Return virtual CON devices of a real CON devices as pairs

Table 7-31. Return virtual CON devices of a real CON device as pairs.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
]	1	Server identification	0X5D
X	1	Command	0x58
Size	2	Total length of telegram (7 bytes) + data	e.g. for ConCnt = 3 0x13 0x00
ConId	2	Number of CON device	e.g. 3 = 0x03 0x00
RConId	2	ID of real CON device	e.g. 3017 = 0xC9 0x0B
VConId	2	ID of virtual CON device	e.g. 4034 = 0xC2 0x0F

Example

Returns virtual CON of a real CON as pairs.

```
RConID[1] = 3017, VConID[1] = 4034;

RConID[2] = 3028, VConID[2] = 4042;

RConID[3] = 3040, VConID[3] = 4045;

0x1B 0x5D 0x58 0x13 0x00 0xC9 0x0B 0xC2 0x0F 0xD4 0x0B
0xCA 0x0F 0xE0 0x0B 0xCD 0x0F
```

7.3.6 Set Virtual CON Devices to Real CON Devices

Request

Telegram

```
ESC [ Y Size ConCnt <RConId, VConId>[1] …
<RConId, VConId>[ConCnt]
```

General Description

Set virtual CON devices to real CON devices.

Table 7-32. Set virtual CON devices to real CON devices.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
Y	1	Command	0x59
Size	2	Total length of telegram (7 bytes + data)	e.g. for ConCnt = 3 0x13 0x00
ConCnt	2	Number of CON device	e.g. 3 = 0x03 0x00
RConId	2	ID of real CON device	e.g. 3017 = 0xC9 0x0B
VConId	2	ID of virtual CON device	e.g. 4034 = 0xC2 0x0F

Example

Set virtual CON devices to real CON devices.

```
RConId[1] = 3017, VConId[1] = 4034;
RConId[2] = 3028, VConId[2] = 4042;
RConId[3] = 3040, VConId[3] = 4045;
0x1B 0x5B 0x59 0x13 0x00 0xC9 0x0B 0xC2 0x0F 0xD4 0x0B
0xCA 0x0F 0xE0 0x0B 0xCD 0x0F
```

Response

```
<ACK> [<ECHO>] or <NAK>
[   ] = Optional elements
```

7.3.7 Get Real CPU Device

Request

Telegram

```
ESC [ Z Size CpuCnt VCpuId[1] … VCpuId[CpuCnt]
```

General Description

Get real CPU devices of virtual CPU devices.

For `CpuCnt` = 0, all virtual CPU devices with assignments to virtual CPU devices will be returned.

Table 7-33. Get real CPU devices of virtual CPU devices.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
Z	1	Command	0x5A
Size	2	Total length of telegram (7 bytes + data)	e.g. for `ConCnt = 3` 0x0D 0x00
CpuCnt	2	Number of CPU devices	e.g. 3 = 0x03 0x00
VCpuId	2	ID of virtual CPU devices	e.g. 2018 = 0xE2 0x07

Example

Get real CPU devices of virtual CPU devices

(VCpuId = 2018, 2030, 2035).

```
0x1B 0x5B 0x5A 0x0D 0x00 0x03 0x00 0xE2 0x07 0xEE 0x07
0xF3 0x07
```

Response

Telegram

```
ESC ] Z Size CpuCnt <VCpuId, RCpuId>[1] …
<VCpuId, RCpuId>[CpuCnt]
```

General Description

Return real CPU devices of virtual CPU devices as pairs.

Table 7-34. Return real CPU devices of virtual CPU devices as pairs.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
]	1	Server identification	0X5D
z	1	Command	0x5A
Size	2	Total length of telegram (7 bytes + data)	e.g. for CpuCnt = 3 0x13 0x00
CpuCnt	2	Number of CPU devices	e.g. 3 = 0x03 0x00
VCpuId	2	ID of virtual CPU device	e.g. 2018 = 0xE2 0x07
RCpuId	2	ID of real CPU device	e.g. 1012 = 0xF4 0x03

Example

Return real CPU devices of virtual CPU devices as pairs.

```
VCpuId[1] = 2018, RCpuId[1] = 1012;

VCpuId[2] = 2030, RCpuId[2] = 1013;

VCpuId[3] = 2035, RCpuId[3] = 1020;

0x1B 0x5D 0x5A 0x13 0x00 0x03 0x00 0xE2 0x07 0xF4 0x03
0xEE 0x07 0xF5 0x03 0xF3 0x07 0xFC 0x03
```

7.3.8 Set Real CPU Devices

Request

Telegram

```
ESC [ a Size CpuCnt <VCpuId, RCpuId>[1] …
<VCpuId, RCpuId>[CpuCnt]
```

General Description

Set real CPU devices to virtual CPU devices.

Table 7-35. Set real CPU devices to virtual CPU devices.

Type	Bytes	Description	Hex Coding
ESC	1	Control character	0x1B
[	1	Server identification	0X5B
a	1	Command	0x61
Size	2	Total length of telegram (7 bytes + data)	e.g. for CpuCnt = 3 0x13 0x00
CpuCnt	2	Number of CPU device	e.g. 3 = 0x03 0x00
VCpuId	2	ID of virtual CPU device	e.g. 2018 = 0xE2 0x07
RCpuId	2	ID of real CPU device	e.g. 1025 = 0x16 0x04

Example

Set real CPU devices to virtual CPU devices.

```
VCpuId[1] = 2018, RCpuId[1] = 1012;
VCpuId[2] = 2030, RCpuId[2] = 1013;
VCpuId[3] = 2035, RCpuId[3] = 1020;
```

```
0x1B 0x5B 0x61 0x13 0x00 0x03 0x00 0xE2 0x07 0xF4 0x03
```

Response

```
<ACK> [<ECHO>] or <NAK>
[    ] = Optional elements
```

7.4 Connector Pinouts

Table 7-36. DB9 connector.

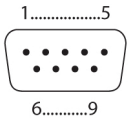
Picture	Pin	Signal	Color	Signal
	1	Not connected	6	DTR
	2	CTS	7	TxD
	3	RTS	8	RxD
	4	DSR	9	Not connected
	5	GND	—	—

Table 7-37. RJ-45 connector.

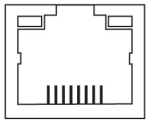
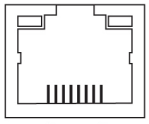
Picture	Pin	Signal	Color	Signal
	1	D1+	5	Not connected
	2	D1-	6	D2-
	3	D2+	7	Not connected
	4	Not connected	8	Not connected

Table 7-38. RJ-45 connector (serial).

Picture	Pin	Signal	Color	Signal
	1	DC D	5	RxD
	2	DSR	6	TxD
	3	RTS	7	CTS
	4	GND	8	DTR

8. Troubleshooting

In Chapters 8–9, support for problems with the DKM FX API is provided. If you have problems with the involved devices, especially the DKM matrix, refer to the respective device manuals.

8.1 Network Error

Diagnosis: Network settings are not accepted after editing.

Possible Reason: Restart of the matrix not yet completed.

Measure: Do a restart.

8.2 Failure at the Matrix

Diagnosis: Serial control impossible or only possible with restrictions.

Possible Reason: Different Baud rate of CPU and matrix.

Measure: Adapt Baud rate in the CPU.

Diagnosis: Serial control via RJ-45 port not possible.

Possible Reason: Wrong network cable.

Measure: Use a crossed network cable.

9. Technical Support

Before contacting technical support, please ensure you have read this manual, and then installed and set up your DKM FX as recommended. Contact Black Box Technical Support at 724-746-5500 or info@blackbox.com.

Support Checklist

To efficiently handle your request it is necessary to provide us with complete information for support and problem cases. Keep the following information available before you call:

- Company, name, phone number and e-mail.
- Type and serial number of the device (see bottom of device).
- Date and number of sales receipt, name of dealer if necessary.
- Issue date of the existing manual.
- Nature, circumstances, and duration of the problem.
- Involved components (such as graphics source/CPU, OS, graphics card, monitor, USB-HID/USB 2.0 devices, interconnect cable) including manufacturer and model number.
- Results from any testing you have done.

10. Glossary

The following terms are commonly used in this guide or in video and KVM technology:

AES/EBU: Digital audio standard that is officially known as AES3 and that is used for carrying digital audio signals between devices.

CATx: Any CAT5e (CAT6, CAT7) cable

CGA: The Color Graphics Adapter (CGA) is an old analog graphic standard with up to 16 displayable colors and a maximum resolution of 640x400 pixels.

Component Video: The Component Video (YPbPr) is a high-quality video standard that consists of three independently and separately transmittable video signals, the luminance signal and two color difference signals.

Composite Video: The Composite Video is also called CVBS and it is part of the PAL TV standard.

CON Unit: Component of a DKM FX or Media Extender to connect to the console (monitor(s), keyboard and mouse; optionally also with USB 2.0 devices).

Console: Keyboard, mouse, and monitor.

CPU Unit: Component of a DKM FX or Media Extender to connect to a source (computer, CPU).

CVBS: The analog color video baseband signal (CVBS) is also called Composite Video and it is part of the PAL TV standard.

DDC: The Display Data Channel (DDC) is a serial communication interface between monitor and source (computer, CPU). It allows a data exchange via monitor cable and an automatic installation and configuration of a monitor driver by the operating system.

Dual Access: A system to operate a source (computer, CPU) from two consoles.

Dual-Head: A system with two video connections.

Dual Link: A DVI-D interface for resolutions up to 2560x2048 by signal transmission of up to 330 Mpixel/s (24-bit)

DVI: Digital video standard, introduced by the Digital Display Working Group (http://www.ddwg.org). Single Link and Dual Link standard are distinguished. The signals have TMDS level.

DVI-I: A combined signal (digital and analog) that allows running a VGA monitor at a DVI-I port – in contrast to DVI-D (see DVI).

EGA: The Enhanced Graphics Adapter (EGA) is an old analog graphic standard, introduced by IBM in 1984. A D-Sub 9 connector is used for connection.

Fiber: Single-mode or multimode fiber cables.

KVM: Keyboard, video, and mouse.

Mini-XLR: Industrial standard for electrical plug connections (3-pole) for the transmission of digital audio and control signals.

Multimode: 62.5µ multimode fiber cable or 50µ multimode fiber cable

OSD: The on-screen display is used to display information or to operate a device.

Quad-Head: A system with four video connections.

RCA (Cinch): A non-standardized plug connection for transmission of electrical audio and video signals, especially with coaxial cables

SFP: SFPs (Small Form Factor Pluggable) are pluggable interface modules for Gigabit connections. SFP modules are available for CATx and fiber interconnect cables.

S/PDIF: A digital audio interconnect that is used in consumer audio equipment over relatively short distances.

Single-Head: A system with one video connection.

Single Link: A DVI-D interface for resolutions up to 1920x1200 by signal transmission of up to 165 Mpixel/s (24-bit). Alternative frequencies are Full HD (1080p), 2K HD (2048x1080) and 2048x1152.

Single-mode: 9µ single-mode fiber cable.

S-Video (Y/C): The S-Video (Y/C) is a video format transmitting luminance and chrominance signals separately. Thereby it has a higher quality standard than CVBS.

TOSLINK®: Standardized fiber connection system for digital transmission of audio signals (F05 plug connection).

Triple-Head: A system with three video connections.

USB-HID: USB-HID (USB Human Interface Devices) allow for data input.

There is no need for a special driver during installation; "New USB-HID device found" is reported.

Typical HID devices include keyboards, mice, graphics tablets, and touchscreens. Storage, video and audio devices are not HID.

VGA: Video Graphics Array (VGA) is a computer graphics standard with a typical resolution of 640x480 pixels and up to 262,144 colors. It can be seen as a follower of the graphics standards MDA, CGA and EGA.

API-Specific Glossary

ACK: Since packet transfer is not reliable, a technique known as positive acknowledgment with retransmission is used to guarantee reliability of packet transfers.

API: An application programming interface (API) is a specification intended to be used as an interface by software components to communicate with each other. An API may include specifications for routines, data structures, object classes, and variables.

Echo: The response of the DKM FX matrix to an external command (optional).

NACK: A transmission control character sent by a station as a negative response to the station with which the connection has been set up.

Serial: In telecommunication and computer science, serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus.

TCP/IP: The Internet protocol suite is the set of communication protocols used for the Internet and similar networks and generally the most popular protocol stack for wide area networks.

Black Box Tech Support: FREE! Live. 24/7.

Great tech support is just 60 seconds away at
724-746-5500 or blackbox.com.

About Black Box

Black Box provides an extensive range of networking and infrastructure products. You'll find everything from cabinets and racks and power and surge protection products to media converters and Ethernet switches all supported by free, live 24/7 Tech support available in 60 seconds or less.