SIEMENS

COROS

Communications Processor
CP 526

Manual

Contents

Warning   C79000-R8576-C217

Information

Suggestions/Corrections

Manual   C79000-B8576-C217-01

Communications Processor CP 526

Contents

Warning

Risks involved in the use of so-called SIMATIC-compatible modules of non-Siemens manufacture

"The manufacturer of a product (SIMATIC in this case) is under the general obligation to give warning of possible risks attached to his product. This obligation has been extended in recent court rulings to include parts supplied by other vendors. Accordingly, the manufacturer is obliged to observe and recognize such hazards as may arise when a product is combined with products of other manufacture.

For this reason, we feel obliged to warn our customers who use SIMATIC products not to install so-called SIMATIC-compatible modules of other manufacture in the form of replacement or add-on modules in SIMATIC systems.

Our products undergo a strict quality assurance procedure. We have no knowledge as to whether outside manufacturers of so-called SIMATIC-compatible modules have any quality assurance at all or one that is nearly equivalent to ours. These so-called SIMATIC- compatible modules are not marketed in agreement with Siemens; we have never recommended the use of so-called SIMATIC-compatible modules of other manufacture. The advertising of these other manufacturers for so-called SIMATIC-compatible modules wrongly creates the impression that the subject advertised in periodicals, catalogues or at exhibitions had been agreed with us. Where so-called SIMATIC-compatible modules of non-Siemens manufacture are combined with our SIMATIC automation systems, we have a case of our product being used contrary to recommendations. Because of the variety of applications of our SIMATIC automation systems and the large number of these products marketed worldwide, we cannot give a concrete description specifically analyzing the hazards created by these so-called SIMATIC-compatible modules.

It is beyond the manufacturer's capabilities to have all these so-called SIMATIC-compatible modules checked for their effect on our SIMATIC products. If the use of so-called SIMATIC-compatible modules leads to defects in a SIMATIC automation system, no warranty for such systems will be given by Siemens.

In the event of product liability damages due to the use of so-called SIMATIC-compatible modules, Siemens are not liable since we took timely action in warning users of the potential hazards involved in so-called SIMATIC-compatible modules."

Safety-Related Guidelines for the User

1 General

This manual provides the information required for the intended use of the particular product. The documentation is written for technically qualified personnel such as engineers, programmers or maintenance specialists who have been specially trained and who have the specialized knowledge required in the field of instrumentation and control.

A knowledge of the safety instructions and warnings contained in this manual and their appropriate application are prerequisites for safe installation and commissioning as well as safety in operation and maintenance of the product described. Only qualified personnel as defined in section 2 have the specialized knowledge that is necessary to correctly interpret the general guidelines relating to the safety instructions and warnings and implement them in each particular case.

This manual is an inherent part of the scope of supply even if, for logistic reasons, it has to be ordered separately. For the sake of clarity, not all details of all versions of the product are described in the documentation, nor can it cover all conceivable cases regarding installation. operation and maintenance. Should you require further information or face special problems that have not been dealt with in sufficient detail in this documentation, please contact your local Siemens office.

We would also point out that the contents of this product documentation shall not become a part of or modify any prior or existing agreement, commitment or legal relationship. The Purchase Agreement contains the complete and exclusive obligations of Siemens. Any statements contained in this documentation do not create new warranties or restrict the existing warranty.

2 Qualified Personnel

Persons who are **not qualified** should not be allowed to handle the equipment/system. Non-compliance with the warnings contained in this manual or appearing on the equipment itself can result in severe personal injury or damage to property. Only **qualified personnel** should be allowed to work on this equipment/system.

Qualified persons as referred to in the safety guidelines in this manual as well as on the product itself are defined as follows:

● System planning and design engineers who are familiar with the safety concepts of automation equipment:

● Operating personnel who have been trained to work with automation equipment and are conversant with the contents of the manual in as far as it is connected with the actual operation of the plant;

● Commissioning and service personnel who are trained to repair such automation equipment and who are authorized to energize, deenergize, clear, ground and tag circuits, equipment and systems in accordance with established safety practices.

3 Danger Notices

The notices and guidelines that follow are intended to ensure personal safety, as well as protecting the product and connected equipment against damage.

The safety notices and warnings for protection against loss of life (the users or service personnel) or for protection against damage to property are highlighted in this manual by the terms and pictograms defined here. The terms used in this manual and marked on the equipment itself have the following significance:

Danger

indicates that death. severe
personal injury or substantial
property damage will result if
proper precautions are not taken.

Warning

indicates that death. severe
personal injury or substantial
property damage can result if
proper precautions are not taken.

Caution

indicates that minor personal injury
or property damage can result if
proper precautions are not taken.

Note

is an important information about
the product, its operation or a part
of the manual to which special
attention is drawn.

> **Important**
>
> If in this manual "Important" should
> appear in bold type, drawing attention
> to any particularly information, the
> definition corresponds to that of
> "Warning", "Caution" or "Note".

4 Proper Usage

- The equipment/system or the system components may only be used for the applications described in the catalog or the technical description, and only in combination with the equipment, components and devices of other manufacturers as far as this is recommended or permitted by Siemens.

- The product described has been developed, manufactured, tested and the documentation compiled in keeping with the relevant safety standards. Consequently, if the described handling instructions and safety guidelines described for planning, installation, proper operation and maintenance are adhered to, the product, under normal conditions, will not be a source of danger to property or life.

> ⚠ **Warning**
>
> - After opening the housing or the protective cover or after opening the system cabinet, certain parts of this equipment system will be accessible, which could have a dangerously high voltage level.
> - Only suitably qualified personnel should be allowed access to this equipment system.
> - These persons must be fully conversant with any potential sources of danger and maintenance measures as set out in this manual.
> - It is assumed that this product be transported, stored and installed as intended, and maintained and operated with care to ensure that the product functions correctly and safely.

5 Guidelines for Planning and Installing the Product

The product generally forms a part of larger systems or plants. These guidelines are intended to help integrate the product into its environment without it constituting a source of danger.

The following facts require particular attention:

⚠ Note

Even when a high degree of safety has been designed into an item of automation equipment by means of multichannel configuration, it is still imperative that the instructions contained in this manual be exactly adhered to. Incorrect handling can render ineffective the preventive measures incorporated into the system to protect it against dangerous faults, and even create new sources of danger.

The following advice regarding installation and commissioning of the product should - in specific cases - also be noted.

⚠ Warning

- Follow strictly the safety and accident prevention rules that apply in each particular case.

- Units which are designed as built-in units may only be operated as such, and table-mounted or portable equipment only with its casing closed.

- In the case of equipment with a permanent power connection which is not provided with an isolating switch and/or fuses which disconnect all poles, a suitable isolating switch or fuses must be provided in the building wiring system (distribution board). Furthermore, the equipment must be connected to a protective ground (PE) conductor.

- For equipment or systems with a fixed connecting cable but no isolating switch which disconnects all poles, the power socket with the grounding pin must be installed close to the unit and must be easily accessible.

- Before switching on the equipment, make sure that the voltage range setting on the equipment corresponds to the local power system voltage.

- In the case of equipment operating on 24 V DC, make sure that proper electrical isolation is provided between the mains supply and the 24 V supply. Only use power supply units to IEC 364-4-41 or HD 384.04.41 (VDE 0100 Part 410).

- Fluctuations or deviations of the power supply voltage from the rated value should not exceed the tolerances specified in the technical specifications. Otherwise, functional failures or dangerous conditions can occur in the electronic modules/equipment.

- Suitable measures must be taken to make sure that programs that are interrupted by a voltage dip or power supply failure resume proper operation when the power supply is restored. Care must be taken to ensure that dangerous operating conditions do not occur even momentarily. If necessary, the equipment must be forced into the "emergency off" state.

- Emergency tripping devices in accordance with EN 60204/IEC 204 (VDE 0113) must be effective in all operating modes of the automation equipment. Resetting the emergency off device must not result in any uncontrolled or undefined restart of the equipment.

4

⚠ **Caution**

- Install the power supply and signal cables in such a manner as to prevent inductive and capacitive interference voltages from affecting the automation functions.

- Automation equipment and its operating elements must be installed in such a manner as to prevent unintentional operation.

- Automation equipment can assume an undefined state in the case of a wire break in the signal lines. To prevent this, suitable hardware and software measures must be taken when interfacing the inputs and outputs of the automation equipment.

---

6 Active and Passive Faults in Automation Equipment

- Depending on the particular task for which the electronic automation equipment is used, both **active** as well as **passive** faults can result in a **dangerous** situation. For example, in drive control, an active fault is generally dangerous because it can result in an unauthorized startup of the drive. On the other hand, a passive fault in a signalling function can result in a dangerous operating state not being reported to the operator.

- This differentiation of the possible faults and their classification into dangerous and non-dangerous faults, depending on the particular task, is important for all safety considerations in respect of the product supplied.

⚠ **Warning**

In all cases where a fault in an automation equipment can result in severe personal injury or substantial damage to property, ie. where a dangerous fault can occur, additional external measures must be taken or equipment provided to ensure or force safe operating conditions even in the event of a fault (e.g. by means of independent limit monitors, mechanical interlocks etc.).

---

7 Procedures for Maintenance and Repair

If measurement or testing work is to be carried out on an active unit, the rules and regulations contained in the "VBG 4.0 Accident prevention regulations" of the German employers liability assurance association (Berufs-genossenschaften) must be observed. Particular attention is drawn to paragraph 8 "Permissible exceptions when working on live parts". Use only suitable electrical tools.

⚠ Warning

- Repairs to an item of automation equipment may only be carried out by **Siemens service personnel or an authorized Siemens repair center**. For replacement purposes, use only parts or components that are contained in the spare parts list or listed in the "Spare parts" section of this manual.
  Unauthorized opening of equipment and improper repairs can result in loss of life or severe personal injury as well as substantial property damage

- Before opening the equipment, <u>always</u> remove the power plug or open the disconnecting switch.

- Only use the fuse types specified in the technical specifications or the maintenance instructions of this manual.

- Do not throw batteries into an open fire and do not carry out any soldering work on batteries (danger of explosion). Maximum ambient temperature 100°C. Lithium batteries or batteries containing mercury should not be opened or recharged. Make sure that the same type is used when replacing batteries.

- Batteries and accumulators must be disposed of as classified waste.

- The following points require attention when using monitors:
  Improper handling, especially the readjustment of the high voltage or fitting of another tube type can result in excessive X-ray radiation from the unit. The license to operate such a modified unit automatically lapses and the unit must not be operated at all.

Guidelines for Handling
Electrostatically Sensitive Devices (ESD)

1 What is ESD?

VSLI chips (MOS technology) are used in practically all SIMATIC S5 and TELEPERM M modules. These VLSI components are, by their nature, very sensitive to overvoltages and thus to electrostatic discharge:

They are therefore defined as
"Electrostatically Sensitive Devices"

"ESD" is the abbreviation used internationally.

The following warning label on the cabinets, subracks and packing indicates that electrostatically sensitive components have been used and that the modules concerned are susceptible to touch:

ESDs can be destroyed by voltage and energy levels which are far below the level perceptible to human beings. Such voltages already occur when a component or a module is touched by a person who has not been electrostatically discharged. Components which have been subjected to such overvoltages cannot. in most cases, be immediately detected as faulty; the fault occurs only after a long period in operation.

An electrostatic discharge
- of 3500 V can be felt
- of 4500 V can be heard
- must take place at a minimum of 5000 V to be seen.

But just a fraction of this voltage can already damage or destroy an electronic component.

The typical data of a component can suffer due to damage, overstressing or weakening caused by electrostatic discharge; this can result in temporary fault behavior, e.g. in the case of

- temperature variations,
- mechanical shocks,
- vibrations,
- change of load.

Only the consequent use of protective equipment and careful observance of the precautions for handling such components can effectively prevent functional disturbances and failures of ESD modules.

2   When is a Static Charge Formed?

One can never be sure that the human body or the material and tools which one is using are not electrostatically charged.

Small charges of 100 V are very common; these can, however, very quickly rise up to 35 000 V.

Examples of static charge:

- Walking on a carpet	up to 35 000 V
- Walking on a PVC flooring	up to 12 000 V
- Sitting on a cushioned chair	up to 18 000 V
- Plastic desoldering unit	up to 8 000 V
- Plastic coffee cup	up to 5 000 V
- Plastic bags	up to 5 000 V
- Books, etc. with a plastic binding	up to 8 000 V

3   Important Protective Measures against Static Charge

● Most plastic materials are highly susceptible to static charge and must therefore be kept as far away as possible from ESDs.

● Personnel who handle ESDs, the work table and the packing must all be carefully grounded.
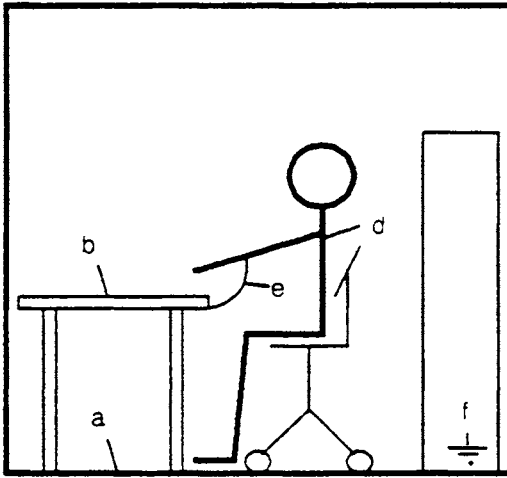
4 Handling of ESD Modules

- One basic rule to be observed is that electronic modules should be touched by hand only if this is necessary for any work to be done on them. Do not touch the component pins or the conductors.

- Touch components only if

    - the person is grounded at all times by means of a wrist strap

    or

    - the person is wearing special anti-static shoes or shoes with a grounding strip.

- Before touching an electronic module, the person concerned must ensure that (s)he is not carrying any static charge. The simplest way is to touch a conductive, grounded item of equipment (e.g. a blank metallic cabinet part, water pipe, etc.) before touching the module.

- Modules should not be brought into contact with insulating materials or materials which take up a static charge, e.g. plastic foil, insulating table tops, synthetic clothing, etc.

- Modules should only be placed on conductive surfaces (table with anti-static table top, conductive foam material, anti-static plastic bag, anti-static transport container).

- Modules should not be placed in the vicinity of monitors, TV sets (minimum distance from screen > 10 cm).

The diagram on the next page shows the required protective measures against electrostatic discharge.

# 5 Measurements and Modification to ESD Modules

- Measurements on modules may only be carried out under the following conditions:

    - the measuring equipment is grounded (e.g. via the PE conductor of the power supply system) or

    - when electrically isolated measuring equipment is used, the probe must be discharged (e.g. by touching the metallic casing of the equipment) before beginning measurements.

- Only grounded soldering irons may be used.
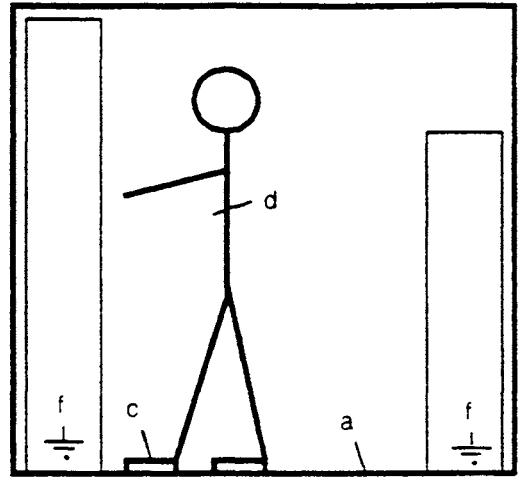
Sitting position   Standing position
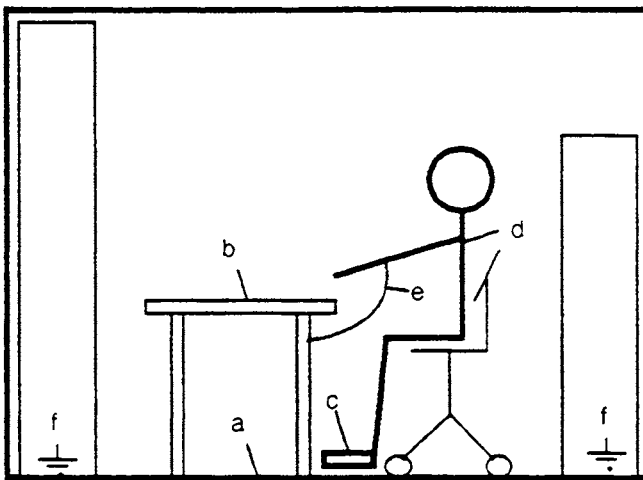
Standing/sitting position

a	Conductive flooring
b	Anti-static table
c	Anti-static shoes
d	Anti-static coat
e	Grounding wrist strap
f	Grounding connection of the cabinets

6 Shipping of ESD Modules

Anti-static packing material must always be used for modules and components, e.g. metalized plastic boxes, metal boxes, etc. for storing and dispatch of modules and components.

If the container itself is not conductive, the modules must be wrapped in a conductive material such as conductive foam, anti-static plastic bag, aluminium foil or paper. Normal plastic bags or foils should not be used under any circumstances.

For modules with built-in batteries ensure that the conductive packing does not touch or short-circuit the battery connections: if necessary cover the connections with insulating tape or material.

4

Preface

The CP 526 Manual gives an overview of the general features and operating principles of the CP 526 module. It provides a description of the hardware configuration of the CP 526 basic board, along with installation instructions for the CP 526 module. In addition, it guides you step-by-step through a practical application example.

This book is intended for engineers, programmers, and maintenance personnel who have a general knowledge of programmable controller concepts.

The following conventions are used in this volume and are listed for your reference:

Convention	Definition	Example
XXX XXXXX	Uppercase characters indicate a command, function, screen form, data handling block, or program name.	SEND-DIRECT
<XXX XXXXX>	Uppercase characters within angle brackets indicate a specific key or function key on the programmer (PG) or process operation keyboard.	<ENTER>
"XXX XXXXX"	Uppercase characters within quotation marks indicate an LED or text that you should enter exactly as shown.	"COLOR"
#####	Pound signs indicate input fields in forms.	######
$$$$$	Dollar signs indicate output fields in masks. These are inputs that were made in previous forms.	$$$$$$$

If you have any questions about CP 526 or the COM software that are not answered in this book, please contact your local Siemens representative.

Contents

6   System Restrictions   6-3

       © *Siemens AG C79000-B8576-C217-01*

Figures

Tables

NOTE

These instructions do not cover all details or variations in equipment or provide for every circumstance that can arise in connection with installation, operation, or maintenance. If you want further information or if particular problems arise that are not covered sufficiently for your purposes, contact your local Siemens sales office.

The contents of this instruction manual shall not become part of or modify any prior or existing agreement, commitment, or relationship. The sales contract contains the entire obligation of Siemens. The warranty contained in the contract between the parties is the sole warranty of Siemens. Any statements contained herein do not create new warranties or modify the existing warranty.

Contents

Communications Processor   C79000-B8576-C217
CP 526
Contents

1   Introduction

1.1   Communications Processors (CPs) for Operator Control and Process Monitoring (O + M)

Even at the level of the programmable controller, i.e. in a machine-oriented environment, automation tasks of today often require that the control and visual display of the process not only be simple but also be user-friendly and efficient. The **communications processor** (CP) is used for operator control and process monitoring (O + M) when the text display unit cannot meet the requirements and the complex operator terminal is not yet necessary.

Communication with the process is implemented with a module which is especially designed for this task and which can be plugged into the programmable controller (PLC). Graphics and texts are stored in the module's own user memory. The module only communicates with the central processor of the programmable controller when data for variable segments is to be exchanged.

The CP is simply plugged into one of the slots of the **SIMATIC S5** programmable controller provided for this purpose. You can thus "upgrade" your system in an easy way and at low cost. Using a CP, your personnel can communicate directly with the process, specify setpoints, verify actual values, evaluate error messages, print out reports and make the process "transparent" by means of plant mimics.

1.2   Configuration of the O + M System with Communications Processor (CP)

This machine-oriented, or direct, operator control and process monitoring system consists of the **communications processor** itself, a **monochrome or color monitor** and a **process keyboard** for input to the process. The CP 526 and CP 527 can be extended by an additional module, - the **CP expansion board** -, which enables process data to be represented in the form of curves and the user memory to be expanded. These functions are already fully integrated in the CP 528 standard module.

Plant displays, measuring scales, tables, curves, message texts, reports etc. are configured using a **SIMATIC S5 programmer**. The COM configuration software enables the user to develop programs simply and efficiently. He need not be familiar with a programming language.

1.3   Available O + M Communications Processors

The "Operator control and process monitoring with communications processors" system comprises modules which differ in complexity and speed to enable the user to configure a system according to his requirements:

- **CP 526 module for monochrome monitor**

    display of error messages;
    configuration of static images;
    without printer connection;
    forced ventilation required (module with heat exchanger can also be used without fan)

- **CP 526 module for color monitor**

    can be used with CP expansion board;
    display of error messages;
    configuration of static images;
    without printer connection;
    forced ventilation required (module with heat exchanger can also be used without fan)

- **CP 527 module for monochrome monitor**

    can be used with CP expansion board
    (only for memory extension);
    with printer connection;
    quick display construction and updating;
    extended scope of functions (e.g. user-configurable reports, input/output fields);
    extended quantity standard (increased number of segments, functions and messages);
    extended user memory;
    additional functions for PLC communications;

password-protected function commands;
pixel-graphics hard copy;
combinable process windows;
forced ventilation not required;
if operated with expansion board, however, forced ventilation required;
BEDI operator control and diagnostic system

● **CP 527 module for color monitor**

can be used with CP expansion board;
with printer connection;
quick display construction and updating;
extended scope of functions (e.g. user-configurable reports,
input/output fields, extended signalling system);
extended quantity standard;
(increased number of segments, functions and messages);
extended user memory;
additional functions for PLC communications;
password-protected function commands;
pixel-graphics hard copy;
combinable process mimics;
forced ventilation not required;
if operated with expansion board, however, forced ventilation required;
BEDI operator control and diagnostic system

● **CP 526/527 interface module (expansion board)**

for the representation of curves;
enlarged texts (up to factor 16);
enlarged value output fields (up to factor 16);
additional user memory;
forced ventilation required

● **CP 528 analog output module for RGB color monitor**

extension of the functions of the CP 527 basic and expansion board,
offered on a single-width module;
2 serial interfaces:
- TTY (20 mA) for programmer (9600 baud), printer and keyboard
  (2400 baud);
- TTY (20 mA) / V.24 for printer (max. of 9600 baud);
further improved display refresh and actualization rate;
extended user memory;
forced ventilation not required;
BEDI operator control and diagnostic system

● **CP 528 digital output module for ELD monitor**

extension of the functions of the CP 527 basic and expansion board,
offered on a single-width module;
2 serial interfaces:
- TTY (20 mA) for programmer (9600 baud), printer and keyboard
  (2400 baud);
- TTY (20 mA) / V.24 for printer (max. of 9600 baud);
further improved display refresh and actualization rate;
extended user memory;
forced ventilation not required;
BEDI operator control and diagnostic system

This volume gives an overview of the general features and operating principles of the CP 526 module and its use with the expansion board.

- Introduction
- Ordering Data
- Technical Description of the CP 526 Module
- Technical Description of the CP Expansion Board
- Application Example for the CP 526
- System Restrictions
- Error Messages
- Reference Literature

The CP 526 is configured using the configuration software COM GRAPH.

Contents

Communications Processor   C79000-B8576-C217
CP 526
Contents

2   Ordering Data

The order numbers for the following components as well as the length key for ordering connecting cables are given in the ST 80 Catalog.

2.1   CP 526 / Expansion Board/Ordering Data

CP 526 Module

Color	forced ventilation
	natural ventilation
Monochrome	forced ventilation
	natural ventilation

CP Expansion Board

2.2   Ordering Data for Software and Manuals

2.2.1   COM GRAPH with Manual

for PG 635-II/685/695/710-II/730/750/770
(S5-DOS; 3 1/2" and 5 1/4 " diskettes)
in German, English, French

2.2.2   Standard Function Blocks (Data Handling Blocks)

(for order numbers refer to the ST 57 Catalog)

AG 135U   S-Processor
(S5-DOS; 3 1/2" and 5 1/4" diskettes)
German, English, French Version

AG 135U   R-Processor/CPU 928
(S5-DOS; 3 1/2" and 5 1/4" diskettes)
German, English, French Version

AG 150U
(S5-DOS; 3 1/2" and 5 1/4" diskettes)
German, English, French Version

AG 155U
(S5-DOS; 3 1/2" and 5 1/4" diskettes)
German, English, French Version

2.2.3   Software for Message Functions

(for order numbers refer to the ST 57 Catalog)

Message functions for up to 4 Standard CPs
        for S5-115U
                135U (CPU 922/928)
                150U
                155U
        (S5-DOS; 3 1/2" and 5 1/4" diskettes)
        memory requirements approx. 9 kbytes in the PLC

Compact Message Functions for a Standard CP
        for S5-115U
                135U (CPU 922/928)
                150U
                155U (CPU 946/947)
        (S5-DOS; 3 1/2" and 5 1/4" diskettes)
        German, English, French Version
        memory requirements approx. up to 4.5 kbytes in the PLC

COM PMC System software
> (S5-DOS; 3 1/2" and 5 1/4" diskettes)
> German, English, French Version
>
> COM PMC is only required for code Message Functions for up to four
> standard CPs

2.3　Ordering Data for Memory Submodule

EPROM   32   kbytes
                     64   kbytes
                     128   kbytes

RAM   32   kbytes
                     64   kbytes
                     128   kbytes

Battery-backed RAM
                     128   kbytes

Backup Battery

2.4　Ordering Data for Connecting Cables

Connecting cable between

> programmer and CP 526
> MP keyboard and CP 526
> process keyboard and CP 526

Connecting cable (coaxial cable with double shielding) between

> monitor (BNC connection) and CP 526
> monitor (DIN connection) and CP 526

2.5   Ordering Data for Process Keyboard

2.5.1   Normal Keyboards

Process keyboard 40:   Alphanumeric keyboard, grade of protection IP 20, desk-top model

2.5.2   Sealed Membrane Keyboard

Process keyboard 20:   Alphanumeric keyboard, grade of protection IP 54
Key assignment

> international
> cyrillic

Process keyboard 10:   Function keyboard, grade of protection IP 54

> desk-top model
> built-in model

2.6   Ordering Data for Monitors

2.6.1   Color Monitors

Desk-top model, office type, 36 cm and 51 cm screen diagonal

Desk-top model, industrial type, 36 cm and 51 cm screen diagonal

Built-in model, industrial type, 36 cm and 51 cm screen diagonal

2.6.2   Monochrome Monitors

Desk-top model, industrial type, 31 cm screen diagonal

2.6.3   Monitor Control Panels

Color version

> MP 20, 25 cm screen diagonal
> MP 40, 36 cm screen diagonal
> MP 14B-527, 36 cm screen diagonal

Monochrome version

> MP 20, 23 cm screen diagonal
> MP 30, electroluminescence display with an active display of
>   192 x 144 mm
> MP 12A-527, 31 cm screen diagonal

2.7.4   Accessories for Monitors

Grounding bar with 4 cable clamps for 4 coaxial cables with double shielding

> for 51 cm process monitors
> for 36 cm process monitors

Cable clamp for attaching the coaxial cable with double shielding on the above grounding bar

Cable clamp for attaching the coaxial cable with double shielding to the monochrome monitor

Passive cable adaptor for monitors up to 15.6 kHz line frequency required for cable length (CP 526 - monitor) > 80 m and < 350 m (for coaxial cables with two shields)

Active cable adaptor for monitors up to 30.84 kHz line frequency required for cable length (CP 526 - monitor) > 80 m and < 200 m (for coaxial cables with two shields)

Contents

Communications Processor   C79000-B8576-C217
CP 526
Contents

3 Technical Description of the CP 526 Module

3.1 Application

The CP 526 communications processor (CP) can be installed in the S5-115U, S5-135U, S5-150U, and S5-155U programmable controllers. The CP 526 provides the video terminal interface for operator-process communication and visualization.

The CP 526 contains the complete electronics to support a video display unit. All you have to connect is a color or monochrome monitor and a process keyboard. With the keyboard, you can call up stored process displays and assign new setpoints.

The power of the CP 526 module is greatly enhanced when it is used with the expansion board. The expansion board allows you to generate the following types of graphic displays of process data:

- curves, as a function of time
- bars, with color change to denote limit violations
- display of digital measured values, in magnified form

3.2   Mechanical Construction

The CP 526 module is a printed circuit board in double-height Eurocard format, which makes it compatible with the 902 modular packaging system. The front panel has the width of a standard module slot (approximately 20 mm).

All CP 526 modules have each of the following connectors:

- Two 48-pin, range 2 backplane connectors

- One 15-contact, female connector for attaching a programmer (PG) or a process keyboard and a printer

- Color version: four coaxial sockets for connecting the monitor; Monochrome version: one coaxial socket for connecting the monitor

- One 60-pin male connector for attaching an expansion board

Figures 3-1 and 3-2 show the major components as well as the plug-in jumpers. The jumper settings made in the factory are highlighted.

Figures 3-3 and 3-4 show the rectangular receptacle on the front panel for a special EPROM (erasable programmable read-only memory) or RAM (random-access memory) submodule for application programs. Additional front panel equipment includes a switch, three LEDs (see subsection 3.8.3), and the interfaces mentioned above for connecting the monitors.
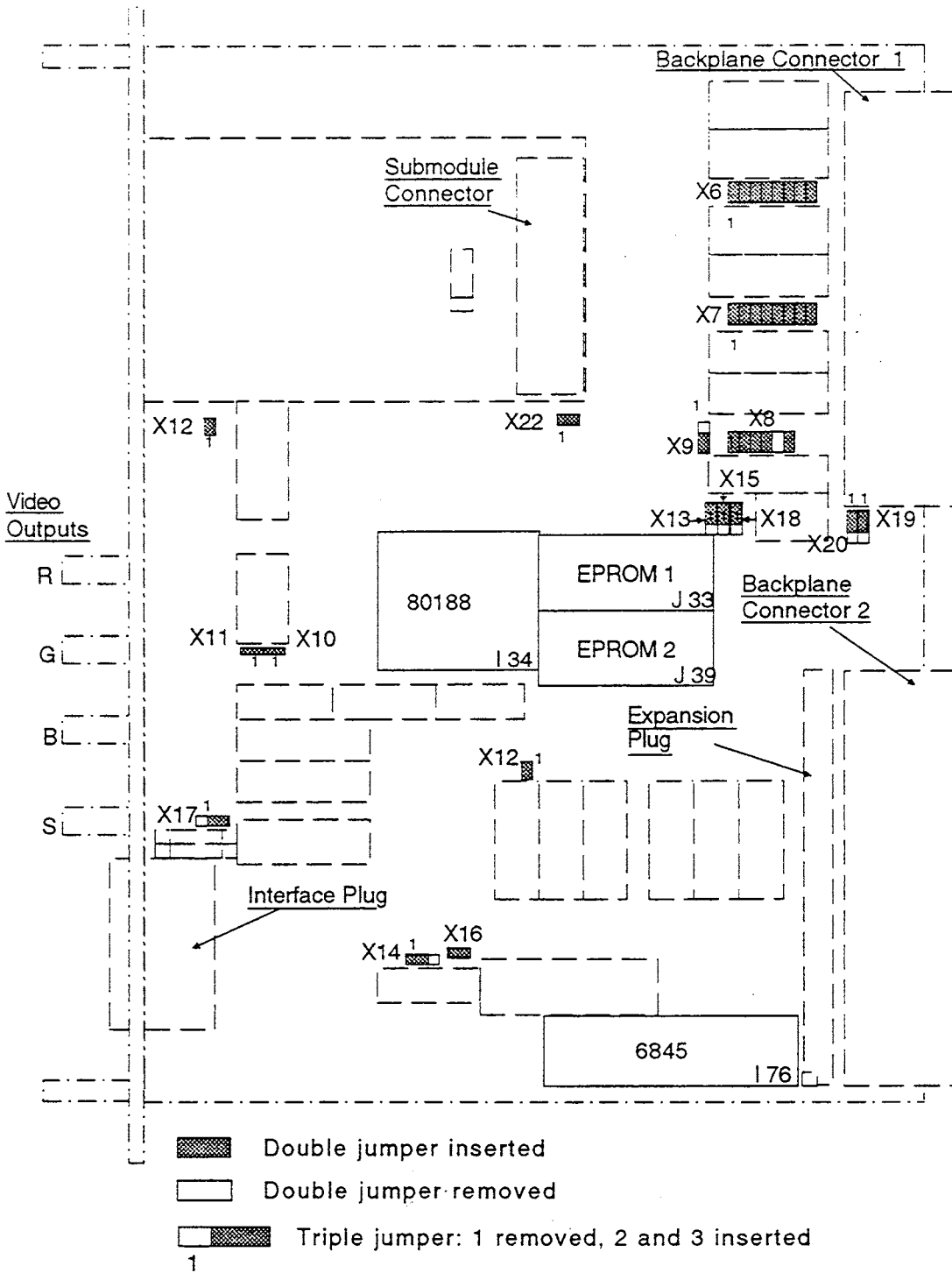
Figure 3-1: Default Jumper Settings of the CP 526 Color Module Printed Circuit
           Board
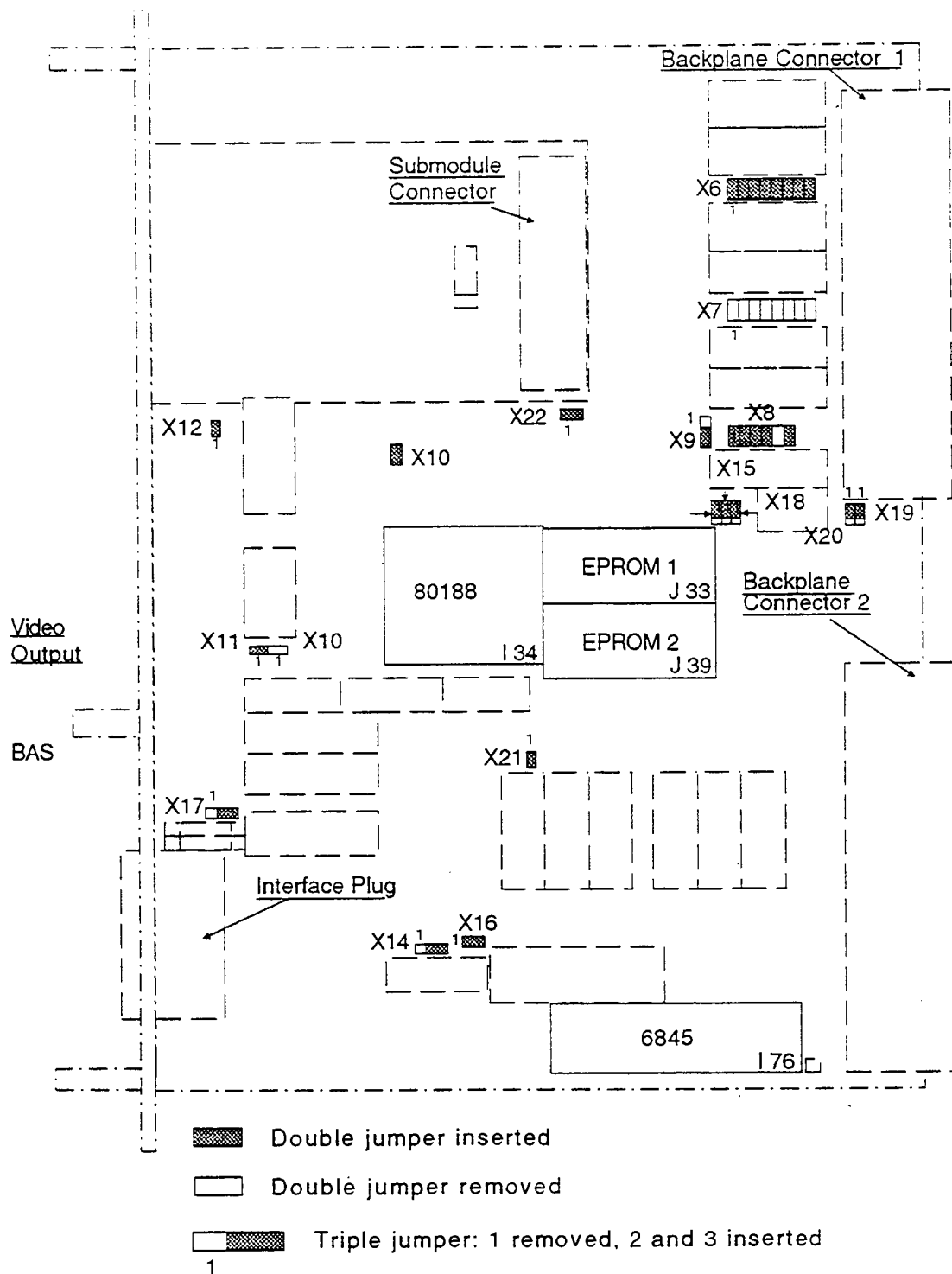
Figure 3-2: Default Jumper Settings of the CP 526 Monochrome Module Printed Circuit Board
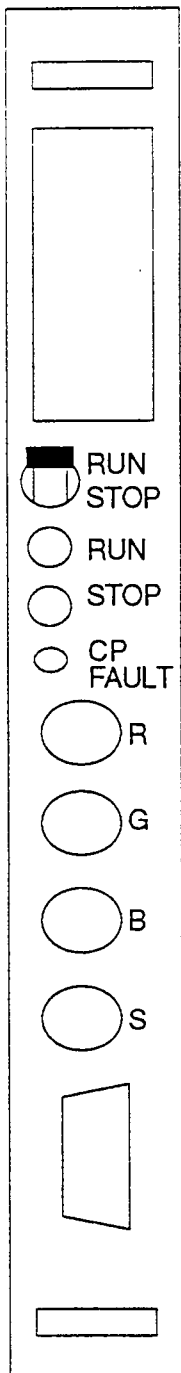
Figure 3-3:
Front Panel of the CP 526
Color Module
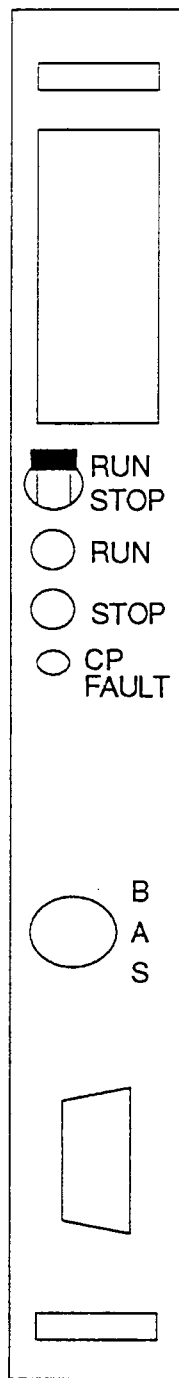
Figure 3-4:
Front Panel of the CP 526
Monochrome Module

3.3   Programmable Controller Interface

Since the user programs that contain the information for graphics displays, message texts, etc. are stored in a memory submodule in the CP 526, only variable data, such as process data, is exchanged between the CPU of the programmable controller[1] and the CP 526.

Data is exchanged through a dual-port RAM, a RAM area of the CP 526. The page of the dual-port RAM is like a mailbox. The CP 526 and the controller can exchange messages, data, requests, etc. through this "mailbox."

The controller always takes the initiative in data exchange. The CPU always "asks" the CP 526 whether or not it has data to transfer.

To avoid data exchange problems between the controller and the CP 526, the controller program calls a function block, or data handling block, which polls the CP 526 every controller cycle. The actual data exchange is executed automatically by the "data handling block."

Figures 3-5 and 3-6 show a schematic representation of the data exchange process.

---

[1]   In this manual the terms 'programmable controller' and 'CPU' (central processor of the programmable controller) are often used synonymously.
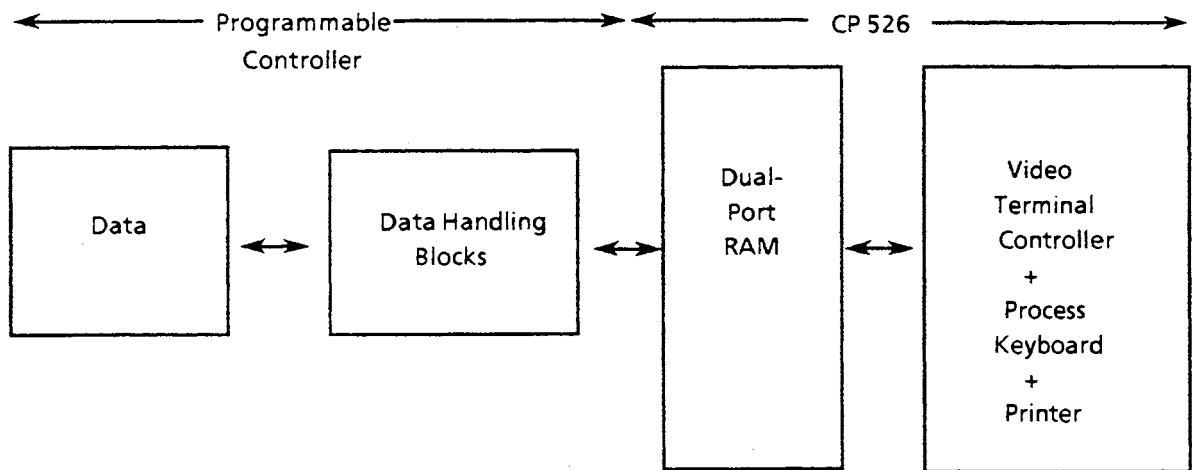
Figure 3-5: Schematic Representation of the Data Exchange Process

The dual-port RAM is divided into three parts:
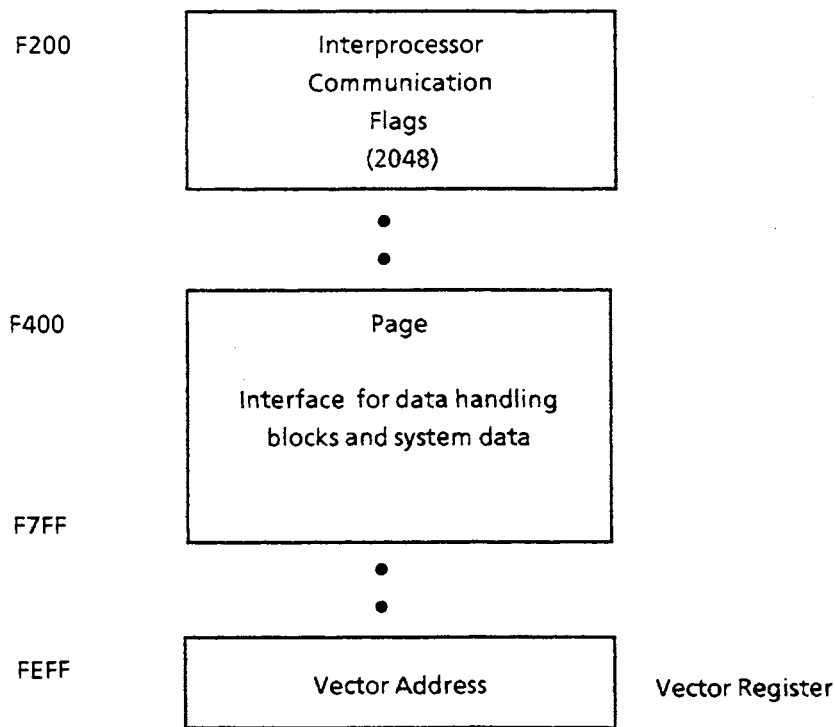
Figure 3-6: The Different Parts of the Dual Port RAM

3.3.1   Interprocessor Communication Flag Area

The interprocessor communication flag area consists of 256 interprocessor communication flag bytes, or 2048 interprocessor communication flags. The controller can cause displays to be output to the monitor by means of these flags.

The interprocessor communication flags are exchanged between the controller and the CP 526 once in each processing cycle of the STEP 5 program.

If several CPs and/or a coordination processor are used in a programmable controller, the interprocessor communication flags must be divided up among the varions CPs (by setting jumpers). This is also true if several central processing units (CPUs) are used in the S5-135U or S5-155U. Note that the total number of all enabled interprocessor communication flags must not exceed 2048 (= 256 bytes).

3.3.2   Data Transfer Area

The data transfer area, or page, is an area within the dual-port RAM in which data is exchanged between the CP 526 and the controller. The data, for example, might be in the form of numerical values to be incorporated in a mimic of a process displayed on the video screen. Unlike the interprocessor communication flag area, the page area can be "multiplied".

Every communications processor installed in the controller, including each CP 526, has its own data transfer area, or page. Each page has its own page number, which is called the vector address.

For the controller, all these pages lie in the same address area in the CP's memory. The vector address is the additional information, which, together with the memory address of the page, enables information to be sent to a particular page (i.e., to a particular CP).

The vector registers all have the same memory address on all the CPs. By writing data (eight bits) into the vector address (using the standard data handling block), all CPs can read the new vector address. Only the CP that has been coded (via its jumper settings) to that vector address will be enabled to receive the page of data. All the other CPs will be inhibited from receiving data.

The controller may now exchange data with the page of the enabled CP. All other CPs and interfaces are completely unaffected by this. Only by writing a new value into the vector address can another CP be selected for data exchange.

3.3.3   Data Handling Blocks

For the controller and the CP 526 to communicate with each other, the SYNCHRON data handling block must be correctly executed.

To execute SYNCHRON, you must specify the parameter SSNR by setting the appropriate jumpers in jumper header X7. Processing of the SYNCHRON data handling block is indicated by the "START" display on the screen.

The organization blocks (OBs) for COLD RESTART (OB20) and WARM RESTART (OB21, OB22) must be programmed in the controller program, and these OBs must call the SYNCHRON data handling block. In the case of the WARM RESTART OBs, you can also program "PLC STOP".

The SEND-ALL data handling block transmits data from the controller to the CP 526. The RECEIVE-ALL data handling block transmits data from the CP 526 to the controller.

SEND-ALL and RECEIVE-ALL must be called at least every 100 ms in the STEP 5 program (max. 2 s permitted). The shorter the time between the calls, the better you can maintain the updating rates programmed into the dynamic displays (see Planning Guide COM GRAPH, Chapter 4, Appendix 1).

The data blocks with values for the input/output fields are also established during programming. You must define them in the controller program, and assign appropriate values to them. To speed up data transfer between the controller and the CP 526, all dynamic values, or as many as possible, should be stored in one data block.

During data transfer with the controller, the CP 526 does not permit blocking, since the maximum frame size prescribed by the CP 526 always corresponds to the size of a standard data block (256 words). The maximum frame size "BLGR," which is sent by the data handling blocks to the CP during synchronization, must also be set at 512 bytes (see also Description of the Data Handling Blocks).

In the case of the S5-135U programmable controller with S processor, you do not need to specify the frame size.

There are two ways to initiate commands on the CP 526 through a controller program:

● By programming the SEND-DIRECT data handling block or

● By setting a particular interprocessor communication flag bit

When programming the SEND-DIRECT data handling block, you only need to know the job number. This is the number entered in the A-NR field in the COM PROGRAMMING FUNCTIONS form. The QTYP, DBNR, QANF, and QLAE parameters are not relevant.

A command (= job) with a particular job number can be reset by programming RESET-DIRECT data handling block with the same job number. In other words, the RESET-DIRECT data handling block deletes any display that was initiated by a SEND-DIRECT with the same job number.

The RESET-ALL data handling block (job number = 0) resets all commands currently being processed on the CP 526.

The status of SEND-DIRECT or RECEIVE-DIRECT jobs is in the job status word. To find out which job is presently being processed by the CP, use the CONTROL data handling block. For this purpose, the job number is passed to the CONTROL data handling block as a parameter.

Bits 0 to 3 of the job status word contain the actual status of the job (command or special job).

Bit 3 2 1 0

0 0 0 0	Status after synchronization and RESET-ALL
0 0 1 0	Job executing
0 1 0 0	Job completed without error
1 0 0 0	Job completed with an error
1 0 1 0	CP 526 not ready for this job

If bit 0 is set, a RECEIVE-DIRECT is possible for the job.

In the case of status = 8 (bit 3 = 1), the high-order byte (bits 8 - 15) in the job status word contains one of the following error identifiers:

Identifier	Meaning
0108H to 0508H	Controller error
0608H	Job not possible now; try again later
0708H	Error because of continuously unacceptable firmware status; subsequent attempts are not recommended
0808H	Interpreter error (error in the execution of a command, or entry/operator error)
0908H	Error connected with message acquisition (SEND-DIRECT 219); message buffer overflow.
0A08H	Error with SEND-DIRECT 218 (send time to the CP 526); job not possible now; try again later
0B08H	Error with RECEIVE-DIRECT 218 (receive time from CP 526); job not possible now; try again later

Any errors flagged in the status word must be interrogated in the STEP 5 application program. If the status indicates that a job is still executing (status = 2), you cannot process a second job with this job number. Only after the job is completed (status = 4) can you start another job with the same job number.

If a command that is to be initiated through a SEND-DIRECT job is not stored in the memory submodule of the CP 526, the job is identified with 080Ah (h = hexadecimal). This job is not processed until the next SYNCHRON data handling block is executed.

You can terminate a command with a new command, a RESET-DIRECT, or a RESET-ALL.

If errors occur in the CP 526 while a command is being processed, the error identifier and the status are set to 8 (job status word = 0808h) upon completion of the command. In addition, the exact error number is stored at address F666h in the system data area (for a description of error numbers, see subsection 3.3.4).

If a command is correctly executed, the status is changed from 2 to 4 (bit 2 = 1) upon completion of the command.

If you specify job number 0 in the CONTROL data handling block, the job status word contains the job number of the command being processed by the CP 526 (but only if this command was assigned a job number when programming).

In addition to the CONTROL data handling block checking the job status word, each data handling block must be checked for parameter assignment errors (monitoring of PAFE byte; see Operator Guide for the data handling blocks of the programmable controller you are using).

Special Jobs

- SEND-DIRECT for messages:
  Job number can be programmed with COM (default: 219),
  Identifier 33H in byte 0 of the transferred data block

- SEND-DIRECT for key code:
  Same job number as SEND-DIRECT for messages, identifier 64H in byte 0 of the
  transferred data block

- SEND-DIRECT and RECEIVE-DIRECT for system time:
  Job number = 218

If the SEND-DIRECT 218 or the RECEIVE-DIRECT 218 are to execute in every cycle of
the STEP 5 application program, they must be programmed after all the other
calls of handling blocks in the cyclic STEP 5 application program. Otherwise, the
controller-to-CP 526 interface is restricted in its functional capacity due to
overload.

Figure 3-7 shows the correct processing sequence of a SEND-DIRECT handling
block in the form of a structure diagram.

FOREVER (cyclic controller program)

.
.
.

Message running

THEN   ?   ELSE

CONTROL DIRECT => status of the job for messages

CP ready
for new
message
(PAFE)

IF " first attempt to transmit these messages"

THEN   ?   ELSE

IF "status of the job for messages"

CASE "status of the job for messages"

?

THEN   ELSE

0002H	(default)	0002H	0004H	0608H	0908H	(default)	THEN	ELSE
Preceding job is still executing	Preceding job complete SEND-DIRECT for messages (Interrogate PAFE byte) First attempt to transmit this message has been completed	Job is still executing	Job complete without error Condition "messages for CP 526" reset	Job complete with an error Repeat job SEND-DIRECT for messages (Interrogate PAFE byte)	Job complete with an error (message buffer overflow)	Job cannot be processed correctly	Activate SEND-DIRECT	

.
.
.

Figure 3-7: Structure Diagram for Transmitting Messages from the Controller to the CP 526 Using SEND-DIRECT

When using the interprocessor communication flag to initiate a command, be sure to define the output interprocessor communication flags in data block DB1 of the controller.

To detect a signal edge from 0 to 1 to be recognized, the interprocessor communication flag must remain at 0 for at least one second and then remain at 1 for at least one second (see subsection 3.3.5).

You must also program the interprocessor communication flag command in the appropriate screen of the COM using the name "MXXX.X". The variables in the name are explained as follows:

MXXX.X

> Bit number (0 - 7) of the interprocessor communication flag byte
>
> Interprocessor communication flag byte number (0 - 255)
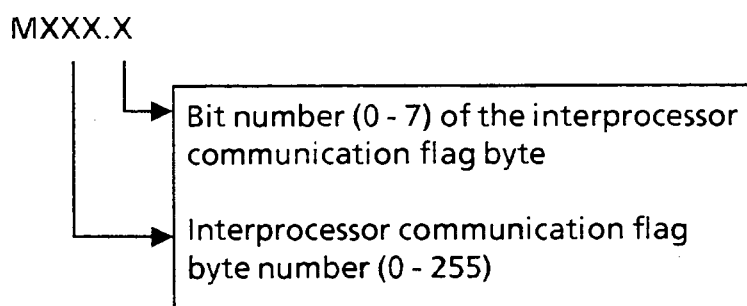
3.3.4   System Data Area

In addition to the data transfer area for the data handling blocks, there is also a memory area in the dual port RAM of the CP 526 containing information about internal system data. Currently, the controller can read this data only through the STEP 5 system operations. Although a write access by the controller is physically possible, it is not recommended.

SYSTAT

The data in this area identifies the operating and error statuses of the interface module. The data fields are rigidly structured and have the same meaning in all communications processors.

The system area SYSTAT of the CP 526 lies within the dual-port RAM beginning at byte address F660h. This area is described as follows:

Byte Address	F660		Operating Status
		Bit 0 = 1	The CP is in the STOP state.
		Bit 1 = 1	The CP is in the RUN state.
	F661	Bits 0 and 1 = 0	Battery Check The CP has no battery backup of its own.
	F662	Bits 0 - 7	Reader byte The CP 526 increments this byte every 100 msec. during trouble-free operation.
	F663	Bit 0 = 0	PG Priority The CP 526 cannot relinquish control to the programmer (PG)
	F664	Unassigned	
	F665	Unassigned	
	F666	Bits 0 - 7	Error Message The CP 526 writes the last message number that was displayed in the right side of the command line (can be set to 0 after evaluation by the controller).
	F667	Unassigned	

SYSID

The SYSID data area begins at byte address F680h and contains a parameter set for the following:

● Describing the characteristics of the CP 526 in the system

● Providing information concerning the hardware module

● Providing information concerning the firmware and software version of the module

Fundamentally:

● The parameters are stored in ASCII code and can therefore be read in plain-text by the controller.

● Each parameter field has a maximum length and is terminated by a carriage return (i.e., the <RETURN> key).

● The parameter field can contain less than the maximum number of characters; in this case, CR follows immediately after the last character.

● An unused field contains no characters, and is identified only by a CR.

● The last parameter is terminated by the SYSID end identifier CR ETX.

● The parameters are passed partly from the CP 526 and partly from the COM. It can therefore happen that the parameters to be passed by COM are still missing at the time of the Read operation.

The SYSID data area is described as follows:

Field	Definition	Max. Length	Example
0	Submodule identifier EPROM: order number 6ES5 xxx-xxxxx RAM: capacity in kwords CP 526 displays RAM capacity of basic board	12	373-0AA41 RAM 16 KW
1	Module identifier	8	CP 526
2	CP firmware version (if expansion board plugged in)	8	4.2 4.2E
3	Plant designation	19	Blast Furnace 47
4	Origination date	8	01/07/91
5	Unassigned		
6	Unassigned		
7	Unassigned		
8	Unassigned		
9	Version of COM software version	8	A01.0
10	Version of PROM software version	8	A02.1
11	Programming date for EPROM user submodules	12	090187084916
12	Number of EPROM submodules belonging to program	1	2

Field	Definition	Max. Length	Example
13	Type of CP 526 module (color, monochrome)	1	C
14	Type of monitor resolution (high/low)	1	H
15	Unassigned		
	End identifier		CR, ETX

3.3.5 Handling Interprocessor Communication Flag

The interprocessor communication flags, that are defined as input flags (CE) in data block DB1 of the controller cannot be processed as output interprocessor communication flags (CA). As a result, interprocessor communication flag bits that are set by the BIT SET command from the keyboard can only be reset by the BIT RESET command.

If you initiate a reset from the controller side without a CP 526 command, you can overwrite the interprocessor communication flags with the STEP 5 system operations.

The addresses of the interprocessor communication flag bytes on the CP 526, and the corresponding relative addresses in the controller, are as follows:

Relative address (decimal) in the controller	Absolute address (hexadecimal) in the CP
FY 0	F200
FY 1	F201
.	.
.	.
FY 255	F2FF

Below are examples of a program in which the interprocessor communication flags are set the by the controller. The following conditions apply:

● The interprocessor communication flags that are being used are enabled on the CP 526 and are programmed as interprocessor communication input flags (CE) in data block DB1 of the controller.

● An interprocessor communication flag byte is evaluated only in a controller.

Example 1

Bits 0 to 7 of the interprocessor communication flag byte 32 are programmed on the CP 526 by the BIT SET command.

DB1: KH: ..., CE00, 0020, ...

OB1 or OB13
.
.
JU FB X
.
.
FBX
.

L FY32	Interprocessor communication flag byte 32 in accumulator 2
L KB 255	Bit pattern 00FF in accumulator 1
AW	FY32 = 0 ?
JZ = END	Jump if FY32 = 0
L KBO	Accumulator 2 = 0
L KHF220	Address of interprocessor communication flag byte 32 in the CP
TIR 3	Interprocessor communication flag byte 32 in the CP overwritten with 0
.	
A F 32.0	Evaluate flag
.	
END:	The flags are reset by the operating system before the next cycle.
.	
BE	

This example refers to the S5-135U and S5-150U programmable controllers; in the case of the S5-115U, replace the TIR 3 operation by the TIR 2 operation.

Example 2

Bit 2 of interprocessor communication flag byte 34 is programmed on the CP 526 by the command BIT SET.

DB1: KH:..., CE00, 0022, ...

FBX
    :

AN F 34.2	Interrogate F 34.2 = 0 ?
JC = END	Jump if F 34.2 = 0
	: (evaluate F 34.2)
R F 34.2	Reset F 34.2
L FY34	Load FY34 into accumulator 2
L KHF222	Address KM34 in CP
TIR 3	Overwrite KM34 in CP with FY34

END:

BE

This example refers to the S5-135U and S5-150U programmable controllers; in the case of the S5-115U, replace the TIR 3 operation by the TIR 2 operation.

3.3.6   Message Processing

Message processing has the following features:

- Texts for a maximum of 1000 messages
- Four message groups distinguished
- Message buffer in the CP for 100 messages
- Indication of message buffer overflow
- Display of the message status through attribute change within the associated message text
  The following four possible conditions apply to messages:

  - Condition 1 →   Message has come (fault) and has not been acknowledged.
  - Condition 2 →   Message has come, and has subsequently gone (back to normal), but has not yet been acknowledged.
  - Condition 3 →   Message has come and has been acknowledged.
  - Condition 4 →   Message has come, has been acknowledged, and has gone.
- Output of current error messages in the message line of the monitor.

Message Processing in the Programmable Controller

The software package "Signaling functions for standard CP" (also refer to the 'Ordering Data' section) evaluates faults/changes arriving from the process in the controller CPU and activates message output. It comprises all necessary standard function blocks and data blocks.

The use of this package in plants with only a small number of messages to be displayed is less advisable, since it requires a considerable amount of memory capacity and computing time. In this case, messages should be started from the PLC. To do this, enter the message number in a data block generated for this purpose and transfer the latter to the CP 526 using the SEND DIRECT data handling block.

The controller determines the status of the message, and writes the message number and status such as 'message come', 'message gone', or 'acknowledged' in the message data block.The date and time are fetched from the hardware clock of the CP 526 by means of the RECEIVE DIRECT data handling block with job number 218 (also refer to subsection 3.3.8) and are written by the PLC into the message data block, data words 3 to 6 (see Fig. 3-10). Transmission is then

initiated by a SEND DIRECT from the PLC (see Fig. 3-8). However, if you enter the value '0' for 'Day' (10th byte in 'MELD' data block), this is interpreted by the CP 526 as a request to fetch the date and time directly from its own hardware clock and add them to the message.

Message acknowledgment is initiated from the CP 526, the corresponding message number is written into the acknowledgment data block, and the controller updates the data block with a RECEIVE-ALL.

Message Processing in the CP 526

The CP 526 receives the message data block from the controller and enters the arriving message/acknowledgment numbers one after the other into the CP internal message buffer, together with the date and time, provided there is an associated message text in the user memory. If messages have already been received, the date and time are updated.

The last message arriving in the data block is displayed in the message line. You can specify if only arriving messages (i.e. the last message to come) or those that have come <u>and</u> gone (i. e. the last message that has come or gone) to be displayed as current message in the message line. You have to make this decision in the CP PARAMETER ASSIGNMENT form.

If the buffer is full, the last message entered in the buffer is overwritten (overflow message). If more messages occur, they overwrite the previous overflow message. The number of overflow messages is determined in the CP 526, and displayed (flashing) in the message line beginning at column 72. There is no further reference in the second part of the command line. The overflow is retained in the message status of the message buffer, and the condition code word 0908h is set in the job status word of the SEND-DIRECT (see subsection 3.3.3).

To remove messages from the message buffer, the messages must be brought into the "back-to-normal" or "gone" state. The messages must then be acknowledged. This deletes them from the message buffer. New messages can then be entered in the message buffer. If the message number already exists in the message buffer, the following conditions are distinguished:

State of message arriving	State of message in CP 526 buffer	New state of message in buffer	Display type
arriving	has come	arriving + time upd.	(1)
arriving	has come and gone	arriving + time upd.	(1)
arriving	has come and has been acknowledged	arriving + time upd.	(1)
going	has come	has come and gone	(2)
going	has come and gone	has come and gone	(2)
going	has come and has been acknowledged	has come, has been acknowl., has gone	(4)
acknowledgm.	has come	has come and has been acknowledged	(3)
acknowledgm.	has come and gone	message is cancelled	
acknowledgm.	has come and has been acknowledged	has come and has been acknowledged	(3)

(Time upd.: Date and time are updated in message buffer)

If a message number is not in the message buffer, any back-to-normal messages or acknowledgments for the message are rejected.

Resetting the Message Buffer

In the event of a cold restart after a PLC-SYNCHRON, and after display editing with the COM, message processing in the CP 526, including the message buffer, is reset.
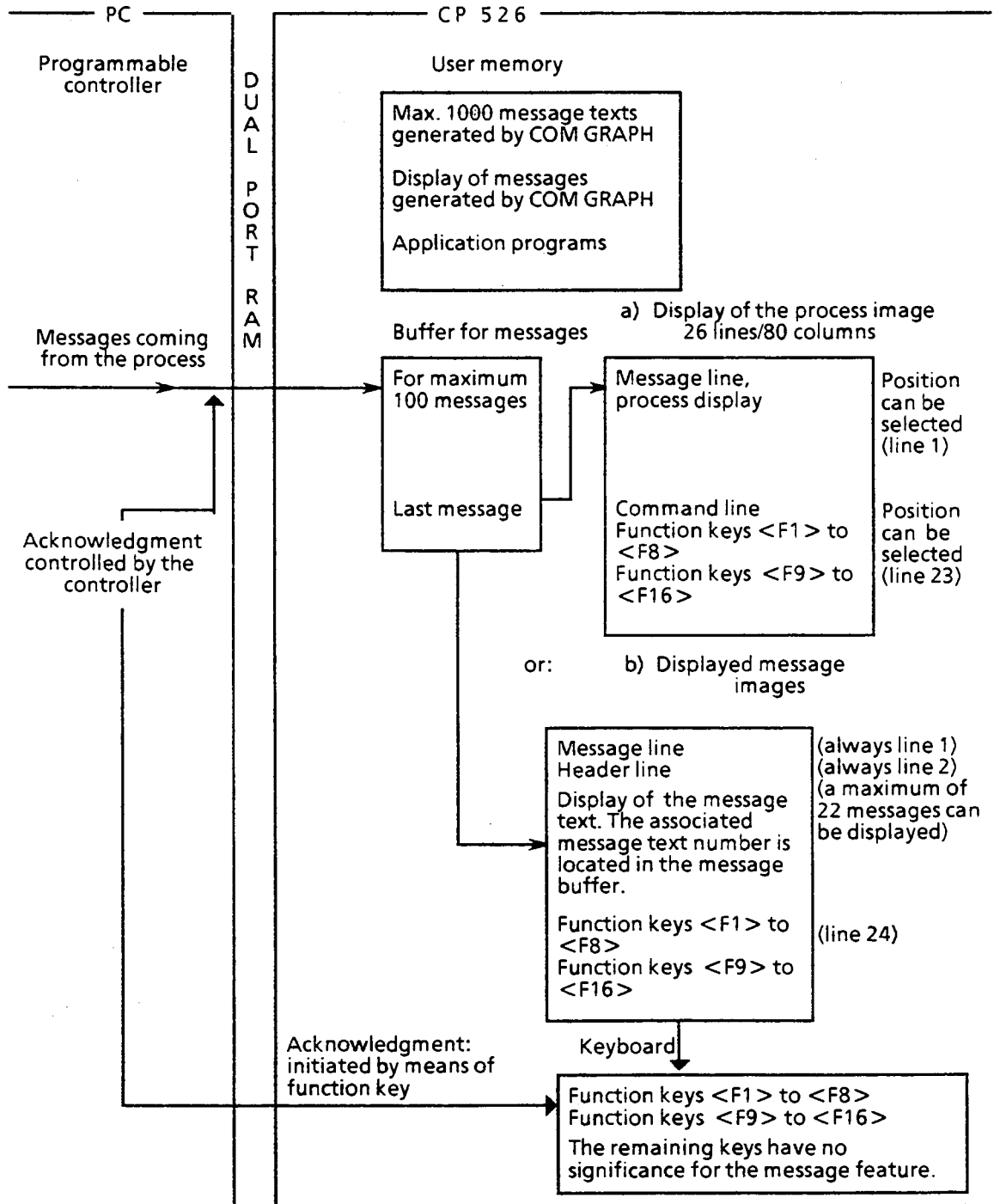
Figure 3-8: Overview

STEP 5 Example of Messages

(See also structure diagram, Figure 3-7)

This function block causes messages to be transferred from the S5-135U/R processor to the CP 526.

The parameters are defined as follows:

- JOB#   Job number of the display on the CP 526 monitor into which the transferred messages are entered (= job number in the corresponding display output command of the CP 526 program). Format: KF

- DB#   Number of the data block that contains the messages to be transmitted. Format: KF

- SIZE   Number of data words being transmitted (including the header of the message record). Format: KF

- INFO   Location providing information concerning the results of processing this function block. Format: FY

  The following meanings apply:

  <INFO> = 0 → Correct processing of the messages
  <INFO> = 1 → Repeat call
  <INFO> = 2 → Messages currently being transferred
  <INFO> = 3 → Basic error. Review the application program structure and message data block.

<INFO> = 4 → CP 526 message buffer has an overflow. Wait
until some of the transferred messages are
acknowledged and the message buffer has free
space again. Then retry.

The function block uses data block DB197, which has the following structure:

0: KF	= + 1	Data words 0, 1, 2, and 3 contain the actual parameters
1: KF	= + 219	for SSNR, ANR and ANZW with indirect parameter
2: KS	= DB	assignment of the SEND-DIRECT data handling block
3: KY	= 197,4	
4: KH	= 0000	Job status word
5: KH	= 0000	Continuation job status word
6: KS	= DB	Data words 6, 7, 8, and 9 contain the actual parameters for
7: KY	= 0,199	the source parameters with the indirect parameter assign-
8: KF	= + 0	ment (QTYP: XX) to the SEND-DIRECT data handling block
9: KF	= + 7	It has to be KF = 7 if no process variable has been stored. If
		one has been stored, it has to be KF = 9 for DW9.
10: KH	= 0	Unassigned
11: KF	= 0	Status word; identical to formal operand INFO
12: KF	= 0	Identical to formal operand JOB#

Data handling blocks CONTROL-ALL, CONTROL-DIRECT and SEND-DIRECT, in each
case with indirect parameters for SSNR, ANR, ANZW and the source parameters,
are used, as well as flag byte FY 197.

FY197, DB197/DW4 and DW11 must be reset in the restart routines.

```
NAME        :MESSAGE
DECL:JOB#   I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
DECL:DB#    I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
DECL:WORD   I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
DECL:SIZE   I/Q/D/B/T/C: D  KM/KH/KY/KS/KF/KT/KC/KG: KF
DECL:INFO   I/Q/D/B/T/C: Q  BI/BY/W/D: BY
```

```
0014            :C      DB197          LOAD ACTUAL PARAMETERS INTO DB198
0015            :LW     = DB#
0016            :T      DW7
0017            :LW     = WORD
0018            :T      DW8
0019            :LW     = SIZE
001A            :T      DW9
001B            :LW     = JOB#
001C            :T      DW12
001D            :
001E            :L      KF + 0         ENTER A-NR = 0 (ALL FUNCTIONS)
0020            :T      DW1            AND
0021            :JU     FB123          CALL CONTROL-ALL
0022 NAME  :CONTROL
0023 SSNR  :        KY255,0
0024 A-NR  :        KY0,0
0025 ANZW  :        DW0
0026 PAFE  :        FY197              PAFE > < 0 (RLO = 1)?
0027            :JC     = M001         YES, ERROR, --> STATUS = 1
0028            :L      DW12           NO, IS THE JOB NUMBER OF THE
0029            :L      KF + 0         DISPLAY RELEVANT?
002B            :! = F
002C            :JC     = M005         NO
002D            :L      DW12           YES, IS IT THE DESIRED DISPLAY?
002E            :L      DW4
002F            :> < F
0030            :JC     = M001         NO, --> STATUS = 1
0031            :
0032 M005  :L      KF + 219           YES, LOAD A-NR = 219 (DIRECT FCT)
0034            :T      DW1            AND
0035            :JU     FB123          CALL CONTROL-DIRECT
0036 NAME  :CONTROL
0037 SSNR  :        KY255,0
```

0038 A-NR	:	KY0,0	
0039 ANZW	:	DW0	
003A PAFE	:	FY197	PAFE > <0 (RLO = 1)?
003B	:JC	= M001	YES, ERROR --> STATUS = 1
003C	:		
003D	:L	DW11	NO, STATUS > 1 ?
003E	:L	KF + 1	
0040	:>F		
0041	:JC	= M002	YES
0042	:A	D 4.1	NO, IS A JOB BEING PROCESSED?
0044	:JC	= M001	YES
0045	:		
0046	:JU	FB120	NO, CALL SEND-DIRECT
0047 NAME	:SEND		
0048 SSNR	:	KY255,0	
0049 A-NR	:	KY0,0	
004A ANZW	:	DW0	
004B QTYP	:	KSXX	
004C DBNR	:	KY0,197	
004D QANF	:	KF + 6	
004E QLAE	:	KF + 0	
004F PAFE	:	FY197	PAFE > <0 (RLO = 1)
0050	:JC	= M001	YES, ERROR, --> STATUS = 1
0051	:L	DW4	NO, JOB IS BEING PROCESSED!
0052	:L	KF + 2	SET STATUS = 2
0054	:JU	= M003	
0055	:		
0056 M001	:L	KF + 1	SET STATUS = 1
0058	:JU	= M003	
0059	:		
005A M002	:A	D 4.1	IS THE JOB STILL BEING PROCESSED?
005C	:JC	= M004	YES
005D	:A	D 4.2	NO, DID IT FINISH WITHOUT ERROR?
005F	:L	KF + 0	SET STATUS = 0
0061	:JC	= M003	YES
0062	:A	D 4.3	NO, DOES AN ERROR EXIST
0064	:AN	D 4.8	BECAUSE OF
0066	:A	D 4.9	CURRENT FIRMWARE STATUS
0068	:A	D 4.10	OF THE CP 526 ?
006A	:AN	D 4.11	

006C	:L	KF + 1	LOAD STATUS 1
006E	:JC	= M003	YES, REPEAT CALL
006F	:A	D 4.3	NO, DOES THE CP 526 MESSAGE
0071	:A	D 4.8	BUFFER HAVE AN OVERFLOW?
0073	:AN	D 4.9	
0075	:AN	D 4.10	
0077	:A	D 4.11	
0079	:L	KF + 4	LOAD STATUS = 4
007B	:JC	= M003	YES
007C	:L	KF + 3	NO, LOAD STATUS = 3
007E	:		
007F M003	:T	= INFO	STORE STATUS
0080	:T	DW11	
0081 M004	:BE		BLOCK END

3.3.6.1   Message Texts

Every message text that is generated contains the name "MXXXX" (where "XXXX" is a number between 0001 and 1000). The message text is stored under this name in the user memory of the CP 526. The maximum number of messages is 1000. If the message number "WXYZ" occurs during operation, the message text MWXYZ is assigned to it. A message text consists of up to 69 characters and appears beginning in column 3 of a message display line and ends at column 71. The message text contains the following information:

● Group number - The message text is displayed only if the appropriate group has been selected.

● Display attribute - Basic color of the message text

● Date and time area (optional)- Mode of representation is programmable

● Area for inserting a process variable.

● Acknowledgment area - Display of the message status in the message buffer of the CP 526

The process variable must be stored in data words DW7 and DW8 of the 'MELD' data block (Figure 3-8).The type of the process variable is defined. in the second byte of the data block. All S5 data formats are permitted, including the type 'character string'. The process variable can be up to four bytes long.

3.3.6.2   Message Structure

After the command DISPLAY MESSAGE has been activated in the CP 526 (keyboard entry or controller job), the screen is cleared and the message screen is displayed. When the screen is cleared, the last incoming message in the message line is deleted, and you cannot enter any new commands. The function keys <F1> through <F16> have new functions for the message screen. Additional controller job commands are stored in the CP 526 and are evaluated after the command line is exited.

A job for transmitting the key codes will continue to be recognized by the CP 526, allowing the message line to be serviced from the controller or keyboard.

Colums 1 and 2, as well as colums 72 to 80, of the message screen are reserved. The cursor is in column 1 (shown as a square at the center of the character), pointing to the message being processed.Columns 2 and 72 are blank columns. Columns 73 to 80 are reserved (see the message screen in Figure 3-9).

The message line (line 1) always shows the text of the last message received or of the last message received and gone. The text corresponds to the text in the process display. The acknowledgment area (message status) is not shown in the message line. If the text does not exist in the CP 526 user memory, the message is discarded. The number of messages in the message buffer is shown in columns 73 to 80. If the message buffer in the CP 526 overflows, the message line blinks and indicates the number of overflow messages.

The second line of the message screen is the header line. Columns 73 to 80 show the number of the first message in the message screen. The current date and time are also shown in this line (columns 3 to 20). The current values of the date and time are continuously updated from the CP 526 internal clock.

The heading programmed with COM appears in columns 21 to 71.

The first 22 messages from the message buffer appear in the remaining lines. The CP 526 cannot yet distinguish between message groups after it displays the message screen.

The cursor is in the first column of the third line, pointing to the first displayed message in the message screen. You can now use the function keys to process the messages.

- 00 -   - 70 -

	1234567890	1234567890

1	12/20/87 10:45:50	Always the last message received	63
2	12/23/87 09:23:56	Heading for message group X	8
3	12/16/87 04:56:56	Fault in Warehouse 23	
4	12/18/87	Close valve number 8 at 10 o'clock	
5	etc.		
6			
7	10/01/88 12:15:14	Furnace 15 overheated: 513 deg. C	
8			
9			
●			
●			
18			
19			
20			
21			
22			
23			
24			
25			
26	F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8

F 9	F 10	F 11	F 12	F 13	F 14	F 15	F 16
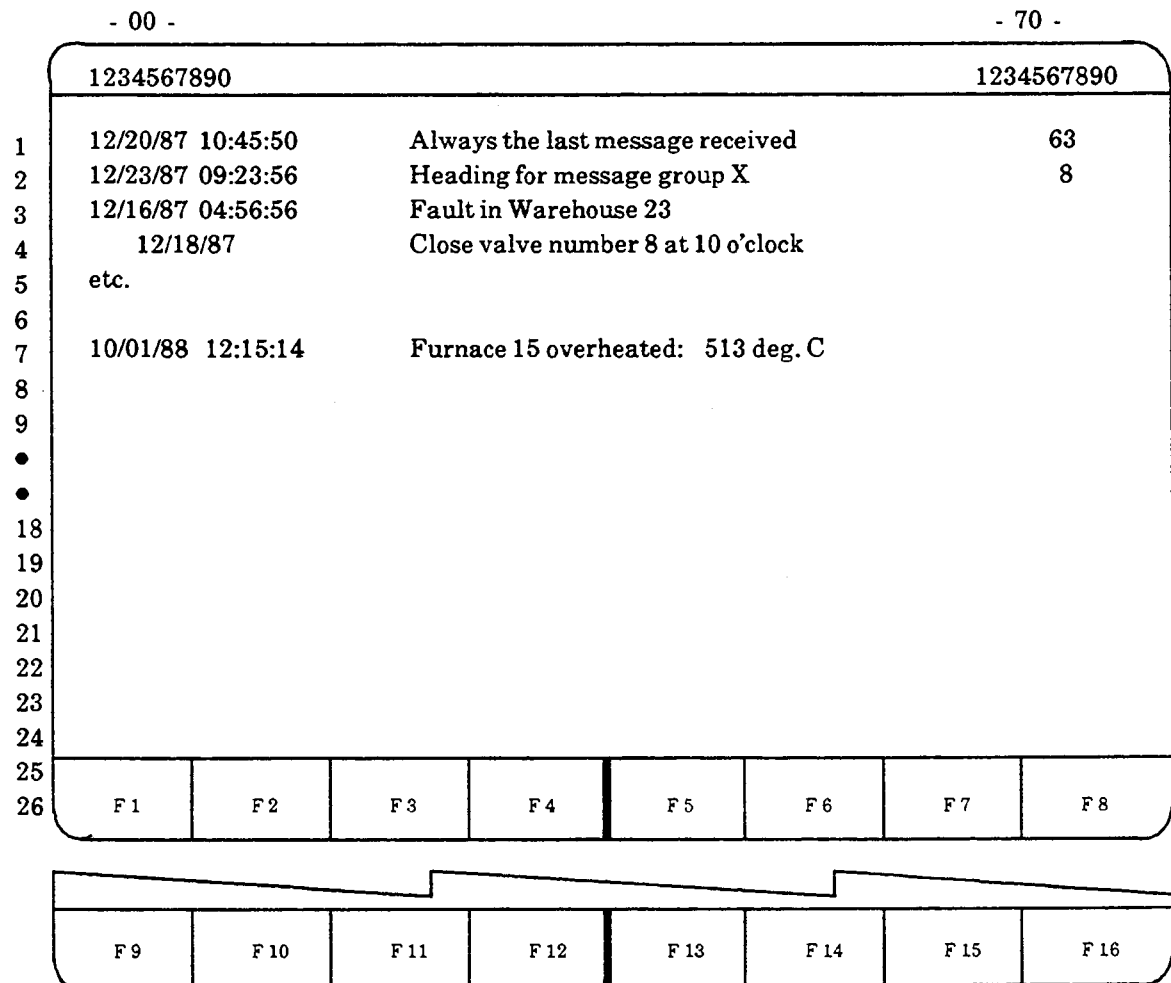
Figure 3-9: Message Screen

The text within the acknowledgment area has a different attribute, depending on the state of the message. There are four different message states, and four different display modes for the acknowledgment area. These are the same for all messages. The date and time area cannot be the acknowledgment area and is always displayed in the basic color of the message text.

If additional messages occur while the message display is being viewed, the additional messages are displayed in the message line (first line in the message screen) and after the last incoming message.

Processing the Message Screen with the Function Keys

The following information describes the default functions of the 16 function keys. You can reprogram each of the function keys with COM.

- <F1> key - performs the same function as the <LINE UP> key. Pressing this key moves the cursor up one line at a time. If the cursor is in the first line of the message screen, pressing the <F1> key causes the messages to move down one line at a time (i. e. F1 performs the function of the F2 key).

- <F9> key - performs the same function as the <LINE DOWN> key. Pressing this key moves the cursor down one line at a time. If the cursor is in the last line of the message image, pressing this key causes the messages to move up one line at a time (i. e. F9 performs the function of the F10 key).

- <F2> key - causes the displayed messages to move down one line at a time. The next (or older) message becomes visible at the top of the display. If the first message of the message buffer is displayed, however, pressing this key has no effect, and the cursor stays where it is.

- <F10> key - causes the displayed messages to move up one line at a time. The next message appears on the bottom of the screen. If the last message of the message buffer is at the bottom of the display, however, pressing this key has no effect, and the cursor stays where it is.

- \<F3\> key - causes the previous 20 messages to be displayed on the screen. The cursor remains at the beginning of the line displaying the first message. If you reach the beginning of the message buffer, however, pressing this key has no effect.

- \<F11\> key - causes the next 20 messages to be displayed on the screen. The cursor remains at the beginning of the line displaying the first message. If you reach the end of the message buffer, pressing this key has no effect.

- \<F4\> key - causes the first 20 messages in the message buffer to be displayed on the screen. The cursor remains at the beginning of the line displaying the first message.

- \<F12\> key - causes the last 20 messages in the message buffer to be displayed on the screen. The cursor remains at the beginning of the line displaying the last message.

- \<F13\> key - causes the number of the message at the current cursor position to be written into the data block (ACK) and transmitted to the controller. After the controller confirms the acknowledgment of the message, the message is identified as "acknowledged" in the communications processor. A message is retrieved from the message buffer if it had the following state before the acknowledgment:

  a) It has come and gone (back to normal).
  b) It has come, been acknowledged, and gone (back to normal).

  The remaining messages then move up one line in the message screen.

- <F5> key - causes the numbers of all messages displayed in the screen to be written into the data block (ACK) and transmitted to the controller. After the controller confirms the acknowledgment of the messages, the messages are identified as "acknowledged" in the CP.
  A message is retrieved from the message buffer if it had the following state before the acknowledgment:
  a)   It has come and gone (back to normal).
  b)   It has come, been acknowledged, and gone (back to normal).
  The remaining messages then move up one line in the message display.

- <F16> key - causes the displayed message screen to be deleted, and the previously displayed picture reappears.

- <F6>, <F14>, <F7>, and <F15> keys
  <F6> causes those messages that belong to message group 1 to be displayed in the message screen. The same holds true for the <F14>, <F7>, and <F15> keys for message groups 2, 3, and 4, respectively.
  The cursor position in the message display does not change.

- <F8> key
  After you press the <GR 1...4> key, all messages are displayed again. There is no distinction by message group.

3.3.6.3 Message Exchange Between the Programmable Controller and the CP 526

Messages are exchanged between the programmable controller and the CP 526 through a data block that is transferred with a SEND-DIRECT and a programmable job number. The execution sequence is shown in the structogram for the "Transfer of Messages between the programmable controller and CP 526" in Figure 3-7.

If more than one CPU is used in a programmable controller the 1000 message texts will have to be distributed to the individual CPUs. This e.g. means:

First CPU: message text nos.   1 through   500
Second CPU: message text nos.   501 through   1000

etc.

This ensures that the message text will be displayed in the corresponding CP if a message is output by a CPU. If a message is acknowledged in the CP 526 it is also ensured that the acknowledgement will be transferred to the corresponding CPU.

When filling up the data block, make sure that you enter the correct number of data words, as well as valid data, into the message blocks. The CP 526 does not check the data.

Data Word

DW 0	Byte 0	33H								Identifier for message
	Byte 1	Length								Number of data words beginning with data word 1
DW 1	Byte 2	P	P	P	P	P	P	P	P	Type of process variable
	Byte 3	E	E	E	E	E	E	E	E	Message status
DW 2	Byte 4						N	N	N	Message number
	Byte 5	N	N	N	N	N	N	N	N	(001...7FFH = 2047D)
DW 3	Byte 6	S	S	S	S	S	S	S	S	1/10 seconds 70H
	Byte 7	S	S	S	S	S	S	S	S	Seconds 16H
DW 4	Byte 8	M	M	M	M	M	M	M	M	Minutes 15H
	Byte 9	H	H	H	H	H	H	H	H	Hours 14H
DW 5	Byte 10	D	D	D	D	D	D	D	D	Day 28H
	Byte 11	M	M	M	M	M	M	M	M	Month 10H
DW 6	Byte 12	Y	Y	Y	Y	Y	Y	Y	Y	Year 87H
	Byte 13									Reserved
DW 7	Byte 14	P	P	P	P	P	P	P	P	Process variable with max. 4 bytes
	Byte 15	P	P	P	P	P	P	P	P	(is not evaluated by the CP 526)
DW 8	Byte 16	P	P	P	P	P	P	P	P	
	Byte 17	P	P	P	P	P	P	P	P	
DW 9	Byte 18	P	P	P	P	P	P	P	P	Type of process variable
	Byte 19	M	M	M	M	M	M	M	M	Message status

Beginning of 1st message

e. g. 28.10.87 14.15 16 sec. 700 msec.

in BCD code — Information about the first message (1st message block)

Beginning of 2nd message

(2nd message block)

Figure 3-10: Structure of the Data Block (MELD)

If byte 2 (**type of process variable**) contains the value FF (hexadecimal), no process variable is contained in the data block. The next message then begins at DW7.

With other assignments of byte 2, the CP 526 does not take the contents of DW7 and DW8 into account.

The message status indicates whether a message has come, gone, or been acknowledged by the controller. Here are the possibilities:

- xxxx xx00 → message gone (back to normal)
- xxxx xx01 → message has come (fault)
- xxxx xx10 → acknowledgment (message acknowledged)

The message number refers to the corresponding message text in the user memory (refer to subsection 3.3.6.1).

Acknowledgment Exchange Between Programmable Controller and CP 526

Acknowledgments are exchanged between the controller and CP 526 through a data block that is transferred with a RECEIVE-ALL. The number of the data block can be programmed when assigning the CP parameters (see Planning Guide COM GRAPH). The job number corresponds to the configurable job number for "Display message screen."

Data word

Byte 0 DW0	05H								Identifier for acknowledgment
Byte 1	Length								Number of words without header
Byte 2 DW1	0	0	0	0		N	N	N	Message number
Byte 3	N	N	N	N	N	N	N	N	

Figure 3-11: Structure of the Data Block (QUIT)

If a message is acknowledged from the keyboard, the message number is written into the data block, and the latter is transferred to the controller with a RECEIVE-ALL. The application program in the controller must then confirm the acknowledgment for the CP 526 with a SEND-DIRECT. Only after this confirmation is received are the acknowledgments processed in the CP 526. If the data block contains an acknowledgment with a message text number that is not in the message buffer, it is discarded by the CP 526.

If all the messages that are displayed in the message screen are acknowledged, a maximum of 22 message numbers is entered in the data block.

3.3.7   Emulation of the Process Keyboard

All key sequences that you can enter via the process keyboard can also be emulated by the controller with a special job number. To do this, you write the required key codes of the process keyboard into a data block, and then transfer this data block by means of a SEND-DIRECT with the job number for messages (default value: 219). The job order numbers can be set when you assign CP parameters with COM.

Data word

Byte 0 DW 0	64H	Identifier for Key Codes
Byte 1	Length	Number of Key Codes < 41
Byte 2 DW 1	31H	Key Code Example
Byte 3	32H	Key Code Example
Byte 4 DW 2	33H	Key Code Example
Byte 5	34H	Key Code Example

Figure 3-12: Structure of the Data Block

You can transmit a maximum of 40 key codes (key activations) with one SEND-DIRECT. The CP 526 interprets the data as if it came from a keyboard, and, starting at the current cursor position, writes the key codes into the input field or into the command line of the process display. At this point, the CP 526 can no longer distinguish between a key code coming from the controller or one coming from the keyboard.

The controller can write all the key codes into the data block, including those for the <F1> through <F16> function keys as well as the cursor control keys.

NOTE:
When transmitting key codes, the number x of data words to be transmitted (parameters of the data handling blocks SEND-DIRECT A-NRxxx) must correspond to the number y of the key code bytes to be transmitted (right-hand byte of the first data word) where number y is specified in the associated data block. The following holds true:
If y is even, $x = y/2 + 1$.
If y is odd, $x = (y + 1)/2 + 1$.
An error can occur if you do not adhere to these calculations.

Only one function key or one enabling character can be contained in a data block. Enabling characters include the following:

- Cursor control keys
- Key code "ODH" (key 11 = <RETURN> key)
- Key code "E7H/A7H" (key 12 = cursor to the left with delete)

If other key codes follow, this will cause errors during processing. ASCII characters can precede the function key code or the enabling character code.

The number of ASCII characters contained in one data block must correspond to the length of the currently addressed input field. If more characters are transferred, they are discarded.

You cannot transfer key codes from the controller while you are entering characters through the process keyboard. This mixed operation can cause illegal operating states in the CP 526.

In addition, you cannot enter key codes from the controller if:

● the CP 526 is in the "PROGRAMMING DISPLAYS" mode, or

● changes in the user memory (delete, transfer) are being made by the COM program.

In these cases, set the CP 526 mode selector to the "STOP" position.

Also note that the CP 526 still needs time to construct a display or verify an input value. The CP 526 can only receive another string of key codes after a job has been completely executed.

Example:
Activating a display output command (BILD XY) through the controller keyboard, and entering the value 1234 in its first input field.

Steps
1. Move cursor to command line   95h
2. Command name   'BILD XY' 91h
3. Move cursor to first input field   97h
4. Input value   '1234' 91h

The third step is accepted by the CP 526 only when the display output command has been completed (the first input field is defined and the cursor is waiting for an input). Otherwise, the controller job (analogous to input via the process keyboard) is discarded and a negative acknowledgment is given. Consequently, the checks shown in the flowchart of Figure 3-11 must be performed in the controller program. The subsequent STEP 5 program shows, for example, one possibility for implementing key codes in the S5-135U with R processor.

NOTE:
The CP 526 recognizes another cursor control character other than the keyboard codes (see Figure 3-24) that allows you to position the cursor at an arbitrary input field. You cannot enter this control character, however, through the process keyboard.

With **7Fh line column** (line: 00..17h, column: 00..4Fh), the cursor is set to the corresponding position within the display, provided an input field has been configured there. Otherwise, the error message "WO4" appears and the cursor waits in the command line for further input.
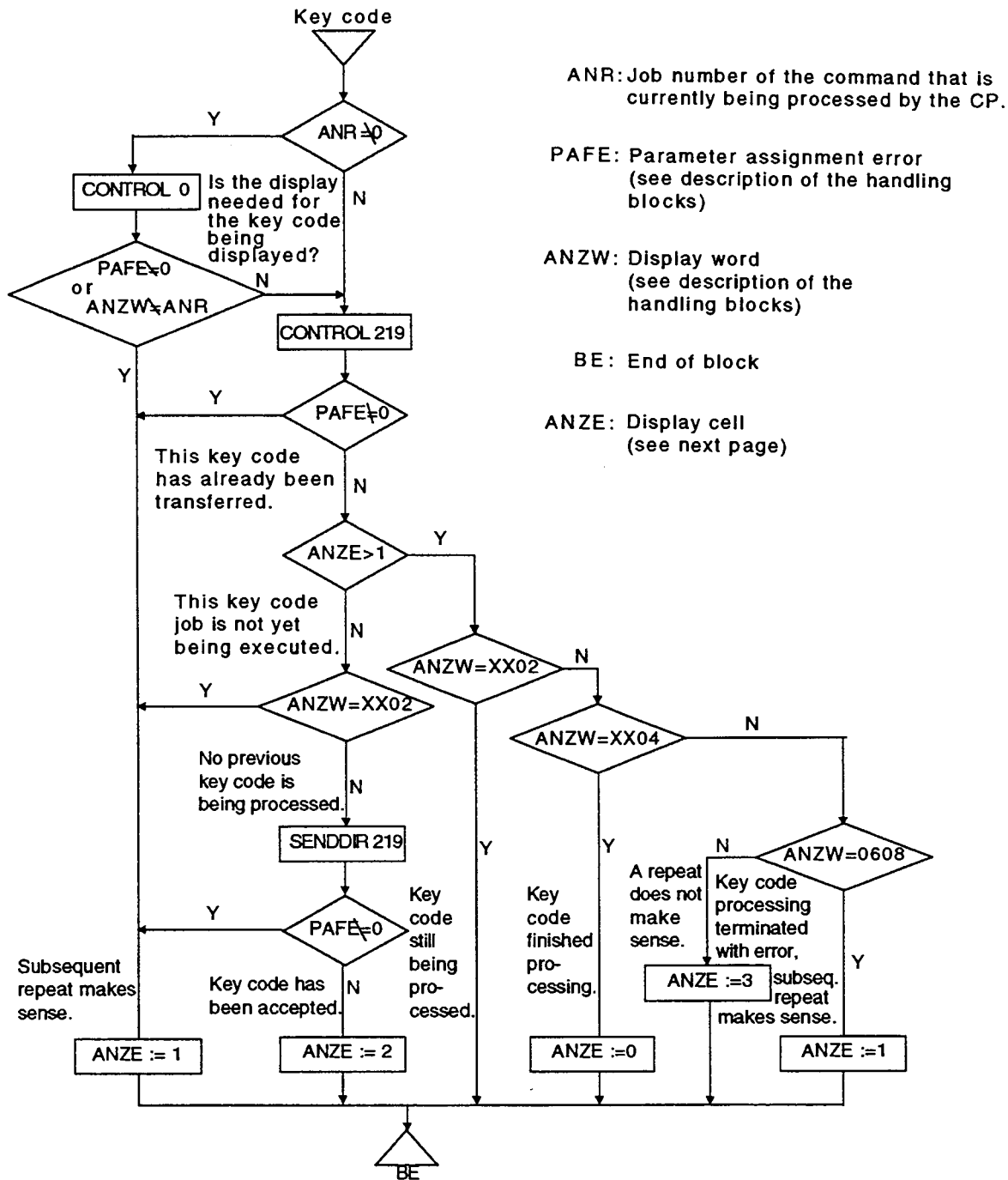
Figure 3-13: Flowchart "Processing a Key Code Job"

STEP 5 Example for Key Code

This function block transmits the key code from the S5-135U with R processor to the CP 526 communications processor.

The parameters are defined as follows:

- Formal parameter:

- **JOB#**   Job number of the display on the CP 526 monitor, on which the key codes that are being transmitted are to act (job number in the corresponding display output command of the CP 526). A-NR = 0 means that the job number of the display selected is irrelevant. Format: KF

- **DB#**   Number of the data block that contains the key code to be transmitted. Format: KF

- **WORD**   Number of the data word in the data block that contains the first data word to be transmitted. Format: KF

- **SIZE**   Number of data words to be transmitted. Format: KF

- **INFO**   Location that provides information concerning results of processing this function block. Format: FY

    The following INFO values apply:

    $<$INFO$>$ = 0 → The last key code has been correctly processed.
    $<$INFO$>$ = 1 → The call should be repeated with the same key code.
    $<$INFO$>$ = 2 → A key code is currently being processed.
    $<$INFO$>$ = 3 → Basic error. Review the application program structure.

The handling block uses data block DB198, which has the following structure:

0: KF = + 1	Data words 0, 1, 2, and 3 contain the actual parameters for
1: KF = + 219	SSNR, A-NR, ANZW, with indirect parameter assignment
2: KS = DB	to the SEND-DIRECT data handling block.
3: KY = 198,4	
4: KH = 0000	Job status word
5: KH = 0000	Continuation job status word
6: KS = DB	Data words 6, 7, 8, and 9 contain the actual parameters
7: KY = 0,199	for the source parameters with indirect parameter assign-
8: KF = + 1	ment (QTYP: XX) to the SEND-DIRECT data handling block
9: KF = + 2	
10: KH = 0	Unassigned
11: KF = 0	Status word, identical with the formal operand INFO
12: KF = 0	Identical with the formal operand JOB#

The data handling blocks CONTROL-ALL, CONTROL-DIRECT, and SEND-DIRECT are always used with indirect assignment of the SSNR, A-NR, and ANZW parameters, and the source parameters, as well as flag byte FY198.

FY198, DB198/DW4, and DW11 must be reset in the restart routines.

NAME	:KEY-CODE	
DECL	:JOB#	I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KF
DECL	:DB#	I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KF
DECL	:WORD	I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KF
DECL	:SIZE	I/Q/D/B/T/C: D KM/KH/KY/KS/KF/KT/KC/KG: KF
DECL	:INFO	I/Q/D/B/T/C: A BI/BY/W/D: BY
0014	:C DB198	LOAD ACTUAL PARAMETERS INTO DB198
0015	:	
0016	:LW = DB#	
0017	:T DW7	
0018	:LW = WORD	
0019	:T DW8	

001A	:LW	= SIZE	
001B	:T	DW9	
001C	:LW	= JOB#	
001D	:T	DW12	
001E	:		
001F	:L	KF + 0	ENTER A-NR = 0 (ALL FUNCTIONS)
0021	:T	DW1	AND
0022	:JU	FB123	CALL CONTROL-ALL
0023 NAME	:CONTROL		
0024 SSNR	:	KY255,0	
0025 A-NR	:	KY0,0	
0026 ANZW	:	DW0	
0027 PAFE	:	FY198	PAFE > < 0 (RLO = 1)?
0028	:JC	= M001	YES; ERROR, --> STATUS = 1
0029	:L	DW12	NO; IS DISPLAYED JOB NUMBER
002A	:L	KF + 0	RELEVANT?
002B	:! = F		
002D	:JC	= M002	NO
002E	:L	DW12	YES, DESIRED DISPLAY?
002F	:L	DW4	
0030	:> < F		
0031	:JC	= M001	NO, --> STATUS = 1
0032	:		
0033 M005	:L	KF + 219	YES, LOAD A-NR = 219 (DIRECT FCT)
0035	:T	DW1	AND
0036	:JU	FB123	CALL CONTROL-DIRECT
0037 NAME	:CONTROL		
0038 SSNR	:	KY255,0	
0039 A-NR	:	KY0,0	
003A ANZW	:	DW0	
003B PAFE	:	FY198	PAFE > < 0 (RLO = 1)?
003C	:JC	= M001	YES, ERROR --> STATUS = 1
003D	:		
003E	:L	DW11	NO, STATUS > 1 ?
003F	:L	KF + 1	
0041	:> F		
0042	:JC	= M003	YES
0043	:A	D 4,1	NO, IS A JOB BEING PROCESSED?
0045	:JC	= M001	YES
0046	:		

```
0047            :AN   D      0.9
0049            :AN   D      0.10
004B            :R    F      98.0
004C            :
004D            :A    F      98.2
004E            :AN   F      98.1
004F            :S    F      98.0
0050            :A    F      98.2
0051            : =   F      98.1
0052            :
0053            :A    F      98.0
0054            :JU   FB120            NO, CALL SEND-DIRECT
0055  NAME :SEND
0056  SSNR  :     KY255,0
0057  A-NR  :     KY0,0
0058  ANZW  :     DW0
0059  QTYP  :     KSXX
005A  DBNR  :     KY0,198
005B  QANF  :     KF + 6
005C  QLAE  :     KF + 0
005D  PAFE  :     FY198             PAFE > < 0 (RLO = 1)?
005E            :JC   = M001         YES, ERROR, -- > STATUS = 1
005F            :
0060            :L    DW4             NO, JOB IS BEING PROCESSED!
0061            :L    KF + 2          SET STATUS = 2
0063            :JU   = M003
0064            :
0065  M001  :L    KF + 1          SET STATUS = 1
0067            :JU   = M003
0068            :
0069  M002  :A    D 4.1           IS THE JOB STILL RUNNING?
006B            :JC   = M004
006C            :
006D            :A    D 4.2           NO, DID IT FINISH WITHOUT ERROR?
006E            :L    KF + 0          SET STATUS = 0
0071            :JC   = M003         YES
0072            :
```

```
0073          :A    D 4.3        NO, DOES AN ERROR EXIST BECAUSE OF
0075          :A    D 4.9        INSTANTANEOUS FIRMWARE STATUS OF
0077          :A    D 4.10       THE CP 526?
0079          :L    KF + 1       LOAD STATUS 1
007B          :JC   = M003       YES
007C          :
007D          :L    KF + 3       NO, LOAD STATUS = 3
007F          :
0080 M003     :T    = INFO       STORE STATUS
0081          :T    DW11
0082          :
0083 M004     :***


Network 2           0084
0084          :BE
```

3.3.8   Exchange of Date and Time Between the Controller and the CP 526

When the date and time are exchanged between the controller and the CP 526, the following occurs:

-   The controller reads the time on the CP 526.
-   The controller writes the time on the CP 526.
-   The CP 526 uses the MASTER clock to set the time.
-   Clocks of several CPs are synchronized.

These exchanges take place exclusively through the SEND-DIRECT and RECEIVE-DIRECT data handling blocks with job number 218.

The data transfer area has the following structure:

Byte addr.	Contents	Remarks
00h	MASTER Identifier	Bit 0 = 1: CP 527 is time master
01h	Unassigned	
02h	1/10 | 1/100 s	All time specifications in BCD
03h	sec.	
04h	min.	
05h	hour	
06h	day	
07h	month	
08h	year	
09h	0	

Figure 3-14: Structure of the Data Transfer Area

With a SEND-DIRECT 218, the following applies:

- A CP 526 can be declared to be the time master.
- The date and time can be transferred to a CP 526 or a CP 526 can simultaneously be declared the time master.

With a RECEIVE-DIRECT 218, the following applies:

- The date and time can be read from the CP 526, if the CP 526 was previously declared the time master.

When transmitting 10 bytes through the controller, the date and time are transferred to the CP 526, and the CP 526 is reinitialized. If the CP 526 was previously the time master, and if this condition continues, the master identifier must be transferred once again with the data.

When the controller reads the date and time from the CP 526, the CP 526 always transfers 10 bytes to the data transfer area, thus allowing the controller to evaluate the master identifier of the CP 526.

After a cold restart, a CP 526 behaves like a time slave and updates its clock only if it receives a job request from the controller. If the CP 526 is then declared to be time master, it keeps the date and time ready so that it can transfer it as quickly as possible with a RECEIVE-DIRECT 218.

Please note that a PLC-SYNCHRON does not change the master identifier on the CP.

3.4   Interface to Keyboard and Programmer

You can connect either the process keyboard or a programmer (PG) to the CP 526
through the 15-pin female "D" type connector on the front panel. (Be sure to use
the correct connecting cable.) The CP 526 automatically recognizes which unit is
connected and automatically switches into the correct operating mode
("OPERATOR COMMUNICATION AND VISUALIZATION" or "PROGRAMMING"
mode).

Either in the "Configuration" as in the " Operator-Process Communication and
Visualization" mode, the data is transferred through a serial current-loop inter-
face (the process keyboard is passive). The power supply for the keyboard (24 V) is
taken from the controller rack power supply.

When you use standardized connecting cables, you do not have to worry about
the connector pin assignments. However, these diagrams are shown in section
3.10 if you wish to refer to them.

3.5   Memory Submodules

The CP 526 memory submodules contain all of the necessary display information
for, as well as the associated executive information. The Planning Guide
COM GRAPH contains specifications for the memory requirements of your
programs.

You can use either EPROM or RAM submodules. When a RAM submodule is used,
it is battery-backed through the backplane bus of the programmable controller.
If the RAM submodule or the entire printed circuit board is pulled out, the
application programs are lost.

3.6   Technical Specifications

3.6.1   Operating and Environmental Conditions

Degree of protection to DIN 40050/IEC 144   IP 00

Permissible ambient temperature

- operating (with forced ventilation)   0 to 55°C

- non operating   - 40 to + 70° C

Humidity class to DIN 40050   Code F, noncondensing

Operating altitude   Up to 1500 m above sea-level

3.6.2   Mechanical and Electrical Data

Weight   0.55 kg

Printed circuit board format   Double-height Eurocard
(160 mm x 233.4 mm)

Front panel width   20.32 mm

Backplane connector   ES 902, range 2, 48-pin

Front connector   Female multipoint conn., 15-pin
4 coaxial BNC connectors (color)
1 coaxial BNC connector
(monochrome)

Supply voltages   + 5 V, tolerance 5%
+ 24 V, tolerance + 25/-15%

Current consumption   + 5 V: 2,2 A
                                            + 24 V: max 40 mA[1]

Battery current (min. 2.7 V)   50 microamp typ.

3.6.3   Logic Data

Microprocessor   INTEL 80188
                                           5 MHz, CMOS

Memory configuration

- RAM   8 kbyte
- dual-port RAM   2 kbyte (1 page frame + interpro-
                                           cessor communication flags)
- character generator   8 kbyte (256 characters/symbols)
- display refresh memory   8 kbyte (characters and attributes)
- memory submodule   128 kbyte (maximum)
- operating system EPROMs   128 kbyte (maximum)

Serial interface   50 - 9600 bits/sec.
                                           TTY(20 mA), max. 1 km

Color CRT controller[2]

- principle   Character graphics
- display refresh frequency   60 Hz
- line frequency   30.87 or 15.44 kHz

---

[1]   The process keyboard draws an additional approximately 200 mA.

[2]   You can modify various parameters with COM (see also subsection 3.10.4.3) in
      order to adapt the display most conveniently to your monitor. Note, however,
      that COM does not support the checking of the symbol set construction. The
      monochrome module is assigned its parameters in a manner similar to the
      color version, but the lower resolution must be selected.

- display format	26 lines, 80 characters/line
- resolution	640 x 468 (234) pixels
- raster field	8 x 18 (9) pixels
- colors	8 colors each for the character foreground and background - black, red, green, blue, yellow, cyan, magenta, white
- gray shades	With the monochrome version of the CP 526, gray shades from 0 to 7 are uniquely assigned to the 8 color designations. This is explained in the following table (the brightness of the gray shades increases from top to bottom):

Color Designation	Gray Shade
black	0
green	1
red	2
yellow	3
blue	4
cyan	5
magenta	6
white	7

- character set	Standard (ASCII) with 7 nationalities + 128 symbols that can be loaded. Various national character sets can be loaded into the CP 526. Even the cyrillic character set can be loaded. In this case, Latin lowercase letters are replaced by cyrillic uppercase letters (see also 3.13.2.3).
- flashing	1 flashing frequency
- display mix	Priority-controlled combination from character and pixel graphics

Video Signals

- colors	Red, green, blue, and SYNC separately Red, green/SYNC, blue
- monochrome	Green/SYNC jointly Standard BAS signal on the coaxial jack of the front panel

Monitor connection	Triaxial cable max. 80 m (high resolution) max. 100 m (low resolution)

3.7   Installation Guidelines

3.7.1   Module Slots for the CP 526 in the Programmable Controller

The CP 526 modules can be plugged into the specified slots of the following programmable controllers :

Subrack/Controller	Slot-in module	Slot No.
PLC S5-115U		
CR 700-2 ER 701-3 CR 700-3	Basic module [1] Basic module [1][2] Basic module [3] Expansion board [4]	0, 1, 2, 3, 4, 5 0, 1, 2, 3, 4, 5,6 0a, 0b, 1a, 1b, 2a, 2b, 3, 4, 5 0b, 1b, 2b
PLC S5-135U		
ZG 135 U EG 185 U	Basic module [7] Basic module [7]	11, 19, 27, 35, 43, 51, 59, 67, 19, 27, 35, 43, 51, 59, 67, 75, 83, 91, 99, 107, 115, 123, 131, 139
PLC S5-150U		
ZG 150 U EG 185 U	Basic module [7] Basic module[7]	3 [5], 11 [5], 19 [5], 107, 115, 123, 131, 139 [6], 147 [6] 19, 27, 35, 43, 51, 59, 67, 75, 83, 91, 99, 107, 115, 123, 131, 139
PLC S5-155U		
ZG 155 U	Basic module [7]	19, 35, 43, 51, 59, 75, 83, 91, 99, 107, 115, 123, 131 139 [6], 147 [6]

[1]   Adapter casing is required! One module per casing.
[2]   Not with Interface Module 311!
[3]   In versions with natural ventilation, 1 module per slot only.
[4]   Only together with basic module in 1 casing.
[5]   Only in conjunction with jumpering module 756.
[6]   Only possible after changing jumpers on the bus PCB.
[7]   The expansion board is inserted into the next highest slot.

Table 3-1: Module Slots in the Programmable Controller

3.7.2   Inserting and Removing the CP 526

CAUTION !
Do not plug in or withdraw the CP 526 module when the power is on.

3.7.3   Installation Guidelines for Keyboard, Programmer and Monitor

You must take the following shielding measures to preclude any future problems with disturbance and faults in your programmable controller system.

If your system is badly grounded or not shielded, low-frequency and/or high-frequency noise signals may be coupled into the internal bus of the programmable controller. Under infavorable circumstances, this may cause the programmable controller to enter the STOP mode.

Noise can be caused, for example, by switching relays or contactors (high rates of current/voltage variation, HF disturbance) or by different Ground potentials between two parts of the system (NF disturbance).

The following installation guidelines cover the following areas:

- Operation and installation of noiseproof cables
- Noiseproof hardware configuration
    - S5 cabinet
    - Monitor connection

3.7.3.1   Operation/Installation of Noiseproof Cables

The shielding of all connecting cables between the CP 526 and the process keyboard/programmer must be grounded at both ends. The standard cables listed in Catalog ST80 and under ORDERING INFORMATION (section 2.4) meet this requirement.

Triaxial cables with double shielding are available for connecting the monitor in environments with severe EMI (electromagnetic interference). These can be used up to a length of 80 m without the need for any additional measures.

NOTE:
Do not run monitor cables or the connecting cables between the CP 526 and the process keyboard/programmer parallel to power cables. Use separate cable racks guaranteeing a clearance of at least 50 cm for the power cables.

Differences in the ground potential between subracks and the monitor housing can cause monitor "humming" (dark horizontal bars across the display). For this reason, and also to provide protection against accidental contact (shocks), run an equipotential bonding conductor between the S5 cabinet and the monitor housing. Design this equipotential bonding conductor on the basis of measurements taken in the system to limit the maximum possible potential difference to 200 mV. Connect the monitor and programmable controller to the same phase of the power system.

3.7.3.2   Noise-proof Hardware Packaging

The general hardware packaging of the control equipment is the basis for noise-proof operation of the programmable controllers. This means suppressing any noise signals emulating from the process.

Make sure that all the shielding measures etc. in the EMC guidelines "hardware installation guidelines or SIMATIC S5 (programmable controllers of the U range)" are observed. If these hardware installation guidelines are not observed, noise signals may propagate into the hardware and thus disturb program execution.

The main measures to be taken are described below.

S5 cabinet

Figure 3-14 gives an example of an S5 cabinet. As can be seen from the figure, devices that could cause noise signals to enter the cabinet from the outside are located at the bottom. The grounding bar must be installed directly at the point where the cable enters the cabinet to connect cables carrying noise signals, e.g. monitor cables, connecting cables for the process keyboard can be connected before they can carry noise signals into the cabinet. All shielded cables (with the exception of coaxial cables with <u>single</u> shielding) must be connected to this bar. In the case of signal cables with 2 shields, **only the external shield** need be connected.

Signal cables must also be run along the cabinet walls.

An EMC-compatible cabinet configuration is important for reducing noise. In this connection, make sure that all ground connections inside the cabinet make **large-area** contact.

CAUTION !
All analog devices in the cabinet must be isolated and grounded at a central point in the cabinet (use a copper strip).

All metal-to-metal connections must be clean (remove any paint). If necessary, use additional angle brackets for attaching the ground bars.
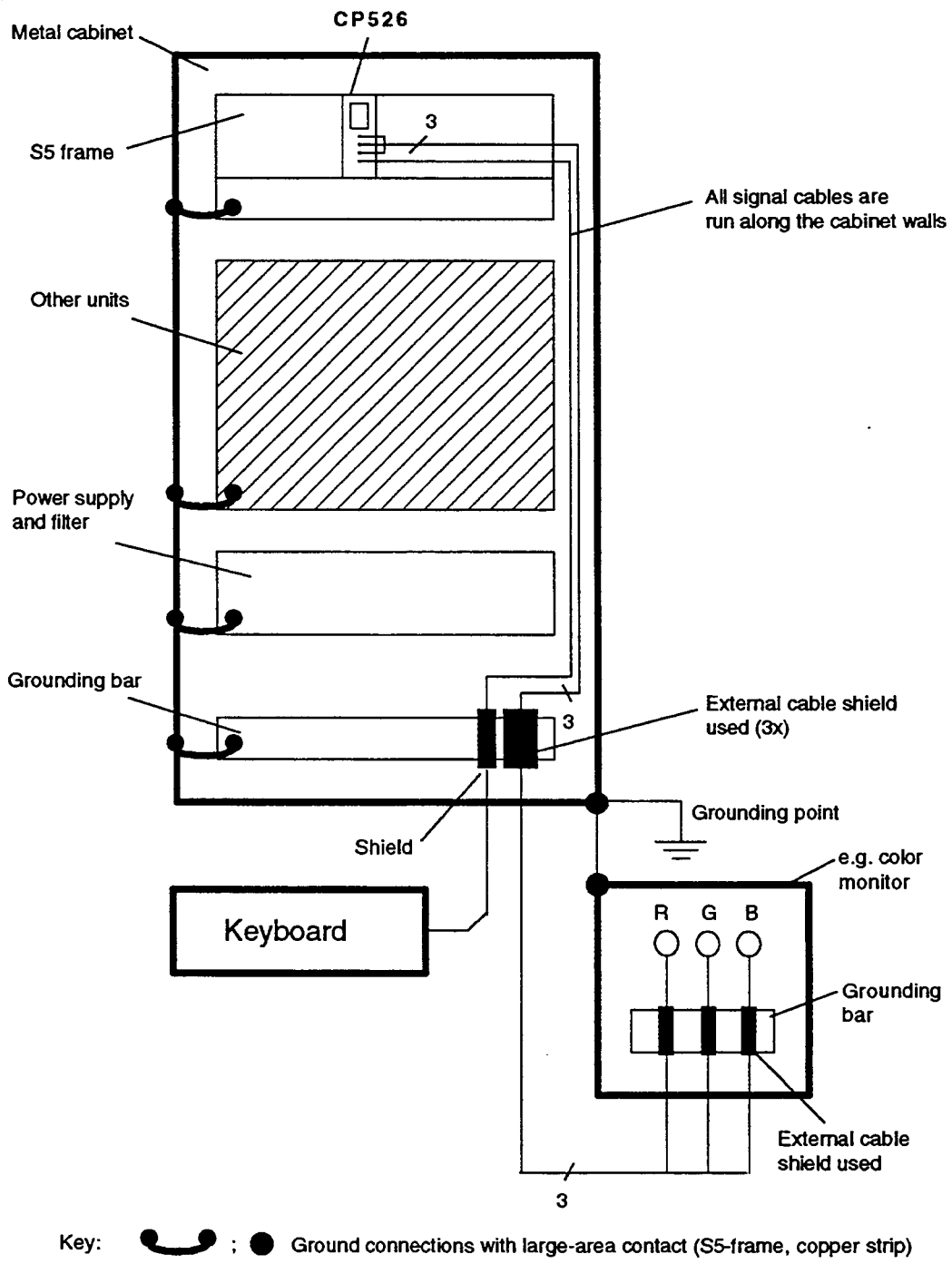
Figure 3-15: Cabinet Schematic

NOTE 1
In metal-to-metal connections always use metals of the same kind (never use aluminium because of the danger of oxidation).

All doors and sheet-metal parts (side panels, rear panel and roof) of the cabinet must be connected to the cabinet frame at least three times by short connectors (remove any paint and ensure large-area contact).

NOTE 2
In the case of plants where high electrostatic charges may be generated (e.g. textile machines, special construction machines), the grounding conductor of the **machine parts generating noise signals** should be run to a seperate system ground conductor isolated from the central grounding point of the S5 cabinet (grid ground with building structure, armouring).

Monitor connection

Coaxial cables with double shieldings should be used in severely contaminated environments. Only the **outer** shield of the monitor cable has to be connected to the frame potential of the process monitor (see Fig. 3-16). In the case of a number of Siemens process monitors, the metal strain-relief clamp can be used for the monitor cables. With other process monitors, on the other hand, a grounding bar must be bolted to the housing (accessory). Make sure that all metal-to-metal connections have a large contact area.

The electronic ground of the monitor (see Fig. 3-16) must be isolated from the frame ground. Jumpers are provided with the Siemens monitors in most cases.

The monitor and programmable controller should be connected to the same system phase.

When using monitors that can be operated only with a separate synchronizing signal, an extra cable (with cable clamp) must be used.

Special measures in connection with office monitors

Monitors with internally metallized plastics housings (office monitors) are not **suitable** for use in environments contaminated with electromagnetic interference since the internal metal surfaces cannot be connected later to the external earthing bar. It is not possible to isolate the electronic ground from the housing ground of the monitor in most cases as would be necessary in such environments. Such monitors can be used only in environments that are also suitable for personal computers.

Such office monitors can be used in connection with single-shielded coaxial cables. Connection to the CP 526/527/528 is therefore only possible under certain conditions since this cable is only suitable for short distances.

Disturbances can be avoided if the monitor and programmable controller are connected to the same reference potential or to the same system phase (via system filters). In the local range, this is only possible by running a system cable from the programmable controller to the monitor.

In connection with PLCs and similar automation equipment, the monitors should therefore always be connected to the system power socket in the S5 cabinet.
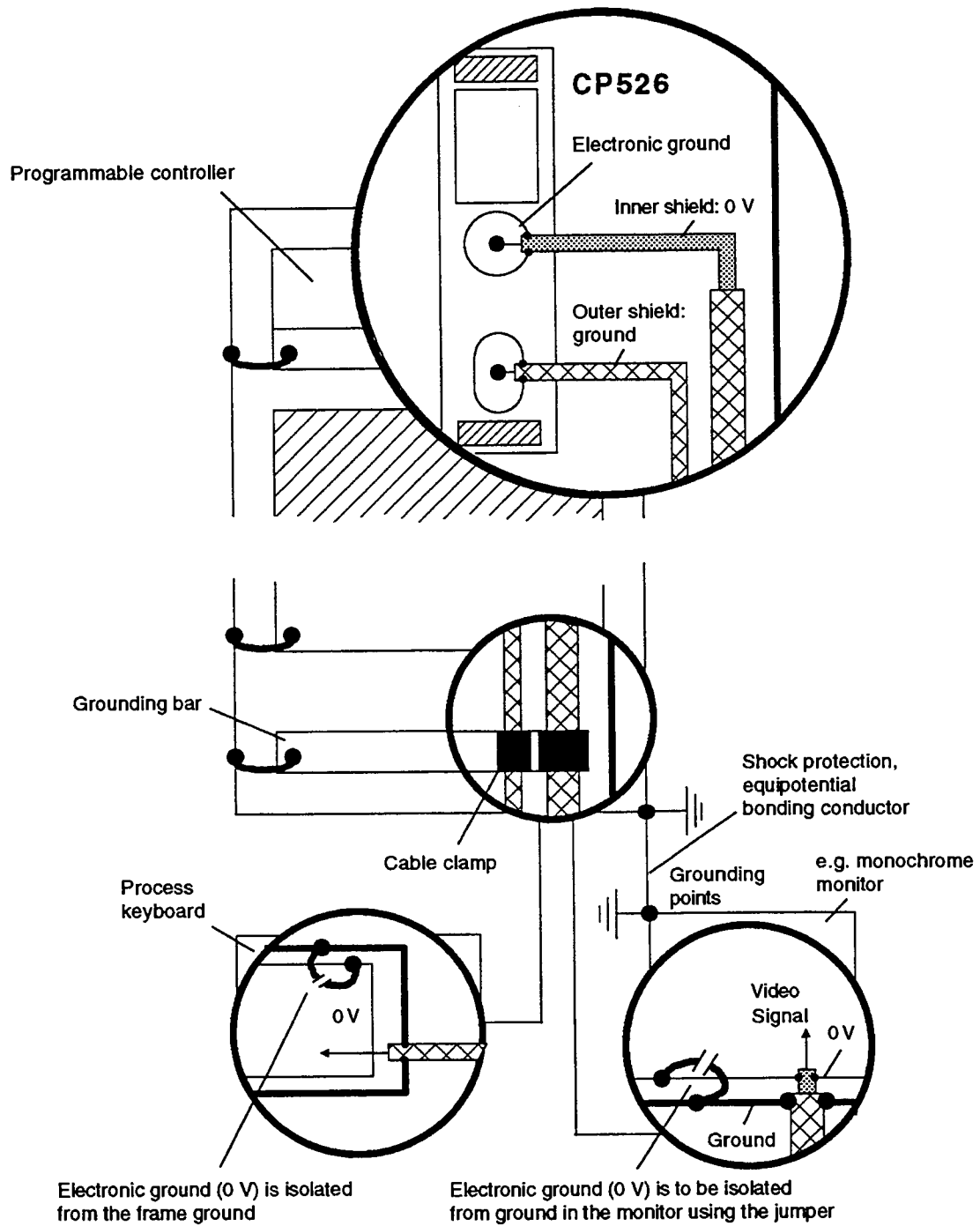
Figure 3-16: Cable Connections to Ground

3.8   Operation

This chapter describes the operator control and display elements, as well as the jumper settings.

3.8.1   Operator Controls and Displays

A switch and three LEDs are located on the front panel of the CP 526. The switch allows you to select the various operating modes, while the LEDs indicate the operating status of the CP 526.

3.8.2   Operating Modes

The switch on the front panel of the CP 526 controls the two operating modes, "STOP" and "RUN". The "RUN" mode allows normal operation. The "STOP" mode prevents data exchange with the programmable controller, even though all other functions are fully operational. For normal operation, this switch must be in the "RUN" position. When programming, the mode selector must be in the "STOP" position.

The device connected to the serial port determines whether the CP 526 selects "Programming" mode or "Operator-process Communication and Visualization" mode. When the process keyboard and/or a printer is connected, the "Operator-process Communication and Visualization" mode is selected. When a programmer (PG) is connected, "Programming" mode is selected. You can switch from "Programming" mode to "Operator-process Communication and Visualization" mode via the COM program.

3.8.3   Fault LEDs

The three LED indicators on the front panel provide information concerning the operating conditions of the CP 526. The following table describes the status and meaning of each combination of LED states:

RUN	STOP	CP Fault	Meaning
ON	ON	ON	Reset condition. This conditon occurs after power on or reset, for the duration of the reset.
ON	OFF	OFF	Normal operating condition. This condition is attained about three seconds after power on and after reset. (No information on the expansion board)
ON	OFF	ON	Break condition. This condition occurs if the cable of the programmer/keyboard serial communications link is interrupted. Exception: If, after resetting the PC, no connection is made to the programmer/keyboard.
OFF	ON	OFF	STOP condition. This condition occurs when the mode selector is in the "STOP" position.
OFF	ON	ON	Fault condition. This condition occurs if a fault is discovered during system startup (self-test). or: Operation mode switch on STOP and connection between programmer/keyboard and CP interrupted.
OFF ON OFF	OFF ON OFF	OFF OFF ON	Undefined states (module fault)

Table 3-2: Fault LEDs

3.8.4   Jumper Settings

This section summarizes the CP 526 jumper functions and settings.

In Figures 3-1 and 3-2, the jumpers and jumper headers are shown in the actual arrangement they have on the module.

The CP 526 has various types of jumpers (e. g. double and triple jumpers).

"X" identifies the jumpers and jumper headers.

The most significant jumper is marked by a " 1 ". Double jumpers are not specially marked.

Inserted jumpers are shown with infill.

NOTE:
The sketches show the jumpers as set in the factory.

3.8.4.1   X6: Enabling or Disabling Interprocessor Communication Flags

Jumper header X6 enables or disables the interprocessor communication flags.

1

| X | X | X | X | X | X | X | X |

Interprocessor Communication Flag Bytes	Address
224 - 255	F2E0 - F2FF
192 - 223	F2C0 - F2DF
160 - 191	F2A0 - F2BF
128 - 159	F280 - F29F
96 - 127	F260 - F27F
64 - 95	F240 - F25F
32 - 63	F220 - F23F
0 - 31	F200 - F21F
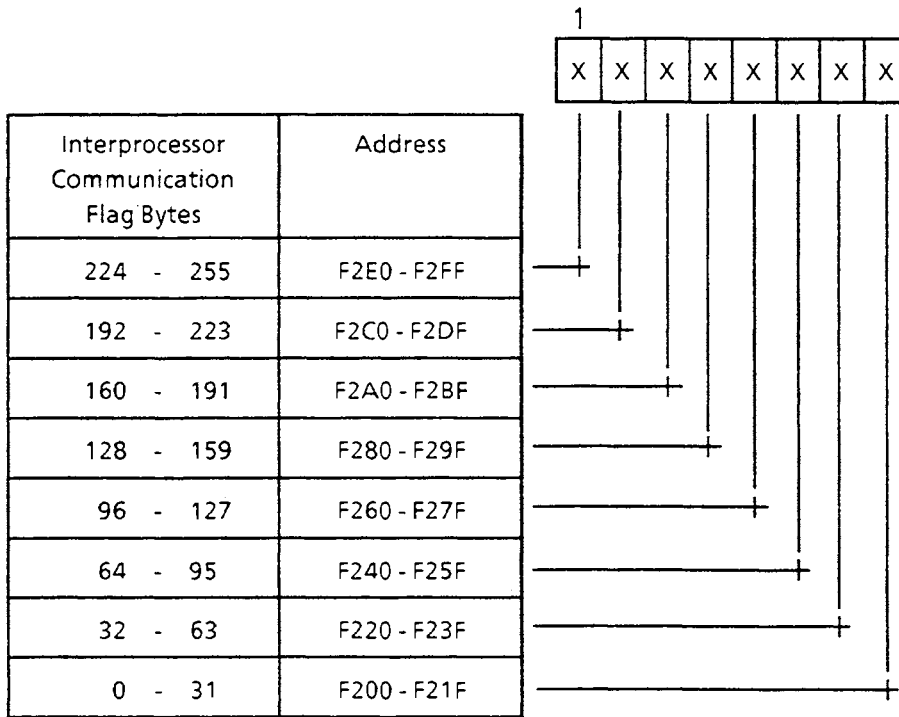
Remarks
- Jumper on = interprocessor communication flags enabled
- Jumper off = interprocessor communication flags disabled

3.8.4.2   X7: Module Address (Page Frame Number)

Jumper header X7 is used to set the page frame number or vector address (see subsection 3.3.2) for the CP 526. There are 256 (0 to 255) possible settings. Using this number, you can address the CP 526 with the data handling blocks (interface number SSNR).

1

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
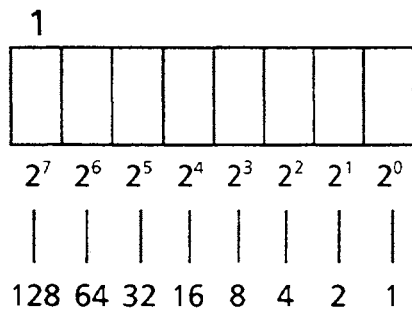
Remarks
- Jumper off = 0 (removed)
- Jumper on = 1 (inserted)

3.8.4.3   X8: Page Window Base Address

For the page window base address, the following assignment between binary weights and the jumpers of header X8 apply.

- Base window address = F * 1K, F = 0, 1, 2 to 63

- Factory setting for 61 k (F = 61)

1

XX	XX	XX	XX		XX
XX	XX	XX	XX		XX

2^5 2^4 2^3 2^2 2^1 2^0

| | | | | | |

32 16 8 4 2 1
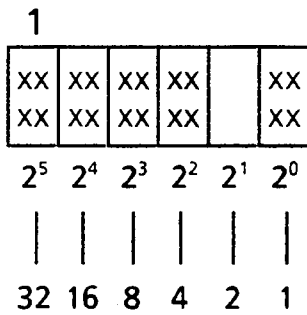
Remarks: - Jumper off = 0
             - Jumper on = 1

The data handling blocks supplied by SIEMENS are tailored to the factory settings.
The jumper settings made in the factory must therefore not be changed.

3.8.4.4   X9: Switching Between Page Frame/Linear Addressing

With jumper header X9 the page frame/linear addressing is selected:

Page Frame Addressing: jumper 1 - 2 = off   1
                           jumper 2 - 3 = on

                                                       X

Linear Addressing: jumper 1 - 2 = on
                           jumper 2 - 3 = off   X
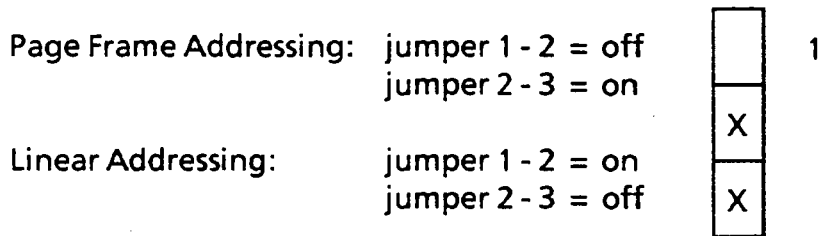
When using the data handling blocks, you must not change this jumper setting.

3.8.4.5   X10, X14: Jumper for Setting the Pixel Cycle

Jumper block X14 is used to specify the pixel cycle:

High-Resolution Monitor Jumper 1 - 2 = on Jumper 2 - 3 = off	Pixel cycle = 26.1818 MHz Controller freq. = 3.2727 MHz
Low-Resolution Monitor Jumper 1 - 2 = off Jumper 2 - 3 = on	Pixel cycle = 13.0909 MHz Controller freq. = 1.6364 MHz

1

X	X	
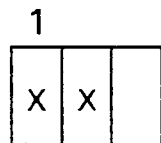
Jumper X10 is used to specify the type of video monitor: a high-resolution or a low-resolution monitor.

Jumper on = high-resolution monitor) | X |

Jumper off = low-resolution monitor | X |

CAUTION !

The setting of jumper X10 must correspond to the setting of jumper X14.

3.8.4.6   X11: Jumper for Setting the Firmware Mode

Jumper on	Module operates as a CP 526/2
Jumper off	Module operates as a CP 526/1

```
┌───┬───┐
│ X │ X │
└───┴───┘
```
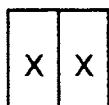
3.8.4.7   X17: Jumper for the SYNC Signal

Jumper header X17 is used to specify the channel on which the synchronization (SYNC) signal is output from the CP 526 monitor.

This jumper allows you to pass the synchronization signal on the green channel (RGB-BAS), or to pass it separately over its own cable. Please follow the operating instructions for your particular monitor.

A: Jumper 1 - 2 = on Jumper 2 - 3 = off	SYNC signal separate
B: Jumper 1 - 2 = off Jumper 2 - 3 = on	SYNC signal on green channel

```
 1
┌───┬───┬───┐
│   │ X │ X │
└───┴───┴───┘
```
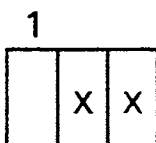
CAUTION:
In the case of the monochrome monitor, the SYNC signal is always the BAS (composite video signal) connection, so that jumper X17 must always be set to B.

3.8.5   Permanently Set Jumpers

All jumpers not mentioned in subsection 3.8.4 are irrelevant. These jumpers are set at the factory and should not be changed.

3.9   CP 526 Start-up

Use the following checklist when starting up your CP 526:

1.   Check the jumper settings on the CP 526 module.

2.   Make sure you have the correct EPROM firmware on your module:

     Location J33 No. S79200-G34-A905
     Location J39 No. S79200-G34-A906

3.   Turn off the power to the programmable controller and plug the CP 526 module into its slot.

4.   Insert a RAM memory module into the receptacle in the front panel of the CP 526 module.

5.   Check the position of the mode selector on the module front panel. The selector must be in the "RUN" position if you want to exchange data with the programmable controller.

6.   Connect up the process keyboard and turn the keylock switch to "ON".

7.   Connect the monitor. Make sure that the jumpers are properly set on the CP 526 for either high or low monitor resolution (X10, X14), and for proper channel specification of the synchronization signal (X17). Follow the operating instructions for your particular monitor.

8. Turn the programmable controller back on. The monitor displays the START command START[1]. Enter START from the process keyboard, and press the <RETURN> key. The CP 526 monitor redisplays the START command. Besides the message (top) and command (bottom) lines, the display contains the current system data.[1]

On the front panel, the red or green LED lights up, depending on the switch position. If this does not happen, check the jumper settings.

9. If everything is O. K., you can start normal operator-process communication procedures or generate new commands for your CP 526 with the programmer.

---

[1] On each cold restart, the CP 526 screens the display output command with the name "START." This display can also be called if no user card is plugged in or if an EPROM or a RAM card is plugged in that does not contain a command with the name "START."

If you want a different display on cold restart, you can program an arbitrary display output command with this name by means of COM, and transfer it into the user memory of the CP 526.

3.9.1   Connecting the Process Keyboard

The process keyboard is directly connected to the serial interface of the CP 526 by means of the appropriate connecting cable (see 'Ordering Data').

The connector required for the serial interface of the CP 526 is described in Table 3-6 (subsection 3.10.3).

The CP 526 recognizes that a process keyboard is connected and automatically switches into the 'OPERATOR CONTROL AND PROCESS MONITORING' mode. As a consequence, the transmission parameters of the interface are adapted. On the process keyboard itself no adjustments have to be made.

3.10   Connector Pin Assignments

The electrical interfaces of the CP 526 are specified in the following sections. These interfaces include the following:

- Backplane connector
- Submodule connector
- Expansion module connector
- Keyboard/programmer connector
- Video output connectors (see Figure 3-1)

3.10.1   Backplane Connector

The CP 526 is connected to the wiring backplane of the programmable controller through two 48-pin, range 2 backplane connectors. The following tables show the pinout of these two connectors.

	d	b	z
2	---	Ground	+ 5 V
4	UBATT	---	---
6	ADB 12	ADB 0	CPKL
8	ADB 13	ADB 1	/MEMR
10	ADB 14	ADB 2	/MEMW
12	ADB 15	ADB 3	/RDY
14		ADB 4	DB 0
16		ADB 5	DB 1
18		ADB 6	DB 2
20		ADB 7	DB 3
22	---	ADB 8	DB 4
24	---	ADB 9	DB 5
26	---	ADB 10	DB 6
28	/DSI	ADB 11	DB 7
30	---	---	---
32	---	Ground	---

Table 3-3: Backplane Connector 1

	d	b	z
2	---	Ground	+ 5 V
4	---	---	---
6	---	---	---
8	---	---	---
10	---	---	---
12	---	---	---
14	---	---	/NAU
16	---	---	---
18	---	---	---
20	/HOLDA	---	/DMAFA
22	/TxD	---	---
24	---	---	
26	---	RxD	---
28	---	---	---
30		Ground for 24 V	Ground for 24 V
32		Ground	+ 24 V

Table 3-4: Backplane Connector 2

3.10.2   Memory Submodule Connector

The CP 526 has a 48-pin, female multipoint connector for a RAM or EPROM memory submodule. The table below shows the pinout of this connector.

	c	b	a
1	SADB 12	Ground	+ 5 V
2	SADB 0	SADB 1	SADB 2
3	SADB 3	SADB 4	SADB 5
4	SADB 6	SADB 7	SADB 8
5	SADB 9	SADB 10	SADB 11
6	SADB 13	SADB 14	/R
7	/W	SDB 8	SDB 9
8	SDB 10	SDB 11	SDB 12
9	SDB 13	SDB 14	SDB 15
10	SDB 0	SDB 1	SDB 2
11	SDB 3	SDB 4	SDB 5
12	SDB 6	SDB 7	---
13	/CS1 submodule 1	/CS3 submodule 1	---
14	/CS2 submodule 1	/STBY; /CS4	---
15	UCMOS	---	---
16	+ 5 V	Ground	K5 (submodule 1)

Table 3-5: Memory Submodule Connector

3.10.3   Keyboard/Programmer Connector

The front panel of the CP 526 has a 15-pin female connector for a programmer. In "Operator Control and Process Monitoring" mode, this serial interface is used to connect a process keyboard (see also Figure 3-3).

Table 3-6 shows the pinout of the female connector.

The serial interface can only be operated in passive mode:

-    Passive mode
     The 20 mA loop current is supplied by the device connected.

Figure 3-18 (serial interface, passive mode) is a schematic diagram of the electrical connections of an external device.

1	GEXT (external ground)
2	TTY IN - (current output)
3	-
4	+ 24 V
5	jumper for process keyboard
6	TTY OUT + (current input)
7	TTY OUT - (current output)
8	GEXT (external ground)
9	TTY IN + (current input)
10	Ground for 24 V
11	20 mA power source of transmitter
12	Ground 0 V
13	20 mA power source of receiver
14	-
15	-

Table 3-6: Serial Interface

The serial interface may either be operated actively or passively:

- Active mode of operation
  The current loop of 20 mA is taken from the power source of the interface connection.

- Passive mode of operation
  The current loop is supplied by the unit connected

Figures 3-17 (serial interface, active) and 3-18 (serial interface, passive) show the electrical connection of an external device for both modes of operation in a diagrammatic representation.
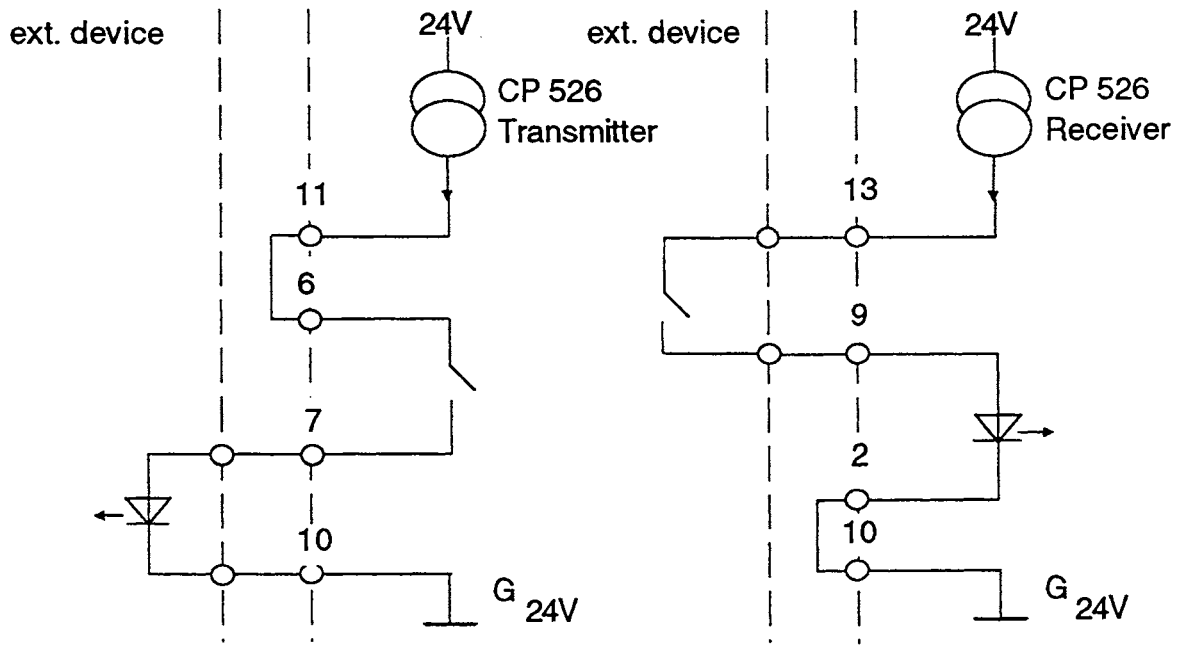
Figure 3-17: Serial Interface, Active
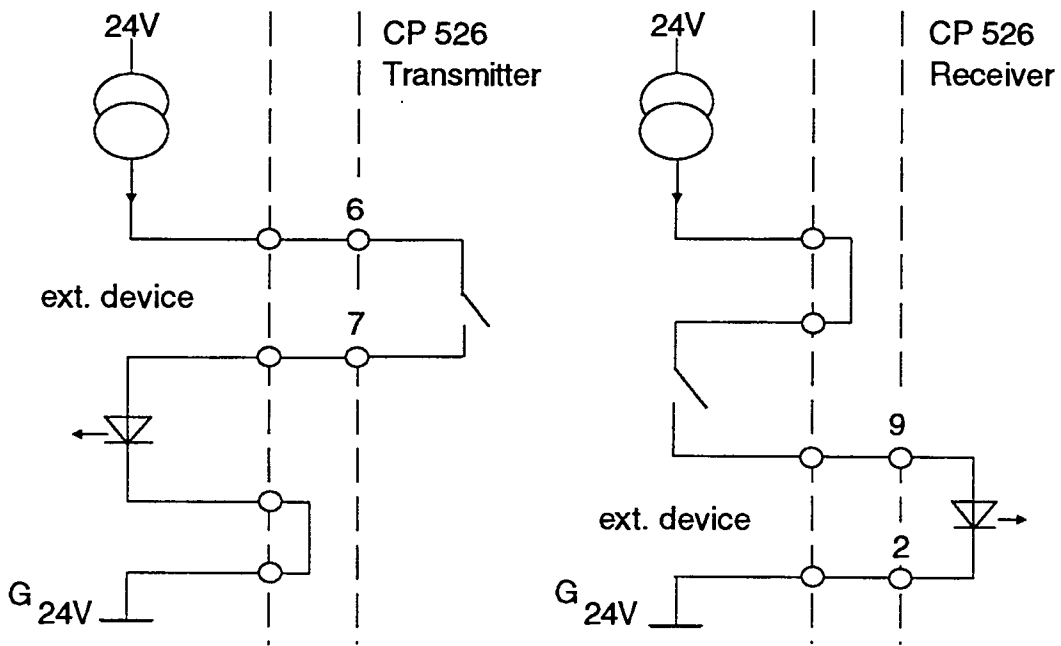
Figure 3-18: Serial Interface, Passive

3.10.4   Video Output Connectors

3.10.4.1 Color Monitor

The four coaxial sockets on the front panel of the CP 526 are used to connect the CP 526 to a video monitor via a triaxial cable with treble shielding. The connectors carry the following signals:

- Red signal   (R)
- Green or green and sync signal   (G) and (G/S)
- Blue signal   (B)
- Synchronization signal   (S)

The SYNC signal, consisting of the horizontal (HSYNC) and the vertical (VSYNC) synchronization signals, can be combined with the green channel.

The following signal voltage specifications are valid for the video signals:

- R, G, B   - level excursion of 0.7 V (75 ohms)
- S   - level excursion of 1.0 V (75 ohms)
- G/S   - level excursion of 0.3 V SYNC portion
                     and 0.7 V Green portion (75 ohms in each case)
- Output impedance: 75 ohms

CAUTION !
The coaxial sockets are not isolated!

3.10.4.2 Monochrome Monitor

In contrast to the color monitor, the monochrome monitor has only one connector for the BAS signal.

-   Level excursion: 0.3 V SYNC portion
                           0.7 V picture portion (75 ohms in each case)

-   Output impedance: 75 ohms

3.10.4.3 Monitor Connection

The CP 526 generates horizontal and vertical synchronization (HSYNC, VSYNC) pulses and a display enable (DISPEN) signal to control the monitor. The DISPEN signal blanks out the screen to obtain clear borders on the top, bottom, and sides.

The timing of these signals, and other properties of the monitor control, are controlled through hardware registers. To obtain an optimal display on the monitor, you can change these settings through the CP parameter assignment function (see Planning Guide COM GRAPH).

The starting point for the timing is the video control frequency (fc) or the time base (tc) = 1/fc, which corresponds to the duration of one character in the horizontal direction. By setting a jumper, you can configure the CP 526 for high- or low-resolution monitors. This changes the fc or tc as follows:

-   High resolution, where fc = 3.2727 MHz, and tc = 305.56 nsec.

-   Low resolution, where fc = 1.6364 MHz, and tc = 611.12 nsec.

Register 0 (R0) specifies the frequency of the horizontal synchronization pulses and, consequently, the line frequency.

Every monitor has a frequency range in which this frequency must be situated to obtain a stationary picture. The following relation is valid between the HSYNC frequency (fh) and R0:

$$fh = fc / (R0 + 1)$$
$$or \quad R0 = (fc / fh) - 1 \text{ (integer)}$$

Register 1 (R1) determines the number of visible characters in a line.

Register 2 (R2) sets the relative length of the HSYNC pulse in relation to the beginning of a line. Increasing this value causes the visible image to shift to the left; decreasing it causes a shift to the right. The following range is available:

$$(R0 + 1) > R2 > R1$$

Register 3 (R3) defines the length of the HSYNC and VSYNC pulses. Here, the left hexadecimal number concerns VSYNC; the right hexadecimal number concerns HSYNC.

The length of HSYNC is specified in multiples of tc; the length of VSYNC is specified in multiples of th (th = 1/fh).

Registers 4 and 5 (R4, R5) set the frequency of the vertical synchronization pulses and the frame frequency (fv). You must adhere to the requirements of the particular monitor (50 or 60 Hz) to get a stationary picture. R4 is understood as a multiple of tzz (time duration of a character line); R5 is understood to be a multiple of th. R4 acts as a coarse adjustment; R5 acts as a fine adjustment.

Register 6 (R6) determines the number of visible lines in the picture.

Register 7 (R7) defines the relative length of the vertical synchronization pulse. The unit is the character line. Increasing this value causes the display to move up on the monitor; decreasing it causes a downward shift.

Range: $(R4 + 1) > R7 > R6$

Register 8 (R8) defines the line jump procedure:

R8 = 00: no line jump
R8 = 01: line jump

Register 9 (R9) determines the number of raster lines per character line.

R9 = raster lines per character line - 1

Low resolution: R9 = 8 (08 hexadecimal)
High resolution: R9 = 17 (11 hexadecimal)

Registers 10 and 11 (R10, R11) set the shape of the cursor and the flashing rate. The number of the first raster line that is to be unblanked should be entered in R10; the number of the last raster line that is to be unblanked should be entered in R11. (Character size = 8 x 18)

		Character line	0
	0	Character line	0
	1	Character line	1
	2	Character line	2
	3	Character line	3
	4	Character line	4
	5	Character line	5
	6	Character line	6
	7	Character line	7
	8	Character line	8
	9	Character line	9
	A	Character line	10
	B	Character line	11
	C	Character line	12
	D	Character line	13
Cursor	E	Character line	14
Cursor	F	Character line	15
Cursor	10	Character line	16
	11	Character line	17

Register 10

```
| 0 | B | P | 0 | 1 | 1 | 1 | 0 |
```
Cursor
Start (0EH)

Register 11

```
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
```
Cursor End
(10H)

B P
0 0: cursor visible
0 1: cursor invisible
1 0: flashing, 1/16 frame frequency
1 1: flashing, 1/32 frame frequency

Figure 3-19: Monitor Register Parameter Assignment

Summary of Abbreviations

fc: controller frequency (base frequency)
tc: 1/fc, character duration, horizontal

fh: horizontal synchronization frequency (fh = fc / (R0 + 1))
th: 1/fh, duration of one raster line

tzz: time required to write one character line

$$tzz = th * (R9 + 1)$$

fv: frame frequency (frequency of the vertical synchronization
     pulses)

Reg. #	Description	Unit	Maximum
R0	Horizontal total[1] *	Character (tc)	255 (FF H)
R1	Horizontal visible	Character (tc)	255 (FF H)
R2	Position HSYNC[1] *	Character (tc)	255 (FF H)
R3	Length VSYNC/HSYNC	Raster line / character	15 / 15
R4	Vertical total[1] *	Character lines (tzz)	127 (7F H)
R5	Vertical total (fine)	Raster lines (th)	31 (1F H)
R6	Vertical visible	Character lines (tzz)	127 (7F H)
R7	Position VSYNC[1] *	Character lines (tzz)	127 (7F H)
R8	Line jump	Raster lines (th)	31 (1F H)
R9	Raster lines / char.[1] *	Raster lines (th)	127 (7F H)
R10	Cursor start / blinking	Raster lines (th)	31 (1F H)
R11	Cursor end		

* Value to be written in = calculated value - 1

Table 3-7: Overview of Registers

	High Resolution 60 Hz	High Resolution 50 Hz	Low Resolution 60 Hz	Low Resolution 50 Hz
R0	69 H	68 H	69 H	69 H
R1	50 H	50 H	50 H	50 H
R2	58 H	58 H	58 H	58 H
R3	65 H	65 H	A7 H	A7 H
R4	1B H	21 H	1B H	21 H
R5	0A H	0B H	05 H	03 H
R6	1A H	1A H	1A H	1A H
R7	1A H	1D H	1A H	1D H
R8	00 H	00 H	00 H	00 H
R9	11 H	11 H	08 H	08 H
R10	6E H	6E H	67 H	67 H
R11	10 H	10 H	09 H	09 H

Table 3-8: Monitor Register Standard Assignment

Determining the Most Important Register Contents

If the standard settings for the registers do not produce a stable picture on the monitor, try to redefine the register contents. To obtain a stable picture on the monitor, you should use the following algorithm:

● Input quantities include the following:

- set by the hardware
  fc (controller frequency)
  R9 (raster lines per character line - 1)

- monitor requirements
  fh.theoretical (line frequency)
  fv (picture change frequency)

● To improve the picture, set these quantities:

  - R0, R4, R5

    a)   fc/fh.theoretical   → auxiliary quantity h0
         The controller frequency divided by the theoretical horizontal
         frequency.

    b)   rounding off h0   → auxiliary quantity r0
         The auxiliary quantity r0 is obtained by rounding off h0.

    c)   r0 - 1   → R0
         The auxiliary quantity r0 decremented by 1 yields R0.

    d)   fc / r0   → fh (actual value)
         The actual value of the horizontal frequency is derived by dividing
         the controller frequency by the auxiliary quantity r0.

    e)   fh / fv   → h4
         The auxiliary quantity h4 is derived by dividing the horizontal
         frequency by the frame frequency.

    f)   rounding off h4   → n-th raster
         By rounding off h4, you can derive the auxiliary quantity n-th raster,
         which specifies the number of raster lines in a picture.

    g)   n-th raster DIV (R9 + 1)   → r4
         The abbreviation DIV symbolizes an integer division in which
         numbers after the decimal point are truncated.

    h)   r4 - 1   → R4
         By reducing the auxiliary quantity r4, you can derive the value to be
         entered for register 4.

    i)   n-th raster MOD (R9 + 1)   → R5
         The abbreviation MOD symbolizes a modulo operation (only the
         remainder of the integer division is used).

The following example shows how some of these formulas can be applied:

f_c = 1.6364 MHz
R9 = 8 (low-resolution monitor)

f_h.theoretical = 15.625 kHz
f_v = 50 Hz

a) 1.6364 MHz / 15.625 kHz   - 104.7296
b) rounding   - 105
c) subtracting 1   - 104 (R0 = 68 hexadecimal)
d) 1.6364 MHz / 105   - 15.585 kHz
e) 15.585 kHz / 50 Hz   - 311.7
f) rounding   - 312
g) 312 DIV 9   - 34
h) subtracting 1   - 33 (R4 = 21 hexadecimal)
i) 312 MOD 9   -   6 (R5 = 06 hexadecimal)

If the picture flickers, try to use a neighboring number in step b or f. Perform the subsequent steps again.

Influence of the Registers on the Timing

This section explains the relation between the registers and the horizontal and vertical timing.
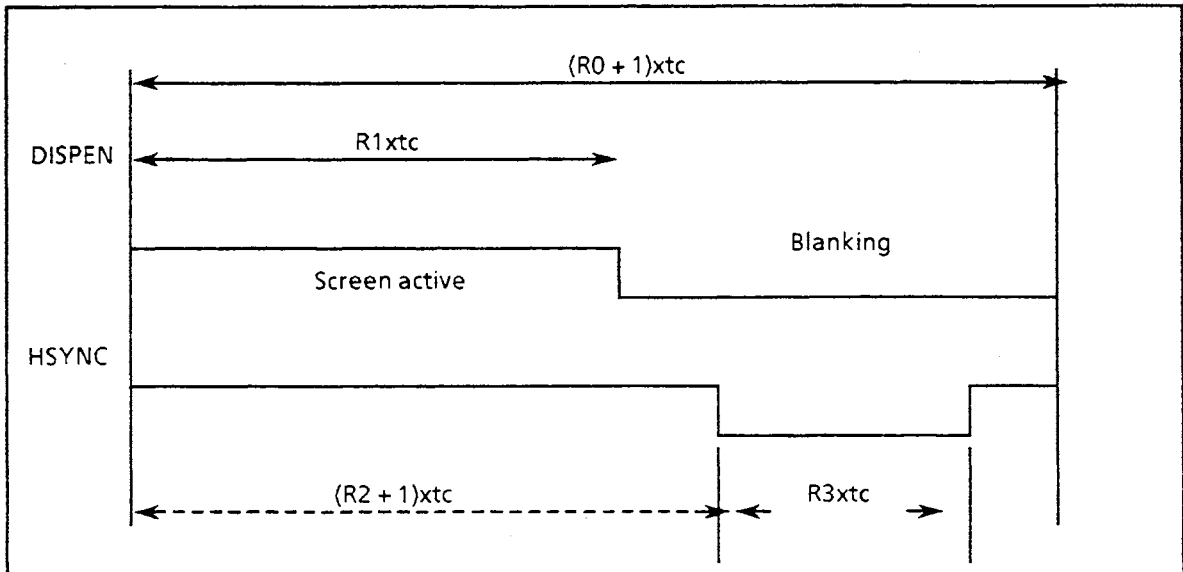
Figure 3-20: Horizontal Timing

The CP 526 software can write characters into the picture buffer memory only during the blanking time. The blanking time must last at least 8 microseconds.
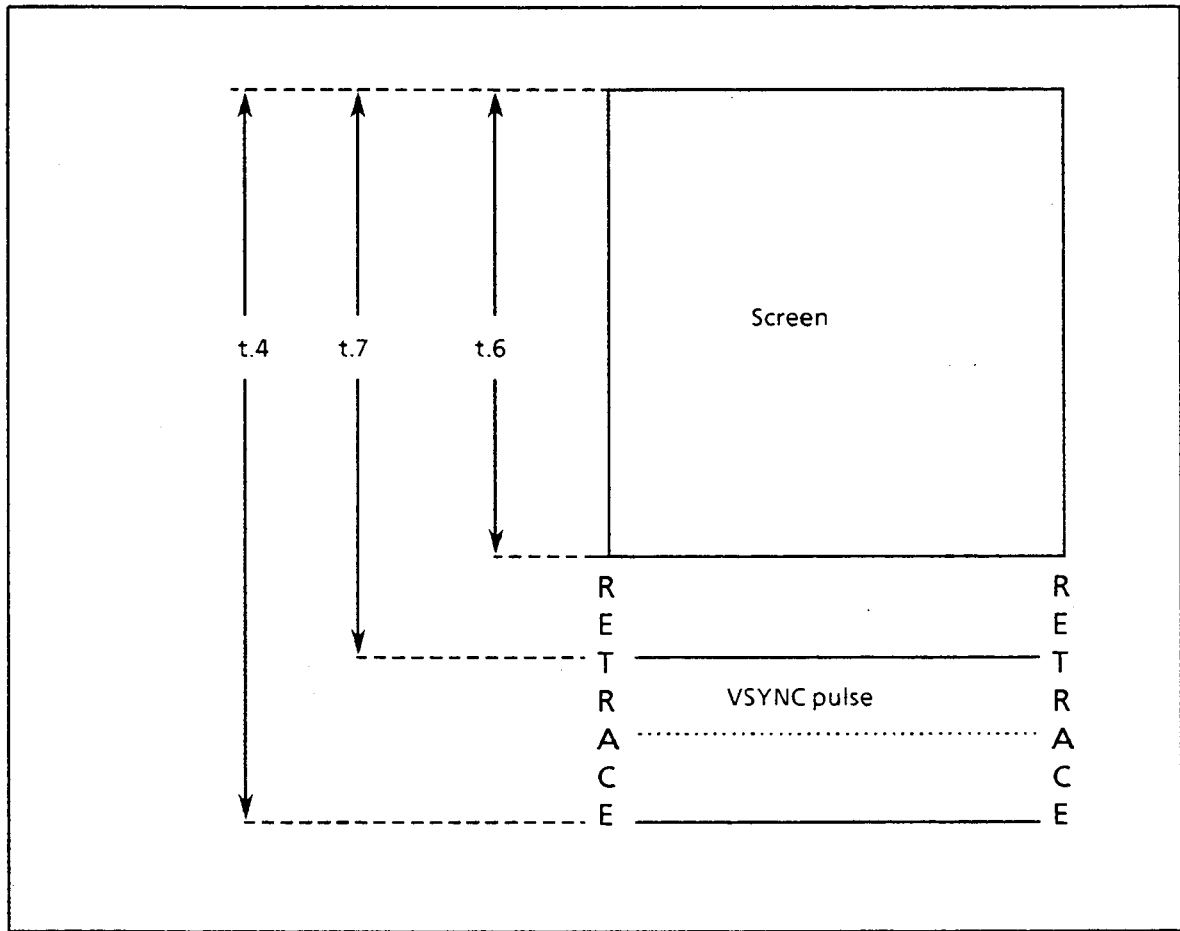
Figure 3-21: Vertical Timing

Explanation of Abbreviations

The frame frequency (fv) is defined by t.4: fv = 1/t.4.
R4 here acts as a coarse adjustment, while R5 acts as a fine adjustment.

- t.4 = (R4 + 1) * tzz + (R5 + 1) * th
- t.6 = R6 * tzz - The number of visible character lines is stored in R6.
- t.7 = (R7 + 1) * tzz - The position of the VSYNC pulse is specified by t.7.

3.11   Process Visualization

3.11.1   Types of Displays

The CP 526 allows you to visualize a process in graphics form. To do this, the CP 526 contains a basic alphanumeric character set (which can be switched over to country-specific variants) and another loadable symbol set that contains 128 characters. The COM diskette contains a standard character set for these 128 characters. In addition, COM GRAPH allows you to generate new character sets or change symbol sets by simple means.

Depending on the type of video monitor you are using, you receive a monochrome or a color display. On color monitors, you can specify the foreground and background color for each separate character. An eight-color spectrum (black, white, red, cyan, blue, yellow, green, magenta) is available for this purpose. In addition, you can make the individual characters flash.

3.11.2   Screen Format

The screen has 24 lines and 80 columns, which means that each line of the screen can have up to 80 characters of text and/or graphics symbols.

The display screen is divided into three areas:

● 22 lines for the actual process display

● A message line (usually at the top of the display, but you can move it or suppress it)

● A command line (usually at the bottom of the display, but you can move it or suppress it)

The lower part of the display (outside the 24 lines) contains the function keys with the functions currently assigned to them.

The figure below shows the standard format of the monitor screen.

Message line (independent of the screen display)		

22 lines

Command line

16 function keys — F1 to F8 / F9 to F16
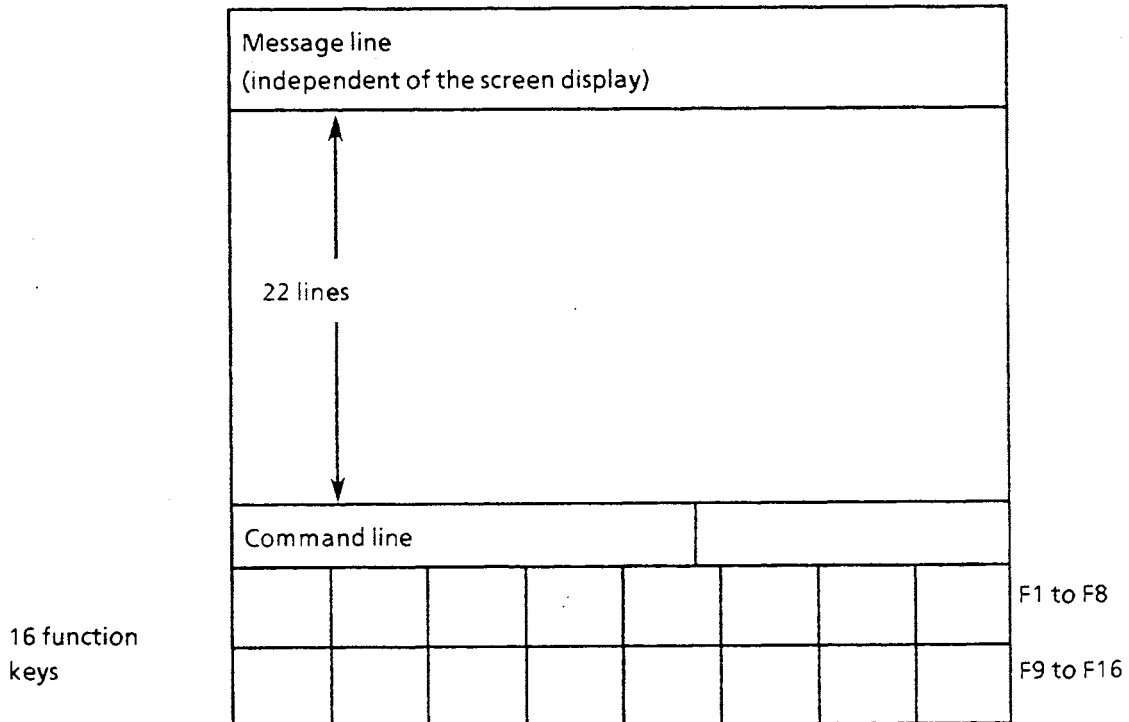
Figure 3-22: Standard Format of the Monitor Screen

3.11.3   Display of Process Data

Data from the controlled process can be visualized in the process display. This data is in the programmable controller data blocks in all known SIMATIC S5 formats (see table 3-9). Process data is displayed in two different modes:

● "DIGITAL" (as a number)
● "ANALOG" (as bars)

Table 3-9 details the different CP 526 display modes.

Format in the Controller	Screen Format	Range of Values	1)			2)		3)	
			F	B	C	d	i	d	i
Floating-point number GPT 32 (32 bits)	Numeric string with max. of 12 places before decimal point (incl. sign) and max. of 10 places after decimal point (incl. decimal point)	-2.147.483.647 to 2.147.483.647	x	x	x	x	x	x	x
Fixed-point number FPT 16 (16 bits)	Numeric string with max. of 12 places including signs	-32 767 to 32 767	x	x	x	x	x	x	x
	Horizontal or vertical bars	Screen display	x	x	x	x	x	x	x
Fixed-Point no. FPT 32 (32 bits)	Numeric string with max. of 12 places including sign	-2.147.483.647 to 2.147.483.647	x	x	x	x	x	x	x
BCD number BCD 16 (16 bits)	Numeric string with max. of 12 places including sign	-999 to 999	x	x	x	x	x	x	x
BCD number BCD 32 (32 bits)	Numeric string with max. of 12 places including sign	-9 999 999 to 9 999 999	x	x	x	x	x	x	x
Byte (8 bits)	Num. string with max. 12 plac.	0 to 255	x	x	x	x	x	x	x
Time TIMER (16 bits)	Field with four places (the decimal point is set automatically so that the time is given in seconds)	0.01 to 9990	x	x	x	-	-	-	-
Count COUNTER (16 bits)	Field with a maximum of 12 places	0 to 999	x	x	x	-	-	-	-
Character string	Char. string with max. 80 char.	ASCII char. set 20H to 7E	x	x	x	-	-	-	-
Binary (1 bit)	Individual ASCII character	0 or 1	x	x	x	-	-	-	-

1) Attribute
F = Foreground color
B = Background color
C = Flashing on/off

2) Scaling   3) Offset
d = Direct specification of scaling/offset
i = Indirect specification (DB address) of scaling/offset

Table 3-9: CP 526 Output Format

NOTE:
The term "fixed-point number" describes an integer with a sign, but no decimal places. With COM GRAPH programming, however, you can specify decimal places for the formatted representation.

In the case of a TIMER word, the time (0 to 999) is in BCD and transferred to the data block (time base in bits 13 and 14).

A SIMATIC floating-point number is represented with a 24-bit mantissa. With large values, inaccuracies of up to $0.6 * 10^{-7}$ in the representation occur relative to the initial value.

Example:
With the number 999,999,872, the least significant bit of the mantissa has the value 128, which means that there is no distinction in the representation of numbers within the range $999,999,872 \pm 64$.

3.11.4   Display of Process Statuses

A process status can be, for example, a plant operating mode or the switching
state of an actuator. The CP 526 can display such processes by the following
means:

- Color change of symbols or groups of symbols (line diagrams); for example,
  the display of a valve symbol in green for open and in red for closed

- Exchange of symbols or symbol groups (e.g., the symbol open switch with the
  symbol closed switch)

- By flashing symbols, symbol groups, numbers, or lines.

The update cycle can be set individually for each output field. Your specific
application determines the best means to implement these capabilities. If you are
not sure of the best implementation, take your time and experiment a little with
different update cycles. COM supports you and helps you to avoid programming
errors.

3.12   Operator-Process Communication

During normal operation, the entering and monitoring of process data is
performed via the process keyboard. This keyboard is especially tailored to
process control and has the following features:

- IP 54 degree of protection
- Standard keyboard field
- Control key block
- Standard key block for numerical inputs
- Function key block
- Keylock switch

Operator-process communication is described below, with reference to the
keyboard shown in Figure 3-23.
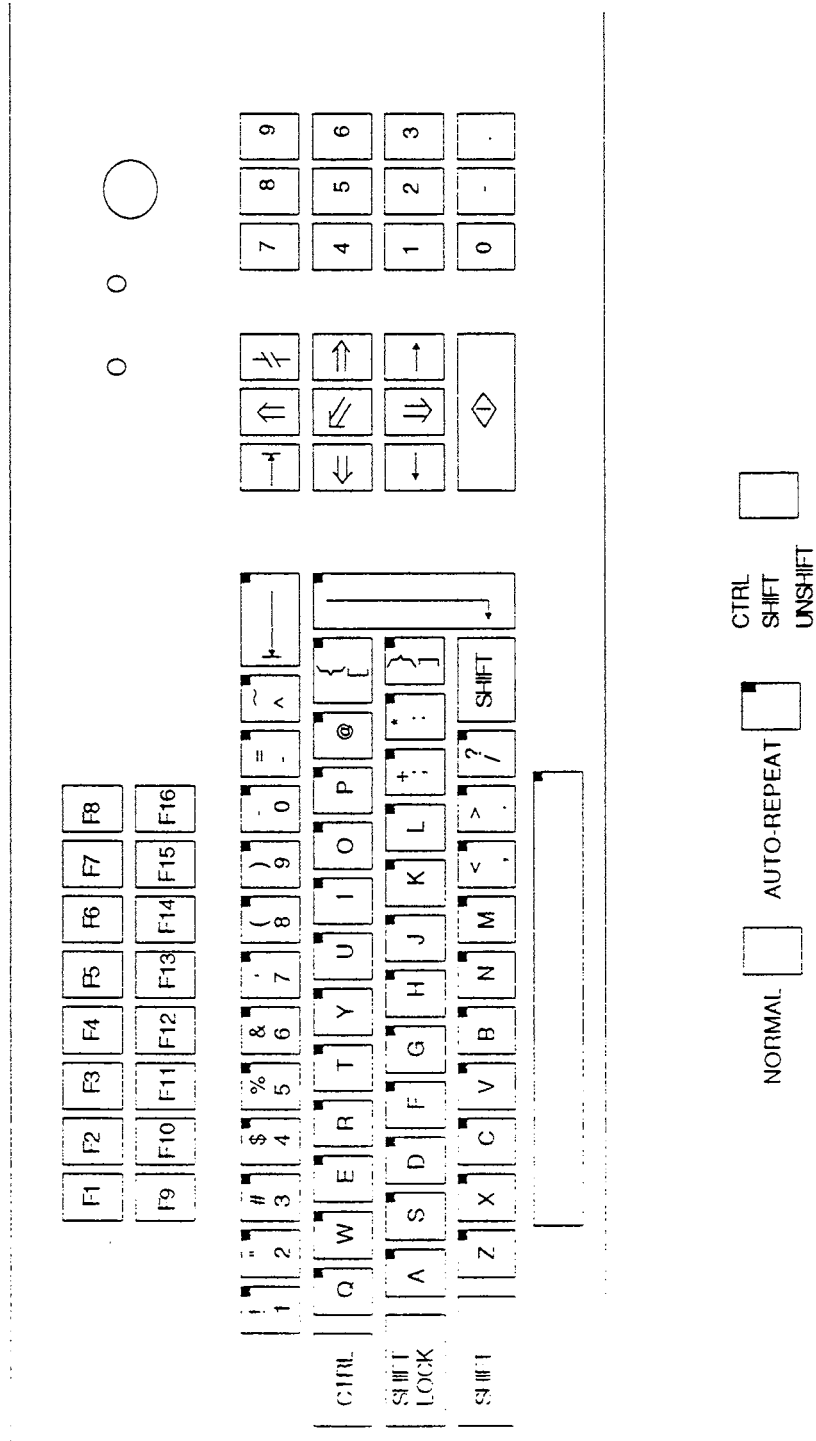
Figure 3-23: Key Layout for the Large Process Keyboard
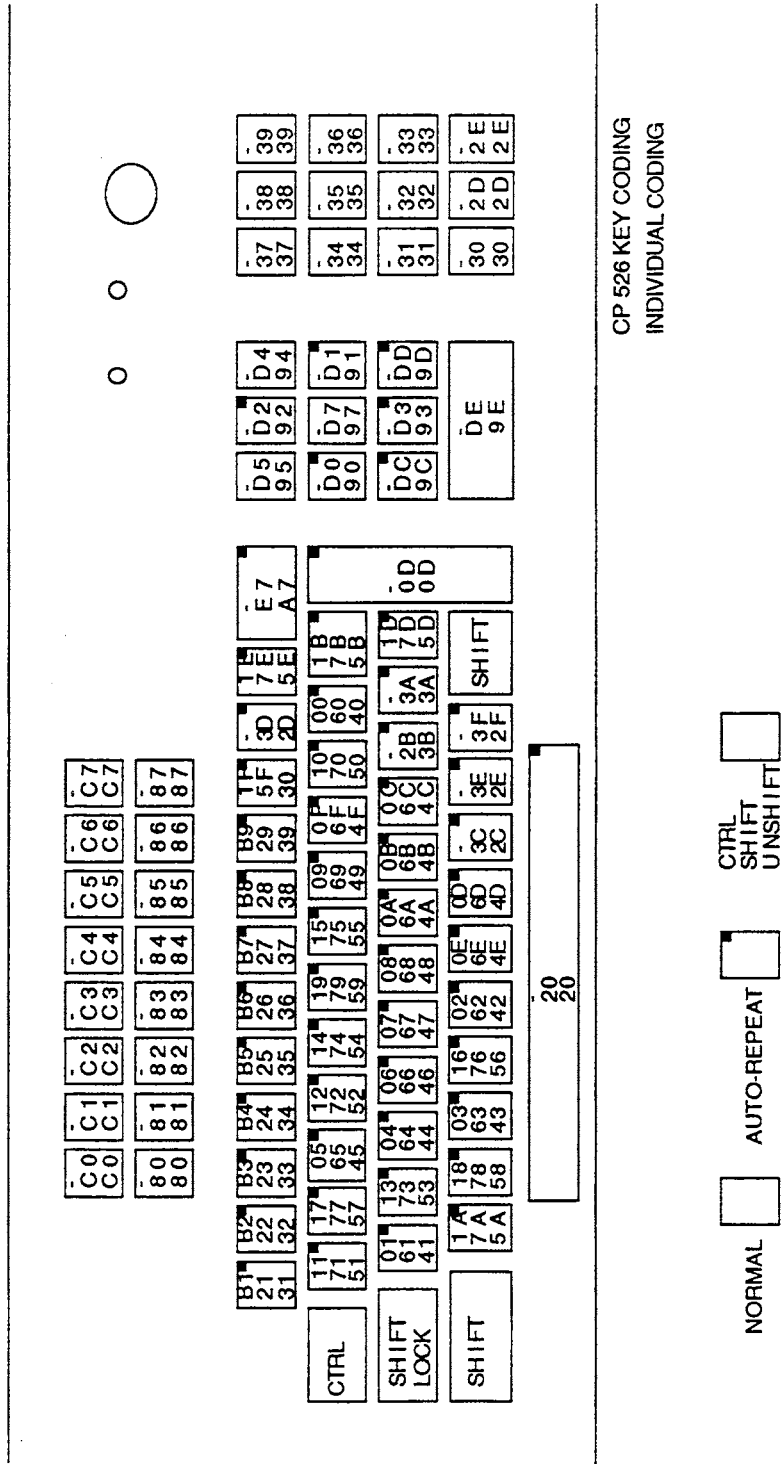
Figure 3-24: Key Codes for the Large Process Keyboard
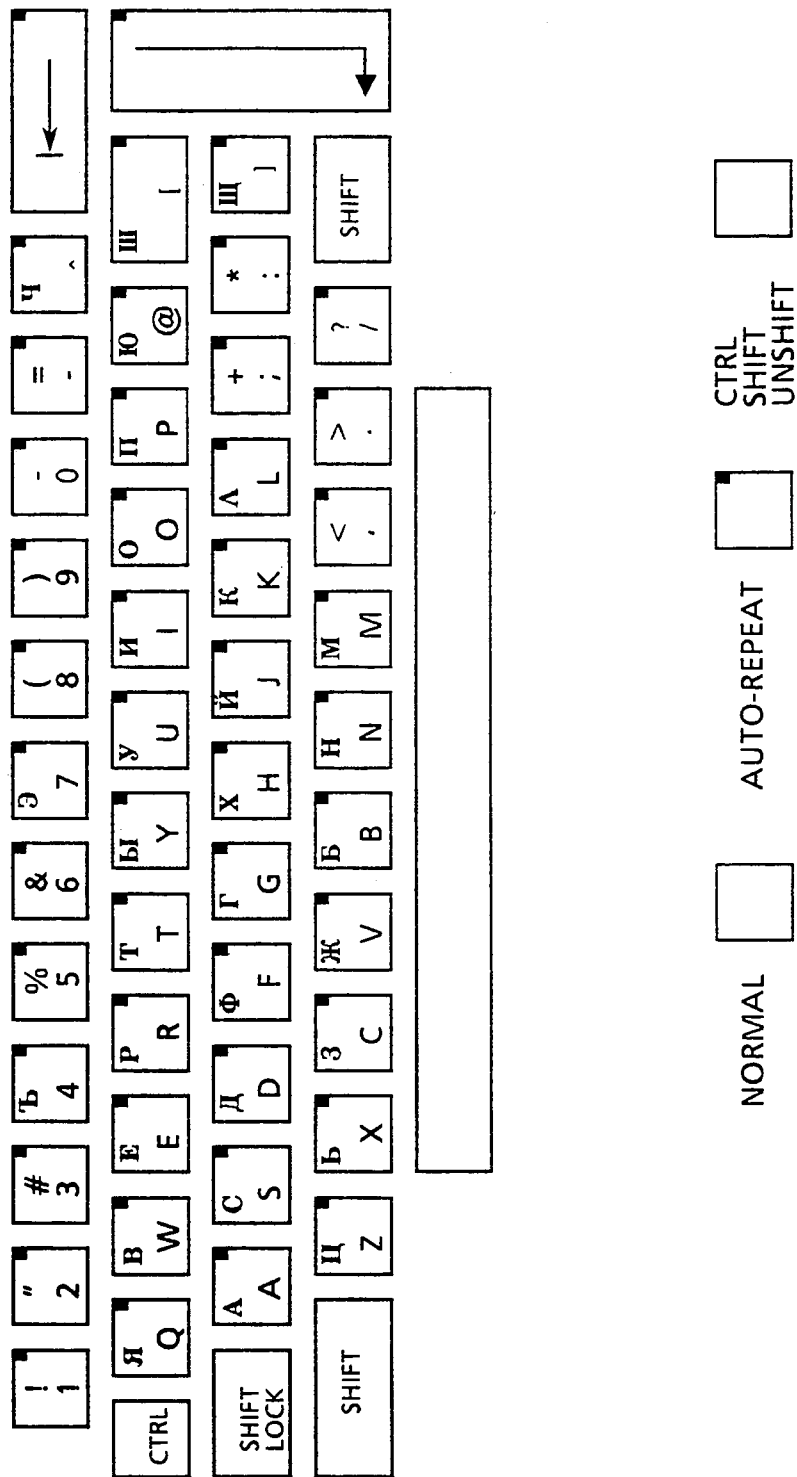
Figure 3-25: Cyrillic Process Keyboard (Section)

0		0
1		1
2		2
3		3
4		4
5		5
6		6
7		7
8		8
9		9
A		A
B		B
C		C
D		D
E		E
F	produces	F
G		G
H		H
I		I
J		J
K		K
L		L
M		M
N		N
O		O
P		P
Q		Q
R		R
S		S
T		T
U		U
V		V
W		W
X		X
Y		Y
Z		Z

		A		А
		B		б
		C		з
		D		Д
		E		Е
		F		Ф
		G		Г
		H		Х
		I		Й
		J		Й
		K		К
		L		Л
		M		М
		N		Н
SHIFT	+	O	produces	О
		P		П
		Q		Я
		R		Р
		S		С
		T		Т
		U		У
		V		Ж
		W		В
		X		Ь
		Y		Ы
		Z		Ц
		7		Э
		^		Ч
		[		Ш
		]		Щ
		4		Ъ
		@		Ю
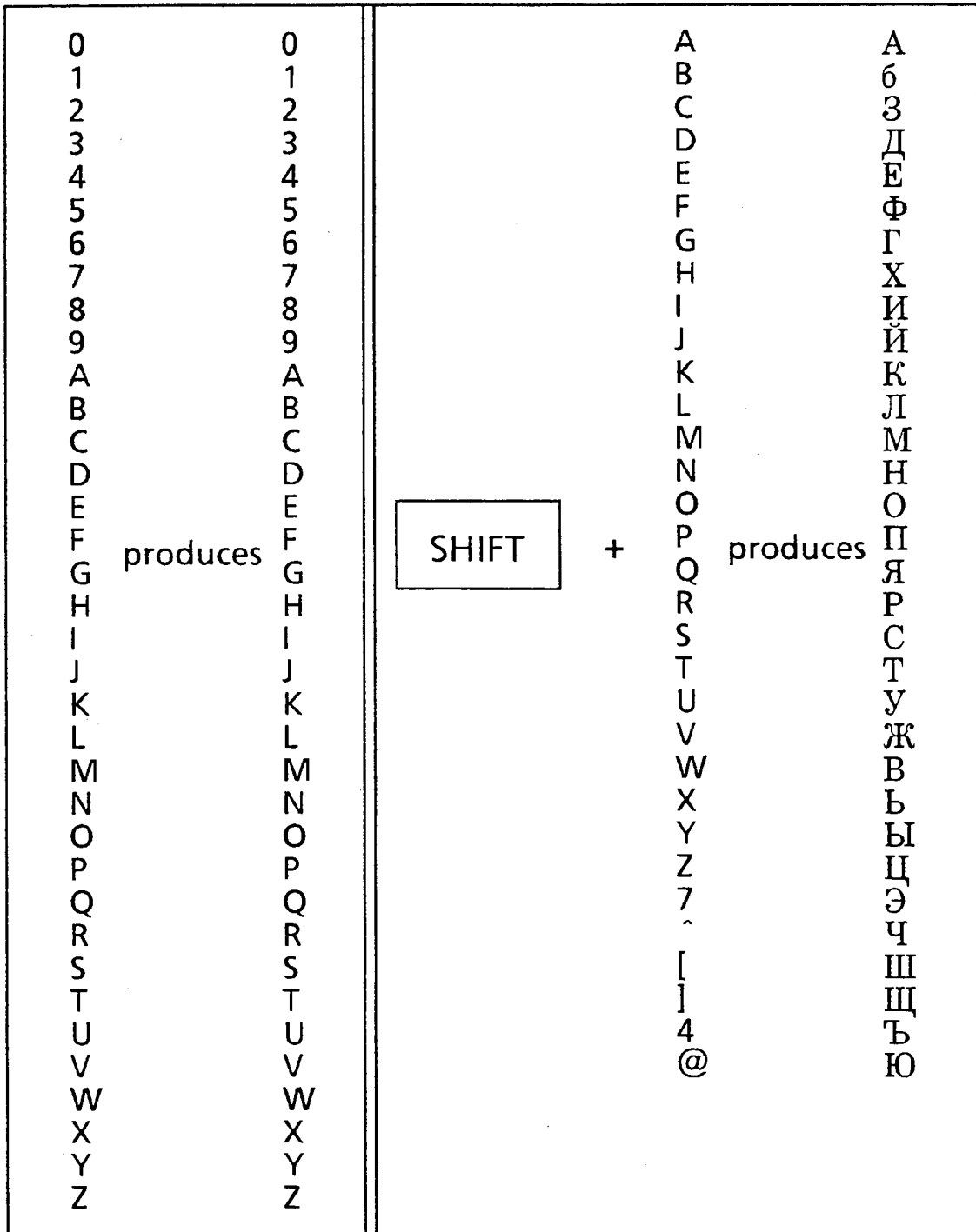
Figure 3-26: Key Assignments for Cyrillic Keyboard
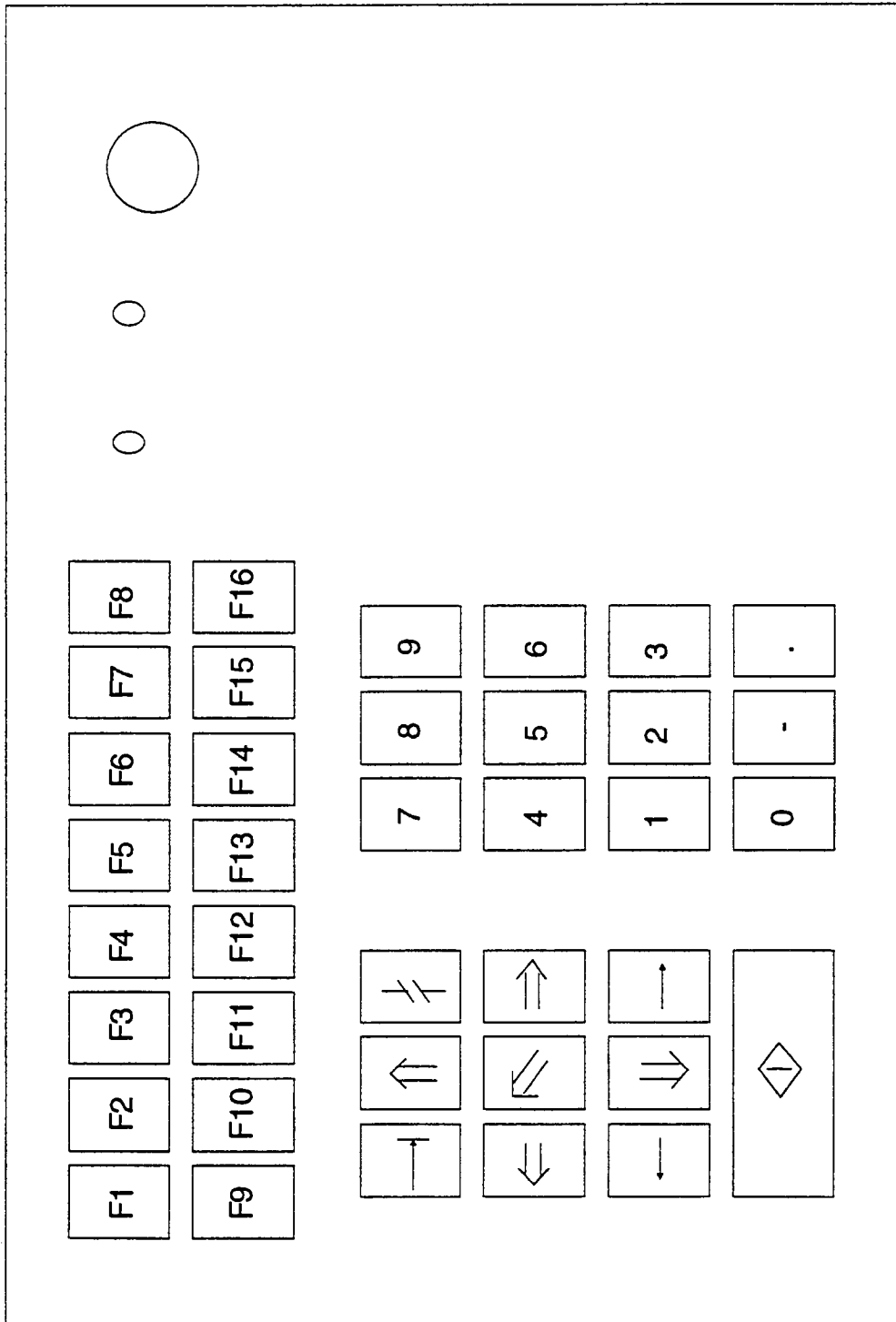
Figure 3-27: Key Layout on the Small Process Keyboard

3.12.1   Command Line

The command line allows you to enter commands to control the process and invoke process displays.

The command line has two parts:

● Input - left half of the command line.

   Enter the name of the appropriate command here. When you press the <CMD> key, the cursor is positioned in the input part of the command line from an arbitrary point in the display. Then enter the name of the command and press the <RETURN> key or the Enter key.

● Output - right half of the command line.

   The CP 526 displays error messages[1] here, e.g. when a command you have entered is unknown (operator error or a CP fault).

A typical user command is to call up a new process display. There are other types of commands (such as entering values); however, these do not alter the current display on the screen (refer to Planning Guide COM GRAPH).

---

[1]   Note that the error texts must be stored in the 'lexicon' (refer to Planning Guide COM GRAPH). If they are not, only error numbers will be displayed (e.g. W15). Explanations to these error numbers can be found in the Chapter entitled "Error Messages" (Appendix A in this Manual).

3.12.2   Using the Function Keys

If you do not want to have to type in a complete command each time you need it, you can program the command into one of the 16 function keys on the process keyboard. A labeling field at the bottom of the screen allows you to program the function keys. You can program the function keys for each display output command, and you can redefine the key functions while the display is on the screen. Refer to Planning Guide COM GRAPH for more information on function keys.

3.12.3   Entering Process Data

You can also use the keyboard to enter data in the process. To do this, you must define data input fields at the programming stage. Enter data into these fields by positioning the cursor at the beginning of each field.

To enter a value, select the correct input field by positioning the cursor into that field. To do this, use the following cursor control keys:

$\nwarrow$ cursor moves to the first input field of the screen

$\Rightarrow$ cursor moves to the next input field (used with the < RETURN > key)

$\Leftarrow$ cursor moves to the preceding input field

$\Downarrow$ cursor moves to the first input field of the following line

$\Uparrow$ cursor moves to the first input field of the preceding line

Use the keyboard to enter data in the input field. To make corrections within a data field, use the < CHAR LEFT > and < CHAR RIGHT > keys to move the cursor one character position at a time.

If you use one of the five cursor keys in the list above to move to another field, all characters to the left of the cursor are checked for their validity. If the results are positive, numerical values are shown right-justified in the input field, and text is shown left-justified. The cursor is positioned at the beginning of the selected field. If the values are not allowed, you cannot leave the input field, and an operator error message appears in the second part of the command line.

You can leave the input field only if you have entered a permissible value. To stop entering data into a particular field, press the <BREAK> key or press the field selection key twice. If you press the field selection key, the input field is filled with the permissible value from the previous procedure.

NOTE:

1. Should you specify a conversion with offset and scaling when configuring input fields, as well as upper and lower limits for the input value, the CP 526 first converts the value entered and checks then whether the converted value is within the specified limits. A nonpermissible value is rejected by the CP 526, and the corresponding error message is displayed.

2. For entries that are converted with offset and scaling (see Planning Guide COM GRAPH), a different value than the value originally entered may appear. With right-justified output, the converted value is represented as it would be when transferred to the programmable controller. If you enter values for offset and scaling that are not particularly meaningful, and if decimal places are suppressed, there may be considerable deviations.

The following examples show how to convert and check a value for input and how to reconvert it for right-justified output:

Example A

Input field in FPT16 format
Stored in the data block at DW10
Character positions 5.2
Scaling = 2.0
Offset = 1.0
Lower limit = 6
Upper limit = 16

1. Keyboard entry: 11.84
   Conversion: $(11.84 - 1.0)/2.0 = 5.42$
   FPT-16 value: 5
   Limit value check: The value '5' is too small! Error!
                        The value is not accepted by the CP 526.

2. Keyboard entry: 15.05
   Conversion: $(15.05 - 1.0)/2.0 = 7.025$
   FPT-16 value:
   Limit value check: OK
   Reconversion for output: $(7 \times 2.0) + 1.0 = 15.00$
   Right-justified output: 15.00

3. Keyboard entry: 16.25
   Conversion: $(16.25 - 1.0)/2.0 = 7.625$
   FPT16 value: 8
   Limit value check: OK
   Reconversion for output: $(8 \times 2.0) + 1.0 = 17.00$
   Right-justified output: 17.00

Example B

Input field in FPT16 format
Character positions 4.0
Scaling = 0.87912
Offset = 0
Lower limit = 2000
Upper limit = 3000

1. Keyboard entry: 2999
   Conversion: $(2999 - 0)/ 0.87912 = 3411.3659$
   Limit value check: The value '3411' is too great!
                                The value is not accepted by the CP 526.

2. Keyboard entry: 2500
   Conversion: $(2500 - 0)/ 0.87912 = 2843.7528$
   FPT16 value: 2844
   Limit value check: OK
   Reconversion for output: $(2844 \times 0.87912) + 0 = 2500.21728$

   Right-justified output: 2500

If the scaling value is located in the data block (indirect specification), it should not be equal to zero. If it is, division by zero during the input check results in error message W20, and you can only exit the input field with the <BREAK> key.

Also, you only need to enter data in the input fields that you want to change. All other entries are transferred to the process, with no further involvement on your part.

When you have changed all the input fields to the appropriate values, press the <TRANSFER> or <ENTER> key. This causes all entries to be processed at the same time.

The CP 526 automatically converts the entered values into the appropriate SIMATIC S5 system formats when they are transferred to the controller. (These formats are specified when programming with COM.) Table 3-10 lists the possible conversions.

Format in the programmable controller	Screen Format	Range of Values	1)			2)		3)		4)	
			F	B	C	d	i	d	i	d	i
Floating-point number GPT 32 (32 bits)	Numeric string with max. of 12 places before decimal point(incl. sign); max. of 10 places after decimal point (incl. decimal point)	-2.147.483.647 to 2.147.483.647	x	x	x	x	x	x	x	x	x
Fixed-point number FPT 16 (16 bits)	Numeric string with max. of 12 places including sign	-32 767 to 32 767	x	x	x	x	x	x	x	x	x
	Horizontal or vertical bars	Screen display	x	x	x	x	x	x	x	x	x
Fixed-point no. FPT (32 bits)	Numeric string with max. of 12 places including sign	-2.147.483.647 to 2.147.483.647	x	x	x	x	x	x	x	x	x
BCD number BCD 16 (16 bits)	Numeric string with max. of 12 places including sign	-999 to 999	x	x	x	x	x	x	x	x	x
BCD number BCD 32 (32 bits)	Numeric string with max. of 12 places including sign	-9 999 999 to 9 999 999	x	x	x	x	x	x	x	x	x
Byte (8 bits)	Num. string with max. of 12 places	0 to 255	x	x	x	x	x	x	x	x	x
Time TIMER (16 bits)	Field with four places (the decimal point is set automatically so that the time is given in seconds)	0.01 to 9990	x	x	x	-	-	-	-	x	x
Count COUNTER (16 bits)	Field with a maximum of 12 places	0 to 999	x	x	x	-	-	-	-	x	x
Character string	Char. string with max. 80 char.	ASCII char. set 20H to 7EH	x	x	x	-	-	-	-	-	-
Binary (1 bit)	Individual ASCII character	0 or 1	x	x	x	-	-	-	-	-	-

1) Attribute   2) Scaling   3) Offset   4) Limits
F = Foreground color   d = Direct specification of scaling/offset
B = Background color   i = Indirect specification (Data Block address) of
C = Flashing on/off   scaling/offset

Table 3-10: CP 526 Input Formats

3.13   CP 526 Character Set

3.13.1   Overview of the CP 526 Character Set

Varoius national charater sets can be loaded into the CP 526. Even the cyrillic character set can be loaded. In this case, Latin lowercase letters are replaced by Cyrillic uppercase letters.

Table 3-11 shows an overview of the standard character set of the CP 526.

Table 3-12 shows an overview of the national character set coding. Please note, that the same character may have to be given a different code in every language it occurs.

Line					b7	0	0	0	0	1	1	1	1
		Column			b6	0	0	1	1	0	0	1	1
					b5	0	1	0	1	0	1	0	1
b4	b3	b2	b1	Hex		0	1	2	3	4	5	6	7
0	0	0	0	0	0			SP	0		P		p
0	0	0	1	1	1			!	1	A	Q	a	q
0	0	1	0	2	2	Re-		"	2	B	R	b	r
0	0	1	1	3	3	served			3	C	S	c	s
0	1	0	0	4	4	for			4	D	T	d	t
0	1	0	1	5	5	CP 526-spe-		%	5	E	U	e	u
0	1	1	0	6	6	cific		&	6	F	V	f	v
0	1	1	1	7	7	char- ac-		'	7	G	W	g	w
1	0	0	0	8	8	ters/		(	8	H	X	h	x
1	0	0	1	9	9	sym- bols		)	9	I	Y	i	y
1	0	1	0	A	10			*	:	J	Z	j	z
1	0	1	1	B	11			+	;	K		k	
1	1	0	0	C	12			,	<	L		l	
1	1	0	1	D	13			-	=	M		m	
1	1	1	0	E	14			.	>	N		n	
1	1	1	1	F	15			/	?	O	—	o	DEL

Table 3-11: Standard Character Set of the CP 526

Inter. ASCII National Standard Char. Set		Hex. Code	ASCII	Swed./Finn.	Danish/Nor.	German	British	Spanish	French/Bel.	International
Col.	Line									
2	3	23	#	§	#	#	£	#	£	#
2	4	24	$	¤	¤	$	$	$	$	¤
4	0	40	@	É	É	§	@	@	à	@
5	11	5B	[	Ä	Æ	Ä	[	¡	'	[
5	12	5C	\	Ö	Ø	Ö	\	Ñ	ç	\
5	13	5D	]	Å	Å	Ü	]	¿	§	]
5	14	5E	^	^	Ü	^	^	^	^	^
6	0	60	`	é	é	`	`	`	`	`
7	11	7B	{	ä	æ	ä	{	´	é	{
7	12	7C	|	ö	ø	ö	|	ñ	ù	|
7	13	7D	}	å	å	ü	}	ç	è	}
7	14	7E	~	ü	ü	ß	―	¨	¨	―

Table 3-12: Special National Characters of the CP 526

3.13.2   Character Set for High-Resolution Monitor (Pixel Representation)

3.13.2.1 Standard Characters

20H = 32	21H = 33	22H = 34	23H = 35

24H = 36	25H = 37	26H = 38	27H = 39

28H = 40	28H = 41	2AH = 42	2BH = 43

2CH = 44	2DH = 45	2EH = 46	2FH = 47
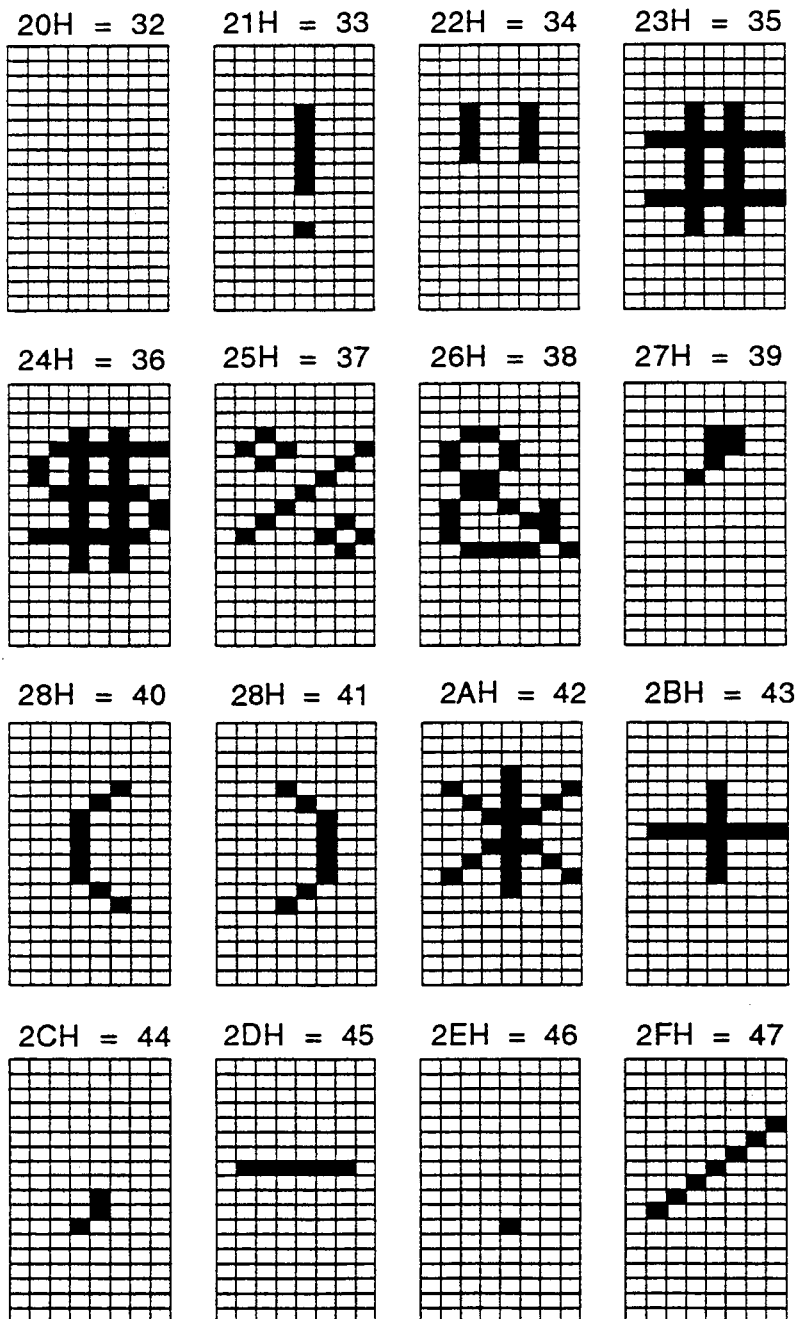
Standard Characters CP 526: 18 * 8
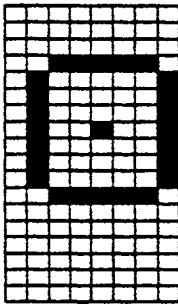
30H = 48	31H = 49	32H = 50	33H = 51
0	1	2	3
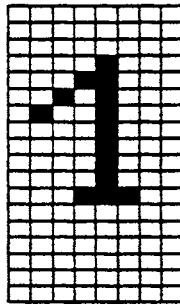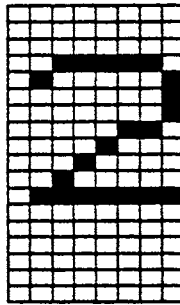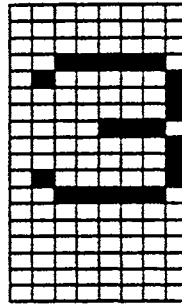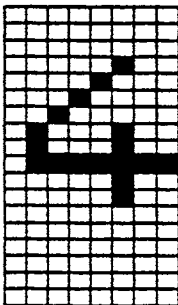
34H = 52	35H = 53	36H = 54	37H = 55
4	5	6	7
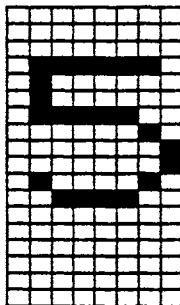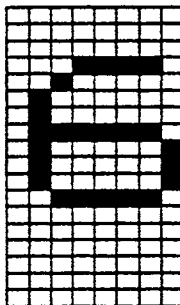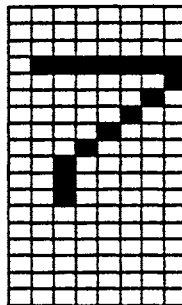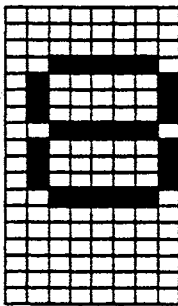
38H = 56	39H = 57	3AH = 58	3BH = 59
8	9	:	;
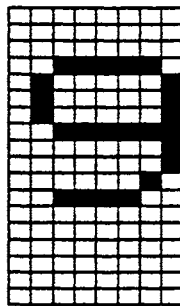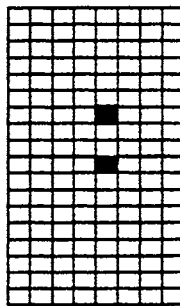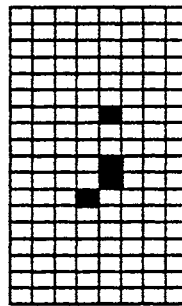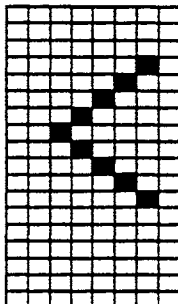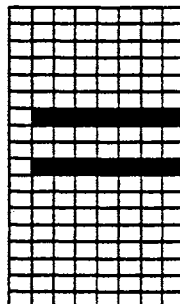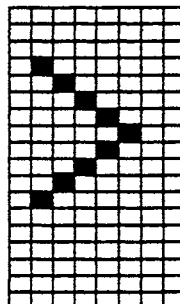
3CH = 60	3DH = 61	3EH = 62	3FH = 63
<	=	>	?
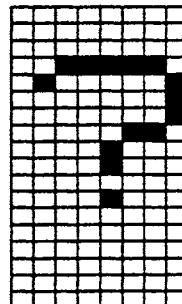
Standard Characters CP 526: 18 * 8

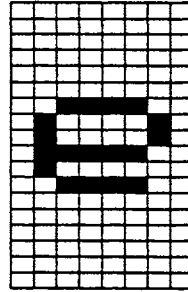40 H = 64   41H = 65   42H = 66   43H = 67
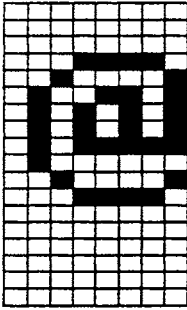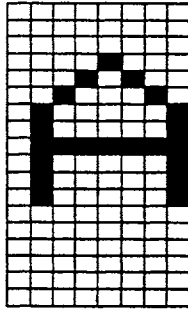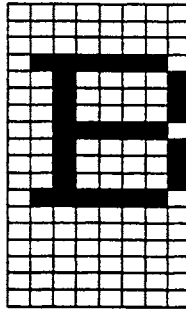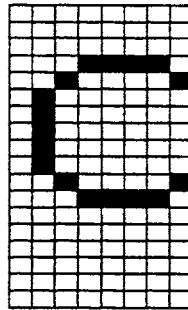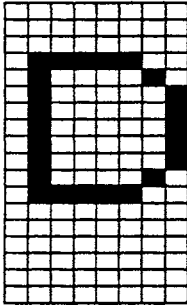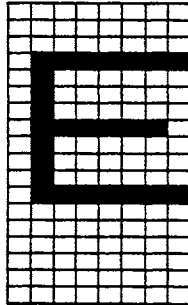
44H = 68   45H = 69   46H = 70   47H = 71
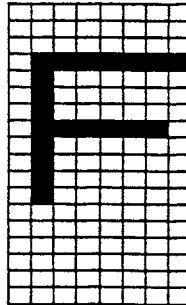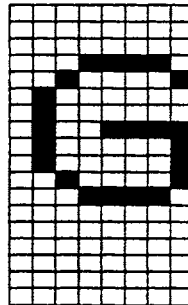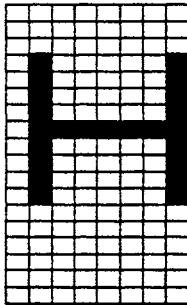
48H = 72   49H = 73   4AH = 74   4BH = 75
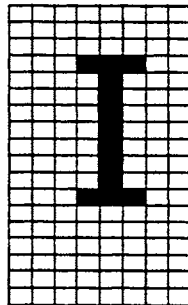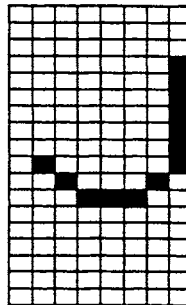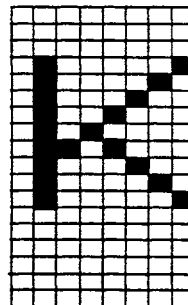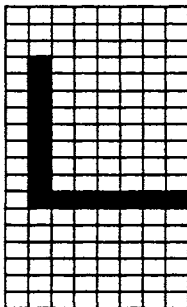
4CH = 76   4DH = 77   4 EH = 78   4FH = 79
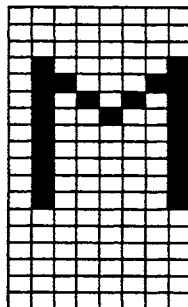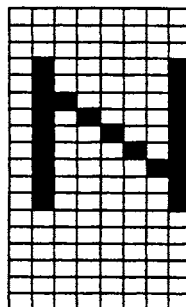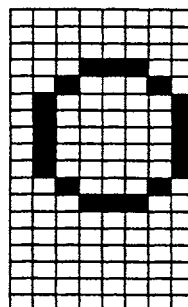
Standard Characters CP 526: 18 * 8

50H = 80	51H = 81	52H = 82	53H = 83
P	Q	R	S
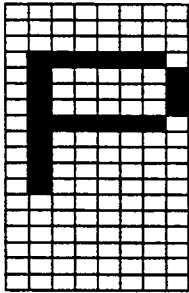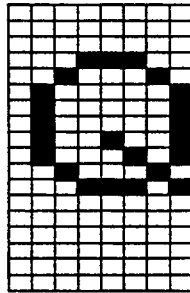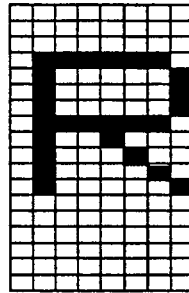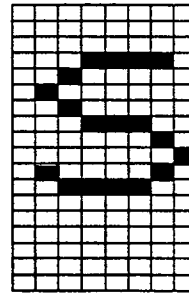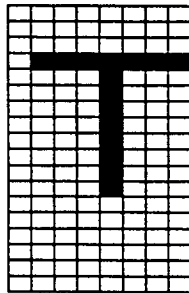
54H = 84	55H = 85	56H = 86	57H = 87
T	U	V	W
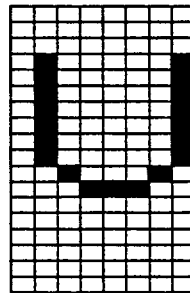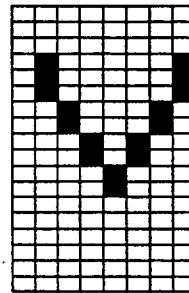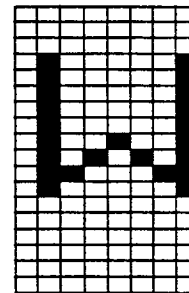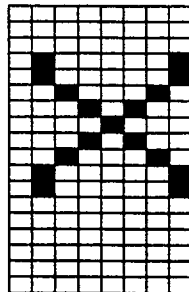
58H = 88	59H = 89	5AH = 90	5BH = 91
X	Y	Z	[
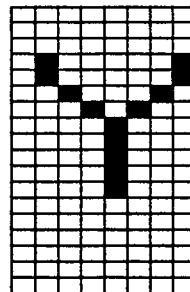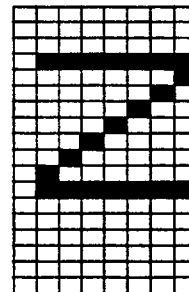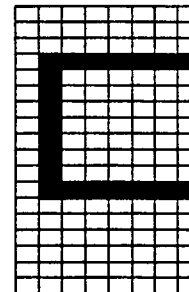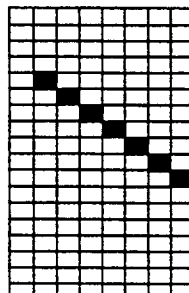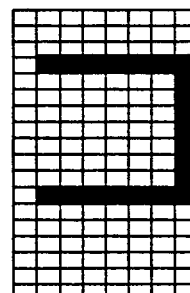
5CH = 92	5DH = 93	5EH = 94	5FH = 95
\	]	^	_

Standard Characters CP 526: 18 * 8

60H = 96	61H = 97	62H = 98	63H = 99
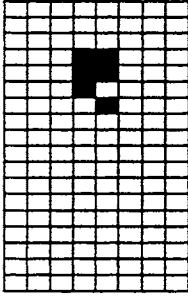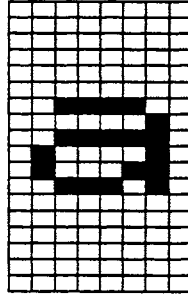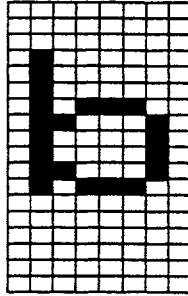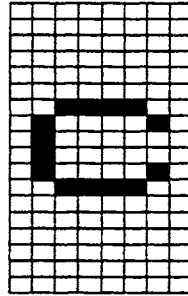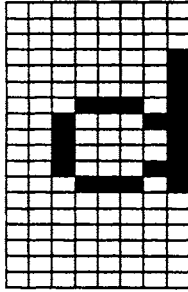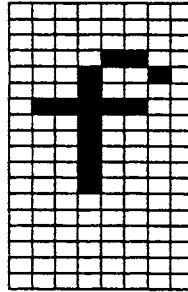
64H =100	65H =101	66H =102	67H =103
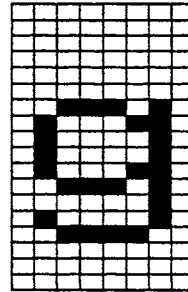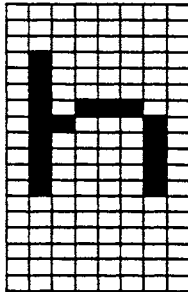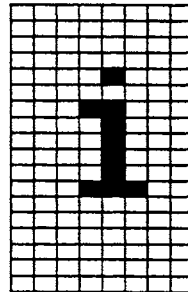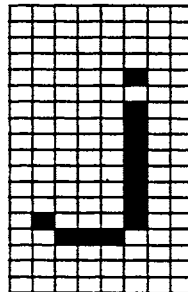
68H =104	69H =105	6AH =106	6BH =107
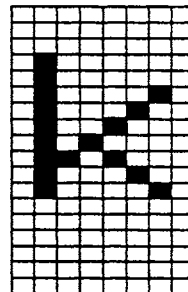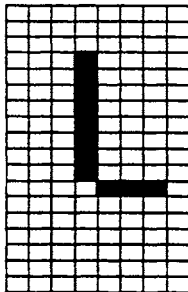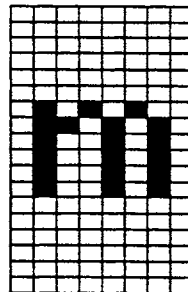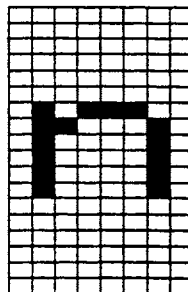
6CH =108	6DH =109	6EH =110	6FH =111
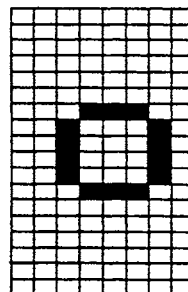
Standard Characters CP 526: 18 * 8
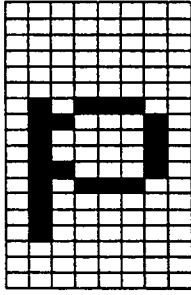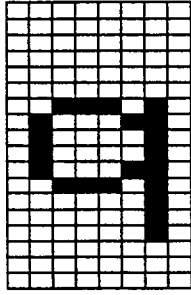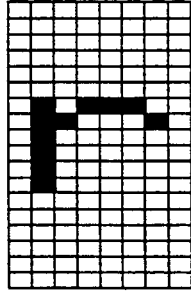
70H =112 **71H =113** **72H =114** **73H =115**
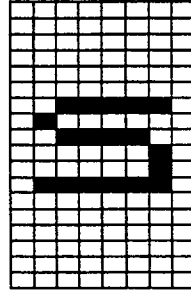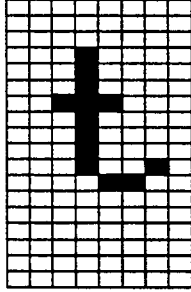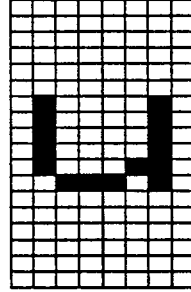
74H =116 **75H =117** **76H =118** **77H =119**
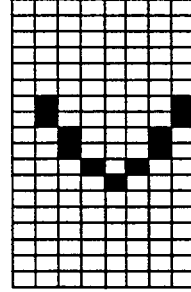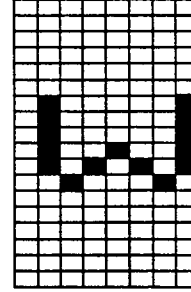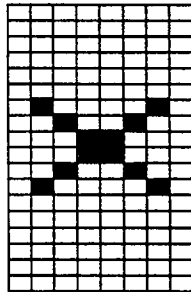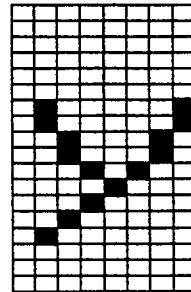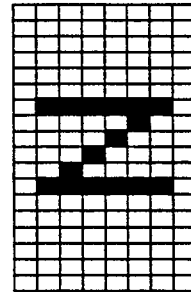
78H =120 **79H =121** **7AH =122** **7BH =123**
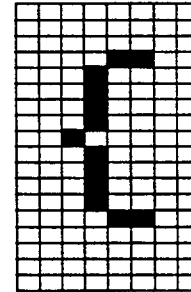
7CH =124 **7DH =125** **7EH =126** **7FH =127**
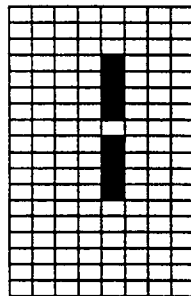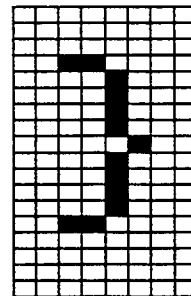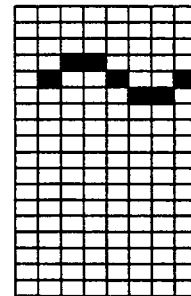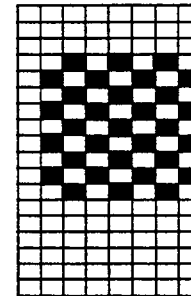
Standard Characters CP 526: 18 * 8

3.13.2.2 Special National Characters
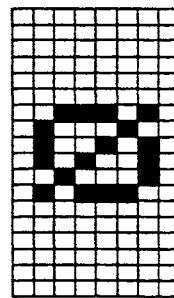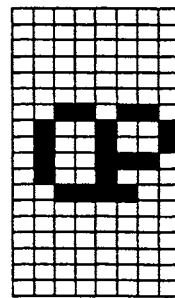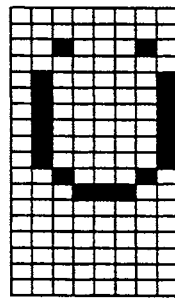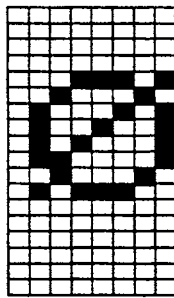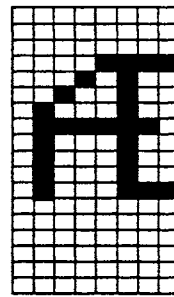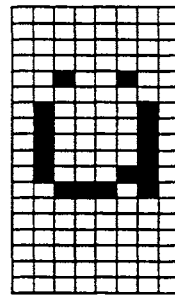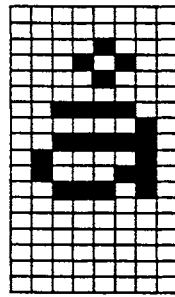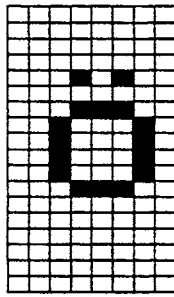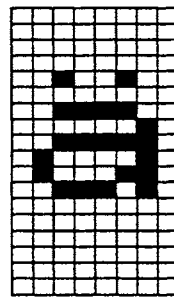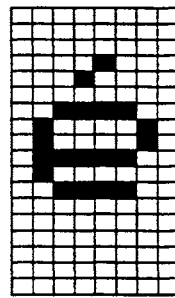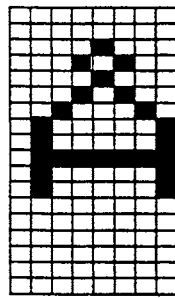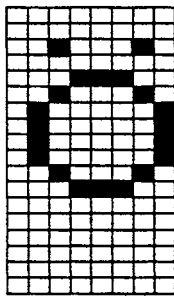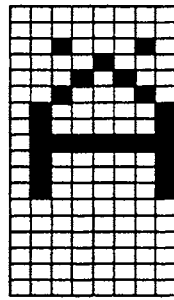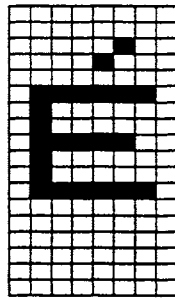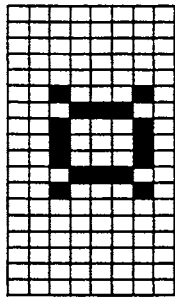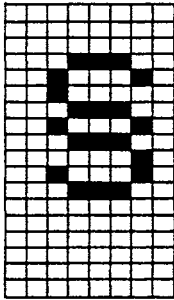
Special National Characters CP 526: 18 * 8
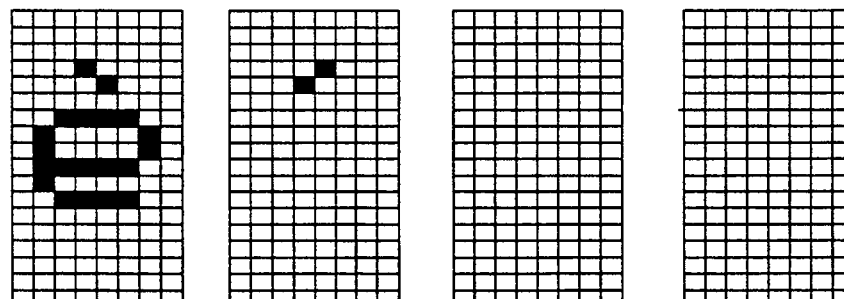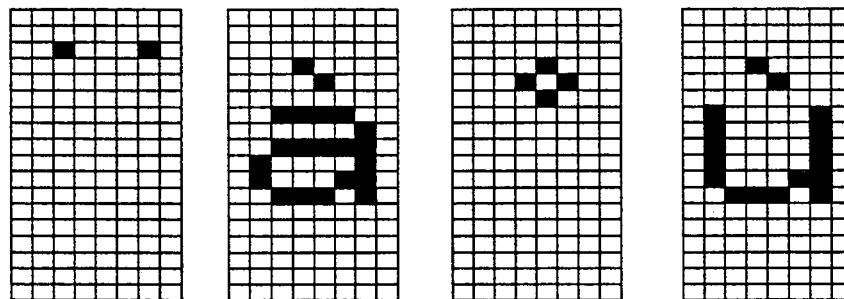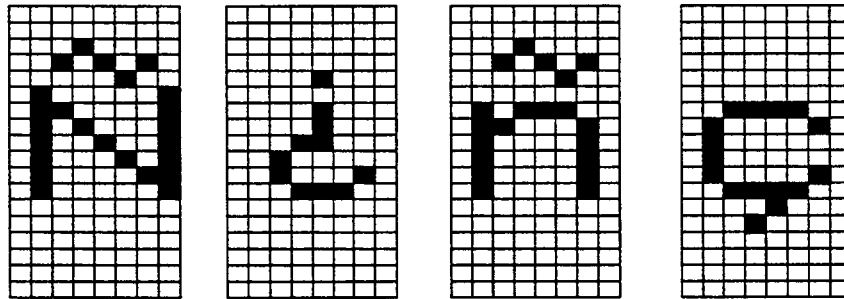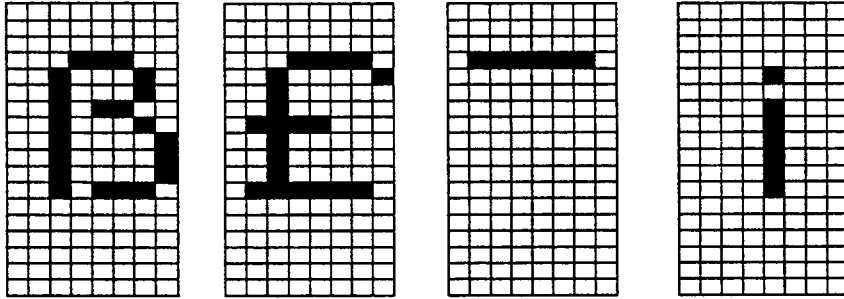
Special National Characters CP 526: 18 * 8

3.13.2.3 Cyrillic Characters
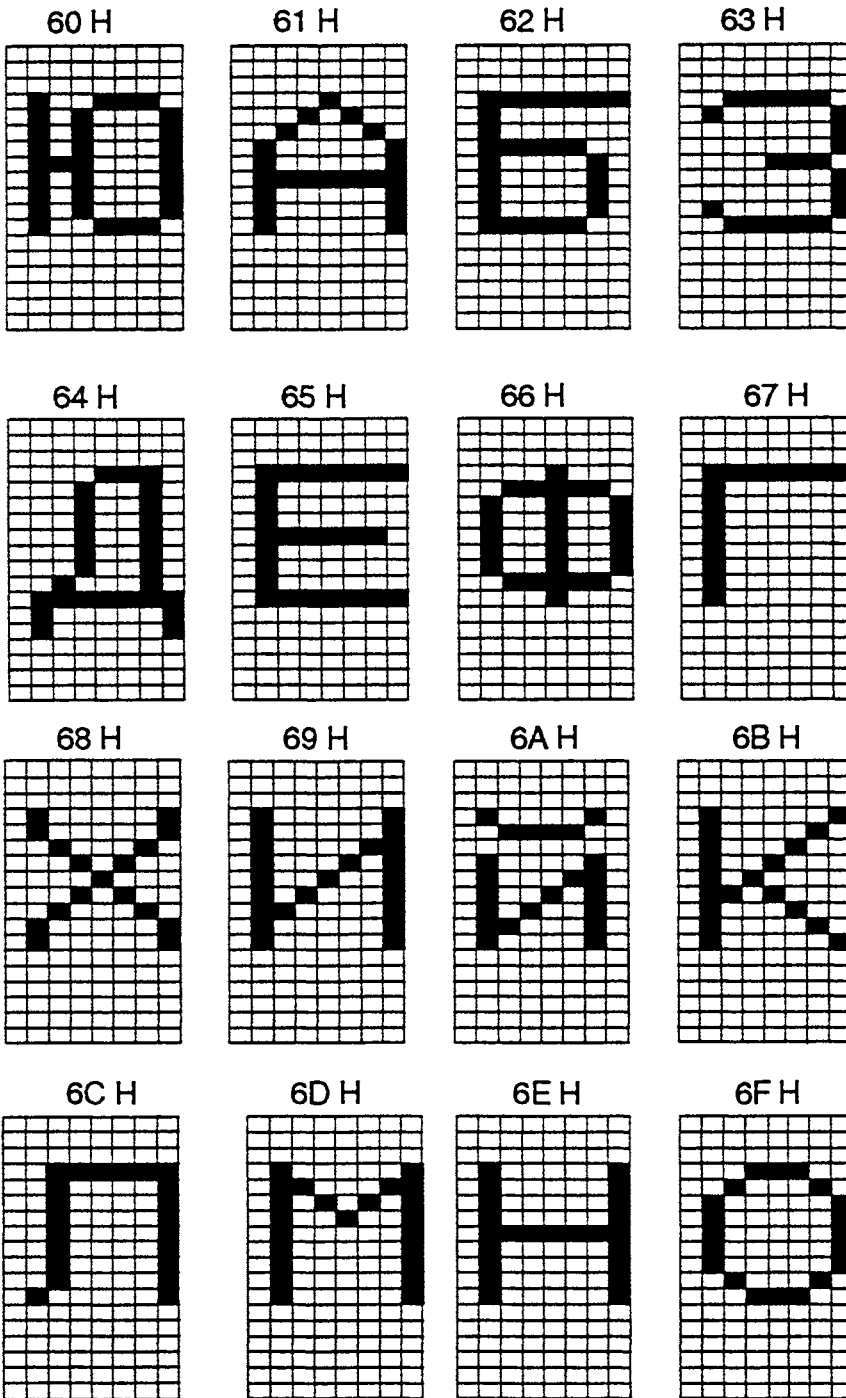
60 H	61 H	62 H	63 H
Ю	Й	Б	З

64 H	65 H	66 H	67 H
Д	Е	Ф	Г

68 H	69 H	6A H	6B H
Ж	И	Й	К

6C H	6D H	6E H	6F H
П	Л	Н	О

Cyrillic Characters CP 526: 18 * 8

70 H	71 H	72 H	73 H
П	Э	Р	С

74 H	75 H	76 H	77 H
Т	У	Ж	Е

78 H	79 H	7A H	7B H
Ь	Ы	Ц	Ш

7C H	7D H	7E H	7F H
Э	Щ	Ч	

Cyrillic Characters CP 526: 18 * 8

24 H   **27 H**
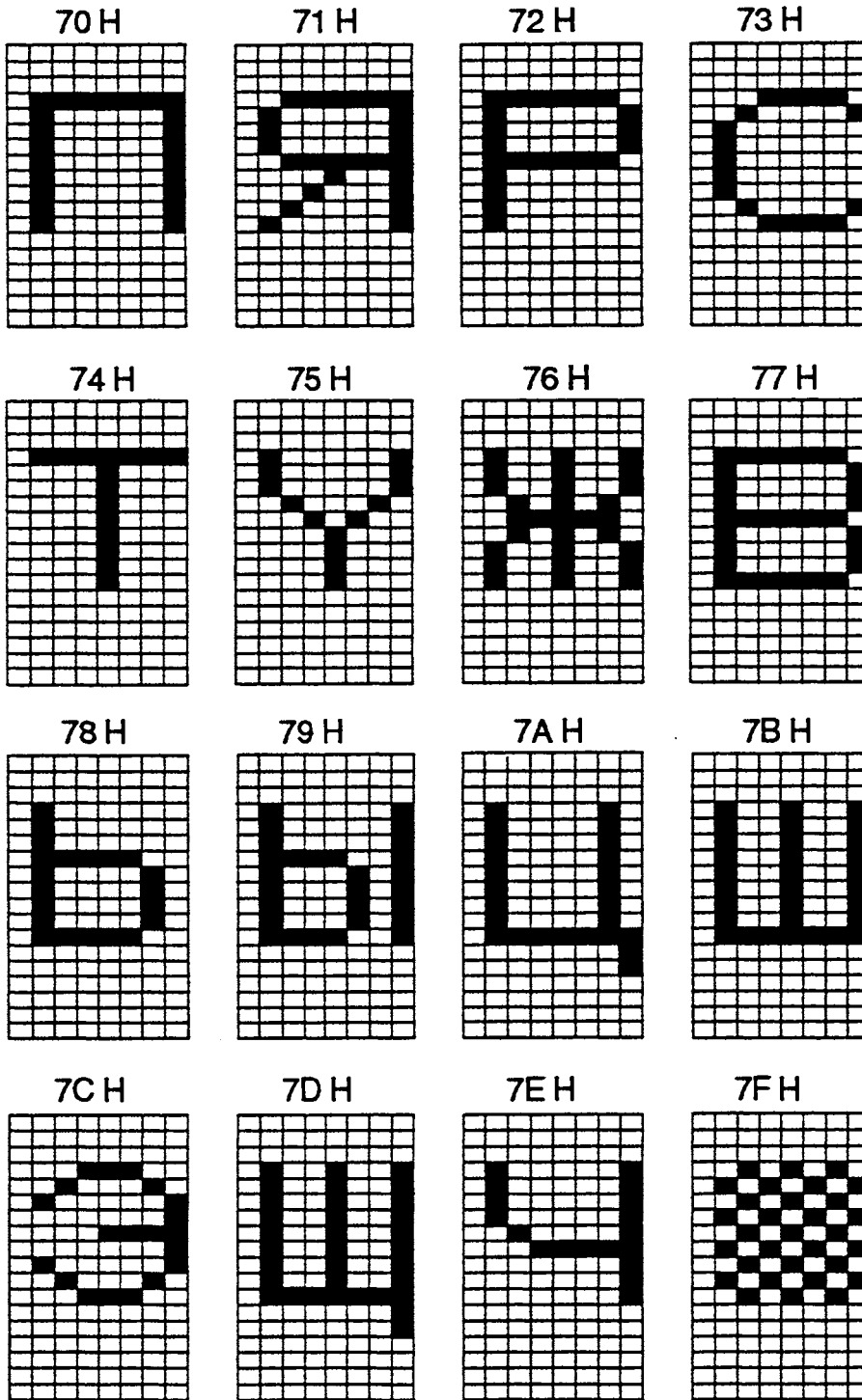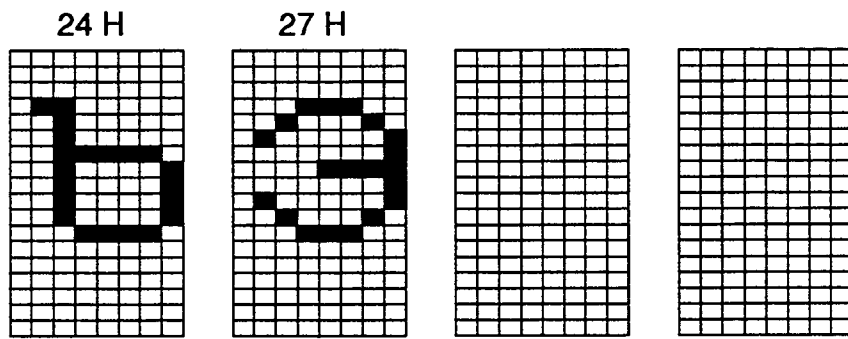
Cyrillic Characters CP 526: 18 * 8

3.13.2.4 Symbols That Can Be Loaded
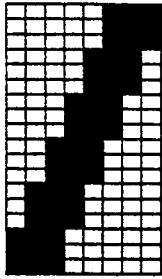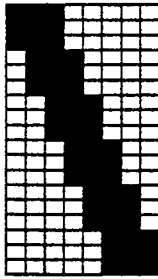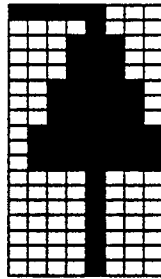
80H =128   81H =129   82H =130   83H =131
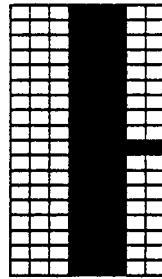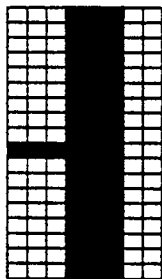
84H =132   85H =133   86H =134   87H =135
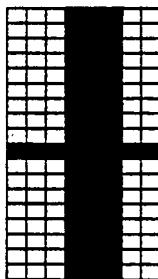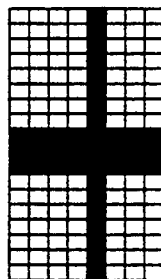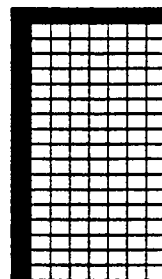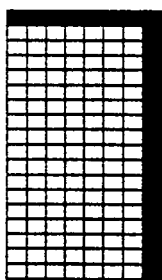
88H =136   89H =137   8AH =138   8BH =139
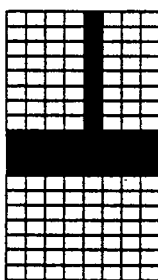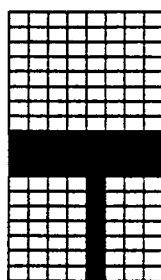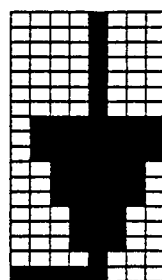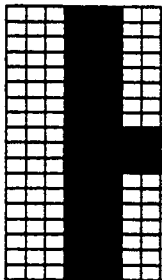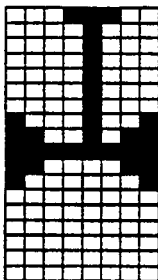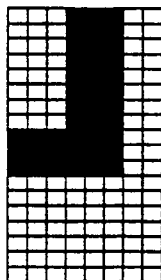
8CH =140   8DH =141   8EH =142   8FH =143
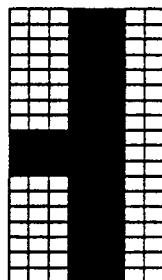
Symbols That Can Be Loaded CP 526: 18 * 8

90H =144   91H =145   92H =146   93H =147
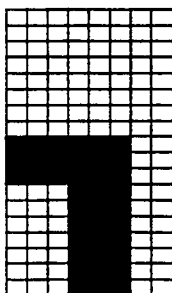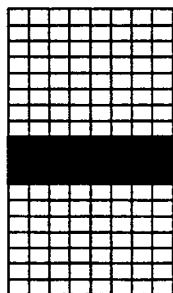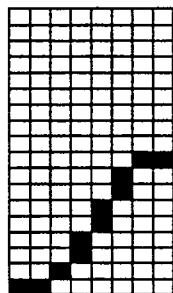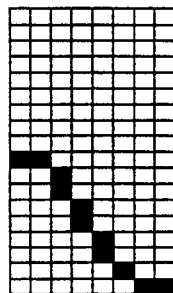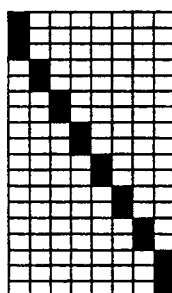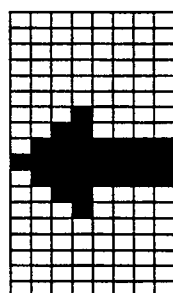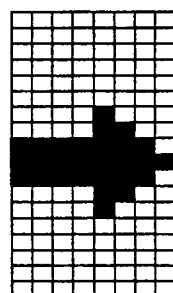
94H =148   95H =149   96H =150   97H =151
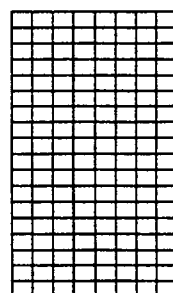
98H =152   99H =153   9AH =154   9BH =155
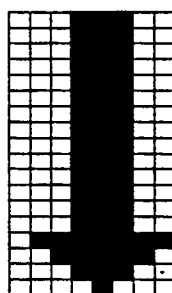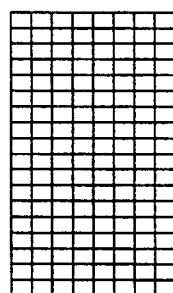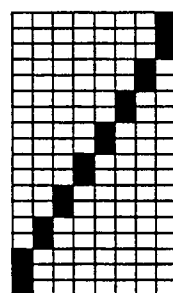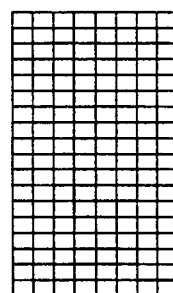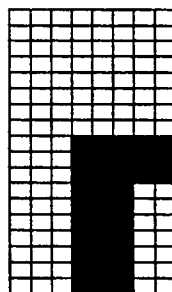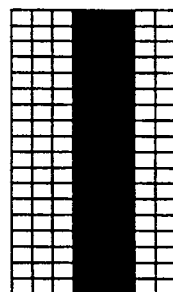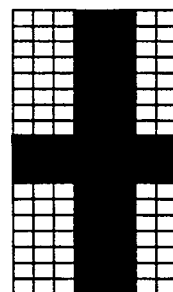
9CH =156   9DH =157   9EH =158   9FH =159
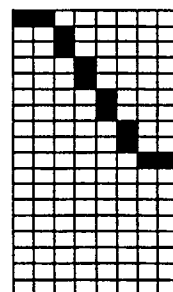
Symbols That Can Be Loaded CP 526: 18 * 8

A0H =160 A1H =161 A2H =162 A3H =163
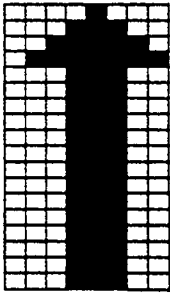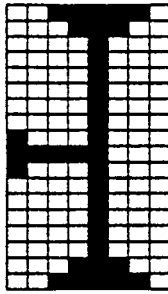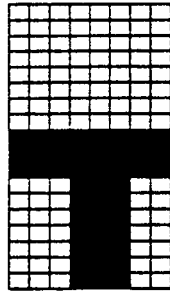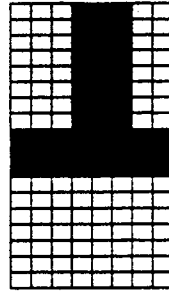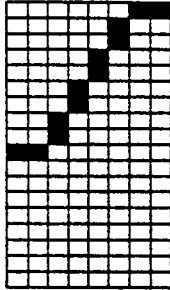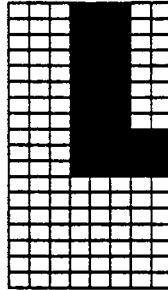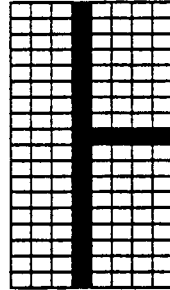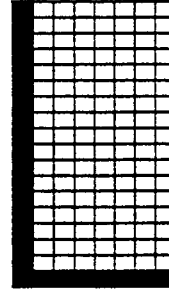
A4H =164 A5H =165 A6H =166 A7H =167
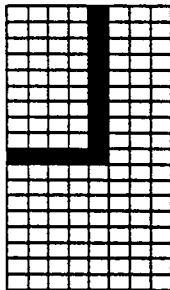
A8H =168 A9H =169 AAH =170 ABH =171
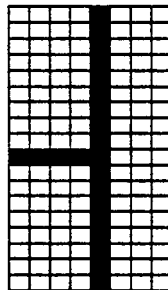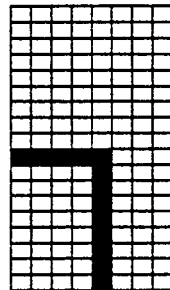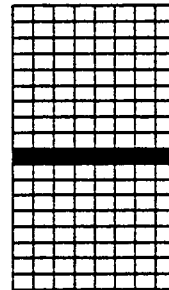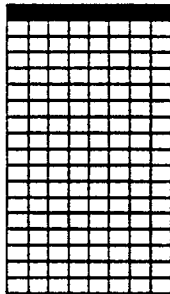
ACH =172 ADH =173 AEH =174 AFH =175
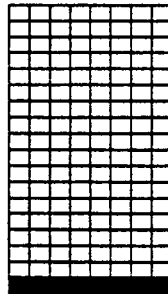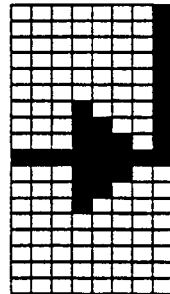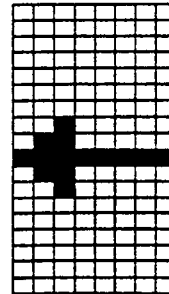
Symbols That Can Be Loaded CP 526: 18 * 8

B0H =176 B1H =177 B2H =178 B3H =179
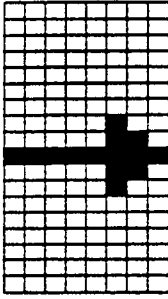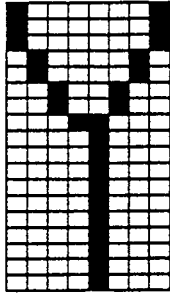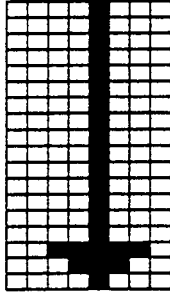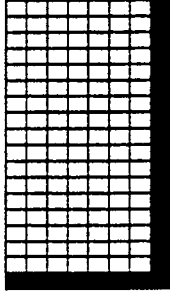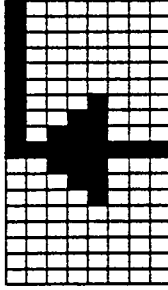
B4H =180 B5H =181 B6H =182 B7H =183
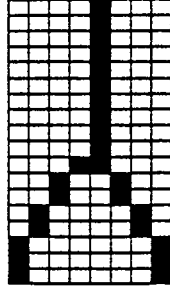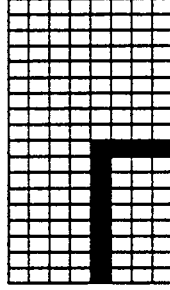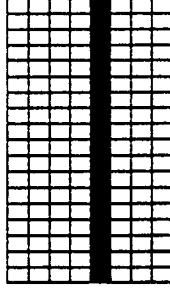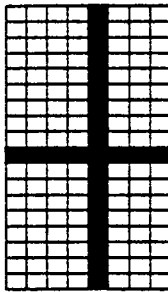
B8H =184 B9H =185 BAH =186 BBH =187
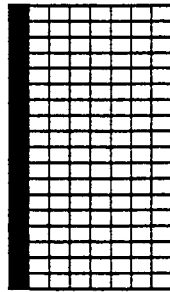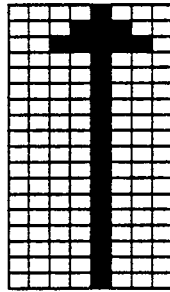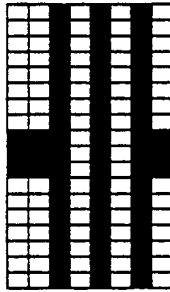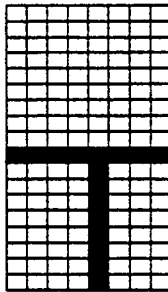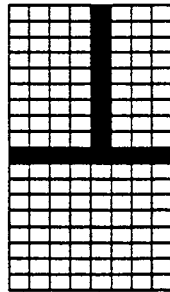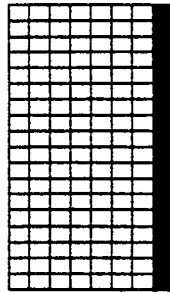
BCH =188 BDH =189 BEH =190 BFH =191
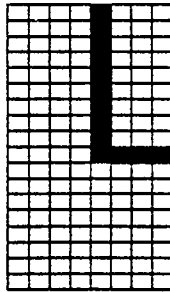
Symbols That Can Be Loaded CP 526: 18 * 8

C0H =192	C1H =193	C2H =194	C3H =195
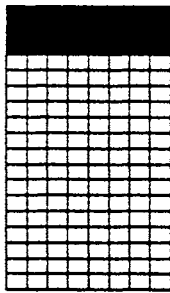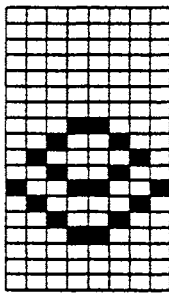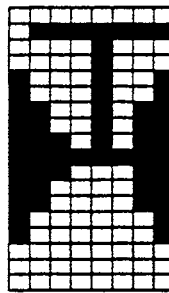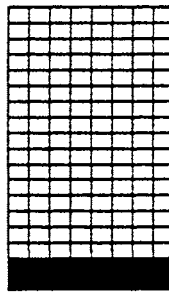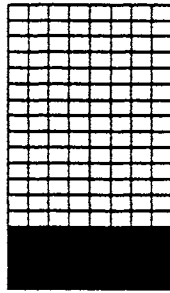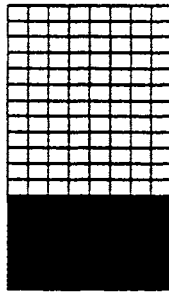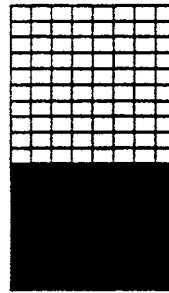
C4H =196	C5H =197	C6H =198	C7H =199
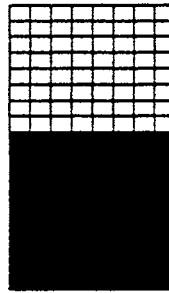
C8H =200	C9H =201	CAH =202	CBH =203
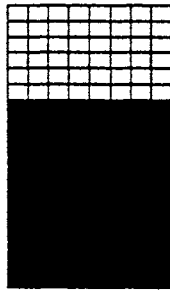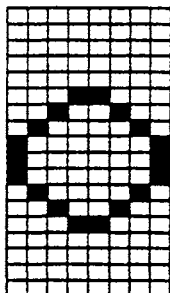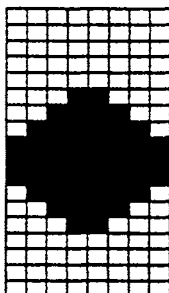
CCH =204	CDH =205	CEH =206	CFH =207

Symbols That Can Be Loaded CP 526: 18 * 8

D0H =208   D1H =209   D2H =210   D3H =211

D4H = 212   D5H =213   D6H =214   D7H =215

D8H =216   D9H =217   DAH =218   DBH =219

DCH =220   DDH =221   DEH =222   DFH =223
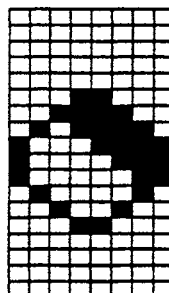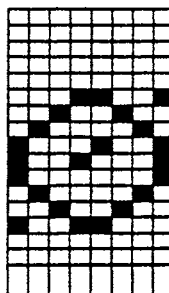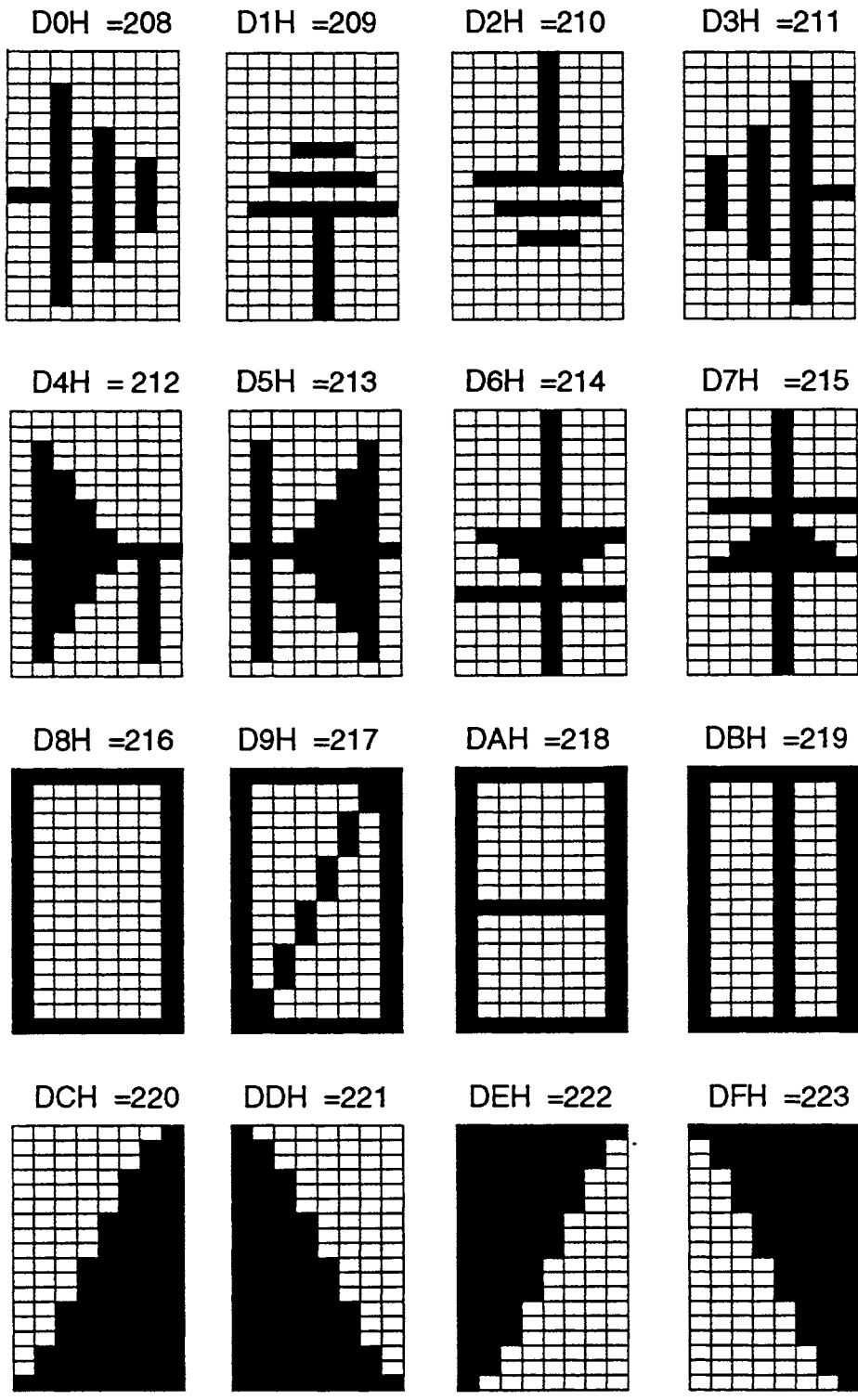
Symbols That Can Be Loaded CP 526: 18* 8

E0H =224	E1H =225	E2H =226	E3H =227
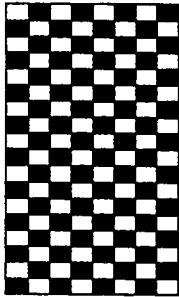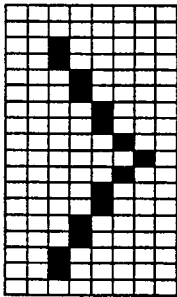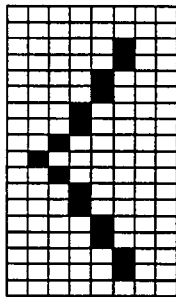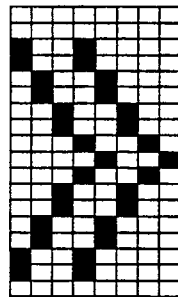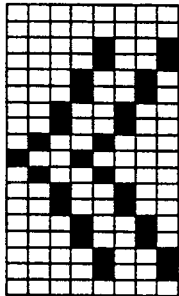
E4H =228	E5H =229	E6H =230	E7H =231
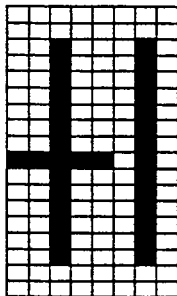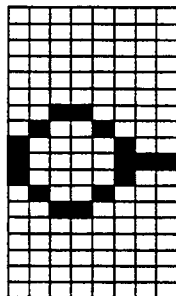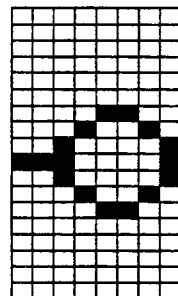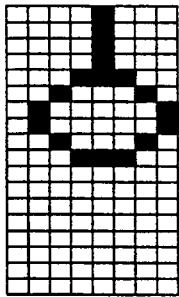
E8H =232	E9H =233	EAH =234	EBH =235
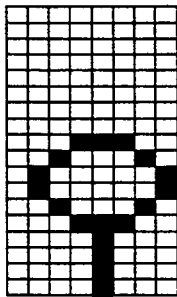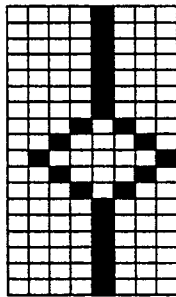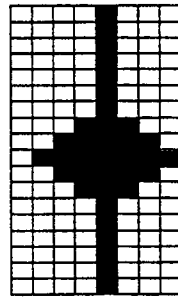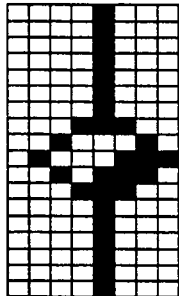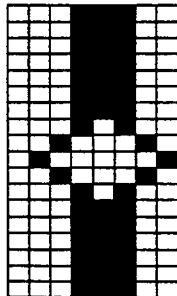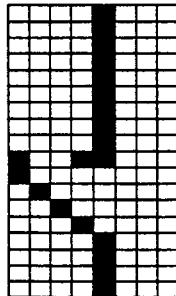
ECH =236	EDH =237	EEH =238	EFH =239
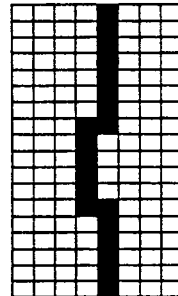
Symbols That Can Be Loaded CP 526: 18 * 8

F0H =240	F1H =241	F2H =242	F3H =243
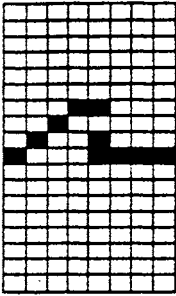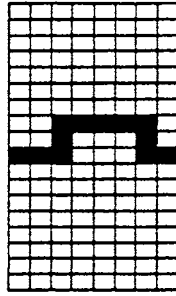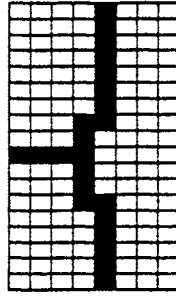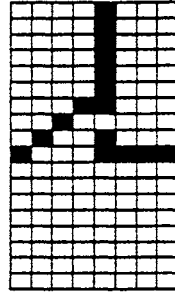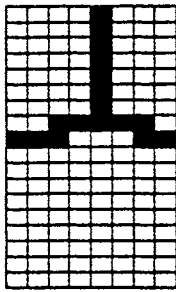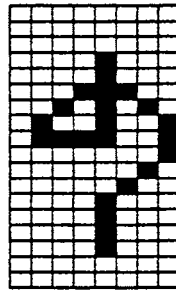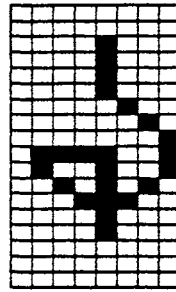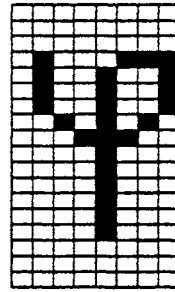
F4H =244	F5H =245	F6H =246	F7H =247

F8H =248	F9H =249	FAH =250	FBH =251
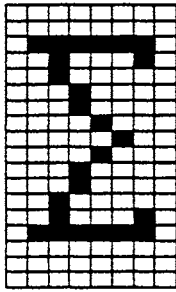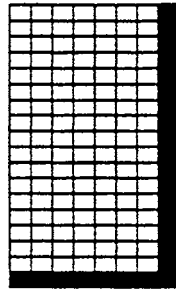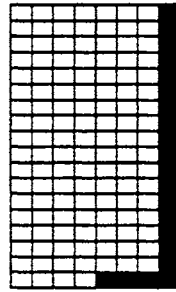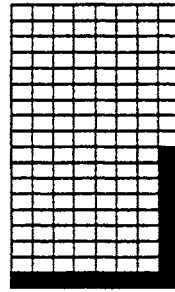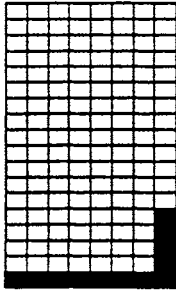
FCH =252	FDH =253	FEH =254	FFH =255
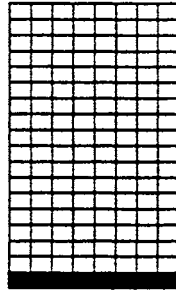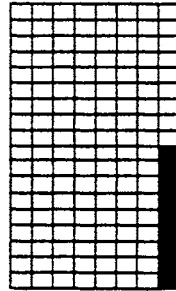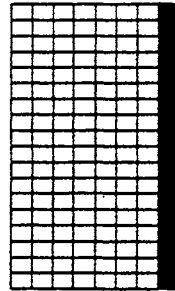
Symbols That Can Be Loaded CP 526: 18 * 8

3.13.3 Character Set for Low Resolution Monitor (Pixel Representation)

3.13.3.1 Standard Characters

| 20H = 32 | 21H = 33 | 22H = 34 | 23H = 35 |

| 24H = 36 | 25H = 37 | 26H = 38 | 27H = 39 |

| 28H = 40 | 29H = 41 | 2AH = 42 | 2BH = 43 |

| 2CH = 44 | 2DH = 45 | 2EH = 46 | 2FH = 47 |

| 30H = 48 | 31H = 49 | 32H = 50 | 33H = 51 |

| 34H = 52 | 35H = 53 | 36H = 54 | 37H = 55 |
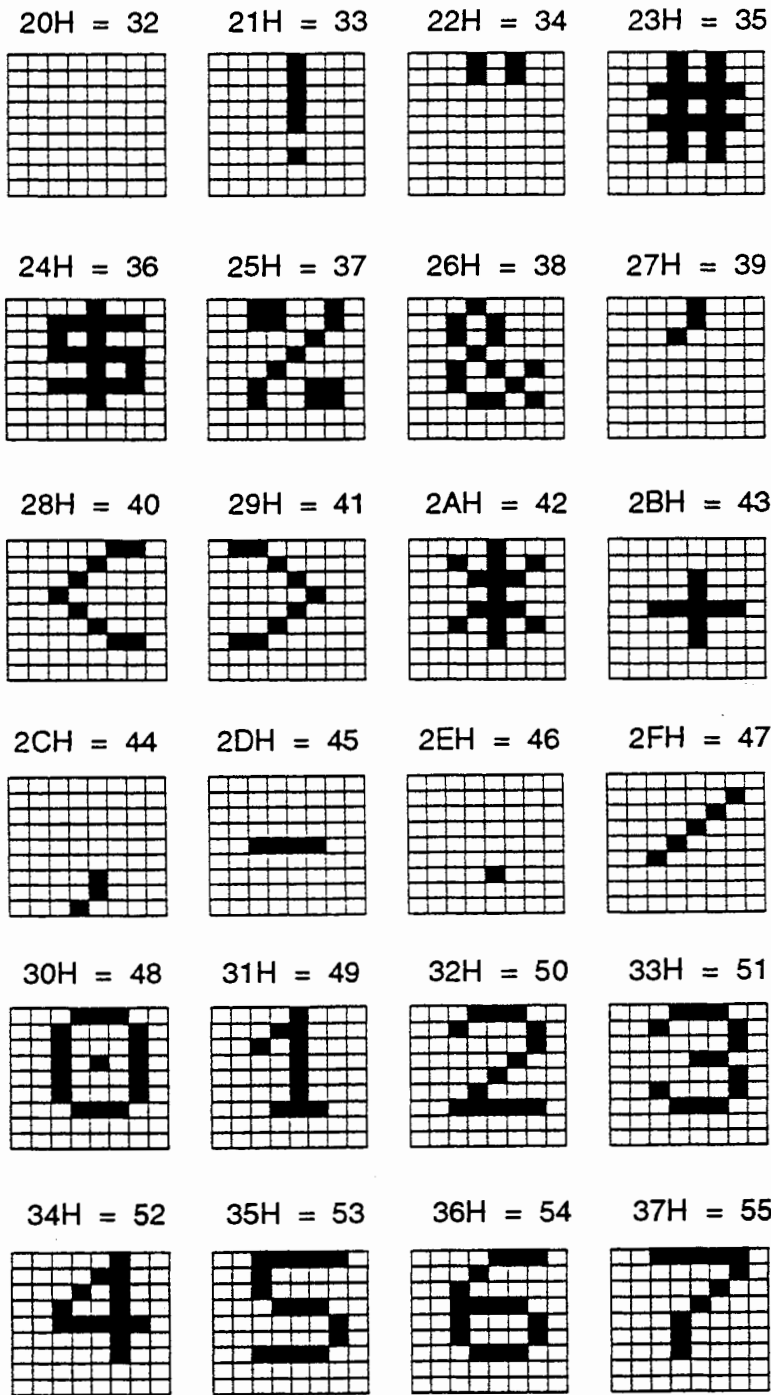
Standard Characters CP 526: 9 * 8
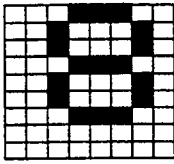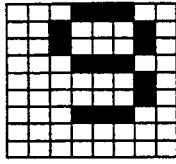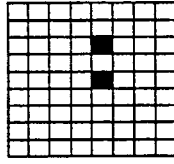
38H = 56   39H = 57   3AH = 58   3BH = 59
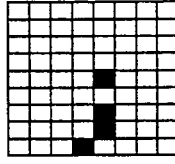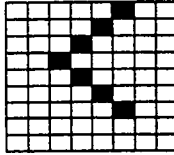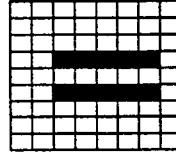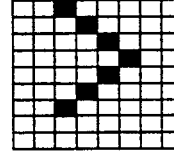
3CH = 60   3DH = 61   3EH = 62   3FH = 63
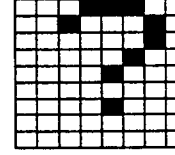
40H = 64   41H = 65   42H = 66   43H = 67
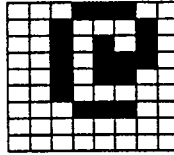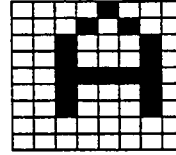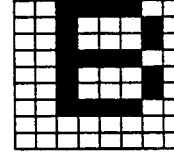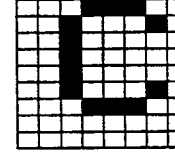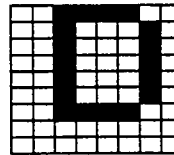
44H = 68   45H = 69   46H = 70   47H = 71
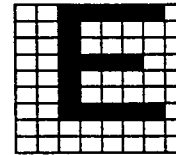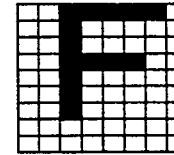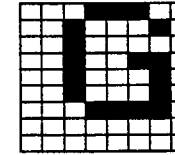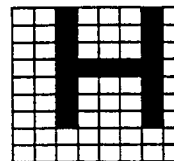
48H = 72   49H = 73   4AH = 74   4BH = 75
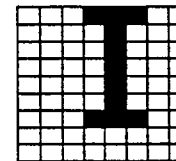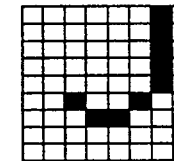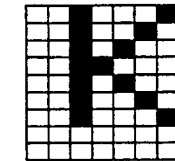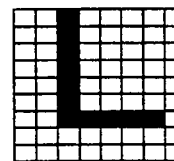
4CH = 76   4DH = 77   4EH = 78   4FH = 79
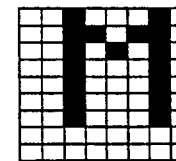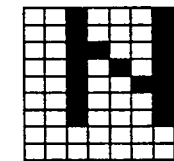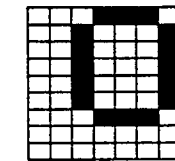
Standard Characters CP 526: 9 * 8
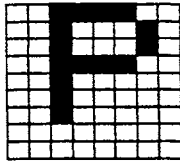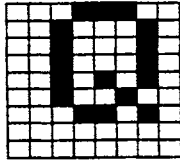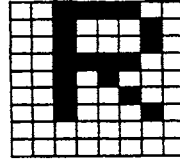
50H = 80   51H = 81   52H = 82   53H = 83
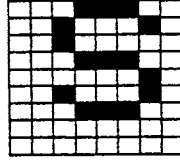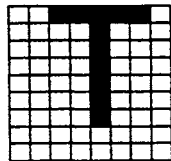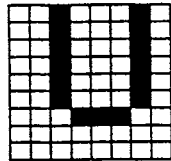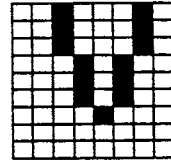
54H = 84   55H = 85   56H = 86   57H = 87
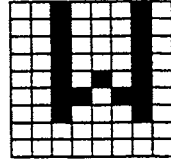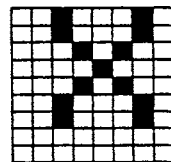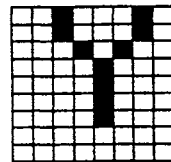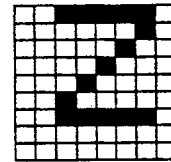
58H = 88   59H = 89   5AH = 90   5BH = 91
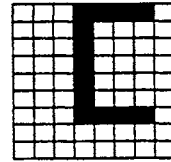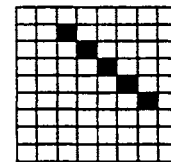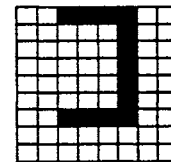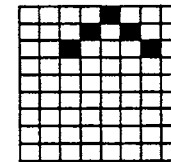
5CH = 92   5DH = 93   5EH = 94   5FH = 95
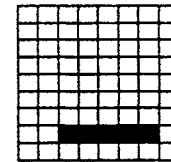
60H = 96   61H = 97   62H = 98   63H = 99
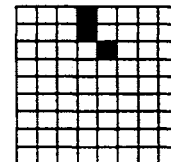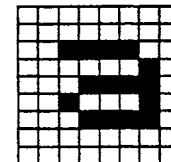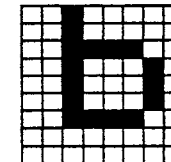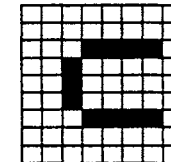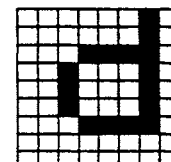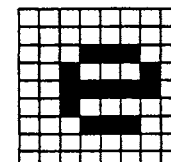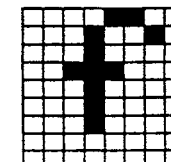
64H =100   65H =101   66H =102   67H =103
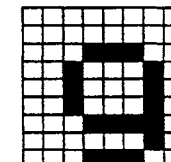
Standard Characters CP 526: 9 * 8

68H =104	69H =105	6AH =106	6BH =107
h	i	j	k
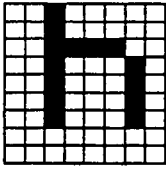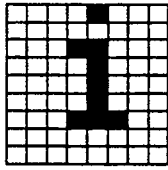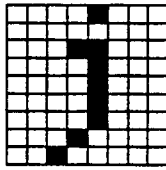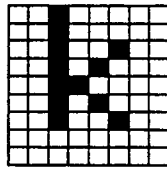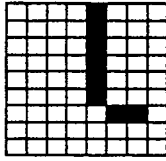
6CH =108	6DH =109	6EH =110	6FH =111
l	m	n	o
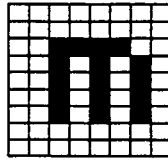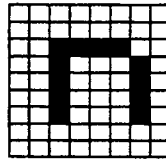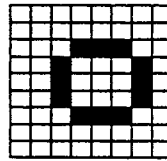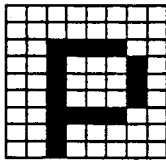
70H =112	71H =113	72H =114	73H =115
p	q	r	s
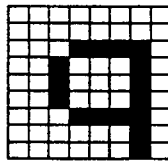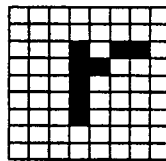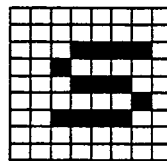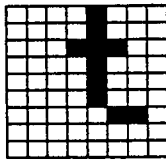
74H =115	75H =117	76H =118	77H =119
t	u	v	w
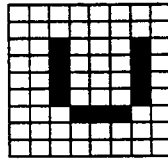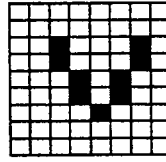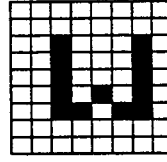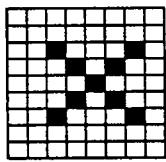
78H =120	79H =121	7AH =122	7BH =123
x	y	z	{
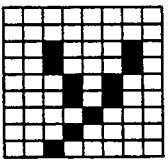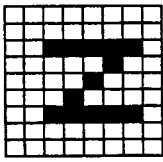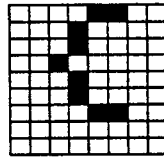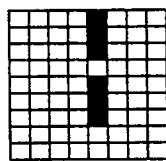
7CH =124	7DH =125	7EH =126	7FH =127
|	}	~	▓
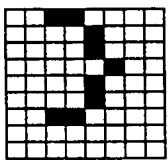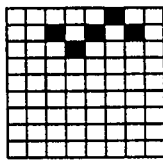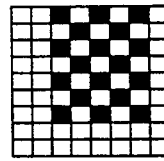
Standard Characters CP 526: 9 * 8

3.13.3.2 Special National Characters
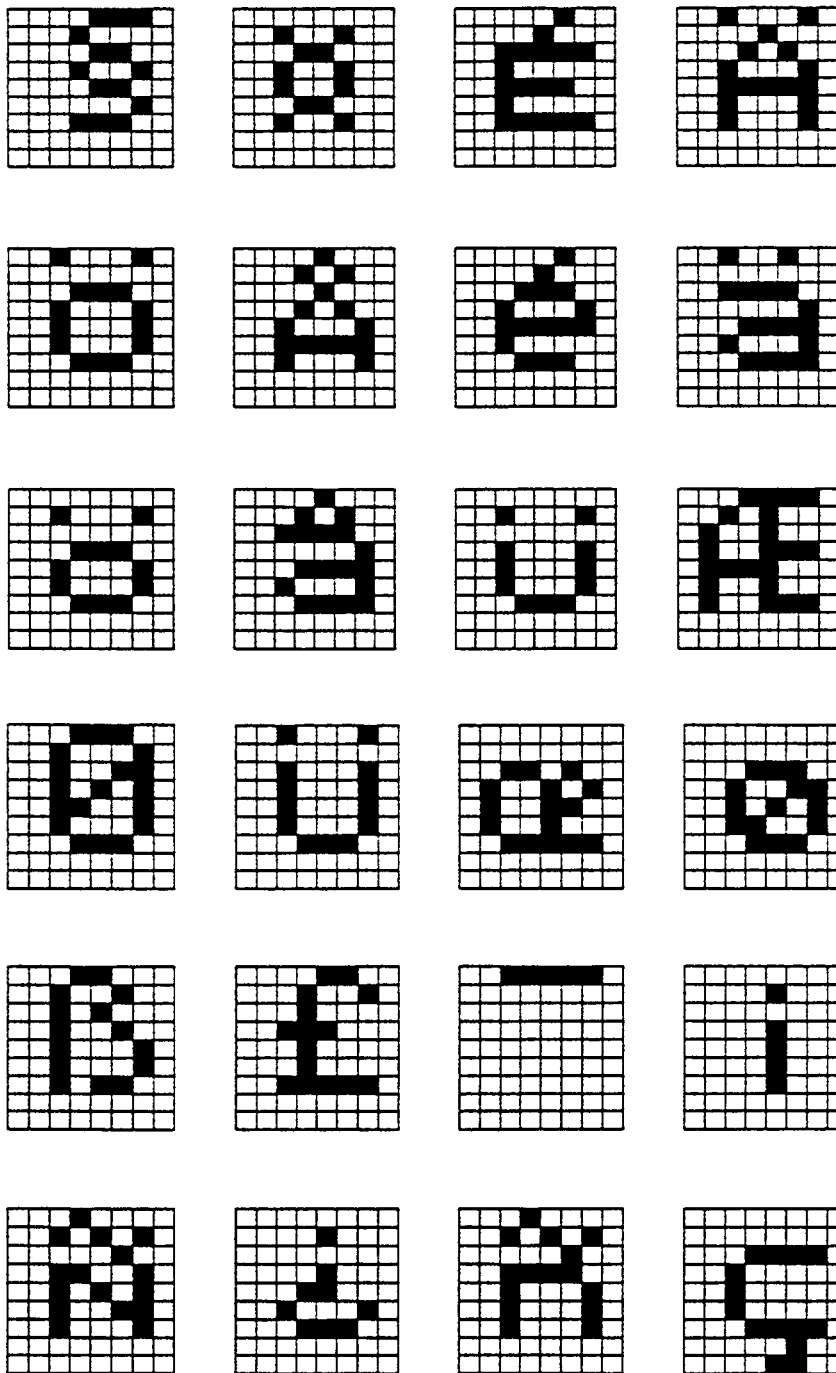
Special National Characters CP 526: 9 * 8
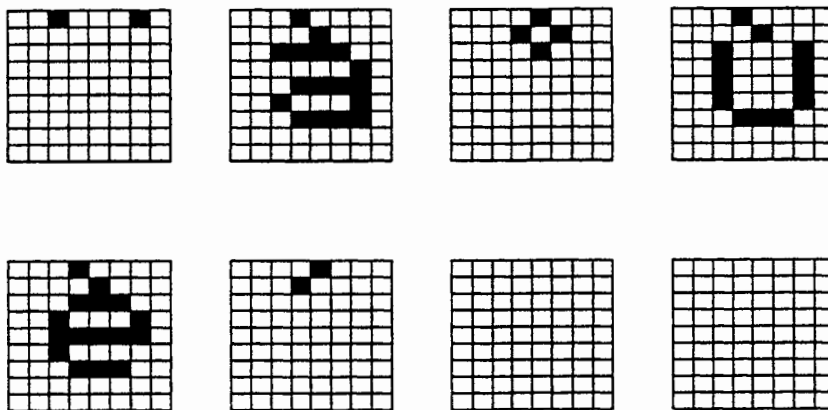
Special National Characters CP 526: 9 * 8

3.13.3.3 Cyrillic Characters

60H	61H	62H	63H
64H	65H	66H	67H
68H	69H	6AH	6BH
6CH	6DH	6EH	6FH
70H	71H	72H	73H
74H	75H	76H	77H
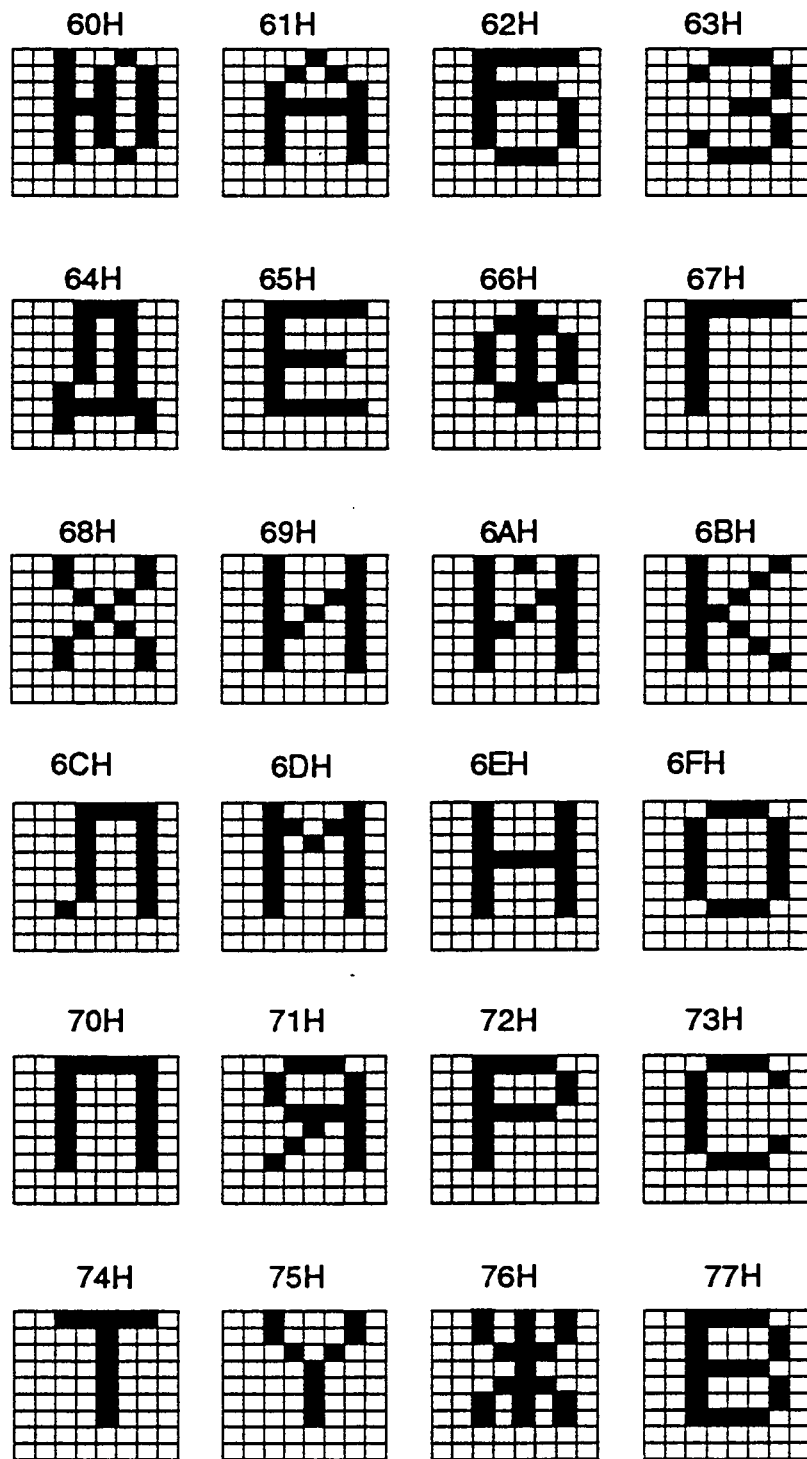
Cyrillic Characters CP 526: 9 * 8
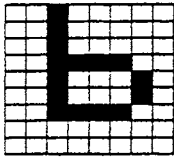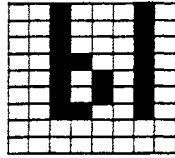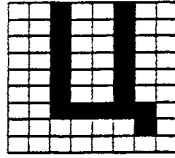
78H	79H	7AH	7BH
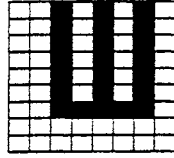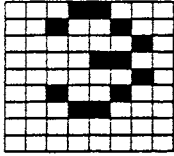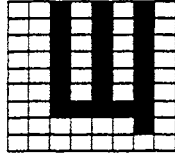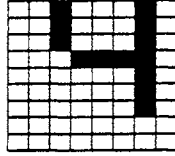
7CH	7DH	7EH	7FH
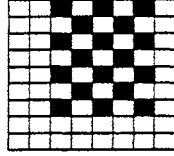
24H	27H		
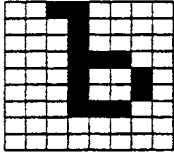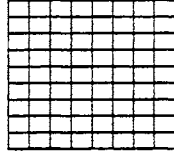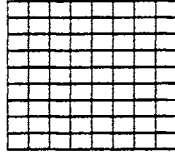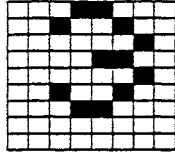
Cyrillic Characters CP 526: 9 * 8

3.13.3.4 Symbols That Can Be Loaded
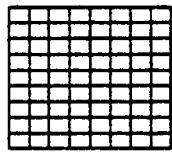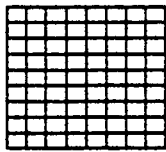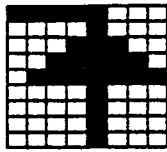
80H =128	81H =129	82H =130	83H =131
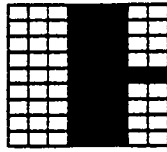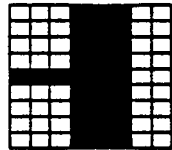
84H =132	85H =133	86H =134	87H =135
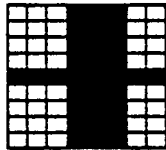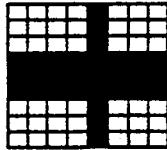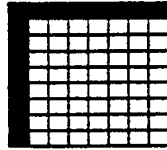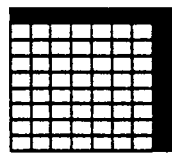
88H =136	89H =137	8AH =138	8BH =139
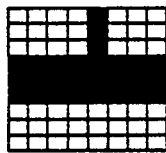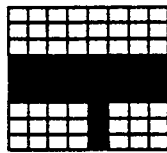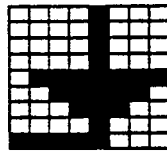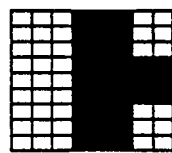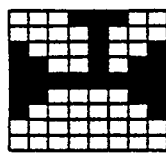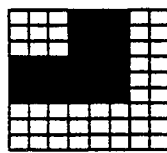
8CH =140	8DH =141	8EH =142	8FH =143
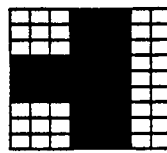
90H =144	91H =145	92H =146	93H =147
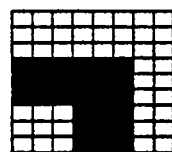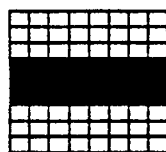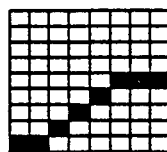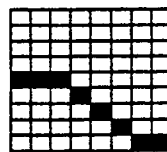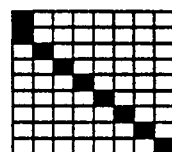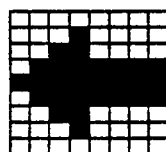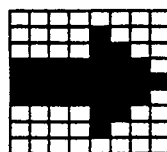
94H =148	95H =149	96H =150	97H =151
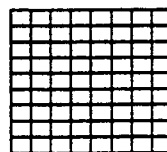
Symbols That Can Be Loaded CP 526: 9 * 8

98H =152	99H =153	9AH =154	9BH =155
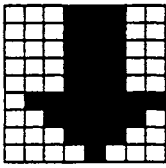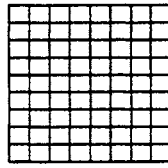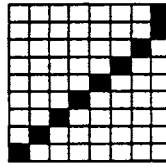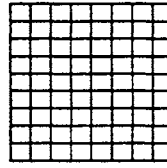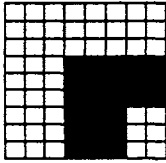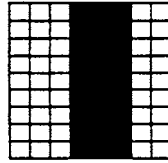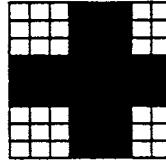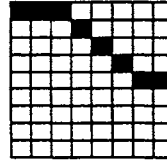
9CH =156	9DH =157	9EH =158	9FH =159

A0H =160	A1H =161	A2H =162	A3H =163
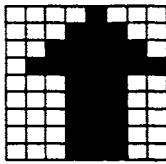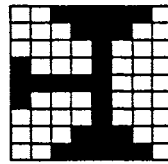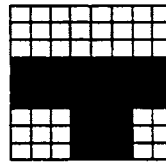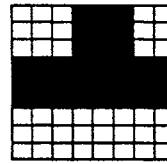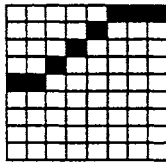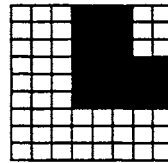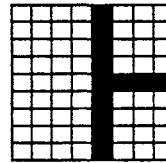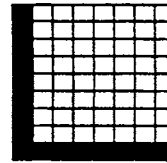
A4H =164	A5H =165	A6H =166	A7H =167
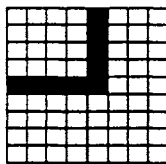
A8H =168	A9H =169	AAH =170	ABH =171
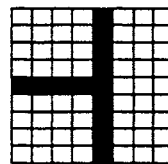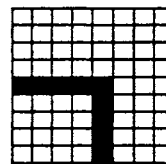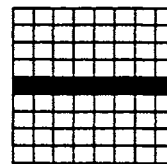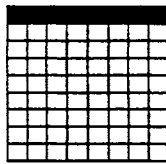
ACH =172	ADH =173	AEH =174	AFH =175
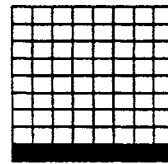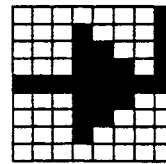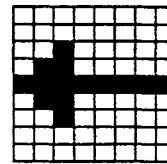
Symbols That Can Be Loaded CP 526: 9 * 8

B0H =176 B1H =177 B2H =178 B3H =179
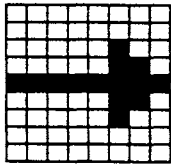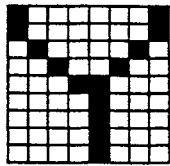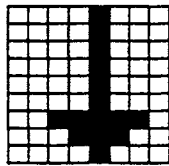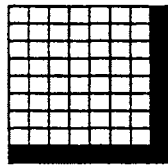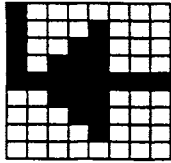
B4H =180 B5H =181 B6H =182 B7H =183
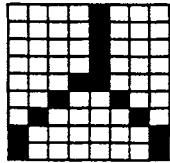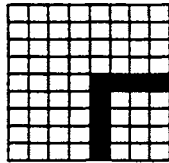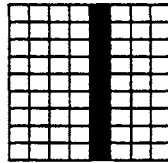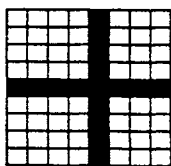
B8H =184 B9H =185 BAH =186 BBH =187
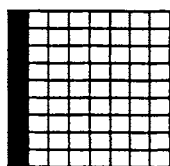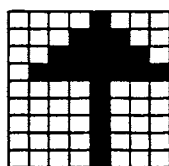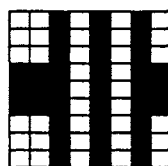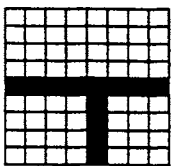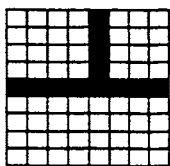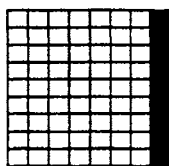
BCH =188 BDH =189 BEH =190 BFH =191
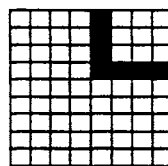
C0H =192 C1H =193 C2H =194 C3H =195
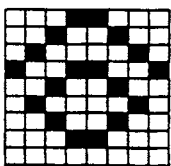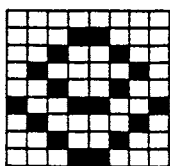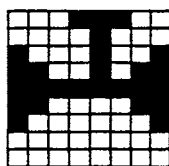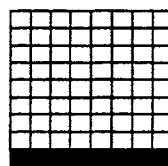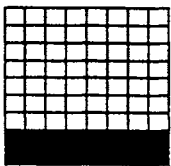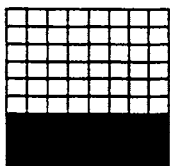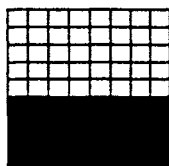
C4H =196 C5H =197 C6H =198 C7H =199
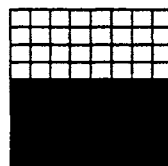
Symbols That Can Be Loaded CP 526: 9 * 8
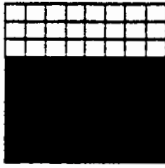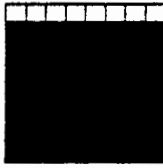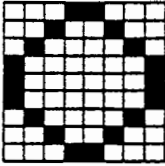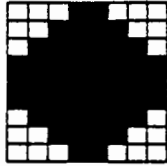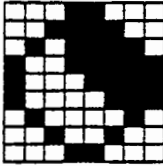
C8H =200	C9H =201	CAH =202	CBH =203
CCH =204	CDH =205	CEH =206	CFH =207
D0H =208	D1H =209	D2H =210	D3H =211
D4H =212	D5H =213	D6H =214	D7H =215
D8H =216	D9H =217	DAH =218	DBH =219
DCH =220	DDH =221	DEH =222	DFH =223
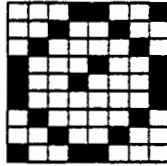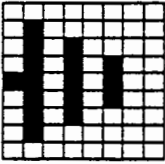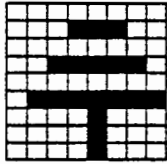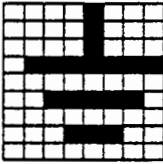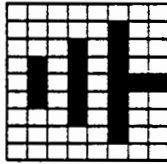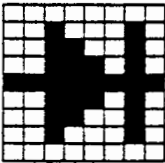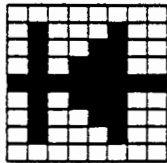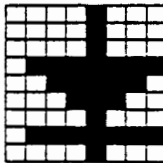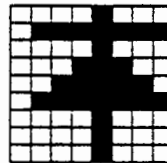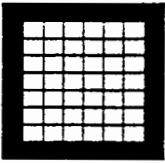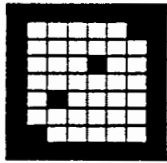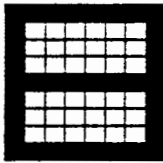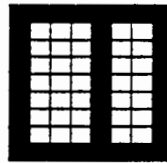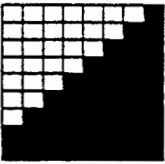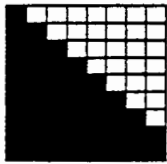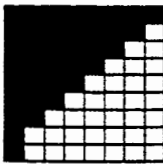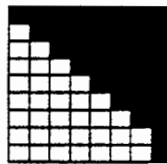
Symbols That Can Be Loaded CP 526: 9 * 8

E0H =224	E1H =225	E2H =226	E3H =227
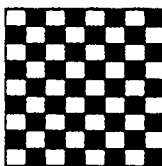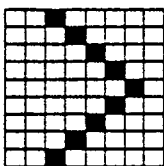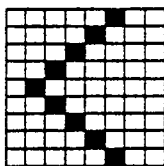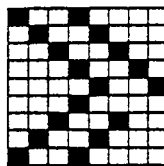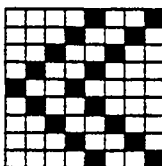
E4H =228	E5H =229	E6H =230	E7H =231
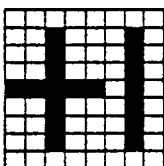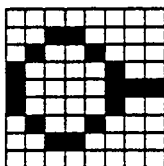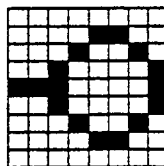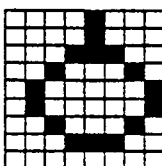
E8H =232	E9H =233	EAH =234	EBH =235
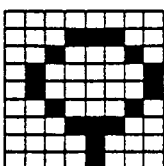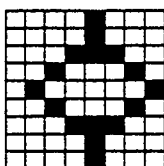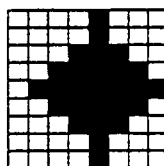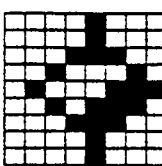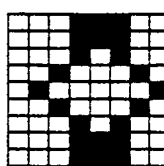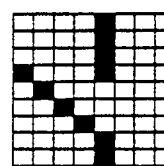
ECH =236	EDH =237	EEH =238	EFH =239
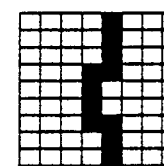
F0H =240	F1H =241	F2H =242	F3H =243
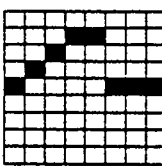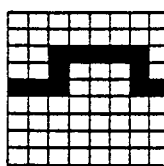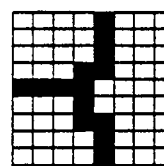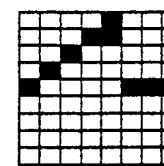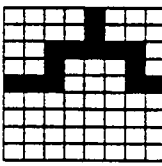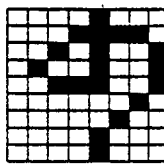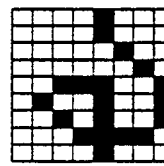
F4H =244	F5H =245	F6H =246	F7H =247
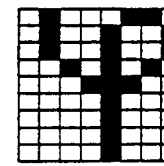
Symbols That Can Be Loaded CP 526: 9 * 8

F8H =248　　**F9H =249**　　**FAH =250**　　**FBH =251**
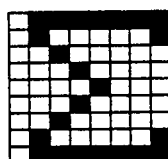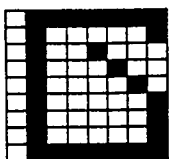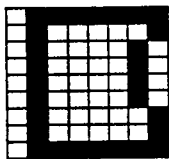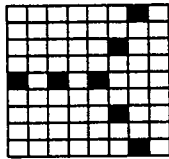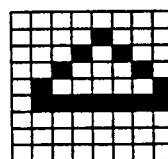
FCH =252　　**FDH =253**　　**FEH =254**　　**FFH =255**
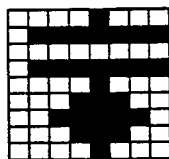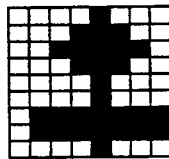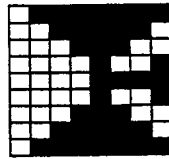
Symbols That Can Be Loaded CP 526: 9 * 8

Contents

Communications Processor
CP 526
Contents

C79000-B8576-C217

4   Technical Description of the CP Expansion Board

4.1   Application

You can use the CP 526 expansion board with the 115 U/135 U/150 U/155 U programmable controllers. The CP expansion board supplements the video terminal interface of the CP 526 basic board .

The expansion board may only be operated in combination with the ventilated basic board (color version).

The expansion board displays measured data in the form of curves or bar charts. It also displays various text sizes and changes color when limit values are exceeded.

4.2   Mechanical Construction

The CP expansion board is a printed circuit board of double-height Eurocard format, which makes it compatible with the 902 modular packaging system. The width of the front panel is approximately 20 mm. The expansion board has an 8-pin backplane connector of range 2, through which the supply voltage is applied. A 64-pin ribbon cable with a plug connects it to the CP 526 basic board. The CP expansion board must be attached to the CP 526 basic board to form one mechanical unit (see 'Installation Guidelines').

There is a rectangular receptacle on the front panel of the CP expansion board for the memory submodule (EPROM or RAM) containing the application program (see Figure 4-3).

Figure 4.1 was deleted.

EPROMᴊ4

Backplane Connector
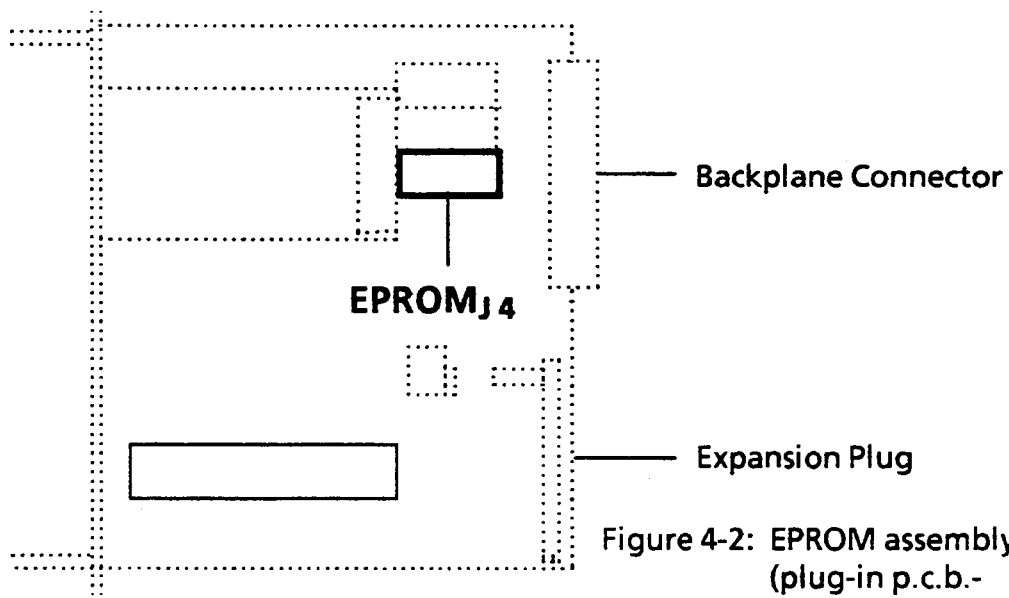
Expansion Plug

Figure 4-2: EPROM assembly
(plug-in p.c.b.-
reduced scale)

Submodule Receptacle
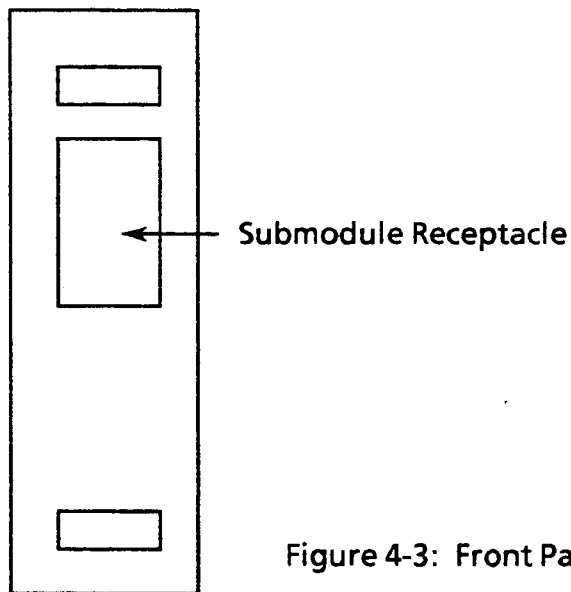
Figure 4-3: Front Panel (reduced scale)

4.3   Memory Submodules

The memory submodules of the CP expansion board contain all of the application programs needed to display graphics, and the associated internal memory management information.

You can use either EPROM or RAM submodules. When it is plugged in, the RAM submodule is battery-backed through the backplane bus of the programmable controller. If either the RAM submodule or the CP 526 is unplugged, all application programs are lost.

4.4   Technical Specifications

4.4.1   Operating and Environmental Conditions

Degree of protection to DIN 40050 /IEC 144	IP 00
Permissible environmental temperature	
- operating (operation possible only with forced ventilation)	0 to 55° C
- nonoperating	- 40 to + 70° C
Humidity class to DIN 40050	Code F, noncondensing
Operating altitude	Up to 1500 m above sea-level

4.4.2 Mechanical and Electrical Specifications

Weight (without memory module)	0.35 kg
Printed circuit board format	Double-height Eurocard (160 mm x 233.4 mm)
Front panel width	20.32 mm
Backplane connector	ES 902, range 2, 8-pin
Ribbon cable with plug for connection to the basic board	64-pin connector
Supply voltage	+ 5 V, tolerance 5%
Current consumption	2.3 A

4.4.3 Logic Data

Graphics processor	HD63484 from Hitachi

Memory configuration

- display refresh memory	3 color levels, 16 kwords each
- RAM	8 kbyte
- operating system EPROM	32 kbyte
- memory submodule	128 kbyte (maximum) for EB 6ES5526-3LA21
	256 kbyte (maximum) for EB 6ES5526-3LA22

Color CRT Control

- principle	Pixel graphics
- display refresh frequency	60 Hz
- line frequency	30.84 kHz
- resolution	640 x 396
- colors	Black, red, green, blue, yellow, cyan, magenta, white

4.5 Installation Guidelines

4.5.1 Module Slots in the Programmable Controller

Subrack/Controller	Slot-in module	Slot No.
PLC S5-115U		
CR 700-2 ER 701-3 CR 700-3	Basic module [1] Basic module [1] [2] Basic module [3] Expansion board [4]	0, 1, 2, 3, 4, 5 0, 1, 2, 3, 4, 5, 6 0a, 0b, 1a, 1b, 2a, 2b, 3, 4, 5 0b, 1b, 2b
PLC S5-135U		
ZG 135 U EG 185 U	Basic module [7] Basic module [7]	11, 19, 27, 35, 43, 51, 59, 67, 19, 27, 35, 43, 51, 59, 67, 75, 83, 91, 99, 107, 115, 123, 131, 139
PLC S5-150U		
ZG 150 U EG 185 U	Basic module [7] Basic module [7]	3 [5], 11 [5], 19 [5], 107, 115, 123, 131, 139 [6], 147 [6] 19, 27, 35, 43, 51, 59, 67, 75, 83, 91, 99, 107, 115, 123, 131, 139
PLC S5-155U		
ZG 155 U	Basic module [7]	19, 35, 43, 51, 59, 75, 83, 91, 99, 107, 115, 123, 131 139 [6], 147 [6]

[1] Adapter casing is required! One module per casing.
[2] Not with Interface Module 311!
[3] In versions with natural ventilation, 1 module per slot only.
[4] Only together with basic module in 1 casing.
[5] Only in conjunction with jumpering module 756.
[6] Only possible after changing jumpers on the bus PCB.
[7] The expansion board is inserted into the next highest slot.

Table 4-1: Module Slots of the Expansion Board

4.5.2   Inserting and Removing the CP 526 Expansion Board

CAUTION !
Be sure that the power is off before inserting or removing the CP 526. Never insert or remove the CP 526 with the power on.

4.5.3   Connecting the CP Expansion Board to the Basic Board

To use the CP 526 expansion board, you must connect it electrically and mechanically to the CP 526 basic board. Before doing this, make sure that all the points described in section 3.9 (Startup) have been checked and that the basic board is fully functional. Then proceed as follows:

1. Unplug the monitor cables from the CP 526 basic board and remove the basic board from the controller rack.

2. Check the pins of the 64-pin plug connector on the basic board, making sure that none are bent.

3. On the CP 526 basic board, loosen the two bolts of backplane connector X1 and the outer bolt of backplane connector X2 and remove the nuts.

4. With the screws that you loosened in step 4, fasten the two enclosed threaded spacer pins No. 1 on backplane connector X1 and the spacer pin No. 2 on backplane connector X2 (see Figure 4.4). The spacer pins must be attached on the component side of the board as shown in Figure 4-4.
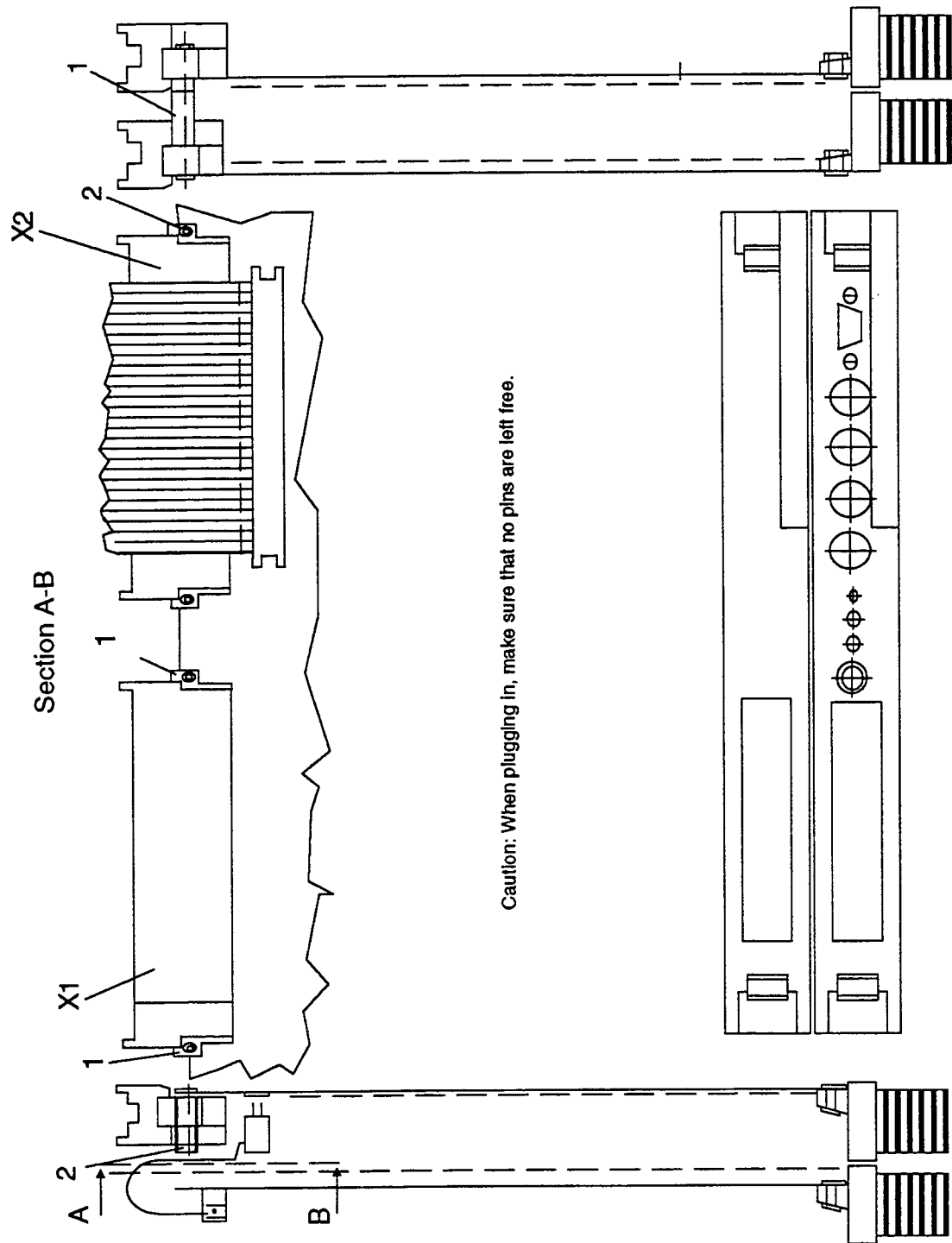
Figure 4-4:   Assembly Directions

5. Plug the 64-pin ribbon connector of the expansion board into the 64-pin plug connector of the basic board as shown in Figure 4-4. Check the plug connection to be sure that every pin is inserted correctly.

6. Place the two boards together, as shown in Figure 4-4. Plug the expansion board spacer pins into the basic board spacer pin holes.

4.6   Jumper Settings

All jumpers are set at the factory and should not be changed.

4.7   Startup

To start up the CP expansion board, follow these steps:

1. Mount the CP expansion board as described in section 4.5 (Installation Guidelines).

2. Check the expansion board jumper settings.

3. Make sure that the expansion board is properly equipped with the EPROM operating system. The EPROM firmware (No. S79200-B1004-A901) is located at J4 on the expansion board.

4. Switch off the power supply of your controller rack, and insert the connected boards (expansion board and basic board) into the specified slot.

5. Insert a RAM submodule into the receptacle on the expansion board. The RAM memory submodule of the basic board must remain plugged in.

6. Check the position of the front panel switch of the basic board. The switch must be in the "RUN" position to exchange data with the controller.

7. Reconnect the process keyboard and turn the keylock switch to the "ON" position.

8. Reconnect the monitor.

9. Turn on the power supply. The CP reports with the START screen form. Type "START" on the process keyboard and press the < RETURN > key. The CP 526 displays the same form again. The current system data is displayed between the message line (top line) and the command line (bottom line).

    A red or green LED lights up on the front panel of the basic board, depending on the position of the switch. If this does not happen, recheck the above settings.

10. Once startup is complete, you can begin normal operator-process communication and visualization, or, using the programmer, you can generate new commands.

4.8   Connector Pin Assignments

The following subsections specify the electrical interfaces of the CP expansion board (shown in Figure 4-2).

4.8.1   Backplane Connector

The expansion board is connected to the backplane bus of the controller by an 8-pin, range 2 backplane connector. This connector is used exclusively for the power supply to the expansion board:

- + 5 V through pin z2
- 0 V through pins b2 and b32

4.8.2   Memory Submodule Connector

The expansion board has a 64-pin female connector to attach it to the memory submodule. The pinout of this connector is shown in Table 4-2.

	c	b	a
1	SADB 12	Ground	+ 5 V
2	SADB 0	SADB 1	SADB 2
3	SADB 3	SADB 4	SADB 5
4	SADB 6	SADB 7	SADB 8
5	SADB 9	SADB 10	SADB 11
6	SADB 13	SADB 14	/RDY
7	/W	SDB 8	SDB 9
8	SDB 10	SDB 11	SDB 12
9	SDB 13	SDB 14	SDB 15
10	SDB 0	SDB 1	SDB 2
11	SDB 3	SDB 4	SDB 5
12	SDB 6	SDB 7	----
13	/CS1 Submodule1	/CS3 Submodule1	----
14	/CS2 Submodule 1	/STBY; /CS4	----
15	UCMOS	----	---
16	+ 5 V	Ground	K5 Submodule1

Table 4-2: Memory Submodule Connector

4.8.3   Expansion Board Connector

A 64-pin ribbon cable with connector establishes the connection between the expansion board and the basic board. The pinout of this connector is shown in Table 4-3.

1	AD 1		2	AD 0
3	AD 3		4	AD 2
5	AD 5		6	AD 4
7	AD 7		8	AD 6
9	AD 9		10	AD 8
11	AD 11		12	AD 10
13	AD 13		14	AD 12
15	/CS 4 Submod.2		16	AD 14
17	/CS 3 Submod.2		18	AD 15
19	/CS 2 Submod.2		20	INT 2
21	/CS 1 Submod.2		22	K5 Submod. 2
23	/WR		24	/SEL-EPROM
25	/RD		26	/VSYNC
27	ERDY		28	Ground
29	/PCS 5		30	+ 5 V
31	RGB		32	Ground
33	CLKPIX		34	Ground
35	GREEN		36	Ground
37	LOAD SR		38	Ground
39	Ground		40	Ground
41	+ 5 V		42	Ground
43	BLUE		44	Ground
45	RED		46	+ 5 V
47	+ 5 V		48	+ 5 V
49	/DSI		50	RESET
51	DB7		52	DB6
53	DB5		54	DB4
55	DB3		56	DB2
57	DB1		58	DB0
59	AD 16		60	/DT-R
61	/ERAM		62	VBATT
63	Ground		64	Ground

Table 4-3: Expansion Board Connector

4.9   Process Visualization with the CP Expansion Board

The CP expansion board enhances the capability of the CP 526 basic board by allowing you to display and observe the following:

- curves, as a function of time
- bars, with limit-dependent color change
- digital measured values in magnified form

The CP 526-E must be used with a high-resolution monitor. The pixel graphics capability is only meaningful with a high-resolution monitor. The display output of the expansion board (pixel graphics) is combined with the display output of the basic board (character graphics). The display of the expansion board has a resolution of 80 characters by 22 lines. It assumes that the standard CP parameters (character size, 8 x 18 pixels) are used. The expansion board display (lines 2-23) is four lines smaller than that of the basic board. The first line of the CP 526 is the standard message line. It can be suppressed, but not used for pixel graphic representation. The last three lines are for function key labels and the command line.

Message line	1
PIXEL GRAPHICS SCREEN 22 lines	2
Command line	24
F1	25
F16	26
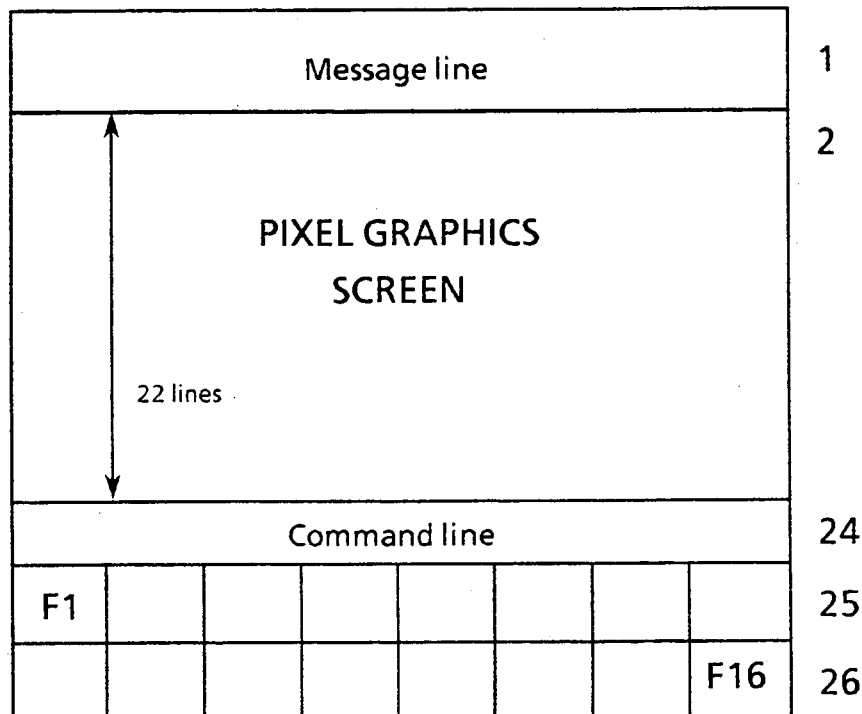
Figure 4-5: Standard Organization of the Screen (not to scale)

Instructions to the expansion board do not always result in a pixel graphics display. Except for the labeling of the axes, curve windows always contain a character graphics display. The current symbol set is used to construct the display. The CP expansion board writes the characters for the axes into the character generator during initialization.

This area also contains the seven characters of the upper half of the symbol set. If you are not using pixel graphics functions, these symbols can be freely defined with COM.

CAUTION !
If you are using the CP 526 expansion board, you should not change the upper seven characters of the upper half of the symbol set (F9H to FFH), because they are assigned the characters required for representing the curve axes.

The axes and their labels for curve windows and pixel graphics bars are white. You can select the background color from the CP parameters. An exception to this rule is the heading for curve windows, which is always displayed in black on a white background.

4.9.1   Curves

The expansion board allows you to define curve windows within a display. A curve window can contain up to six curves. Each curve window displays a set of coordinate axes with definable labels. The following are examples of possible labels:

●   Header text with a maximum of 24 characters (only 20 characters for the smallest window)

●   Curve text, in the defined curve color for the individual curve (the number of possible texts depends on the window size). If the number of curves in a curve window exceeds the number of possible texts, only the first curve texts are displayed.

●   Text on the time axis with a maximum of three characters

●   Axis labeling on the horizontal time- and the vertical y-axis, with a maximum of four digits per displayed value (with negative values only three digits are possible, e. g.
labeling of the time axis in the range -999 to 0,
and labeling of the y-axis in the range -999 to 9999)

●   Coordinate axes can be displayed with a grid network, so that the measured values can be read off quickly and reliably

A detailed description of curve windows is given in the Application Example (Chapter 5 in this Manual).

The expansion board updates all curves in a window at cyclic intervals. During updating, the CP 526 fetches the data block (DB) that has been prepared by the controller. The measured values displayed up to that time are shifted to the left by the CP 526, and the most recent values are added to the curves. Displayed measured values that extend beyond the left edge of the window are not stored by the CP 526.

4.9.1.1   Interface to the Programmable Controller

The CP 526 does not store old curve values. You can only observe the measured values displayed in the form of curves as long as the curve window appears on the screen. To display old values while you are constructing a new display, make sure that the controller program saves the required values and then makes them available to the CP 526 at the right time.

All measured values (both current and old) are in a data block (DB). The DB contains data words (DWs) and has the same structure for each curve window. The DB has the following structure:

- DW0   - check word (number of curve and number of old values)
- DW1-DW6   - current curve values in fixed-point format, 16 bits.

The following data words are needed if old values are to be displayed. To store values in this data block, use function block FB10:CURVE (see Application Example, Chapter 5).

- DW7   - reserved
- DW8   - counters managed by FB10: CURVE. These are used by the
                    CP 526 for time checking
- DW9   - used by FB10:CURVE

Old values are stored beginning at DW10. A DB with a maximum length of 256 words can be transferred to the CP 526 with expansion board. This limits the number of values that can be stored. If there are six curves, a maximum of 40 old data points per curve is possible in 16- bit, fixed-point format. If only the first curve exists, the number is increased to 240. If only the sixth curve exists, the ranges of old values for curves 1 through 5 will still exist, but there will be only 40 old values.

A detailed specification of this data block is given in the Appendix.

When displaying curves, the CP 526 allows three types of updating:

- Updating without displaying old values
- Updating while displaying old values
- Updating initiated by the controller

4.9.1.2   Updating without Displaying the Old Value

If you do not need to display old values, the CP 526 only requires the first seven data words (DW0 - DW6) from the controller. This is an easy way of ensuring that the required values are stored in the appropriate DWs. The right-hand byte of DW0 (DR0) contains the number of old values and must be set to zero; the left-hand byte of the DW0 (DL0) contains the number (in binary) of curves.

Bit	15	14	13	12	11	10	9	8
DL0	0	0	Curve 6	Curve 5	Curve 4	Curve 3	Curve 2	Curve 1
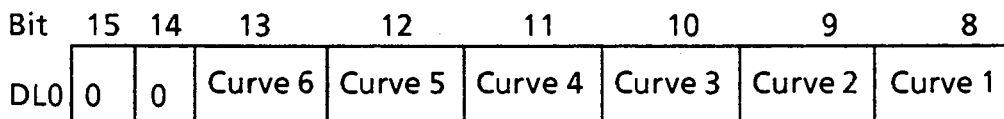
To display , for example, only one curve, use two data words as follows:

- DW0: KM 00000001 00000000   - first curve exists with no old values
- DW1: KF 100   - measured value to be displayed = 100

Note that the bit position in the DL0 above corresponds to the DW number of the displayed measured value as follows:

- DW0: KM 00100000 00000000   - sixth curve exists with no old values
- DW1 to DW5   - not evaluated by the CP 526
- DW6: KF 200   - measured value to be displayed = 200

A value is displayed only if the curve with this number has also been defined with COM.

4.9.1.3   Updating While Displaying the Old Value

The CP 526 with expansion board displays the old values collected by the programmable controller once while the display is being constructed. The CP 526 needs the entire data block (DW0 - DW255) that holds the old values. As soon as the old values are displayed, the CP 526 starts to update. Once updating begins, the CP 526 requests the current curve values and any required supplementary information (DW0 - DW9) from the controller.

The DB must be structured as specified in the Appendix. When updating old values, use function block FB10:CURVE. As an input, FB10:CURVE receives the current values (in DB: DW1- DW6), stores the old values in the specified format, and maintains the counters required by the CP 526. This function block is described in detail in the Application Example (Chapter 5).

Example

In DB111, the most recent measured values are in DW1-DW6. The function block stores these values every second. It stores every tenth measured value as old information. The area containing the old values is filled after 400 seconds. After that, the oldest values are overwritten by the more recent values.

```
         :JU   FB10
NAME     :CURVE
DB       :    DB111                    TRANSFER DB
N        :    KM00000000 00111111      ALL SIX CURVES EXIST
AZ       :    KF + 40                  40 OLD VALUES
W1       :    KY111,1                  MEASURED VALUES ARE IN DB111
W2       :    KY111,2
W3       :    KY111,3
W4       :    KY111,4
W5       :    KY111,5
W6       :    KY111,6
TA       :    KT100,1                  ENTER EVERY SECOND
F        :    KF + 10                  EVERY 10TH VALUE IS OLD VALUE
TI       :    T 10                     TIMER 10 IS USED
```

The example above describes a curve window configured with COM. DB111 begins at DW0 and is specified as the transfer block. With curve windows using COM you can specify the values in the controller that you want to be dis-played as curves.

4.9.1.4   Updating Initiated by the Controller

This is a special case and is not common in practice. COM allows you to define a curve window and then specify the controller as the updating device. To do this, use the PIXELGRAPH. BACKGROUND.UPD function (see Application Example, Chapter 5). You must specify the display, the defined window number and the DB that contains the old values. Update the curve window using an interprocessor communication flag, or by calling the SEND-DIRECT data handling block (see subsection 3.3.3).

If the controller initiates this job, the CP 526 displays the collected old values in the curve window.

Before starting, set the DB containing the old values to its full length (0 to 255). Of course, this is only meaningful if the relevant display with the curve window is onscreen.

This type of updating is used when an updating rate of one second is not fast enough for visualizing fast processes. These rapid events are sent to the controller and stored as old values. The old values are then collected for display by the CP 526. You can observe rapid processes, at least at random points, using this method.

The DB that contains the old values has the same structure as the curve described in section 4.9.1.3, and can be processed with FB10:CURVE (see Application Example). Only the right-hand byte of DW7 (DR7) is required additionally by the CP 526. This byte contains the number of pixels that should separate two measured values. In other types of updating, this value is determined from the ratio of the updating time to the observation time. However, since this type of update is intended specially for fast processes, there is no point in working with update rates in the seconds range. The update time is therefore treated separately from the observation period and, instead, the numbers of pixels are transferred in the DB.

Example:

Defining a curve window with COM:

Form: CURVE WINDOW

Pxlgra. disp.:   DEMO
Window: 1

WINDOW SIZE: 80 x 22
HEADER: PLC upd. (time in 100 msec)
UPDATE RATE: PC
STEP SIZE of time axis: 4   TIME UNIT: irrelevant

Function F36.0 is then defined as PIXELGRAPH. BACKGROUND. UPD:
(see Application Example)

Form: PIXELGRAPH. BACKGROUND. UPD

Window 1 of the pixel graphics display DEMO
The data are fetched by CPU No. 1 from DB111.

The DEMO function is defined as display output command with job number 10.

Form: DISPLAY OUTPUT COMMAND

Name: DEMO   A-No.: 10

The PC program then updates the display automatically.

SEND-DIRECT with A-No. = 10 (job number) invokes the defined curve window.

To update the curve window every 2 seconds, the interprocessor communication
flag is inverted every second. The flag therefore changes from 0 to 1 every two
seconds, i. e. the display is updated every 2 seconds with the collected old data.

You must define flag byte FY36 as an interprocessor communication flag in DB1.

```
                 :
          :AN    T 10            THE INTERPROCESSOR COMMUNICATION
          :L     KT100.0         FLAG IS INVERTED EVERY SECOND
          :SE    T10
          :JC    M001
          :JU    M002
M001      :AN    F36.0           THE INTERPROCESSOR COMMUNICATION
          : =    F36.0           FLAG IS INVERTED
M002      :
          :
```

The most recent value is always in DB111, DW1, and is updated every 100 msec.

```
                 :
          :JU    FB10
NAME      :CURVE
DB        :      DB111
N         :      KM00000000 00000001    ONLY THE FIRST CURVE IS SET UP
                 :
AZ        :      KF + 240        240 OLD DATA
W1        :      KY111,1         ALREADY STORED IN DB111
W2        :      KY0,0           THERE IS NO REMAINDER
W3        :      KY0,0
W4        :      KY0,0
W5        :      KY0,0
W6        :      KY0,0
TA        :      KT010.0         EVERY 100 MSEC., EVERY VALUE
F         :      KF + 1          CHANGES TO AN OLD VALUE
TI        :      T 11            TIMER 11 IS USED
```

The spacing, in pixels, between two measured values need to be entered only once:

```
          :C     DB111           THE SPACING IN PIXELS ( = 2) BETWEEN
          :L     KF + 2          OLD VALUES IS IN DB111, DR7
          :T     DR7
```

In this example, you must be sure that the labeling of the time axis agrees with the actual time.

The following relationships are valid and are prescribed by the controller program:

- Number of old values   - 240
- Spacing between two old values in pixels   - 2
- Updating time in the controller (dt)   - 100 msec.

The CP 526 specifies that there lie 80 pixels between the present time (i.e., 0) and the step size of the time axis (i.e., T1). This is true regardless of the window size.

To display all the old values, a curve window with 480 horizontal pixels (number of old values * spacing = 240 * 2) is necessary.

From the present time to T1, 40 measured values (number of pixels/spacing = 80/2) are entered.

This corresponds to a controller time of 4 sec. (dt * 40).

To be sure that the labeling of the time axis is correct, you must specify a corresponding value for T1 in COM. With T1 = 4, a labeling unit of 1 sec. must be used.

The time unit is not taken into account with this kind of updating.

4.9.1.5   Time Checks of the CP 526

The curve display is the only application where past values are displayed. You must therefore make sure that the values over the time axis agree with real time. Problems can occur if the values cannot be transferred from the controller with every updating of the CP 526, or if the CP 526 cannot adhere to the updating time due to the large amount of updating required. For this reason, the expansion board makes a time check during the updating of the curves.

The following are two examples:

● The controller carries old values and tests the counters in the DB.

● The controller does not carry old values and tests on the internal CP 526 clock.

If the controller carries old values, as specified above, the counters controlled by FB 10:CURVE (DW8) are used for time checking. The updating time configured with COM is not applicable here. This time determines at what time intervals the CP 526 reads the DB with the current curve values. The expansion board compares the current counter values with those of the previous update. It assumes that the measured values in the controller are updated in the time intervals that were defined with COM. If this is not the case, the curve update depends on the controller. If the measured values in the controller are updated faster than specified in COM, the expansion board interpolates the measured data it could not read. When this happens, error message W40 appears on the screen. If the measured values are updated by the controller slower than specified, this update time is adhered to.

CAUTION !
If the update time that you specify in COM is not identical to the update time of the measured values (call of FB10:CURVE), the time axis labeling no longer agrees with the real time.

If the controller is not carrying any old values, the counters in the DB do not exist, and you can only check for the internal CP clock. This check is made to see whether the update time defined with COM is strictly observed. If it is not, the controller interpolates the measured values, and error message W40 appears.

4.9.1.6   Displaying the Measured Values On the Screen

You can define up to six curves for one curve window. This section discusses the range of values of the curves and the associated y-axis labeling.

In the simplest case, all the measured values to be displayed as a curve have the same range of values, and this range of values is used to label the y-axis. Beginning at the time axis, the defined infill pattern should appear below the curves. No limit values are defined in COM. The form below allows you to define three curves in the range of values 0 to 400, and select the largest curve.

Form: CURVE WINDOW

WINDOW SIZE: 80 x 22
UPDATE RATE: 1 sec
STEP SIZE of time axis: 80   TIME UNIT: sec
SMALLEST SCALE VALUE: 0   STEP SIZE of y-axis: 100

the GREATEST SCALE VALUE is calculated as follows:

GREATEST SCALE VALUE =
SMALLEST SCALE VALUE + Y x STEP SIZE of y-axis
where

Y = 4 for window sizes 80 x 22 and 40 x 22,
i. e. GREATEST SCALE VALUE = 400

and

y = 2 for window sizes 80 x 11, 40 x 11 and 26 x 11,
i. e. GREATEST SCALE VALUE = 200

Form: CURVE

The following holds good for all three curves:

SMALLEST OUTPUT VALUE: 0
GREATEST OUTPUT VALUE: 400

If you call this CURVE form to the screen during operation, and the PC provides the OUTPUT VALUES as specified in COM (see subsection 4.9.1.1). The current values in the DB are displayed in the curve window and you can simply read them off from the screen. If a curve value is not in the defined range of values, the message W35 appears and reduces or increases this value to the smallest or largest output value, respectively.

If you want the measured values displayed on a different range during output, follow one of these procedures:

●   Specify the desired y-axis labeling for the curve window.
●   Specify the curve range values in the controller. The CP 526 handles the display for you.

The smallest output value of the curve is displayed at the lower window edge, and the largest output value is displayed at the upper window edge. The CP 526 calculates the necessary values for length and offset. The following example illustrates this application:

The measured values are in the range of values 0 to 400 in the controller, but the display has a temperature of -20 to + 20° C. The desired filling pattern should lie beneath the curves, beginning at 0° C . Below -10 and over + 10° C, the curve should be completely filled in. To display the following PROGRAM LIMITS form, press the < F2 > function key from the CURVE WINDOW form.

Form: CURVE WINDOW

SMALLEST SCALE VALUE: -20   STEP SIZE of y-axis: 10

Form: PROGRAM LIMITS

REF. VALUE: 0
LOWER LIMIT: -10
UPPER LIMIT: 10

Form: CURVE

The following holds good for all three curves:

SMALLEST OUTPUT VALUE: 0
GREATEST OUTPUT VALUE: 400

To define three curves in the example, the following is true:

- smallest output value: 0
- greatest output value: 400

If you call this CURVE form to the screen during operation, the current values in the DB are displayed.

Values from the programmable controller
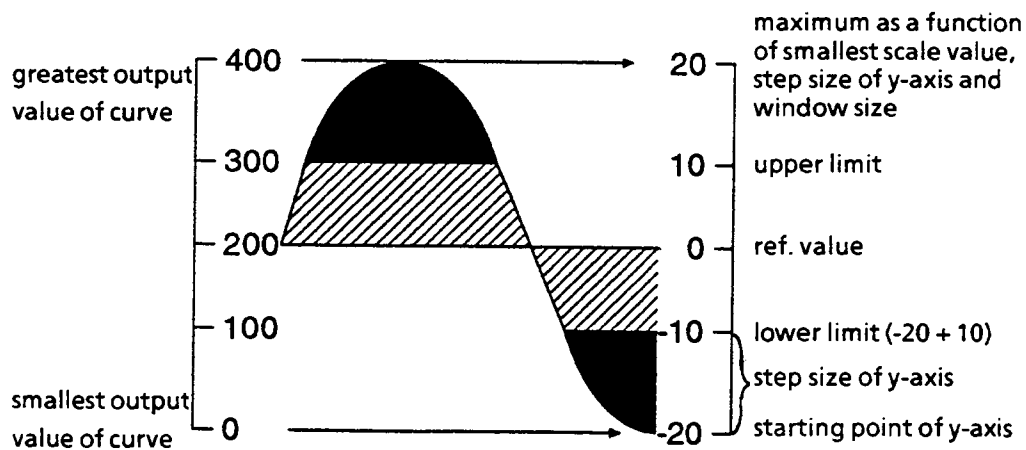
Figure 4-6: Displaying the Measured Values on the Screen

You can also define curves with different value ranges, to make up one curve window. The CP 526 automatically projects the range of values in the curve window. The y-axis labeling depends on the ordinate values that you specified for the curve window. If you want an extra label on the y-axis, you can enter the labeling yourself with an appropriate background display.

4.9.1.7   Updating Time and Visualization Period

When defining curve windows with COM, you can select the update time and the visualization period. This section provides information on defining the curve window most suitable for your curve display. The visualization period is not specified directly, but is calculated from the window size, the step size on the time axis, and the time unit.

Visualization period = X * step size, in time units, where:

- X = 6   for window size 80 by 22 and 80 by 11,
- X = 3   for window size 40 by 22 and 40 by 11,
- X = 2   for window size 26 by 11,

The following are examples:

Window Size	Step Size	Time Unit	Visualization Period
80 x 22	1	min.	6 min.
80 x 11	80	sec.	480 sec.
40 x 22	1	hr.	3 hr.
40 x 11	100	sec.	300 sec.
26 x 11	120	sec.	240 sec.

The visualization period is not important in these situations. This applies to all window sizes. What is important is the interaction of the update time (dt) and the step size x time unit (T1 in sec.). These specifications determine how many pixels (k) there are between two measured values. To obtain the desired display, you must configure the display so that 'k' lies between 1 and 4. 'k' does not need to be an integer.

For example, if you plot a 'k' of 4/3, the CP 526 shifts the curves twice by one pixel, and once by two pixels. In general, the following formula holds:

$k = 80 \times dt / T1$

Examples:

Configured with COM GRAPH			Result	
Window size	dt	T1	T_{max}	k
80 x 22	1 sec	10 secs	60 secs	8
80 x 22	2 secs	40 secs	240 secs	4
80 x 11	5 secs	50 secs	300 secs	8
40 x 11	2 secs	20 secs	60 secs	8
80 x 22	5 min	100 min	600 min	4
40 x 22	10 secs	2 min	6 min	6.67
26 x 11	30 secs	10 min	20 min	4
80 x 11	2 secs	100 secs	600 secs	1.6
40 x 22	5 secs	10 min	30 min	0.67
80 x 22	1 min	1 hr	6 hr	1.33
80 x 22	1 sec	80 secs	480 secs	1
80 x 22	10 secs	10 min	60 min	1.33
26 x 11	30 secs	40 min	80 min	1
40 x 22	30 secs	40 min	120 min	1
40 x 11	5 min	1 hr	3 hr	6.67

Table 4-4: Example of Defining Curve Windows

4.9.2   Bar Display

The bar display with pixel graphics allows you to define a color change that depends on a limit value. The position of the bar on the screen is defined by the values you enter in the ROW and COLUMN fields, and by the window size, and the type of bar.

If the type of bar is vertical, the maximum bar length is calculated as follows:

> window size * 4 characters

If the type of bar is horizontal, the maximum bar length is calculated as follows:

> window size * 10 characters

The largest output value is calculated as follows:

> smallest output value + window size * step size

The pixel graphics bar is displayed on the screen as follows:

- Smallest output value is the position of the bar (row, column)
- Largest output value is the position of the bar + maximum bar length

The other y-axis values (reference value, lower and upper limit) are also displayed.

The bar is displayed, starting at the reference point. If you have not defined a reference point on the screen, the CP sets the reference value equal to the smallest output value.

4.9.3   Displaying Values

Pixel graphics displays allow you to magnify the values so that you can easily read them from a distance.

The value display uses the character graphics symbol set. The characters have a resolution of 8 x 18 pixels. During enlargement, individual pixels are magnified. This includes pixels that have not yet been set. (Keep this in mind when positioning an image.) The position of a character should always be in the upper left-hand corner.

4.9.4   Pixel Graphics Texts

Pixel graphics text displays can be magnified in four different directions.

Appendix: Structure of the Data Block for Curve Measured Values

This appendix lists the parameters for a data block containing measured values to be displayed with time curves. The contents of the data block are shown on the next page, with the explanation of each parameter on the following pages.

Bit Number	15	8	7	0
Offset + 0	Number of curves n		Number of curves N	
1	Current value of the first curve			
2	Current value of the second curve			
3	Current value of the third curve			
4	Current value of the fourth curve			
5	Current value of the fifth curve			
6	Current value of the sixth curve			
7	Reserved		Only if PC updating	
8	Counter z		Update old value address	
9	Factor f		Reserved	
10	Old value of curve 1			
	Old value of curve 2			
	Old value of curve 3			
	Old value of curve 4			
	Old value of curve 5			
	Old value of curve 6			

Controlled by FB10: CURVE

Offset: Relative starting address of the data within the data block. In the manual this offset has been assumed to be zero. But this is not absolutely necessary. You can also set up the structure of the data block (DB) beginning at an arbitrary data word (DW) address. The addresses specified in the manual are relative to the offset. This allows you to keep the measured data for several curves within one data block. This is particularly useful if you are working without displaying the old values.

n: Number of curve plots displayed in the window as a bit field.

   The contents of data word 0 (DW0) are as follows:

   Bit 15: old values valid/ old values invalid   (0/1)
   Bit 14: old values format FPT16/BYTE   (0/1)
   Bit 13: curve plot 6 present /not present   (1/0)
   Bit 12: curve plot 5 present /not present   (1/0)
   Bit 11: curve plot 4 present /not present   (1/0)
   Bit 10: curve plot 3 present /not present   (1/0)
   Bit 9: curve plot 2 present /not present   (1/0)
   Bit 8: curve plot 1 present /not present   (1/0)

N: Number of old values per curve ($N = 0$ means that there are no old values).

   A total of 240 old values are possible with offset 0 and FPT16 format. N determines the maximum number of old values that can be stored per curve. It makes a difference whether a particular curve has been set up as curve 1 or as curve 6. The number of possible old values depends on the most significant curve bit set (bit 8 through bit 15 in DW0). Here, it makes no difference whether curves with lower-order curve numbers have been selected. The old value number must have an even number of digits.

Example:

- If only curve plot 1 exists, and the old values are in FPT16 format, the following applies:

$0 <= N <= 240$

- If curve plot 6 exists, and the old values are in FPT16 format, the following applies:

$0 <= N <= 40$

f: Factor that can be used to extend the visualization period (at the cost of resolution). $f = 1$ means that each measured value is also stored as an old value by the controller. Otherwise, each f-th measured value is acquired by the controller. When old values are displayed, therefore, the time between two displayed measured values is equal to f * update time.

Value range: $1 \leq f \leq 15$

z: Current counter value when acquiring old values. If it is incremented per measured value, and if $z = f$, an old value is stored.

Value range: $1 \leq z \leq f$

address: Relative address of the most recently stored old value within the memory reserved for old values. The following is the range of values:

Value range: $0 \leq \text{address} \leq N-1$

Current value of the i-th curve plot:

Must be in FPT16 format. These values must be kept current for the curves that are set in the DL0 (number of curves n).

Old values for curve i:

The old values for curve i are stored here. Through the format bit in n, you can store the old values in byte format. This allows you to store twice as many old values as with FPT16. With the valid bit in n, you can prevent old values from being output before the controller program has filled this buffer.

The old values are stored in a cyclic buffer. This means that the current value overwrites the oldest one. ADR refers to the most recent old value. The range of old values for curve i, with FPT16, is N words long, and begins at the following DW address:

$10 + i * N$

Example:

For curve 4 with 40 old values, the range of old values is DW170 to DW209.

Only if PC updating:

See subsection 4.9.1.4, (Updating Initiated by the Controller). For other applications, this byte has no significance.

Contents

Communications Processor   C79000-B8576-C217
CP 526
Contents

5 Example of an Application Program for the CP 526

The following example shows you, step-by-step, how to create a complete application program that runs on COM. Due to space limitations, all programmable functions cannot be explained here.

An overview of all programming possibilities, detailed information on screen forms (permissible entries, limit values, etc.) and a description of error messages are given in the Planning Guide for the Configuration Software COM GRAPH.

CP-526-specific error messages are explained in Appendix A of this Manual.

5.1 Set-up Requirements

To follow this example, you must set up the application program named "EXAMPLE". At least the standard symbol set with the appropriate resolution for your monitor (i.e.,SS1 for high and SS2 for low resolution) must be present in your program. Note that in the example, SSH and SSN are renamed to SS1 and SS2, respectively, during transfer.

The lexicon data is not necessary, but helpful. It enables plain text messages, instead of error numbers, to be output while you are testing the program.

5.2 Functional Description

This section describes how the finished application program should run on the CP.

Since the application program functions overwrite the standard CP 526 command START, the following display appears immediately after a total program transfer (TRANSFER FD -> CP function) and after you have either plugged in the process keyboard or selected the COM PROCESS KEYBOARD EMULATION function.
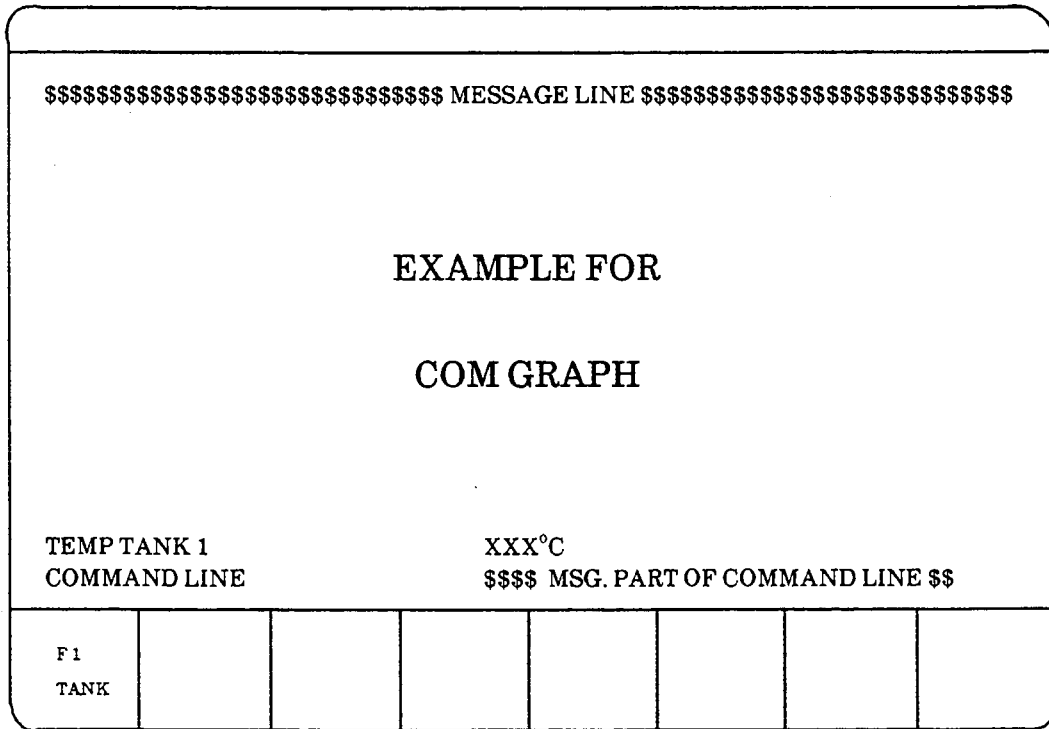
```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ MESSAGE LINE $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$



                        EXAMPLE FOR


                        COM GRAPH



 TEMP TANK 1                      XXX°C
 COMMAND LINE                     $$$$ MSG. PART OF COMMAND LINE $$

  F 1
  TANK
```

This display shows an example of a text output scaled up by means of the CP expansion board ("EXAMPLE FOR COM GRAPH").

The message line is just below the upper border of the display. The command line is just above the lower border.

The message line contains the latest message sent from the programmable controller. This definition is to be made when assigning parameters to the CP/OP. You can output all messages in the next display by either entering the command "MESSAGE DISPLAY," or by pressing the <F7 HELP> function key.

The lower part of the display shows a scaled up variable value (in this case the temperature of tank 1). This kind of representation is useful especially in the case of critical actual values which have to be recognized quickly by the operator.

) You reach the next display by pressing the <F1 TANK> function key.

The display on the CP monitor shows a process with two tanks. Over each tank is a value input field in which you may preset the liquid level and a value output field that shows the actual liquid level.

In addition, there are bars that show the liquid level in the tanks. To the right of the tanks, the tank temperatures are represented as a moving curve.

Note that in the controller, a program must be running that supplies the data blocks used in the program EXAMPLE with process values. Also, the CP and the controller must be connected to each other.

You can operate the display without curves and measured values, when the controller is not connected. In this case, the error message "Problems on PC line!" appears in the command line.

)

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
VENT1 C	VENT1 O	VENT2 C	VENT2 O	ENTER VALUE	ENTER TIME	MESSAGE	EXIT

You can open and close the left valve on the display with the <F1> and <F2> function keys. This also works without the controller. The color of the valve and the flashing attribute should change. At the same time, the display attribute of the value output field above the left tank changes. The <F3> and <F4> function keys change the right valve and the display attribute of the value output field above the right tank.

You can enter "VALUE = " into the command line with the <F5> function key.

You can enter a numerical liquid level for tank 1, and send this value to the controller by pressing the <RETURN> key.

The <F8 EXIT> function key brings you back to the START display.

You can find the meanings of all function keys on the process keyboard (e.g., <FIELD RIGHT>) in the CP description (see subsection 3.12).

The <F7> function key in the tank display causes the message display to be output. You can use the function keys to display messages (possibly divided into groups) and if desired, to acknowledge them.

) You can find specific instructions on operating the message display in the CP description (see subsection 3.3.6).

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
00.00.00 22:59:30  TITLE FOR GROUP 1




                                   $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```
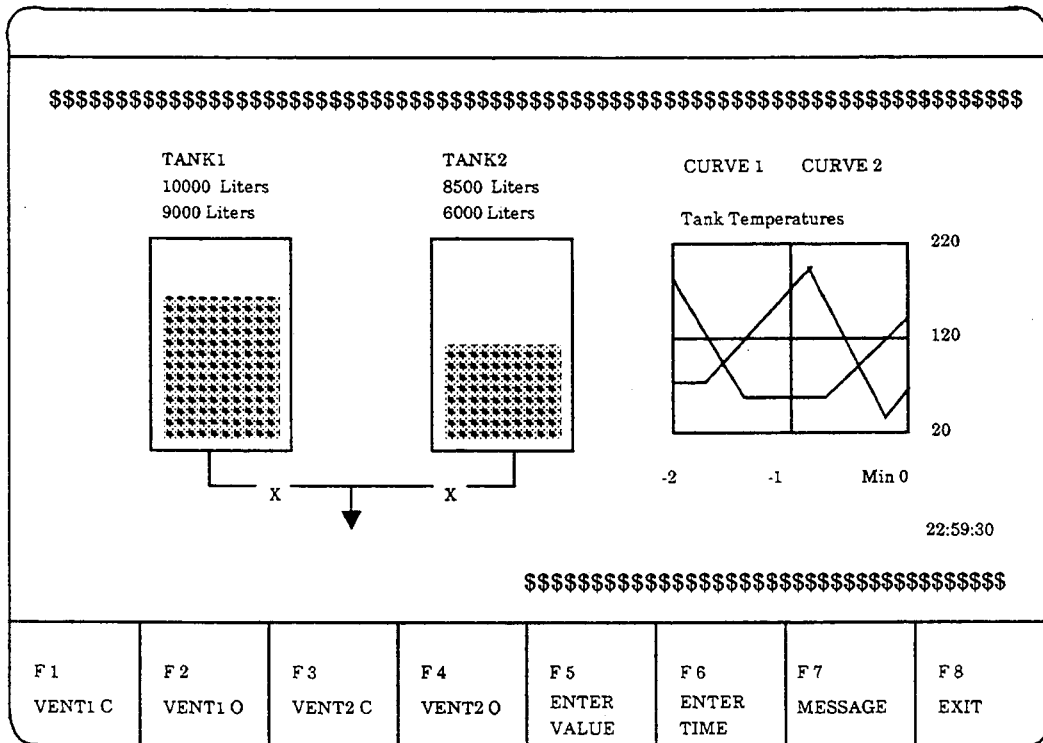
↑ Curs. ↑ ↓ Curs. ↓	↑ ↓	↑↑↑ ↓↓↓	↘ ↙	Q+++ Q+	Group 1 Group 2	Group 3 Group 4	Gr 1..4 End

To display the title as shown, press the <F6 GROUP 1> function key. You can display messages only if the controller sends messages to the CP with the send direct command. These messages are also present in the application program.

)

5.3 Assigning Parameters to the CP 526

Before you can begin programming, you need to assign the CP parameters.

You reach the ASSIGN CP/OP PARAM. form by pressing the <F6 CONTINUE> and <F4 ASSIGN CP/OP PAR> function keys in the PROGRAM USER DATA form.

It is not necessary to fill in the fields in the form to get to the CP/OP PARAM. ASSIGNMENT form. They are only needed when you program displays (e.g., segments).

NOTE:
In all screen forms entries can be made directly or selected by means of the <F7 HELP> function key.

```
-> PROGRAM SELECTION -> SELECTION ->                 COROS / COM GRAPH
 U S E R    D A T A                                        DEVICE: CP526
 ------------------------------------------------------------------------
                                             DRIVE: $   PROGRAM: $$$$$$$$


                     STATIC DISPLAY: ############

                     SYMBOL SET NAME: ######

                     DYNAMIC DISPLAY: ############

                  PIXELGRAPHIC DISPLAY: ############


 The symbol set (SS) need only be specified when first generating static displays
 and segments or when the SS is to be changed. If no SS is specified when
 changing existing displays, then the SS with which the display was generated
 will be entered and used.


    F 1       F 2       F 3      F 4      F 5      F 6      F 7      F 8
             STATIC    DYNAMIC
 SEGMENTS   DISPLAYS  DISPLAYS  SYMBOLS  FUNCTIONS CONTINUE  HELP     EXIT
```

```
    F 1       F 2       F 3      F 4     F 5      F 6      F 7      F 8
             LEXICON   PXLGRA.  ASSIGN
             DATA     DISPLAYS  CP/OP PAR       CONTINUE  HELP     EXIT
```

Before you begin generating segments, decide on a background color for the entire program. Define the color in the CP/OP PARAM. ASSIGNMENT form. You may also determine other central values here e.g., position and color of message and command lines, nationality of ASCII character set on the CP, parameters for connecting the monitor
) (monitor register) etc..

You can determine the exact meanings of the input fields in assigning CP/OP parameters in the Planning Guide COM GRAPH (Chapter 4).

For the example, use the following parameter assignments:

```
-> SELECTION -> USER DATA ->                            COROS / COM GRAPH
C P / O P - P A R A M.  A S S I G N M E N T                 DEVICE: CP526
--------------------------------------------------------------------------
                                             DRIVE: B    PROGRAM: EXAMPLE

Message line:      Backgr: RED       Foregr: BLACK   Flashing: M    Line: 01
Command line:      Backgr: BLUE      Foregr: CYAN    Flashing: M    Line: 24

Delimiter:    Time:        Date:        Binary Represent.:    ZERO:        ONE: 1
Decimal point:        Nationality: INTERNATIONAL

Screen:        Backgr: BLUE    Foregr: WHITE    Flashing: M    Char:
Fct. Keys:     Backgr: WHITE   Foregr: BLACK    Flashing: M    Char:
If '&' is specified as character, a centralized point is used.

Monitor resolution: HIGH                  Signal in case of op. error: 2

Send Direct for messages: 199             Receive All for messages DB: 202




    F 1        F 2        F 3        F 4        F 5        F 6        F 7        F 8
 MONITOR
 REGISTER                                             STORE      HELP       EXIT
```

)

By pressing the <F1 MONITOR REGISTER> function key you reach the ASSIGN PARAMETERS form. Here you can enter the parameters necessary for the connection of your monitor. These parameters depend on the monitor used. A detailed description of the individual registers and how to define them can be found in subsection 3.10.4.

By pressing the <F8 EXIT> function key you return to the previous parameter assignment form. You are now prompted to save the parameters just entered using <F6 STORE>. The CP parameters in the program EXAMPLE are stored on drive B.

If you try to exit the form by pressing the <F8 EXIT> function key before storing the CP parameters, COM responds DATA LOSS...EXIT FORM?. If you press the <F1 YES> function key, you exit the form without saving any of the input values. If you press the <F3 NO> function key, you remain in the form.

5.4   Generating Displays

5.4.1   Symbols

) You have already transferred the standard symbol set (i.e., SS1 or SS2, depending on your
monitor) to the program EXAMPLE. To better understand symbol set generation, you
will, in this subsection, redefine one character of this symbol set.

In the SELECTION form (for information on entering this form, see 'Use of Libraries'), press
the <F1 PROGRAM USER DATA> function key. In the next form, USER DATA, enter the
name "SS1" in the SYMBOL-SET NAME field. Do not fill in any other fields in this form.

The <F4 SYMBOLS> function key in the first part menu of this form brings you to the
SYMBOLS form.

If the CP 526 is on-line while you select static display generation, the CP monitor responds
by switching off the START display. The START display contains, besides product designat-
ion and version number, information on the system (e.g., capacity of the memory
submodule used).

The color of the screen, the color and position of the message and command lines, as well
as the color of the function keys, now correspond to the values preset in the CP 526
PARAM. ASSIGNMENT form; the screen is completely blank, except for the row of
function keys.

Beginning with the SELECTION form, the drive and program name are displayed in the
upper right-hand corner of most forms, under the solid line. This reminds you that all Save
and Delete operations refer to the specified program.

The fields in the SYMBOLS form, which comes up next, show the following default values:

)

```
-> SELECTION -> USER DATA ->                      COROS / COM GRAPH
S Y M B O L S                                         DEVICE: CP526
-------------------------------------------------------------------
                                        DRIVE: B   PROGRAM: EXAMPLE



            SYMBOL SET NAME FOR SYMBOL SELECTION:   SS1

                   SYMBOL SET NAME TO BE STORED:    SS2

                        MONITOR RESOLUTION:   HIGH







   F 1        F 2        F 3      F 4      F 5      F 6      F 7      F 8
                      SYMBOL                      STORE
                      SELECTION                   SYMB. SET  HELP     EXIT
```

The MONITOR RESOLUTION field is preset to either HIGH or LOW, depending on the monitor, or on the entry in the program-ID header. In the SYMBOL GENERATION form, the field for a symbol is either 18 x 8 pixels for HIGH resolution, or 9 x 8 pixels for LOW resolution (a pixel is one image element).

The <F3 SYMBOL SELECTION> function key brings you to the SYMBOL SELECTION form.

To understand the inputs required in the SYMBOL SELECTION form, you must first know how segment generation proceeds with a finished symbol set.

ASCII and symbol set characters are available for segment generating. You can activate the symbol set during segment generation by pressing a function key to access these characters. If you press the <A> key on the PG keyboard while the symbol set is turned on, the symbol assigned to the character "A," not the character "A" itself, appears on the CP monitor.

The assignment of ASCII characters to symbols occurs in the SYMBOL SELECTION form. The following ASCII characters may be assigned two symbols each:

"A" - "Z" / "a" - "z" / "0" - "9" / "!" / " " / ","

For clarity, the symbol set is divided into two halves: upper and lower sets. You can use a function key to switch between the upper and lower sets while generating segments.

```
-> USER DATA -> SYMBOLS ->                          COROS / COM GRAPH
S Y M B O L   S E L E C T I O N                       DEVICE: CP526
------------------------------------------------------------------------
                                    DRIVE: B   PROGRAM: EXAMPLE

       SYMBOL SET NAME:    SS1


       SOURCE SYMBOL:      0
       FROM  upper    SET

       TO BE STORED AS:    P
       IN    lower    SET




    F 1        F 2        F 3        F 4        F 5       F 6       F 7        F 8
  GENERATE                                    CHANGE    STORE
  SYMBOL                                      SET       SYMBOL               EXIT
```

If a symbol set already exists, the symbols are displayed in their authentic sizes in the last line on the CP monitor. To make it easier to differentiate between symbols, the sizes are displayed in alternating red and green colors. The ASCII character assigned to each symbol is located above the symbol. The same representation occurs during segment generation, if you select the symbols.

If, in the SYMBOLS form, you chose a symbol set that does not yet exist, all symbols would be preassigned blank characters.

However, since the standard symbol set already exists, you can choose an available symbol in the SOURCE SYMBOL field, modify it any way you like, and store it under the same or a different ASCII character in the TO BE STORED AS field. You can also select a character from the upper set, modify, and store it in the lower set.

In the SYMBOL SELECTION form, select source symbol "O" in the lower set, and store it as "P" in the upper set. You can change the set half by pressing the < F5 CHANGE SET> function key. The switch occurs independent of the cursor position for either the SOURCE SYMBOL or the TO BE STORED AS fields.

After you press the <F1 GENERATE SYMBOL> function key, the form should appear as follows:

```
-> SYMBOLS -> SYMBOL SELECTION ->
G E N E R A T E    S Y M B O L
                                        * * * * * * * * * * * * * * * * * * * * * * *
                                                        * * *
                                                        * * *
          SYMBOL SET NAME:  SS1          * * *          * * *          * * *
                                         * * *          * * *          * * *
          SOURCE SYMBOL:      O          * * * * * *    * * *          * * *
          FROM  upper    SET             * * * * * *    * * *          * * *
                                         * * * * * * * *  * * *        * * * * * *
          RESOLUTION:      HIGH          * * * * * * * * * * * * * * * * * * * * * * * *
                                         * * * * * * * *  * * *        * * * * * *
          The high resolution monitor    * * * * * *    * * *          * * * * * *
          has 18 pixels and the low      * * * * * *    * * *          * * * * * *
          resolution monitor 9 pixels    * * *          * * *          * * *
          available in a vertical        * * *          * * *          * * *
          direction.
          Delete pixels with the space bar.


     F 1        F 2        F 3        F 4        F 5        F 6        F 7        F 8
   CURSOR     CURSOR     CURSOR     CURSOR     SYMBOL     MIRROR     MIRROR
   LEFT       UP         DOWN       RIGHT      INVERSE    HORIZONT.  VERTICAL    EXIT
```


Edit the symbol until it looks like the one described in the next form. Experiment a little. You cannot destroy any data. The symbol is only saved to the symbol set when you press the <F6 STORE SYMBOL> function key in the SYMBOL SELECTION form. To do this, exit the SYMBOL GENERATION form by pressing the <F8 EXIT> key.

You can draw the symbol with the <F1 CURSOR LEFT>, <F2 CURSOR UP>, <F3 CURSOR DOWN>, and <F4 CURSOR RIGHT> function keys, and delete parts of the symbol with the space bar. INVERT and MIRROR functions are also available for creating symmetrical symbols.

```
-> SYMBOLS -> SYMBOL SELECTION ->
G E N E R A T E   S Y M B O L
                                         *********************
                                               ***
                                               ***
        SYMBOL SET NAME:  SS1          ***     ***        ***
                                       ***     ***        ***
        SOURCE SYMBOL:    0            ******  ***        ***
        FROM  upper   SET              ******  ***        ***
                                       ******* ***     ******
        RESOLUTION:     HIGH           **************************
                                       ******* ***     ******
        The high resolution monitor    ******  ***     ******
        has 18 pixels and the low      ******  ***     ******
        resolution monitor 9 pixels    ***     ***        ***
        available in a vertical        ***     ***        ***
        direction.
        Delete pixels with the space bar.


     F 1      F 2      F 3      F 4      F 5      F 6      F 7      F 8
   CURSOR   CURSOR   CURSOR   CURSOR   SYMBOL   MIRROR   MIRROR
   LEFT     UP       DOWN     RIGHT    INVERSE  HORIZONT. VERTICAL   EXIT
```

Note that in the enlarged symbol representation on the PG screen, three stars placed horizontally, "***", represent one pixel to eliminate distortion.

After you store the symbol, the new symbol appears immediately in the last line on the CP monitor, as long as the correct set half is turned on, and you can see the symbol in its actual size.

Until this point, this modification takes place only in the PG memory. After you generate all symbols, exit the SYMBOL SELECTION form by pressing the <F8 EXIT> function key, and then press the <F6 STORE SYMB. SET> function key in the SYMBOLS form. This saves the symbol set on the data medium (i.e., floppy diskette or hard disk).

Since SS1 (or SS2) is still entered in the SYMBOL SET NAME TO BE STORED field, COM asks whether you want to overwrite the existing symbol set.

The message STORE ALTHOUGH ELEMENT SS1 ALREADY EXISTS? appears. The function key menu in the SYMBOLS form is replaced by the acknowledgment menu. Press the <F1 YES> function key to answer yes, or the <F3 NO> function key to answer no.

If you respond affirmatively with the <F1 YES> function key, the old symbol set is overwritten by the new one on the data medium.

If you respond negatively with the <F3 NO> function key, nothing is stored. You do not exit the SYMBOLS form and you have an opportunity to modify the SYMBOL SET NAME TO BE STORED field (i.e., change the name to "SS3"). Then press the <F6 STORE SYMB. SET> function key again. After a short time, the message ELEMENT STORED appears.

Exit the SYMBOL GENERATION form by pressing the <F8 EXIT> function key. You again enter the PROGRAM USER DATA form.

5.4.2   Segments

To construct segments, designate a symbol set in the PROGRAM USER DATA form. For this example, use symbol set SS3 (i.e., enter "SS3" in the SYMBOL SET NAME field).

Pressing the <F1 SEGMENTS> function key brings you to the SEGMENTS form.

Fill in the SEGMENT field as shown. If you do not enter another name, the name from the SEGMENT field is copied into the NAME TO STORE & ASSIGN field.

Segments that already exist appear on the CP screen. When you create a new display, the screen is cleared.

```
-> SELECTION -> USER DATA ->                          COROS / COM GRAPH
S E G M E N T S                                         DEVICE: CP526
------------------------------------------------------------------------
                                           DRIVE: B   PROGRAM: EXAMPLE



                        STATIC DISPLAY:
                        SYMBOL SET NAME: SS3


                        SEGMENT: PART 1

              Name of SEGMENT to
                STORE & ASSIGN: PART 1


                     CURSOR: LINE 01 COLUMN 01.
          Change between PG and CP cursor by pressing enter key



   F 1       F 2        F 3       F 4      F 5       F 6       F 7       F 8
           GENERATE   REPOS.   STAT.DISP ASSIGN    STORE
           SEGMENT    SEGMENT  OFF       SEGMENT   SEGMENT    HELP      EXIT
```

The < F2 GENERATE SEGMENT > function key brings you to the segment editor (i.e., the SEGMENT GENERATION form).

Press the < F4 STAT.DISP ON-OFF > and < F5 ASSIGN SEGMENT > function keys to fit a segment into an existing static display. Designate the static display in the PROGRAM USER DATA form. You can switch it on and off with the < F4 STAT.DISP OFF > function key.

The < F5 ASSIGN SEGMENT > function key is intended to supplement static display generation. First, the segment is stored, just as when you press the < F6 STORE SEGMENT > function key; however, when you exit the form, the static display, which changed through the segment assignment, is also stored. The message STORE, ALTHOUGH ELEMENT (ELEMENT NAME) ALREADY EXISTS? appears if the static display already exists (i.e., an acknowledgment is required). The current cursor position in the LINE and COLUMN fields determines the segment position in the static display (see also static displays - segment beginning).

The < F3 REPOS. SEGMENT > function key allows you to reposition a preexisting, modified segment within a static display, without having to branch to the segment editor.

```
-> USER DATA -> SEGMENT  ->                        COROS / COM GRAPH
G E N E R A T E   S E G M E N T                         DEVICE: CP526
-------------------------------------------------------------------------
                                           DRIVE: B   PROGRAM: EXAMPLE

  STAT. DISP.:              SYMBOL SET: ss3     SEGMENT: PART 1


  CURSOR:  LINE 01 COLUMN 01. Change between PG operation and segment
           generation at CP/OP with ENTER key (only in main menu).

  Char. attributes:  BACKGROUND: BLACK    FOREGROUND: WHITE    FLASHING: H
  Attr. for colors:  BACKGROUND: BLACK    FOREGROUND: WHITE    FLASHING: H

  BLOCK OPERATIONS:  Block start:        LINE: 01    COLUMN: 01
                     Block end:          LINE: 01    COLUMN: 01
                     Start of new block: LINE: 01    COLUMN: 01
  STATUS:
  SYMBOLS OFF  -  ASCII   SET,   WRITE VERTICALLY OFF,   BACKGROUND ON



   F 1       F 2       F 3       F 4      F 5      F 6      F 7      F 8
 CHANGE
 STATUS     CHAR.     LINE     COLUMN    BLOCK    COLOR    HELP     EXIT
```

The segment editor is a tool that is easy to use for constructing segment. Try out the various functions. You cannot destroy any data. The segment is only saved to the data medium when you exit the form by pressing the < F8 EXIT > and < F6 STORE SEGMENT > function keys, or when you press the < F5 ASSIGN SEGMENT > function key in the SEGMENT form.

When you either overwrite or exit the SEGMENTS form, the system prompts for verification.

NOTE:
When you press the < ENTER > key, you can move the cursor on the CP screen with the cursor control keys (for further information on the "CP- CURSOR" mode, see Planning Guide COM GRAPH, subsection 3.6).

Try to edit the segment in the example as described. The individual editor functions are explained later in this chapter.

You do not need to think about the position of the segment on the screen. Segments are variable in position. Their final placement on the screen is determined in the STATIC DISPLAY form.

The segment, PART1, should resemble the following figure:

```
┌──────────┐
│          │
│          │
│          │
│          │
│          │
│          │
│          │
│          │
│          │
└──── T ───┘
```
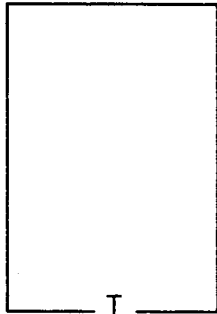
First select the character attributes according to the following steps:

In the GENERATE SEGMENT form, in the "Character attributes" line, place the cursor in the BACKGROUND field. Choose "BLUE" as the background color to conform to the CP parameters already set. You can either type in the word "BLUE", or enter it by pressing the <F7 HELP> function key. Enter "CYAN" in the FOREGROUND COLOR field, and "N" in the FLASHING field (i.e., not flashing).

Switch to the "CP-CURSOR" mode by pressing the <ENTER> key. You can use the cursor control keys, which usually move the cursor in the PG forms, to move the cursor on the CP screen. Note the tab function (for further information, see Planning Guide COM GRAPH, subsection 3.6). You can use the alphanumeric keyboard of your programmer for the CP monitor, but no longer for the programmer.

Write a few words. You can easily erase any characters with the REMOVE LINE function. The **ACTIVE** message flashes with each keystroke in the bottom right-hand corner of the programmer screen, as the yes signal is transferred to the CP.

) To access symbols in symbol set SS3, press the <F1 CHANGE STATUS> function key. The following submenu appears:

F 1	F 2	F 3	F 4	F 5	F 6	F 7	F 8
SYMBOLS	CHANGE	VERTICAL	BACKG.				MAIN
ON-OFF	SET	ON-OFF	ON-OFF				MENU

Among other functions, this submenu contains the SYMBOLS ON-OFF function. Pressing the <F1 SYMBOLS ON-OFF> function key brings you back to the main menu. In the form, the "Status" line changes from **SYMBOLS OFF - ASCII SET** to **SYMBOLS ON - LOWER SET**. In the last two lines on the CP monitor, the symbols of the lower set appear with their assigned ASCII characters.

When you type the same words as before, it is no longer the ASCII characters, but their corresponding symbols, that appear in the assigned color on the CP screen. Pressing the <F2 CHANGE SET> function key in the CHANGE STATUS submenu displays the upper symbol set on the screen.

Turn on the lower symbol set and press the <CAPS LOCK> function key. Use the following keys to construct the tank in this example:

<Q>	Upper left corner	(tank size is 15 characters x 12 characters)
<E>	Upper right corner	
<Z>	Lower left corner	
<C>	Lower right corner	
<F>	Horizontal line	(13)
<R>	Vertical line	(10)
<W>	T-part	

)

The VERTICAL ON-OFF function in the CHANGE STATUS submenu allows you to draw vertical lines easily. The cursor is not placed, as usual, in front and to the right of the character last entered, but below it. By pressing the <F3 VERTICAL ON-OFF> function key once more, you can return to the normal "WRITING" mode.

You will not need the <F4 STAT.DISP ON-OFF> function key until you generate a static display and want to fit a segment into it. Designate the static display in the PROGRAM USER DATA form. Use the <F4 STAT.DISP ON-OFF> function key in the SEGMENTS and the GENERATE SEGMENT forms to turn the static display on or off.

Use the following function keys to edit your segments:

● <F2 CHAR.> - brings you to the following submenu:

```
   F 1       F 2      F 3       F 4       F 5       F 6       F 7       F 8
 INSERT    REMOVE            DELETE                                    MAIN
 CHAR.     CHAR.             CHAR.                                     MENU
```

● <F3 LINE> - brings you to the following submenu:

```
   F 1       F 2      F 3       F 4       F 5       F 6       F 7       F 8
 INSERT    REMOVE                                                      MAIN
 LINE      LINE                                                        MENU
```

●
<F4 COLUMN> - brings you to the following submenu:

```
   F 1       F 2      F 3       F 4       F 5       F 6       F 7       F 8
 INSERT    REMOVE                                                      MAIN
 COLUMN    COLUMN                                                      MENU
```

© *Siemens AG C79000-B8576-C217-01*

If you designate a static display, it is not affected by any Delete or Insert operations.

You do not automatically return to the main menu from the CHARACTER, LINE, and COLUMN submenus when carrying out a function, as is the case when you press the <F1 CHANGE STATUS> function key. This is because you usually want to delete more than one character, line, or column.

NOTE:
You cannot do any editing while you are in a submenu. For example, if you press the <A> key while in a submenu, COM responds by displaying a CP message. Return to the main menu by pressing the <F8 MAIN MENU> function key to continue generating a segment.

Experiment with the functions; when the segment looks like the one in the figure, exit the form by pressing the <F8 EXIT> function key and save it by pressing the <F6 STORE SEGMENT> function key in the SEGMENT form.

The Block and Coloring operations are especially helpful when generating several similar segments.

In the SEGMENT form, the name of a segment to be edited is specified in the SEGMENT field. The NAME OF SEGMENT TO STORE & ASSIGN field does not need the same name, so you can modify an existing display and save it under a different name, the same way you do with symbol selection.

Coloring Operation

For this example, you need a second tank exactly the same as the first one except for the color. In the NAME TO STORE field, enter "PART2", and in the SEGMENT field, enter "PART1". Then, call up the segment editor again by pressing the <F2 GENERATE SEGMENT> function key. Segment PART1 reappears. Change the color of the display from "BLUE/CYAN" to "BLUE/RED" and store it.

The COLOR submenu offers the following functions:

```
   F 1       F 2       F 3       F 4       F 5       F 6       F 7      F 8
  COLOR     COLOR     COLOR     COLOR     COLOR     COLOR     VISU.    MAIN
  LEFT      UP        DOWN      RIGHT     BLOCK     TOT.DISP. ATTRIB.  MENU
```

When coloring, you must differentiate between two types of operation:

● conditional coloring
● unconditional coloring

The COLOR BLOCK and COLOR TOTAL DISPLAY functions color the display conditionally; the COLOR LEFT, COLOR UP, COLOR DOWN, and COLOR RIGHT functions color unconditionally.

Unconditional Coloring

The segment takes on the attributes specified in the "Attr. for colors" line, regardless of the attributes that you previously assigned.

Conditional Coloring

Only the characters that have the same attributes as in the "Character attributes" line are changed. They take on the new attribute that you assign in the "Attr. for colors" line. For example:

Characters Attributes for All Characters in a Line

	BACKGROUND	FOREGROUND	FLASHING
123456789	BLUE	CYAN	N
ABCDEFGHI	BLUE	YELLOW	Y
JKLMNOPQR	YELLOW	RED	N

In the "Attr. for colors" line, enter "YELLOW" in the BACKGROUND field, "GREEN" in the FOREGROUND field, and "N" in the FLASHING field.

Place the CP cursor on 3 in the first line.

© *Siemens AG C79000-B8576-C217-01*

Press the <F6 COLOR> function key, and then the following function keys in the order specified:

<F4>, <F4>, <F4>, <F3>, <F3>, <F4>, <F4>, <F4>. Then press the <F8 MAIN MENU> function key to return to the main menu.

The screen should have the following appearance:

```
123456789
ABCDEFGHI
JKLMNORQR
```

The characters 3456, F, and OPQ are green on a yellow background.

The COLOR TOTAL DISPLAY function works conditionally.

If in the "Char. attributes" line, you enter "YELLOW" in the BACKGROUND field, "GREEN" in the FOREGROUND field, and "N" in the FLASHING field, and in the "Attr. for colors" line, you enter "WHITE" in the BACKGROUND field, "RED" in the FOREGROUND field, and "N" in the FLASHING field, then when you execute the COLOR TOTAL DISPLAY function by pressing the <F6 COLOR TOT.DISP> function key, all parts formerly green on a yellow background are displayed in red on a white background.

The <F5 COLOR BLOCK> function key has the same effect, except that you can limit the color change, independent of the starting color condition, to a rectangular display segment.

Block Operation

The Block submenu offers the following functions:

```
  F 1        F 2        F 3        F 4        F 5      F 6      F 7      F 8
  BLOCK      BLOCK      REPOS.     DELETE     COPY                       MAIN
  START      END        BLOCK      BLOCK      BLOCK                      MENU
```

With the Reposition Block, Delete Block, and Copy Block operations, you can define the block in one of two ways:

Either by directly entering the line and column numbers in the appropriate fields in the "Block start" and "Block end" lines, or by using the <F1 BLOCK START> and <F2 BLOCK END> function keys.
Pressing the <F1 BLOCK START> function key causes the CP cursor position coordinates to appear in the LINE and COLUMN fields in the "Block start" line. Pressing the <F2 BLOCK END> function key causes the CP cursor position parameters to appear in the LINE and COLUMN fields in the "Block end" line. For example:

```
123456789
ABCDEFGHI
JKLMNOPQR
```

Place the cursor on the C and press the <F5 COPY BLOCK> function key and then the <F1 BLOCK START> function key. Then place the cursor on the P and press the <F2 BLOCK END> function key.

The area **CDEFG** and **LMNOP** is now defined as a block.

To perform the Reposition and Copy operations, you must specify the position to which you want to copy or move the block. Do this either by entering the line and column numbers in the "Block start" line, or by moving the CP cursor to the desired location. If you use the CP cursor for positioning, the numbers in the LINE and COLUMN fields in the new "Block start" line change as the cursor moves.

The Reposition Block operation causes the block to be deleted at the old location, and appear at the new one. The Copy Block operation leaves the block intact.

In this example, you can create the second tank by using the Copy Block operation.

However, you should create a second segment since you can use segments in many static displays. The space is only occupied once in CP user memory. Also, a single segment is easier to reposition within a static display.

Remember that you can also store often used segments in a library.

Create the remaining segments so that they resemble the following. The figures on the left show how the segments will look; the figures on the right show the keys you should press to create these symbols:

PART2: Use PART1 and change the color of the entire display from a "BLUE" background, "CYAN" foreground, and "N" for flashing, to a "BLUE" background, "RED" foreground, and "N" for flashing. Store the recolored display under the name PART2.

Segment name: PART 3 Symbols used: uppercase letters from the "lower" set	
Symbols	Keys to be used

T
 	
↓
	R R
	Z FFFF FFFFF W FFFFF FFFF C
	                                   R
	                                   M

Note: the segment consists of 3 areas 1 ◄---► 2 ◄---► 3	

Area:	1	2	3
Background color:	BLUE	BLUE	BLUE
Foreground color:	CYAN	MAGENTA	RED
Flashing:	N	N	N

Segment name:	PART 4	
Symbols used:	ASCII set	
Symbols		**Keys to be used**
LITERS		**LITERS**
Background color:	BLUE	
Foreground color:	WHITE	
Flashing:	**N**	

Segment name:	PART 5	
Symbols used:	"upper" set	
Symbols		**Keys to be used**
		O
Background color:	BLUE	
Foreground color:	CYAN	
Flashing:	**N**	

| Segment name: PART 6 | |
Symbols used: "lower" set	
Symbols	Keys to be used
	P
Background color: BLUE foreground color: YELLOW Flashing: **Y**	

| Segment name: PART 7 | |
Symbols used: "upper" set	
Symbols	Keys to be used
	O
Background color: BLUE Foreground color: RED Flashing: **N**	

Segment name:	PART 8	
Symbols used:	ASCII set	
Symbols		**Keys to be used**
TANK 1		**TANK 1**
Background color:	BLUE	
Foreground color:	CYAN	
Flashing:	**N**	

Segment name:	PART 9	
Symbols used:	ASCII set	
Symbols		**Keys to be used**
TANK 2		**TANK 2**
Background color:	BLUE	
Foreground color:	RED	
Flashing:	**N**	

Use the same procedure as described in the tank example to generate the remaining displays.

5.4.3   Static Displays

In the static display, you define segments, their positions, and the symbol set with which they are displayed.

You must declare the name of static display BACK1 and symbol set SS3 in the PROGRAM USER DATA form. Pressing the < F2 STATIC DISPLAY > function key brings you to the STATIC DISPLAY form.

The symbol sets that you used to generate the segments must correspond to the one specified. However, once you generate a segment, you can later change the symbol set you used to generate the display (i.e., with an acknowledgment). You may want to do this when the symbol sets differ only in details (e.g., representing switches in different norms).

To determine the final position of the segment, enter the coordinates in the LINE and COLUMN fields manually or by using the CP cursor. If you use the CP cursor, line and column coordinates are input automatically. Pressing the < F1 INSERT SEGMENT > function key enters the next consecutive number available (i.e., beginning with 1) in the DISP-NO. field. To insert a segment at a particular location, simply enter the number.

The spot where you insert the segment is important if there are gaps in it. Gaps are places in the segment that have no characters. If displays are layered, then the order of specification in the static display determines which segment is in front and which is behind on the CP screen. A segment neither needs any particular form, nor must it cover a continuous area.

The character at the farthest upper left location represents the beginning of the segment.

For example:

```
    segment beginning
      |
      |
      v
    + *********        *********
    *********            *********
        *********  *********
```

If during segment generating, you use the space bar to enter blanks between the asterisks, and you assign this segment last in order, blank spaces cover the displays behind the segment. However, if you jump over the blank space using the CP cursor keys, you can see the displays behind it.

The name of the segment is only important with the <F1 INSERT SEGMENT> function key. You can enter the name either in the P-D NAME field manually, or by pressing the <F7 HELP> function key.

The DISP-NO field identifies the number of the segment that the <F2 REPOS. SEGMENT>, <F3 SEGMENT> and <F4 DELETE SEGMENT> function keys use to perform their functions.

Use the <F3 SEGMENT> function key to set the CP cursor to the starting point of the segment.

NOTE:
Please note that you can also mark the starting point with a blank space. In this case, you could mistake this for a segment beginning. If the segment that you insert or reposition does not seem to appear at the right spot, try to determine whether the display really starts where you think it does.

Define the position for segment PART1 as shown on the next page and insert it in display number 1 by pressing the <F1 INSERT SEGMENT> function key.

```
-> SELECTION -> USER DATA ->                         COROS / COM GRAPH
S T A T I C   D I S P L A Y                             DEVICE: CP526
---------------------------------------------------------------------
                                          DRIVE: B   PROGRAM: EXAMPLE

STAT.DISP: BACK1        SYMBOL SET: SS3     DISP-NO: 01 SEGMENT: PART 1
LINE 05  COL.04  (ENTER) Store: STAT.DISP: BACK1      SEGMENT:

           SEGMENT POS.    |    SEGMENT  POS.    |     SEGMENT POS.
   _____|_____|_____
        01 PART 1   05, 04 |02              ,    |03                ,
        04              ,   |05              ,    |06                ,
        07              ,   |08              ,    |09                ,
        10              ,   |11              ,    |12                ,
        13              ,   |14              ,    |15                ,
        16              ,   |17              ,    |18                ,
        19              ,   |20              ,    |21                ,
        22              ,   |23              ,    |24                ,
        25              ,   |26              ,    |27                ,
        28              ,   |29              ,    |30                ,
        31              ,   |32              ,    |

   F 1        F 2        F 3        F 4       F 5       F 6     F 7      F 8
 INSERT     REPOS.    SEGMENT    REMOVE   STORE AS   STORE
 SEGMENT    SEGMENT    START     SEGMENT   SEGMENT   STAT.DISP  HELP     EXIT
```

Insert one segment after another into the static display. Make deliberate mistakes when entering the line and column numbers, so you can practice with the < F2 REPOS. SEGMENT > and < F4 REMOVE SEGMENT > function keys.

When the static display is finished, the screen should have the following appearance:

```
-> SELECTION -> USER DATA ->                        COROS / COM GRAPH
S T A T I C   D I S P L A Y                            DEVICE: CP526
-----------------------------------------------------------------------
                                           DRIVE: B    PROGRAM: EXAMPLE

STAT.DISP: BACK1        SYMBOL SET: SS3    DISP-NO: 01 SEGMENT: PART 9
LINE 17  COL.11  (ENTER) Store: STAT.DISP: BACK1       SEGMENT:

       SEGMENT POS.    |    SEGMENT  POS.    |    SEGMENT POS.
    ------------------- | ------------------- | -------------------
       01 PART 1  05, 04 | 02 PART 2  05, 26 | 03 PART 4    03, 12
       04 PART 4  03, 34 | 05 PART 3  17, 11 | 06 PART 8    02, 08
       07 PART 9  02, 30 | 08           ,    | 09             ,
       10          ,    | 11           ,    | 12             ,
       13          ,    | 14           ,    | 15             ,
       16          ,    | 17           ,    | 18             ,
       19          ,    | 20           ,    | 21             ,
       22          ,    | 23           ,    | 24             ,
       25          ,    | 26           ,    | 27             ,
       28          ,    | 29           ,    | 30             ,
       31          ,    | 32           ,    |
    ----------------------------------------------------------------
     F 1       F 2       F 3       F 4       F 5       F 6      F 7       F 8
   INSERT    REPOS.    SEGMENT   REMOVE   STORE AS   STORE
   SEGMENT   SEGMENT    START    SEGMENT  SEGMENT   STAT.DISP  HELP      EXIT
```

Store the display in the program EXAMPLE on drive B by pressing the <F6 STORE
STAT.DISP> function key. The name of the display to store is taken from the STORE:
STATIC DISP. field. Therefore, you can modify, just as with segments, an existing static
display, and store it under a different name. If you try to overwrite an existing display,
COM requests an acknowledgment to the prompt STORE, ALTHOUGH ELEMENT BACK1
ALREADY EXISTS?.

Now generate static display BACK0 using symbol set SS3.

To do this, change the display name to BACK0 in the PROGRAM USER DATA form, call up
the STATIC DISPLAYS form by pressing the <F2 STATIC DISPLAYS> function key, and
store static display BACK0 by pressing the <F6 STORE STAT.DISP> function key, without
inserting any segment.

Static display BACK0 contains no segments and is therefore empty.

It may seem pointless to have an empty static display, but you need it for the display output command START that is described in section 2.4.

You can store up to 32 segments in a static display. If you use extremely small segments, so that 32 segments are not enough, or if you want to join three segments together to make a new segment, use the <F5 STORE AS SEGMENT> function key.

The static display that you generate this way is stored in the form of a segment. You can use the segment editor to further edit this segment.

COM takes the name of the segment to store from the SEGMENT field.

5.4.4   Dynamic Displays

5.4.4.1   General

With dynamic displays, you can represent values from data blocks in the programmable controller in either digital or analog form on the CP monitor. You can make the display attributes (e.g., background color, foreground color) dependent upon the status of a flag bit, or upon a data block value. You can therefore represent measured value overranges or underranges in an easily recognizable form.

You can also output segments, depending on flag conditions or data block values. This is how the program EXAMPLE represents a valve in the two states, open and closed.

Since the flag status and the values in the controller are constantly changing (i.e., the control program runs cyclically), the values needed to represent the display must be repeatedly fetched from the controller, at predetermined time intervals. These time intervals are called the "update rate" in COM.

The dynamic display consists of several fields. The exact number of maximum permissible fields varies greatly, and depends on the type of fields programmed.

You can represent each of these fields by a value from another data block. To do this, a part of the block must be fetched, which takes a relatively high amount of time, regardless of the number of words to be transferred. Therefore, you should enter as many values as needed to represent a dynamic display in **one** data block. COM organizes the dynamic display so that all values needed in a display that are located in one data block are loaded simultaneously.

NOTE:
You can only load a block of words.

For example, in a dynamic display there are two value output fields. Field 1 represents the value from CPU 1, data block DB2, and data word DW20; field 2 represents the value from CPU 1, DB2, and DW150.

While the application program processes this display, DB2 is fetched from CPU 1. In multiprocessor operation, you can specify which CPU the DB belongs to in the CPU NO. field.

Since DW20 and DW150 must be represented, the entire data block range from DW20 to DW150 is fetched. This takes less time than retrieving one word twice. Therefore, organize your data blocks as compactly as possible, to avoid transferring unnecessary data.

Transferring in blocks is especially important with input fields.

For an output field, you must define from which data block and at which starting address the display value should be fetched.

For an input field, you must define in which data block and at which starting address the input value, entered via the process keyboard, should be stored.

The input fields are supplied with the current values in the blocks when the display is called (e.g., by a display output command or PC job order). All input fields that relate to a common block are again grouped together, as shown in the following example:

    input field 1: CPU 2, DB3, DW45
    input field 2: CPU 2, DB3, DW110

In this case, the block from DW 45 to DW110 is fetched.

When you make inputs in both fields via the process keyboard, these inputs have as yet, in the data block itself, no effect and can therefore still be corrected as desired.

The input values are not transferred to the controller until you press the <ENTER> key.

An effective way to optimize the speed of your dynamic displays is to adjust the update rate, which can be determined separately for each output field. The grading can range from 0.1 second to 5 minutes (refer to Planning Guide COM GRAPH, Chapter 4, Appendix 1).

In programming, try to keep the image on the screen as similar as possible to the actual process. With quickly changing values, you must therefore reach an update rate of 0.1 seconds.

The data blocks must be fetched from the controller within this amount of time (depending on the organization), and the values transformed from the format stored in the controller into recognizable characters. Before transformation, the values can be converted using gain and offset (see the description of value input and value output fields later in this section).

If you are displaying many values (e.g., with tables), the updating time cannot be maintained.

There are three solutions to this problem:

- Represent few fields in a display, do without gain and offset conversion, and organize data in the controller so that all necessary values are located in one data block without gaps.

- Assign reasonable update rates.

- Occasionally switch off updating of certain fields by using flags.

For example, in a display representing a motor test stand, you have a bar that symbolizes the motor speed and ten temperatures from different sections of the motor.

The speed of the motor can change relatively quickly, and should therefore be updated at a rate of ten times per second. For the temperatures, depending on the application, updating every ten seconds would probably be enough.

The ten temperature fields are therefore updated after every hundredth update of a speed bar, which makes it possible to maintain the updating time of the bar. Try to assign a slow update rate wherever possible, to enable fast update rates to be maintained where they are necessary.

Another way to increase the speed is to make the updating of slow fields (e.g., temperatures) dependent upon an interprocessor communication flag bit by designating a flag in the WHEN UPDATING DEPENDENT ON FLAG BIT: ###.# field. Then the field is only updated if the flag is set. The flag can be set and reset either by the control program, or you can set and reset it by entering a set bit command with the process keyboard.

5.4.4.2   How to Generate DYN1

The dynamic display DYN1 is used in the program EXAMPLE. It consists of two value input fields, two value output fields, two bar fields, two figure fields, and one time output field.

The SWITCHOVER field consists of two value output fields; one of the fields is output, depending on the status of a flag. A possible application might be to convert data block values one way when the flag is set, and another way when it is not set, such as displaying a temperature either in Celsius or in Fahrenheit.

In the PROGRAM USER DATA form, specify static display BACK1 and dynamic display DYN1. Branch to the PROGRAM DYNAMIC DISPLAY form by pressing the <F3 DYNAM. DISPLAYS> function key.

The static display is now on the CP monitor. With a preexisting dynamic display, the dynamic fields would be marked by the symbols "?", " + ", "&", and " >" (see form). You determine the position of a field either with the CP cursor by pressing the <ENTER> key or manually, in the LINE and COLUMN fields. You determine the type of a field (e.g., input, output, etc.) by pressing one of the function keys in any of the three part menus, that may be reached by pressing the <F6 CONTINUE> function key.

```
-> SELECTION -> USER DATA ->                        COROS / COM GRAPH
P R O G R .   D Y N A M I C   D I S P L A Y            DEVICE: CP526
--------------------------------------------------------------------
                                            DRIVE: B   PROGRAM: EXAMPLE

DYN. DISP: DYN1          STAT. DISP: BACK1          SYMBOL SET: ss3


CURSOR:               LINE: 03 COLUMN: 06
                      Change between PG and CP/OP cursor by pressing ENTER key

FIELD NO. 001         For CHANGE FIELD,  DELETE FIELD,  and MOVE FIELD the
                      field number is valid. The line and column only apply
                      if no entry has been made for the field number.

NAME to be stored:    DYN1


Field representation: 7 Input    + Output    & Bar    > Shape


    F 1        F 2        F 3      F 4       F 5      F 6      F 7      F 8
  VALUE      VALUE                         STORE
  INPUT      OUTPUT       BAR     SHAPE   DYN. DISP CONTINUE   HELP    EXIT
```

```
    F 1        F 2        F 3      F 4       F 5      F 6      F 7      F 8
  PREVIOUS   NEXT       CHANGE   DELETE    MOVE
  FIELD      FIELD      FIELD    FIELD     FIELD    CONTINUE   HELP    EXIT
```

```
    F 1        F 2        F 3      F 4       F 5      F 6      F 7      F 8
  ENTER      ENTER      OUTPUT   OUTPUT   SW.OVER
  DATE       TIME       DATE     TIME     FIELD    CONTINUE   HELP    EXIT
```

After the display is completed, you can store it on the data medium by pressing the <F5 STORE DYN. DISP> function key. The display name to be stored is taken from the PROGRAM DYNAMIC DISPLAY form. The field is preassigned with the input from the DYNAMIC DISPLAY form. You can decompile and edit an existing display, and store it under a different name, just as you can with segment and static displays.
If the display already exists, an acknowledgment is requested before overwriting.

The second part menu contains the functions necessary to edit existing displays.

The <F3 CHANGE FIELD> function key redisplays a selected, already existing field that you can edit and store again. During editing, the field is optically removed from the CP monitor, and the static display reappears at this location, if it was specified.

Fields can be selected in several ways:

-   By pressing the <F1 PREVIOUS FIELD> or <F2 NEXT FIELD> function keys - The line, column, and field numbers are input automatically.
    If no fields exist, an error message results.

-   By specifying the line and column numbers: The field number is input automatically.
    If no field exists at this position, an error message results.

-   By specifying the field number - You may not make an entry in the LINE or COLUMN field. The line and column numbers are entered automatically. If no field with this number exists, an error message results.

The <F4 DELETE FIELD> function key causes the selected field to be deleted after a positive acknowledgment.

With the <F5 MOVE FIELD> function key, you can change the position of a field. Select the field, and then press the <F5 MOVE FIELD> function key. The message **FIELD/WINDOW COUPLED - USE F5 TO STORE!** indicates the dual function of this function key. Place the cursor at the desired new location, and press the <F5 MOVE FIELD> function key once more. The field is deleted at the old position, the background reappears, and the field is output at the new position.

The <F1 VALUE INPUT>, <F2 VALUE OUTPUT>, <F3 BAR>, and <F4 FIGURE> function keys in the first part menu and the <F1 DATE INPUT>, <F2 TIME INPUT>, <F3 DATE OUTPUT>, <F4 TIME OUTPUT>, and <F5 SWITCHOV. FIELD> function keys in the third part menu determine the field type.

You must place the cursor on a "free" position. Only one field can be represented per location. The error message **Generation not possible; position reserved!** tells you when a field already exists at a given location.

The error message **Field no longer fits in dynamic display!** means that the field exceeds the maximum length of a dynamic display. The number of possible fields depends to a great extent upon the type of fields programmed. In general, approximately 100 fields are possible.

5.4.4.3   Value Input Field

Position the cursor for the program EXAMPLE on line 3, column 6, and press the
< F1 VALUE INPUT > function key.

```
-> USER DATA -> PROGR. DYNAMIC DISPLAY ->                  COROS / COM GRAPH
V A L U E   I N P U T   F I E L D                              DEVICE: CP526
------------------------------------------------------------------------------
                                              DRIVE: B   PROGRAM: EXAMPLE

DYN. DISP: DYN1           STAT. DISP: BACK1           SYMBOL SET: ss3
generate  FIELD: 001                 LINE: 03   COLUMN: 06

Input is stored in CPU-No.1, DB 020 from ADDRESS DW 000.  , FORMAT 16BIT FXP
CHAR. POSITIONS 05.0. The actual value stored is calculated from
VALUE=(INPUT - (0                          v   )) / (1                  v   )

Limit values: UPPER LIMIT:  direct  10000                or indirect
              LOWER LIMIT:  direct  0                    or indirect
Indirectly specified limit values and conversion factors reside in the input DB,
indirect limit values use the same type of memory access as the input value.

COLORS & ATTRIBUTES: STATIC                  Only fill in for static C&A:
BACKGROUND COLOR:  WHITE       FOREGROUND COLOR:  BLACK       FLASHING:  N


   F 1       F 2       F 3       F 4       F 5       F 6       F 7       F 8
                                                     STORE
                                                     FIELD     HELP      EXIT
```

Besides the drive and program name, the form contains the following additional
information on the programming environment: the name of the dynamic display, static
display, and symbol set; the code **GENERATE** or **EDIT** before the field number; the field
position (i.e., line and column numbers). This information is contained in all forms that
you can reach by pressing any of the above mentioned keys.

If you experience any difficulty in operating the forms, or if you get error messages that
give no direct clue as to the cause of the error, look in the Planning Guide (Chapter 4).
There you can find complete information on permissible inputs.

In the VALUE INPUT FIELD form, you must input the CPU number, DB number, and starting address of the field.

The CPU-NO. field defines, in multiprocessor operation, which CPU the data block belongs to. You can find further information on multiprocessor operation in the CP description (Chapter 3).

The DB field designates the number of the data block from which data is fetched when the display is called or to which input data is written after you press the <ENTER> key on the process keyboard, thus activating the inputs in the process. The DB field is preassigned with current values.

The ADDRESS field has three parts: ## ###. ##

The first part (##) designates the memory access type, and the second part (###) designates the actual address in the data block. You only need to fill in the third part (.##) if bit (BI) was assigned in the memory access part, in which case it designates the bit within the word. The Planning Guide (Chapter 4, Appendix) contains a table with the allowed combinations of memory access, format, and character locations.

The FORMAT field designates the STEP 5 format in which the corresponding address in the data block is specified. All STEP 5 formats are acceptable (i.e., byte, character, binary-coded (BCD), fixed-point (FXP), floating-point (FLP), timer, and counter).

The CHAR. POSITIONS field determines the length of the field displayed on the CP monitor. There are restrictions in the combination of format and memory access type (see also Planning Guide, Chapter 4, Appendix).

The line `VALUE = (INPUT - (0 v ###)) / (1 v ###)` in the CONVERSION field means that the value you enter on the process keyboard is reduced by an amount you can specify (e.g., offset, presetting 0), and then divided by another amount (e.g., gain, presetting 1), before being entered in the data block. Detailed information on converting with gain and offset is given in subsection 3.11.3, 'Display of Process Data'.

You can specify the value either directly in the form, by overwriting the presetting, or indirectly, by entering an address. The advantage of indirect specification is that the controller can influence the conversion during run time. If you specify an address for indirect conversion, the address refers to the CPU number and the data block specified in the "Input is stored in" line. The memory access type for indirect conversion factors is always data double word (DD).

Direct and indirect values for gain or offset cannot be specified simultaneously. However, gain can be assigned a direct value and offset an indirect value, and vice versa.

The following formula appears in the VALUE INPUT FIELD form:

    `VALUE = (INPUT - (0 v ###)) / (1 v ###)`

It is the supplement to the following formula, which appears in the VALUE OUTPUT FIELD form:

    `OUTPUT = VALUE x (1 v ###) + (0 v ###)`

Example:
In DB4, starting at DW10, there is a measured value available in BCD format. The value corresponds to the voltage of a digital measuring device, which is connected to the controller via AD transformers. Since the voltage of a tachometer is being measured, the value is directly proportional to motor speed. If 500 rpm corresponds to the value 100 in the DB, then 5 must be specified as the gain, and 0 as the offset.

When the value is output, it is first multiplied by 5, and added to 0 before the result is output on the screen.

For inputs (e.g., to preset the speed), the input value is reduced by 0 and divided by 5, so that the controller receives the value the same as it sends it.

Inputs on the process keyboard can be checked for validity, similar to the way incorrect inputs are rejected by COM. You can specify limits that are fetched either directly or indirectly from the controller for this purpose (see also subsection 3.11.3, 'Display of Process Data').

Indirect limits, like conversion factors, are located in the same data block as specified in the "Input is stored in" line. The type of memory access that applies to the input also applies to the limit value (conversion factors are always DD).

The COLORS & ATTRIBUTES field is where the parameters for background color, foreground color, and flashing are set. These should be entered the way the field is later to appear on the screen. These entries have no influence on a dynamic display while it is being generated. A dynamic display is always generated with the shield attributes that were assigned during CP parameter setting.

Store the field in the form by pressing the <F6 STORE FIELD> function key. The field then appears on the CP monitor in its actual length.

To generate the second value input field, position the cursor on line 3, column 28, and press the <F1 VALUE INPUT> function key.

The second value input field should look just like the first one, but the input values should be stored in data block 20, data word 1. Since the VALUE INPUT FIELD form is still filled in with the values that you entered for the first value input field, change the entry in the ADDRESS field, and store the value input field by pressing the <F6 STORE FIELD> function key.

5.4.4.4   Value Output Field

The first value output field for the example is positioned in line 4, column 6.

```
-> USER DATA -> PROGR. DYNAMIC DISPLAY ->                    COROS / COM GRAPH
V A L U E   O U T P U T   F I E L D                              DEVICE: CP526
-------------------------------------------------------------------------------
                                                   DRIVE: B   PROGRAM: EXAMPLE

DYN. DISP: DYN1           STAT. DISP: BACK1             SYMBOL SET: ss3
generate    FIELD: 003            LINE: 04    COLUMN: 06

Output is fetched from CPU-No.1, DB 020 from ADDRESS DW 000.  , FORMAT 16BIT FXP
CHAR. POSITIONS  05.0.   The actual value output is calculated from
OUTPUT = VALUE * (1                        v    ) + (0                    v    )

Indirectly specified conversion factors reside as DD in the output DB.        .

UPDATE RATE: 1    s  If updating dependent on FLAG BIT: F     .


COLORS & ATTRIBUTES: FLAG-CONTROLLED        Only fill in for static C&A:
BACKGROUND COLOR:             FOREGROUND COLOR:             FLASHING:



   F 1        F 2        F 3        F 4        F 5        F 6      F 7      F 8
 DYNAM.                                                  STORE
 C&A                                                     FIELD    HELP     EXIT
```

For information on the CPU-NO., DB, ADDRESS, FORMAT, and CHAR. POSITIONS fields, and conversion factors, see p. 4-43, which specifies the same attributes for value output fields.

The value specified in the UPDATE RATE field, and the flag (which is optional) specified in the IF UPDATING DEPENDENT ON FLAG BIT field, control cyclic rejuvenation of the values output on the screen. You should choose updating times that are possible to maintain. Especially avoid specifying small time intervals where unnecessary (see also 5.4.4.1).

The COLORS & ATTRIBUTES field works somewhat differently with output fields than with input fields. With an output field, you can make CP screen representation conditional either upon a flag (i.e., two states) or upon a value in the data block (i.e., eight states maximum).

The value must be located in the same data block as specified in the "Output is fetched from" line. The memory access type is always data word (DW).

For the program EXAMPLE, specify "FLAG CONTROLLED" in the COLORS & ATTRIBUTES field, and branch to the C & A FLAG CONTROLLED form by pressing the <F1 DYNAM.C&A> function key. You get to the C&A DB CONTROLLED form by specifying "DB CONTROLLED" and pressing the <F1 DYNAM.C&A> function key.

Fill in the C&A FLAG CONTROLLED form as shown on the next page, and exit the form by pressing the <F6 ENTER DATA> function key. You are once again in the VALUE OUTPUT FIELD form. Then store the field in the dynamic display by pressing the <F6 STORE FIELD> function key.

Position the second VALUE OUTPUT field in line 4, column 28. The output value should be fetched from data block 20, data word 1, and the colors and attributes should be controlled by flag F 20.1. To do this, change the ADDRESS field in the VALUE OUTPUT FIELD form, and the flag in the C&A FLAG CONTROLLED form. Then store the field by pressing the <F6 ENTER DATA> function key twice.

```
-> PROGR. DYNAMIC DISPLAY -> SPEC. FELD ->              COROS / COM GRAPH
C & A   F L A G   C O N T R O L L E D                       DEVICE: CP526
--------------------------------------------------------------------------
                                             DRIVE: B    PROGRAM: EXAMPLE

DYN. DISP: DYN1            STAT. DISP: BACK1          SYMBOL SET: ss3
generate   FIELD: 003               LINE: 04   COLUMN: 06


                          FLAG:            020.0

                          BACKGROUND COLOR  FOREGROUND COLOR  FLASHING
            FLAG BIT  = 0 :      BLACK            WHITE           N
            FLAG BIT  = 1 :      BLACK            WHITE           Y




      F 1       F 2       F 3       F 4       F 5       F 6       F 7       F 8

                                                      SAVE      HELP      EXIT
```

5.4.4.5   Bar Output

Numeric values are represented in analog form using bars.

In the example, the liquid level in the tank is represented by a bar. For the first bar, position the cursor in line 14, column 8, and press the < F3 BAR> function key in the PROGRAM DYNAMIC DISPLAY form

Value output with bars is structured similarly to digital value output; you only need to enter a few additional parameters.

)

```
-> USER DATA -> PROGR. DYNAMIC DISPLAY ->                    COROS / COM GRAPH
B A R   O U T P U T                                               DEVICE: CP526
-------------------------------------------------------------------------------
                                              DRIVE: B   PROGRAM: EXAMPLE

DYN. DISP: DYN1          STAT. DISP: BACK1           SYMBOL SET: ss3
generate   FIELD: 005              LINE: 15   COLUMN: 05

Output is fetched from CPU-No.1, DB 020 from ADDRESS DW 000.$, FORMAT 16BIT FXP
             The actual value output is calculated from
OUTPUT = VALUE * (1                      v   ) + (0                    v   )
Indirectly specified conversion factors reside in the output DB.

UPDATE RATE: 1   s    If updating dependent on FLAG BIT: F   .

BAR TYPE: VERTICAL     WIDTH: 13    max. lengths:  NEGATIVE 00  POSITIVE 10
Max. value in positive direction: 10000 (In negative direction, if POSITIVE=00)

    COLORS & ATTRIBUTES: STATIC       Only fill in for static C&A:
    BACKGROUND COLOR:  BLUE    FOREGROUND COLOR:  CYAN       FLASHING:  N


   F 1        F 2        F 3        F 4        F 5        F 6        F 7        F 8
  DYNAM.                                                 STORE
  C&A                                                    FIELD      HELP       EXIT
```

The BAR TYPE field determines the direction of the bar. Enter either "HORIZONTAL" or "VERTICAL".

The WIDTH field determines the width in characters of a vertical bar or the height in characters of a horizontal bar.

The MAX. LENGTHS field determines the character length of a bar in a positive or negative direction.

The MAXIMUM VALUE IN POSITIVE DIRECTION field shows the maximum numeric value with reference to the maximum bar length.

Fill in the fields as shown, and store the output field by pressing the <F6 STORE FIELD> function key. You then return to the PROGRAM DYNAMIC DISPLAY form. The bar is represented on the monitor by a series of "&" symbols.

The second bar is positioned in line 15, column 27. The output should be fetched from data block 20, data word 1. Change the foreground color to red; the remaining inputs can be taken from the first bar. Change the inputs in the ADDRESS and FOREGROUND COLOR fields, and store the field by pressing the <F6 STORE FIELD> function key. Then store it in the dynamic display by pressing the <F5 STORE DYN. DISP> function key.

5.4.4.6   Shape Output

You use the SHAPE OUTPUT function to output segments, depending either on a flag bit (i.e., two states), or on the value in a data block (i.e., 16 states maximum).

In the example, this is the way a valve is displayed either in an open or a closed state.

The segments can be any size (e.g., the valve in the example consists of only one symbol).

The specified position of the shape field in the dynamic display refers to the beginning of the segment (see also segment beginning).

Note that the segment is not represented in its full size, but as a single " >." The position of the SHAPE field in the dynamic display indicates the start of the segment. With complex segments, use the <F2 REPOS. SEGMENT> function key in the SEGMENT form to determine whether the segment fits into the static display at this location.

Generate two shape fields for the left and right tanks in the static display. Press the <F4 FIGURE> function key to enter the SHAPE OUTPUT form. The data you should input is shown in the following two forms:

```
┌─────────────────────────────────────────────────────────────────────────┐
│                                                                           │
│   -> USER DATA -> PROGR. DYNAMIC DISPLAY ->              COROS / COM GRAPH │
│   S H A P E   O U T P U T                                   DEVICE: CP526  │
│   --------------------------------------------------------------------     │
│                                                 DRIVE: B   PROGRAM: EXAMPLE │
│                                                                           │
│   DYN. DISP: DYN1            STAT. DISP: BACK1         SYMBOL SET: ss3     │
│   generate   FIELD: 007          LINE: 18   COLUMN: 16                     │
│                                                                           │
│                                                                           │
│   UPDATE RATE: 1    s  When updating dependent on FLAG BIT: F    .        │
│                                                                           │
│   If alternative dependent on SWITCHOVER FLAG: F 020.0 (Alternatives 0.1), if not, │
│   the alternative (0 .. 15) is fetched from CPU-No.   DB:      ADDRESS:    │
│                                                                           │
│   Note: * = Segment does not exist, 7 = name wrong (deleted with YES ack.) │
│                                                                           │
│    0: PART6           1: PART5          2:              3:                 │
│    4:                 5:                6:              7:                 │
│    8:                 9:               10:             11:                 │
│   12:                13:               14:             15:                 │
│                                                                           │
│                                                                           │
│     F 1        F 2        F 3        F 4        F 5        F 6     F 7     F 8 │
│                                                          STORE            │
│                                                          FIELD     HELP   EXIT │
│                                                                           │
└─────────────────────────────────────────────────────────────────────────┘
```

The UPDATE RATE, WHEN UPDATING DEPENDENT ON FLAG BIT, CPU, DB, and ADDRESS fields have the usual meanings.

You may specify either the SWITCHOVER FLAG field, or the CPU, DB, and ADDRESS fields.

In the ALTERNATIVES 0,1 field, enter segments that are output depending on the condition of a flag bit. If you specify the CPU number, DB, and address, then you can enter from 2 to 16 segments. The display that is then output depends on the respective value in the data block (i.e., 0 to 15).

```
┌─────────────────────────────────────────────────────────────────────────────┐
│   -> USER DATA -> PROGR. DYNAMIC DISPLAY ->              COROS / COM GRAPH     │
│   S H A P E   O U T P U T                                    DEVICE: CP526     │
│   ---------------------------------------------------------------------------- │
│                                               DRIVE: B   PROGRAM: EXAMPLE      │
│                                                                                │
│   DYN. DISP: DYN1          STAT. DISP: BACK1           SYMBOL SET: ss3          │
│   generate   FIELD: 008              LINE: 18   COLUMN: 28                      │
│                                                                                │
│                                                                                │
│   UPDATE RATE: 1   s  When updating dependent on FLAG BIT: F    .              │
│                                                                                │
│   If alternative dependent on SWITCHOVER FLAG: F 020.1 (Alternatives 0.1), if not, │
│   the alternative (0 .. 15) is fetched from CPU-No.   DB:     ADDRESS:          │
│                                                                                │
│   Note: * = Segment does not exist, ? = name wrong (deleted with YES ack.)     │
│                                                                                │
│     0: PART6         1: PART7         2:              3:                        │
│     4:               5:               6:              7:                        │
│     8:               9:              10:             11:                        │
│    12:              13:              14:             15:                        │
│                                                                                │
│                                                                                │
│      F 1      F 2      F 3      F 4      F 5      F 6      F 7      F 8          │
│                                                 STORE                           │
│                                                 FIELD    HELP     EXIT          │
└─────────────────────────────────────────────────────────────────────────────┘
```

5.4.4.7   Time Output

The CP 526 has a built-in clock that you can integrate into the display at any location.

The output format is determined in the TIME OUTPUT FIELD form. Parts of the specification can be missing, so that only hours, minutes, or seconds are output. Double specifications or gaps are not permitted.

NOTE:
With date and time input, you must enter the complete specification, in any order .

You can change the dividing characters between H, M, and S (and between M, D, and Y for date) in the CP/OP PARAM. ASSIGNMENT form.

The US standard is the ":" symbol for time and the "/" symbol for date.

```
-> USER DATA -> PROGR. DYNAMIC DISPLAY ->              COROS / COM GRAPH
T I M E   O U T P U T   F I E L D                         DEVICE: CP526
-------------------------------------------------------------------------
                                            DRIVE: B   PROGRAM: EXAMPLE

DYN. DISP: DYN1            STAT. DISP: BACK1          SYMBOL SET: ss3
generate   FIELD: 09                LINE: 23   COLUMN: 73



           TIME:  H  M  S     H - Hour      M - Minute      S - Second





  BACKGROUND COLOR: RED        FOREGROUND COLOR: WHITE        FLASHING: N


    F 1        F 2        F 3        F 4        F 5        F 6        F 7        F 8
                                                          STORE
                                                          FIELD      HELP       EXIT
```

5.4.5   Pixel Graphics

5.4.5.1   General

You can only use the pixel graphics functions with the expansion board.

The pixel graphics functions substantially increase the possibilities for process visualization. The following four pixel graphics window types are available:

- curve windows
- bar windows
- pixel graphics value output windows
- pixel graphics text output windows

In curve windows, you can represent measured values from the controller as a function of time. They are defined as a rectangle that covers a part of the screen. All data necessary to represent one or more curves is contained in one curve window.

Bar windows are, as in the character-based graphics, a special form of representation for numeric values in bar form. In addition to the character-based graphics, a value can be defined which, when exceeded, causes the color of the bar to change.

The pixel graphics value output function allows you to view process data on the video screen in up to 16 sizes, 4 directions, and 8 colors.

Pixel graphics text outputs are static texts that describe the other windows.

You can assign an individual update rate for each curve, bar window, and value output. Each specification may be selected, due to speed rates, from preset time intervals. Eight different update rates are possible (with curves, the controller is an added alternative).

For each window, the values to be output can be fetched from any controller data block.

The transfer of data blocks between the CP and controller takes considerable time. Therefore, to avoid slowing down display updating, you should limit access to only one data block, whenever possible.

A pixel graphics display can contain up to 40 windows.

The program EXAMPLE consists of the pixel graphics displays PXG1 and PXG2.

5.4.5.2 Description of the Pixel Graphics Display PXG1

The following paragraphs describe how to generate the pixel graphics display PXG1, consisting of two pixel graphics text outputs.

In the PROGRAM USER DATA form, enter "PXG1", the name of the first display, in the PIXEL GRAPHICS DISPLAY field.

All other fields in this form remain blank to generate PXG1.

The STATIC DISPLAY, SYMBOL SET NAME, and DYNAMIC DISPLAY fields in the PROGRAM USER DATA form define parts of the pixel graphics display that you are generating or editing.

To make it easier to fit a pixel graphics display into a total display, you can preassign the background a static background display and a dynamic display (input with the < F7 HELP> function key is possible).

Once the display names are entered, the CP screen is preprocessed with the specified displays. Dynamic value outputs and time outputs are marked by " + ," dynamic value inputs by "?," bars by "&," and figures by ">," on the CP screen. Press the <F3 PIXEL GRAPHICS> function key to enter the PIXEL GRAPHICS DISPLAY form.

In this form, you define the positions of the windows in a pixel graphics display on the CP screen. (The location of the reference point of a window is explained in the detailed description of the individual windows on the following pages).

```
┌──────────────────────────────────────────────────────────────────────────────┐
│                                                                                │
│    -> SELECTION -> USER DATA ->                        COROS / COM GRAPH        │
│    P I X E L G R A P H I C   D I S P L A Y                  DEVICE: CP526       │
│    --------------------------------------------------------------------------  │
│                                                    DRIVE: B   PROGRAM: EXAMPLE  │
│    PXLGRA.D.:PXG1              DYN.D.:             STAT.D.:              SS.:    │
│                                                                                │
│                                                                                │
│                                                                                │
│    CURSOR:                     LINE: 10 COLUMN: 18                              │
│                                Change between PG and CP cursor by pressing ENTER key │
│                                                                                │
│    WINDOW NO.: 01              For CHANGE WINDOW, DELETE WINDOW and MOVE WINDOW │
│                                the window number is valid. The line and column only │
│                                apply if no entry is made for the window number. │
│                                                                                │
│                                                                                │
│    NAME to be stored:          PXG1                                            │
│                                                                                │
│    Window representation:     * Curve      ! Bar   8 Pxlgr. value   % Stat. text │
│                                                                                │
│                                                                                │
│       F 1        F 2        F 3        F 4        F 5      F 6      F 7      F 8 │
│    CURVE      PXLGRA.   PXLGRA.   STATIC    STORE                               │
│    WINDOW     BAR       VALU.OUT  TEXT      DISPLAY   CONTINUE  HELP     EXIT   │
│                                                                                │
└──────────────────────────────────────────────────────────────────────────────┘
```

```
┌──────────────────────────────────────────────────────────────────────────────┐
│                                                                                │
│       F 1        F 2        F 3        F 4        F 5      F 6      F 7      F 8 │
│    PREVIOUS   NEXT      CHANGE    DELETE    MOVE                                │
│    WINDOW     WINDOW    WINDOW    WINDOW    WINDOW    CONTINUE  HELP     EXIT   │
│                                                                                │
└──────────────────────────────────────────────────────────────────────────────┘
```

Again, you can control the cursor position (i.e., line and column numbers) either by pressing the <ENTER> key and using the cursor control keys, or by entering numbers directly in the LINE and COLUMN fields.

With the WINDOW NO. field, if several windows are defined in one pixel graphics display, then the windows are numbered in increasing, consecutive order. The assignment of window numbers is, in contrast to dynamic displays, entirely dependent on the order that they are generated.

To identify a window, you can either enter the window number directly in the WINDOW NO. field, or press the <F1 PREVIOUS WINDOW> and <F2 NEXT WINDOW> function keys in the second part menu.

Pressing the <F1 PREVIOUS WINDOW> or <F2 NEXT WINDOW> function keys in the second part menu writes the number of the window positioned either before or after the current cursor position to the field.

Press the <F4 STAT.TEXT> function key in the first menu to get to the PIXEL GRAPHICS TEXT OUTPUT form.

Fill in the form as shown:

```
 -> USER DATA -> PXLGRA. DISPLAY ->                          COROS / COM GRAPH
 P X L G R A .  T E X T  O U T P U T                         DEVICE: CP526
 ------------------------------------------------------------------------------
                                                DRIVE: B   PROGRAM: EXAMPLE
 PIXELGRA.DISP: PXG1              WINDOW:  01  generate    LINE: 10  COLUMN: 18




 TEXT:
 EXAMPLE TO

 COLOR: MAGENTA                             ENLARGEMENT FACTOR:   4

 The text is output at an ANGLE of 0    degrees. Only multiples of 90
 degrees are allowed (0 to 270 degrees)

                                   90
                            180  +  0
                                  270


   F 1        F 2        F 3        F 4        F 5        F 6       F 7       F 8
                                                         STORE
                                                         WINDOW    HELP      EXIT
```

In this form, you can define a static output text. You can output this text in pixel graphics in 16 sizes and 4 directions, and you may use it to label other windows or to generate headings.

The reference point for pixel graphics text outputs is always in the upper left-hand corner, and is labeled "**o**" in the following shape:

```
o ┌──────────────────┐
  │  TEXT            │
  │                  │
  └──────────────────┘
```

COM checks, based on the inputs made in the LINE, COLUMN, ANGLE, and ENLARGEMENT FACTOR fields, and the text length, whether the text exceeds the display screen limits. It does not check for overlapping with other windows.

You can make alternative entries in the COLOR and ANGLE fields by pressing the < F7 HELP > function key.

The COLOR field indicates the color in which the text is output on the CP monitor.

The ANGLE field determines the direction in which the text is output. The pivot of this field is always the first character in the upper left-hand corner.

In the ENLARGEMENT FACTOR field, the numbers from 1 to 16 determine text enlargement in multiples of 8 x 18 pixels. A pixel is one display element (a character-based graphics use 8 x 18 pixels).

If you enter no enlargement factor, or "1", the text is not enlarged.

By pressing the < F6 STORE WINDOW > function key, you store the pixel graphics window in the pixel graphics display, after a validity check. At the same time, the program returns to the PIXEL GRAPHICS DISPLAY form, in which the message **WINDOW STORED** appears. The area on the CP screen covered by the static text is marked by a series of "%" symbols.

)

To install a second static text, repeat the preceding steps. Input the reference point of the second static text in the PIXEL GRAPHICS DISPLAY form as follows:

LINE: "15"
COLUMN: "18"

Pressing the < F4 STAT. TEXT> function key brings you once again to the PIXEL GRAPHICS TEXT OUTPUT form.

```
-> USER DATA -> PIXELGRA. DISPLAY ->                    COROS / COM GRAPH
P X L G R A .   T E X T   O U T P U T                        DEVICE: CP526
----------------------------------------------------------------------------
                                            DRIVE: B   PROGRAM: EXAMPLE
PIXELGRA.DISP: PXG1              WINDOW: 02   generate    LINE: 15  COLUMN: 18



TEXT:
COM 5 2 6

COLOR: MAGENTA                            ENLARGEMENT FACTOR:   4

The text is output at an ANGLE of 0    degrees. Only multiples of 90
degrees are allowed (0 to 270 degrees)

                                  90
                          180   +   0
                                 270


   F 1        F 2        F 3        F 4        F 5        F 6        F 7        F 8
                                                        STORE
                                                        WINDOW      HELP       EXIT
```

After storing the second window by pressing the <F6 STORE WINDOW> function key, store the display under the name "PXG1" to drive C by pressing the <F5 STORE DISPLAY> function key in the first menu of the PIXEL GRAPHICS DISPLAY form.

The message **ELEMENT STORED** appears. Pressing the <F8 EXIT> function key brings you back to the PROGRAM USER DATA form.

5.4.5.3   Description of the Pixel Graphics Display PXG2

The following paragraphs describe how to generate pixel graphics display PXG2, consisting of one curve window:

In the PROGRAM USER DATA form, enter the name "BACK1" in the STATIC DISPLAY field and the name "DYN1" in the DYNAMIC DISPLAY field or press the <F7 HELP> function key.

Now enter the name "PXG2" in the PIXEL GRAPHICS DISPLAY field.
Pressing the <F3 PIXEL GRAPHICS> function key in the PROGRAM USER DATA form brings up the static display BACK1 and the dynamic display DYN1 on the CP screen.

To construct the coordinate system of the curve window, you must define certain symbols in the upper part of the symbol set during output (for further information, see the CP description, Chapter 3).

)

Curve Window

After entering the reference point for the curve window in the PIXEL GRAPHICS DISPLAY
form (i.e., LINE = 5, COLUMN = 50), press the <F1 CURVE> function key to arrive in the
CURVE WINDOW form.

```
-> USER DATA -> PXLGRA. DISPLAY ->                          COROS / COM GRAPH
C U R V E   W I N D O W                                         DEVICE: CP526
-----------------------------------------------------------------------------
                                                  DRIVE: B   PROGRAM: EXAMPLE
PIXELGRA.DISP: PXG2               WINDOW: 01  generate    LINE: 05  COLUMN: 50
                                                  ^ GREATEST SCALE INT:
   +---------------------------------------------+  - 220
   | WINDOW SIZE:  26x11 Characters              |
   | LABELING of window:    Tank Temperature     |
   | output OLD VALUE: Y    UPDATE RATE: 1   s   |
   | output GRID: Y                              |
   |                                             |  STEP SIZE:
   | OUTPUT VALUES fetched from:                 |  - direct           100
   | CPU: 1, DB: 021 START ADDRESS DW 000        |    or indirect DW
   |                                             |
   | INDIRECT VALUES are fetched from: CPU:  , DB: |  SMALLEST SCALE INT:
 < +---------------------------------------------+  - direct           20
LABELING of                  STEP SIZE:   1        0  or indirect DW
Time axis:   Min             TIME UNIT:   M

CURVE NUMBER: 1


    F 1        F 2       F 3       F 4       F 5       F 6      F 7       F 8
 PROGRAM    PROGRAM                                   STORE
 CURVE      LIMITS                                    WINDOW   HELP      EXIT
```

This form defines a flowing curve diagram in which a maximum of six curves can be
depicted. After each refresh cycle (i.e., update time) the curves represented in the
diagram are shifted to the left. This shifting is dependent upon the update time and the
selected scaling for the curve window.

If you exit the fields GRID, ADDRESS, INDIRECT VALUES ARE FETCHED FROM: DB, STEP SIZE DIRECT and INDIRECT DW, SMALLEST SCALE INT. DIRECT and INDIRECT DW, or STEP SIZE OF TIME AXIS by simply pressing the <RETURN> key, or by pressing the <FIELD RIGHT> key, then a nonstandard type of cursor control is used.

First, the input fields that are located in the coordinate system are processed (with a window size up to the indirect value address), followed by the y-axis data, and last the time-axis data.

If you exit the INDIRECT VALUES ARE FETCHED FROM: CPU field by pressing the <FIELD RIGHT> or <RETURN> keys without entering any information, then the cursor moves instantly to the STEP SIZE DIRECT field. At the same time, any contents of the INDIRECT VALUES ARE FETCHED FROM: DB field are erased.

If you do not make any entry in the STEP SIZE DIRECT field, the cursor moves to the STEP SIZE INDIRECT DW field; if you do, it moves to the SMALLEST SCALE INT. DIRECT field.

The same applies for the SMALLEST SCALE INT. DIRECT field; if you make an entry in the SMALLEST SCALE INT. DIRECT field, the cursor jumps to the LABELING OF STEP SIZE field after you press <RETURN>.

Upon exiting the STEP SIZE and SMALLEST SCALE INT. fields that are needed for labeling the ordinates, with direct specification of both values, COM computes and outputs the greatest scale integer. If at least one value is specified indirectly, the word INDIRECT appears in the GREATEST SCALE INT. field.

Scaling in X-Direction

Time is depicted on the x-axis. The past is represented in a negative direction on the x-axis, (i.e., the curve moves from right to left). The present is represented by the farthest point to the right .

You determine the total time to be represented by selecting a scale classification and its associated step size. The total time is calculated from these values.

Each tenth scaling mark on the x-axis is elongated and labelled with the value defined by the specified step size. Scaling marks are always output in equal intervals (i.e., in intervals of eight pixels), regardless of the window size . Therefore, the number of scaling marks differs according to the window size (see Table 5-1).

Scaling in Y-Direction

You can read the current measured value at any given time from the y-axis ordinates.

In y-direction, scaling marks are output in intervals of 18 pixels. Each fourth scaling mark is elongated and labelled with the value defined by the specified step size.

In addition, the smallest scale integer that the curve can display must be specified.

As an example of smallest scale integer and step size specification on the y-axis, with a selected window size of 26 x 11, according to Table 5-1, the number of scaling marks on the y-axis is two.

If the estimated measured value is between 120 and 180, then you should use the following examples of scaling: (see over)

Example 1: Smallest scale integer 100, step size 50

```
                          ↑
                         ─┼200
                          │
                    *     │
                          │
               *         ─┼150
                          │
                          │
                          │100
      ←───────┼───────┼───────┼─────┘
          30      20      10      0
```
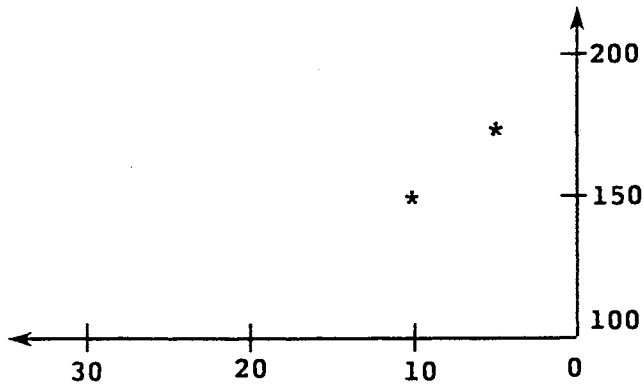
Example 2: Smallest scale integer 0, step size 100

For the same window size and number of scaling marks, the reading is less accurate in this example.

```
                          ↑
                         ─┼200
                    *     │
               *          │
                          │
                         ─┼100
                          │
                          │
                          │0
      ←───────┼───────┼───────┼─────┘
          30      20      10      0
```
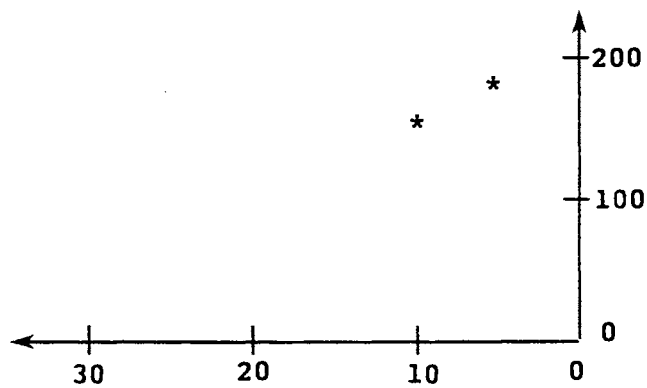
You can make alternative entries in the WINDOW SIZE, GRID, OLD VALUE, UPDATE RATE, TIME UNIT, and CURVE NUMBER (i.e., if curves are already set up) fields by pressing the <F7 HELP> function key.

In the GREATEST SCALE INT. field, with direct specification of step size and smallest scale integer, you can calculate and display the greatest scale integer for the specified window size ; otherwise the word **INDIRECT** appears.
If calculation is possible, the following formulas apply:

greatest scale integer - smallest scale integer < = 32767

greatest scale integer < = 32767

If these conditions are not fulfilled, then no greatest scale integer is displayed.

The error message **Value range not permissible** does not appear, however, until you press the <F1>, <F2>, or <F6> function keys.

If you change the window size, then the greatest scale integer is also updated.

With the WINDOW SIZE field, allowed inputs are dependent on the selected cursor position, since COM does not allow you to store a window that exceeds the video screen limits.

If the cursor is positioned incorrectly when you exit the WINDOW SIZE field or press the <F7> function key, the warning **Window not correctly positioned** appears.

COM does not check for overlapping with other windows.

When you select the smallest curve window, remember that you can only represent six curve diagrams on the screen at one time. If you select the largest size, then the curve window covers the entire screen.

NOTE:
With a window size of 80 x 22, the window label at the right border overwrites the ordinate value of the scaling when the maximum label length is used.

Example 1: Window size 80 columns x 22 lines

```
o
        Curve 1      Curve 2      Curve 3      Curve 4      Curve 5      Curve 6

        Heading
                                                                          400

                                                                       ┼  300

                                                                       ┼  200

                                                                       ┼  100

                                                                          0
       - 60       - 50       - 40       - 30       - 20       - 10       0 s
```
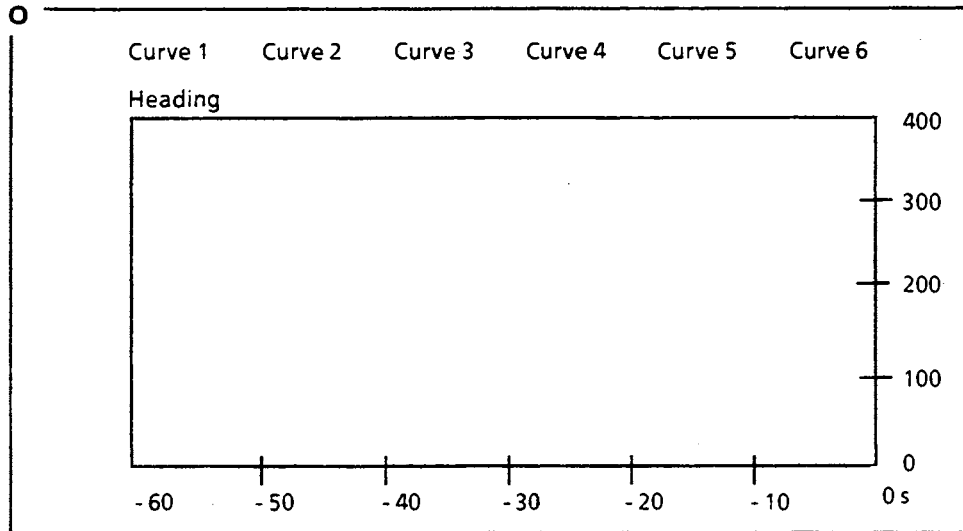
Example 2: Window size 26 columns x 11 lines (minimum size)

```
o
     Curve 1      Curve 2
     Heading
                              800

                           ┼  400

                              0
   - 20       - 10       0 s
```
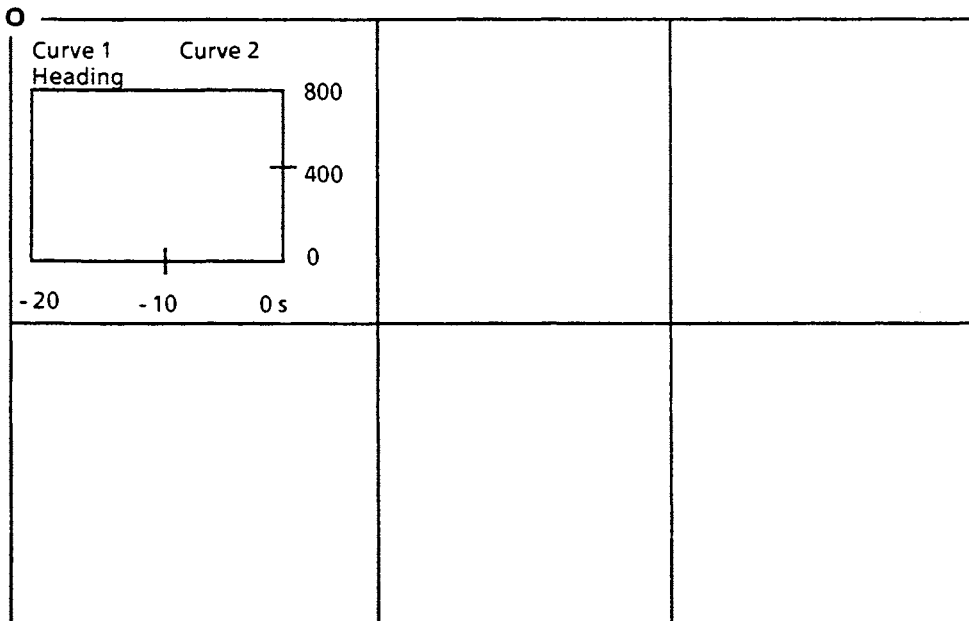
The reference point for a curve window is located in the upper left-hand corner, and is labeled "o" in the figure.

Window Size Column x Line	Scaling Marks on X-Axis	Scaling Marks on Y-Axis
80 x 22	6	4
80 x 11	6	2
40 x 22	3	4
40 x 11	3	2
26 x 11	2	2

Table 5-1: **Number of Elongated Scaling Marks (labeled) for Various Window Sizes**

The specified size in lines and columns refers to the entire window
(i.e., the coordinate system with labels). The size of the coordinate system in which the curves are represented may be taken from the following table:

Window Size	Coordinate System in Characters	Coordinate System in Pixels
80 x 22	60 x 16	480 x 288
80 x 11	60 x 8	480 x 144
40 x 22	30 x 16	240 x 288
40 x 11	30 x 8	240 x 144
26 x 11	20 x 8	160 x 144

Table 5-2: **Sizes of Coordinate Systems in Characters and Pixels for Various Window Sizes**

With the LABELING OF WINDOW field, you can output a heading that functions simultaneously as the ordinate label. The heading is output in the color white.

You can enter either "Y" or "N" in the OLD VALUES field. If you specify "Y", then a complete curve diagram, stored by the controller in a data block, can be output with old measured values. Afterwards, the curve is updated as usual (except with PC updating). The data block layout may be found in the description of the CP expansion board.

With the UPDATE RATE field, you can assign cycle times in which the curve windows are updated, specifically for each curve window.

The total monitoring time span is determined by the selected step size on the x-axis, the time unit, and the window size.

Example: Determining the Monitoring Time Span

TIME UNIT   "S"
STEP SIZE   "10"
WINDOW SIZE   "80 x 22"

(according to Table 5-1, there are six scaling marks on the x-axis)

Monitoring interval = number of scaling marks x step size = 6 x 10 = 60 s

You must coordinate the choice of update rate and desired monitoring time span.

If you choose slow updating, then the monitoring time span must be correspondingly long.

You should not represent too many curves with a short refresh time on the video screen at the same time, because it may not be possible to maintain the refresh time.

)

In exceptional cases, you may select the controller as an alternative. In such a case, no cyclic updating is done by the firmware, and it is left up to the controller to decide when to update. The collected old values are output with this kind of updating.

The shorter the update time with the same monitoring time span, the more vertices the curve has.

Example 1: Step size 10, Time Unit s, Update Rate 10 Seconds
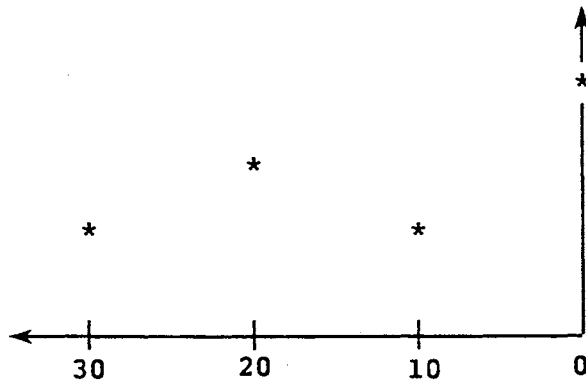
Example 2: Step Size 10, Time Unit s, Update Rate 5 Seconds
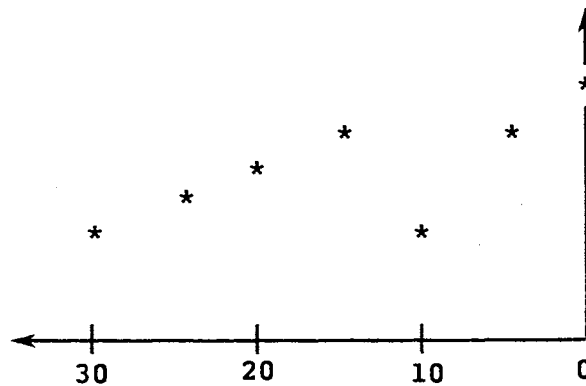
The GRID field allows you to read the function values more easily by giving the window a predefined background grid.

With the STEP SIZE field, you can input the value for the step size on the y-axis either directly in numbers, or indirectly. With indirect input, you must fill in both the INDIRECT VALUES ARE FETCHED FROM: CPU and DB fields .

You must fill in the CPU field because a controller can have several CPUs. This number indicates which CPU (i.e., 1 - 8) the value is fetched from.

If you specify "N" in the OLD VALUES field, and select PC updating, then the CPU, DB, and ADDRESS fields can remain blank.

In the DB field, you specify the data block within the prespecified CPU, from which the value is fetched (i.e. 2 - 255).

In the ADDRESS field, you must enter the data block address (i.e., 0 - 255). You can find information on the total layout of the data block in the controller in the description of the CP expansion board.

In the SMALLEST SCALE INT. field, you can input this value either directly in numbers, or indirectly. With indirect input, you must fill in both the INDIRECT VALUES ARE FETCHED FROM :CPU and DB fields.

The step size you enter in the STEP SIZE OF TIME AXIS field becomes the value at the first elongated scaling mark on the axis. The step size for the time axis is defined positive, but is output as negative time when the coordinate axes are output. If you specify an "S" for seconds in the TIME UNIT field, then only inputs in multiples of ten are allowed in this field.

With the LABELING OF TIME AXIS field, a three-character long text can stand next to the time axis, which specifies the dimension of the entered values.

The entry you make in the CURVE NUMBER field specifies the number of curves to appear in a window. There can be a maximum of six curves per curve window. However, you should not assign this many to one window, as they become increasingly difficult to differentiate with decreasing size.

You must define at least one curve per window before it can be stored.

Pressing the <F2 PROGRAM LIMITS> function key brings you to the LIMITS form, if you
have made the correct inputs in the SMALLEST SCALE INT. and STEP SIZE fields. In this
form, you have a choice between determining upper limit, reference value, and lower
limit. You can, but do not have to specify all of these values.

```
┌─────────────────────────────────────────────────────────────────────────────┐
│  -> PXLGRA. DISPLAY -> CURVE WINDOW ->                    COROS / COM GRAPH    │
│  L I M I T S                                              DEVICE: CP526        │
│  --------------------------------------------------------------------------   │
│                                          DRIVE: B    PROGRAM: EXAMPLE          │
│  PIXELGRA.DISP: PXG2           WINDOW: 01  generate      LINE: 05  COLUMN: 50  │
│  The following values fix the limits for the fill-in pattern changeover in    │
│  the window. Indirectly specified values are fetched from  CPU:  , DB:    .   │
│                                                                               │
│                                            ┌──────────────────────┐  - 220    │
│                                            │ ╫╫╫                   │           │
│  UPPER LIMIT:    direct      150           │                      │           │
│                  or indirect DW            │                      │           │
│                                            │                      │           │
│  REF. VALUE:     direct      100           │                      │  - 120    │
│                  or indirect DW            │                      │           │
│                                            │   ╫╫╫╫╫╫╫╫╫           │           │
│  LOWER LIMIT:    direct       50           │   ╫╫╫╫╫╫╫╫╫           │           │
│                  or indirect DW            │   ╫╫╫╫╫╫╫╫╫           │           │
│                                        <   └──────────────────────┘  - 20     │
│                                                                               │
│    F 1        F 2        F 3        F 4        F 5        F 6      F 7     F 8  │
│                                                          STORE                 │
│                                                          LIMITS          EXIT  │
└─────────────────────────────────────────────────────────────────────────────┘
```

This form also shows the greatest and smallest scale integers for direct output.

Depending on the window size entered, the other scaling marks are also labeled.

When you specify a value in the REFERENCE VALUE field, curves are layered under one another, starting from the reference value line (see also the description of the FILL-IN PATTERN NO. field in the CURVE form).
If no reference value is specified, then layering starts with the smallest scale integer.

The UPPER LIMIT and LOWER LIMIT fields specify limit values. If these limits are exceeded, the fill-in type changes to solid shading; the color remains the same.

You can enter values either directly in numbers, or indirectly. With indirect input, you must fill in the INDIRECT VALUES ARE FETCHED FROM CPU and DB fields in the CURVE WINDOW form.

The following figure is an example of limit value and/or reference value definition:
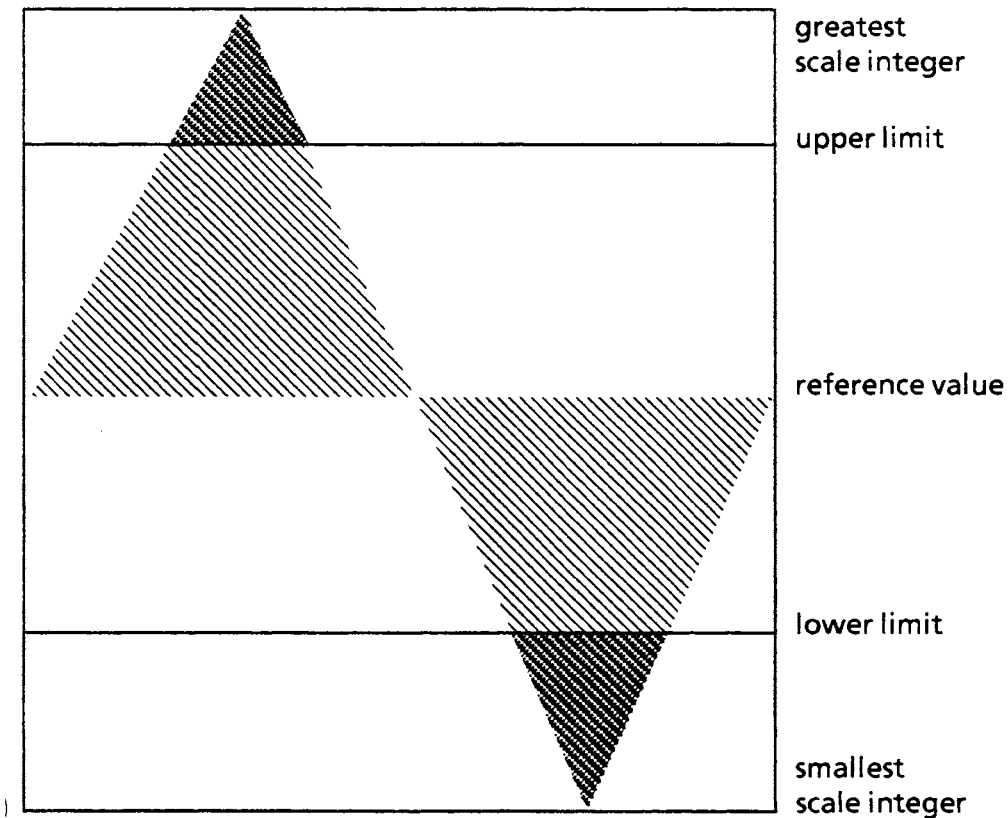
Pressing the < F6 STORE LIMITS > function key in this form stores the limit values after a validity check.

The validity check is done according to the following formula:

smallest scale integer < = lower limit < = reference value < = upper limit < = greatest scale integer

The difference between the smallest and the greatest scale integer must not exceed 32767.

If one of these values is specified indirectly, the condition is considered fulfilled and the remaining direct values are checked.

Then press the <F1 PROGRAM CURVE> function key in the CURVE WINDOW form.

```
-> PXLGRA. DISPLAY -> CURVE WINDOW ->                    COROS / COM GRAPH
C U R V E                                                    DEVICE: CP526
--------------------------------------------------------------------------
                                              DRIVE: B   PROGRAM: EXAMPLE
PIXELGRA.DISP: PXG2            WINDOW: 01  generate      LINE: 05  COLUMN: 50

                                        ^
                                      - 220   GREATEST OUTPUT VALUE:
    ┌──────────────────────────────────┐     direct        220
    │ Curve      No.:1                  │     or indirect  DW
    │                                   │
    │ LABELING of the curve   : Tank 1  │
    │ COLOR: CYAN       LINE TYPE NO.: 1 │
    │ FILL-IN PATTERN NO.:    1          │
    │                                  - 120
    │                                   │
    │ INDIRECT VALUES are fetched from: │
    │ CPU:  , DB:                       │
    │                                   │     SMALLEST OUTPUT VALUE:
<   └──────────────────────────────────┘- 20   direct         20
                                              or indirect   DW


    F 1       F 2       F 3       F 4     F 5      F 6      F 7      F 8
                                DELETE           STORE
                                CURVE            CURVE    HELP     EXIT
```

You define the individual curve parameters in the CURVE form

Using nonstandard cursor control (<RETURN> and <FIELD RIGHT> keys), pass over first the fields within the coordinate system, and then the GREATEST and SMALLEST OUTPUT VALUE fields.

With the LABELLING field, you can output a label for each diagram, in which a text of up to 10 characters is output in the corresponding color for each curve.

For each window size, a maximum number of texts supportable by the firmware is determined. For the smallest window, the maximum is two; for the next larger window, three. If there are more curves than texts, then the specified texts refer to the curves defined first. If you need further texts, you must decide whether to insert these in the static display as segments or output them as pixel graphics texts.

With the COLOR field, you can display curves in all colors that are also available in the character graphics (red, green, blue, cyan, magenta, yellow, and white).

With the FILL-IN PATTERN NO. field, when curves are layered, each point of the curve is joined to the point that is vertically below or above it on the reference value line.

In the LINE TYPE NO. field, line type 1 corresponds to a solid line.

With fill in pattern 2, layering is full. You can find further alternatives for fill in patterns and line types in the Planning Guide COM GRAPH (Chapter 4).

With the GREATEST OUTPUT VALUE and SMALLEST OUTPUT VALUE fields, since measured values for each of the six curves can have different value ranges, you can define curve-specific values, such as the smallest and the greatest output value, for each curve separately.

When the curve is output, each measured value, reference value, and lower and upper limit is recalculated to the full diagram height.

)

Again, the following condition must be fulfilled:

smallest output value < = greatest output value
greatest output value - smallest output value < = 32767

Example:
You might want to represent the time waveform of the primary voltage (-220 to + 220 Veff) and the secondary voltage (-12 to + 12 Veff) of a transformer.

If no conversion were done, you would not be able to read the value for the secondary voltage, since the curve would only deviate slightly from the x-axis.

Therefore, you should select a greatest output value of + 320 and a smallest output value of -320 for the primary voltage wave, and a greatest output value of + 17 and a smallest output value of -17 for the secondary voltage.
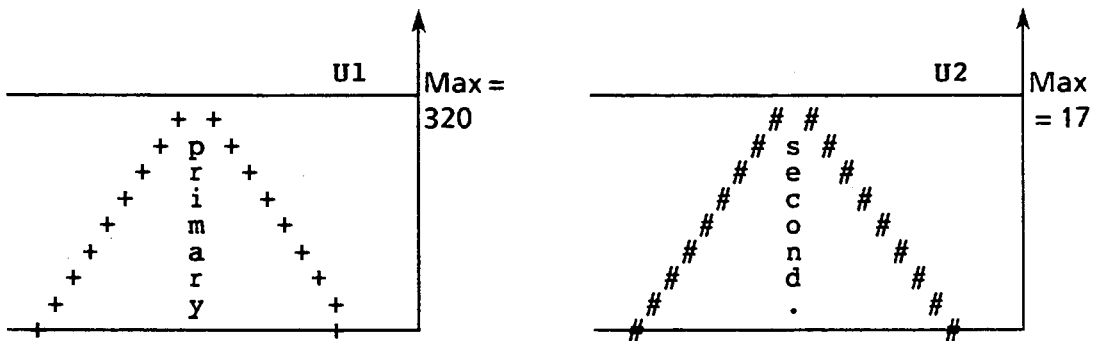
With indirect input, you must fill in the CPU NO. and DB fields in the CURVE WINDOW form.

Store this curve by pressing the < F6 STORE CURVE > function key. In the CURVE WINDOW form, enter "2" for the curve number and press the < F1 > function key again. Now enter the data as shown in the form on the following page.

```
-> PXLGRA. DISPLAY -> CURVE WINDOW ->                        COROS / COM GRAPH
C U R V E                                                     DEVICE: CP526
--------------------------------------------------------------------------------
                                                DRIVE: B    PROGRAM: EXAMPLE
PIXELGRA.DISP: PXG2              WINDOW: 01  generate       LINE: 05  COLUMN: 50

                                            ˙           GREATEST OUTPUT VALUE:
   ┌────────────────────────────────────────┐  - 220    direct       220
   │                                         │          or indirect   DW
   │  Curve No.:2                            │
   │                                         │
   │  LABELING of the curve: Tank 2          │
   │  COLOR: RED          LINE TYPE NO.: 1   │
   │  FILL-IN PATTERN NO.:    2              │
   │                                         │  -
   │                                         │
   │  INDIRECT VALUES are fetched from:      │
   │  CPU: , DB:                             │
   │                                         │
   │                                         │          SMALLEST OUTPUT VALUE:
<  └────────────────────────────────────────┘  - 20     direct        20
                                                        or indirect   DW

   F 1         F 2         F 3         F 4     F 5     F 6      F 7       F 8
                                       DELETE          STORE
                                       CURVE           CURVE    HELP      EXIT
```

Store curve 2 by pressing the < F6 STORE CURVE > function key in this form. Then store it in the window by pressing the < F6 STORE WINDOW> function key in the CURVE WINDOW form. On the CP screen, the area covered by the curve window is marked by a series of "*" symbols.

5.4.5.4 Bar Windows

Bars are a special representation form for numeric values. When a lower or upper limit value is reached, the fill-in pattern type for the exceeded part becomes automatically full, and the color changes to the selected signal color. You can, but do not have to, define such upper or lower limit values.

Otherwise, you can freely select the fill-in type for values within the allowed measured-value range (i.e., between upper and lower value limits).

Using CP cursor control, place the cursor on the first blank space in the upper left-hand corner inside tank 1. Then press the <F2 PXL GRA. BAR> function key.

You can make alternative entries in the BAR TYPE, BAR LENGTH, COLOR, AXIS, and UPDATE RATE fields by pressing the <F7 HELP> function key.

First process the BAR TYPE, BAR LENGTH, BAR WIDTH, COLOR, PATTERN NO., and AXIS fields using the special cursor control (i.e., <FIELD RIGHT> and <RETURN>) keys, then assign the ordinates, and last the output value addresses.

The **BAR TYPE** Field

The following figure is an example of a vertical bar diagram:
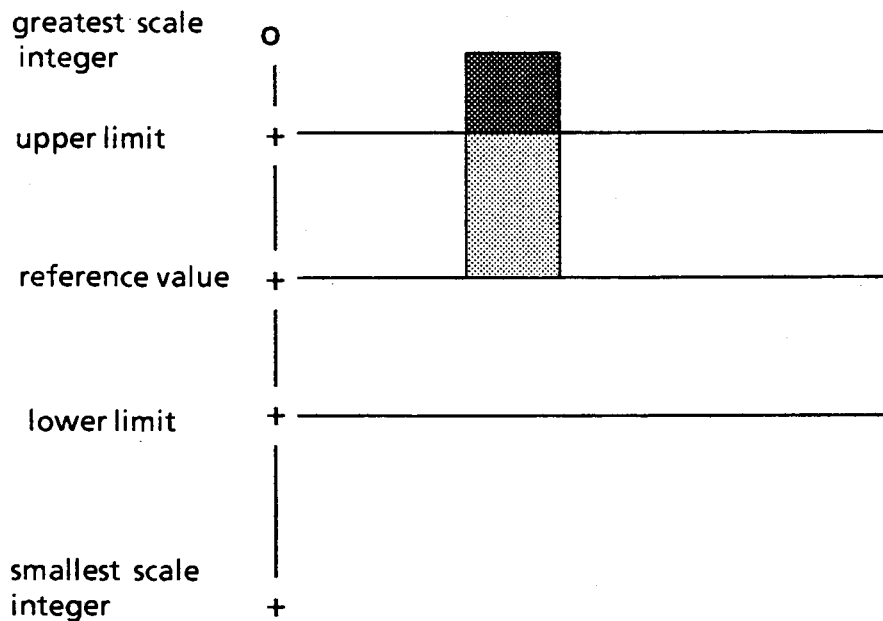
Example of a horizontal bar diagram:

gr. sc. int.   low. lim.   ref. val.   up. lim.   sm. sc. int.
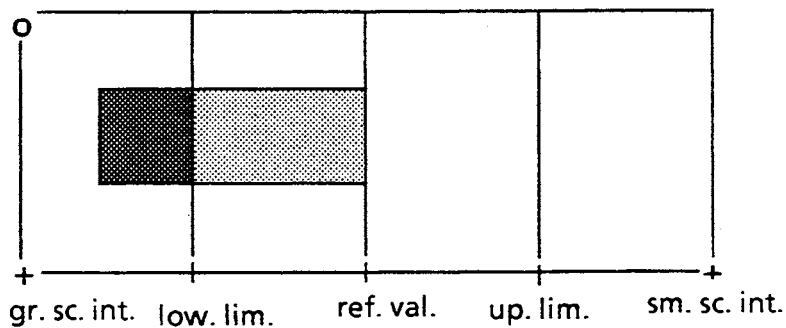
The reference point of the bar is located in the upper left-hand corner of the window and is labelled "o".

With the GREATEST SCALE INT. field, COM calculates and outputs the greatest scale integer according to window size (if the smallest scale integer is specified directly) and step size.

If you specify either one of the values indirectly, then INDIRECT is output in this field.

The BAR LENGTH Field

The entry made in this field depends on the bar type.

Either manually or with the <F7 HELP> function key, you can enter the lengths 4, 8, 12, 16, or 20 for a vertical bar, and the lengths 10, 20, 30, 40, or 50 for a horizontal bar.

)

Vertical Bar	Horizontal Bar
4 characters	10 characters
8 characters	20 characters
12 characters	30 characters
16 characters	40 characters
20 characters	50 characters

Table 5-3:   Maximum Span (distance between greatest and smallest scale integer)

Bar Length	Vertical Bar	Horizontal Bar
4/10	1	1
8/20	2	2
12/30	3	3
16/40	4	4
20/50	5	5

Table 5-4:   Number of Elongated Scaling Marks (labeled) for Various Bar Lengths

With the BAR WIDTH field, the input depends on the inputs in the BAR LENGTH and BAR TYPE fields, and on the position of the bar on the CP screen. COM does not check for overlapping with other windows.

With the AXIS OUTPUT field, the firmware generates an associated axis with corresponding scaling.

If you enter "Y" in this field, the axis is output in white. However, you must take this into account for positioning as this causes the bar dimensions to be enlarged and the software does not take this into account when visualizing the bar.

In the UPDATE RATE field, you assign a rate at which the bar can be updated in cyclic time intervals (i.e., the length of the bar is adjusted with every refresh cycle).

In the STEP SIZE field, you can enter either a direct or indirect value. With indirect input, you must fill in the INDIRECT VALUES ARE FETCHED FROM CPU and DB fields.

In the SMALLEST SCALE INT. field, you can specify the smallest scale integer either directly or indirectly. With indirect input, you must fill in the INDIRECT VALUES ARE FETCHED FROM CPU and DB fields.

Again, the following formula applies:

> greatest scale integer $< = 32767$
> greatest scale integer - smallest scale integer $< = 32767$

For information on all other input fields, see the CURVE WINDOW form and the CURVE form.

Pressing the < F2 PROGRAM LIMITS > function key brings you to the LIMITS form, in which you can define the upper limit, reference value, and lower limit.

The COLOR field in the LIMITS form indicates the color in which the part of the bar that exceeds the limits is displayed.

The REFERENCE VALUE field in the LIMITS form assigns a value to a reference line for bar output. The bar is drawn starting from the reference line; layering also starts at this line. If no reference value is specified, layering starts at the lower border of the window.

The specified values for smallest and greatest scale integer, upper limit, and reference value must fulfill the following condition:

sm. scale int. < = lower limit < = ref. val. < = up. limit < = gr. scale int.

After the bar window is stored, the area covered by the bar window on the screen is marked by a series of "!" symbols.

5.4.5.5   Pixel Graphics Value Output

Use pixel graphics value output to define dynamic texts in the "PIXEL GRAPHIC" mode. By enlarging the value, you can read it from greater distances.

Place the CP cursor on the spot where you want to place a pixel graphics value output field and press the < F3 PIXEL GRA. VALU. OUT> function key.

You can determine length, color, size, and direction of the value in this form.

In contrast to dynamic value outputs, these values cannot be converted by specifying gain and offset.

You can make alternative inputs in the UPDATE RATE, ADDRESS, FORMAT, COLOR, and ANGLE fields by pressing the <F7 HELP> function key.

The UPDATE RATE field specifies the cyclic time interval in which the output field is updated.

The reference point for pixel graphics value output is located in the upper left-hand corner and is labelled "o" in the folllowing figure:

```
O ┌─────────────────────────────────┐
  │    V A L U E   O U T P U T       │
  └─────────────────────────────────┘
```

For the FORMAT field, you can find all possible formats for value representation in the Planning Guide COM GRAPH (Chapter 4).

The number of character positions depends on the format.

COM checks whether the window exceeds the limits of the screen. Avoid exceeding the limits when you choose the inputs in the LINE, COLUMN, ANGLE, and ENLARGEMENT fields, as well as the number of character positions.

With the ANGLE field, the pivot of the output field is always the first character in the upper left-hand corner.

Pressing the <F6 STORE WINDOW> function key incorporates the pixel graphics window into the pixel graphics display, after a validity check. At the same time, the screen area covered by the pixel graphics value is marked by a series of "8" symbols.

To delete a window, press the <F1> and <F2> function keys in the second part menu of the PIXEL GRAPHICS DISPLAY form to set the number of the window (you can also enter the window number directly in the WINDOW field). When you press the <F4 DELETE WINDOW> function key, the window is deleted, after an acknowledgment.

There are three ways to identify a window for repositioning:

1. Place the CP cursor on the reference point of the window to be repositioned.

2. Enter the number of the window to be repositioned in the WINDOW field.

3. Enter the coordinates of the window to be repositioned in the LINE and COLUMN fields.

If you press the <F5 MOVE WINDOW> function key in the second part menu, the message FIELD/WINDOW COUPLED, USE F5 TO STORE! appears.

At the same time, the window is erased from the CP screen.

Place the CP cursor at the desired position (either by pressing the <ENTER> key and using the arrow keys, or by entering the coordinates of the new reference point in the LINE and COLUMN fields).

When you press the <F5 MOVE WINDOW> function key again, the message FIELD/WINDOW STORED is displayed and the window appears at its new position on the CP screen.

NOTE:
Remember that COM does not check whether a window exceeds the limits of the video screen when repositioning.

5.4.6   Functions

5.4.6.1   General

Functions are divided into commands and PC job orders.

You initiate commands by entering them on the process keyboard (either by pressing a function key that has been assigned the particular function, or by entering the command name directly).

The program initiates PC job orders depending on the status of a flag bit that corresponds to the name of the PC job order.

Commands and PC job orders assist in data exchange between the programmable controller and the CP 526, as well as in visualization of process data.

Although messages are not actually functions, message programming is also discussed in this chapter since you can only branch to the PROGRAM MESSAGE form from the FUNCTION form.

The following types of commands are available:

-   Display output command - outputs the associated process display on the CP screen

-   Function key command - assigns command names or cursor movements to function keys

-   Value input command - allows entering of new setpoint values for process variables with the process keyboard

-   Value output command - displays values in process variables on the right side of the command line

-   Set bit or reset bit command - sets or resets a flag bit

-   Status display command - outputs a text on the right side of the command line, depending on the status of a flag bit

-   Shape output command - outputs a shape at any desired position on the screen, depending on the status of a flag bit.
-   Date input or time input command - enables date and time to be input in the command line

-   Date output or time output command - loads the parameterized values for date and time into the CP clock block

-   Message display command - outputs all messages that have arrived. Note the triple meaning of the term "message" in COM (see Planning Guide, Appendix B)

The following types of job orders are available:

- Display output job order - outputs a process display on the screen, depending on the status of a flag bit that identifies the PC job order

- Date output and time output job order - fetches date and time from the controller, triggered by an interprocessor communication flag bit (same as the name of the PC job order), and loads them into the CP clock block

- Pixel graphics background updating job order - outputs old values of curves, triggered by the status of an interprocessor communication flag bit (i.e., name of the PC job order), as long as the associated pixel graphics display is visible on the CP screen. If the pixel graphics display is not visible on the screen, then COM ignores this job order.

The program EXAMPLE consists of the following functions:

- Display output commands START and DISP1
- Set bit commands SET1 and SET2
- Reset bit commands RESET1 and RESET2
- Value input command VALUE =
- Time input command TIME =
- Message display command MESSAGE DISPLAY
- Message text
- Value output command VAL.OUTP.

5.4.6.2   Value Input Command

This section explains how to generate the value input command named VALUE = , starting from the FUNCTION form.

```
-> SELECTION -> USER DATA ->                            COROS / COM GRAPH
 F U N C T I O N                                            DEVICE: CP526
 -------------------------------------------------------------------------
                                          DRIVE: B    PROGRAM: EXAMPLE


              FUNCTION NAME: VALUE =           JOB NUMBER:

 "COMMAND":   Initiate a CP526 function by entering
              a command name in the command line.

              The command name must not be a flag name.
              Input commands must be ended with "=".

 "PC JOB" :   Initiate a CP526 function by setting
              an IPC flag bit.

              The name of the PC job must be a flag name.
              Flag name: F(byte).(bit)   byte 0 - 255  bit 0 - 7



      F 1       F 2       F 3       F 4       F 5      F 6       F 7       F 8
  DISP.OUTP   FCT       VALUE     VALUE     SHAPE
  COMMAND     COMMAND   INPUT     OUTPUT    OUTPUT   CONTINUE   HELP      EXIT
```

```
      F 1       F 2       F 3       F 4       F 5      F 6       F 7       F 8
  DISP.OUTP   PC DATE/            PXLGRA.   PRINT
  JOB         TIME      MESSAGES  UPD.BACK  COMMANDS CONTINUE   HELP      EXIT
```

```
      F 1       F 2       F 3       F 4       F 5      F 6       F 7       F 8
   ENTER      ENTER     DISPLAY   S/R BIT   STATUS
   DATE       TIME      DATE/TIME COMMAND   DISPLAY  CONTINUE   HELP      EXIT
```

With the FUNCTION NAME field, the name of a value input command must end (as with all input commands) with an " = " symbol as the last character.

Names of output commands may not contain any " = " symbols. You may find permissible formats of all other commands in the Planning Guide (Chapter 4).

The name of PC job orders must designate an interprocessor communication flag bit.

An address range of F (0-255).(0-7) allows a possible 2048 interprocessor communication flag bits.

In the JOB ORDER NUMBER field, specifying a job order number is optional (except with value input commands, where it is not permitted). If you specify a job order number greater than zero, then the controller can trigger the function by sending this number to CP 526 (for information on the send direct job order, see the CP description, Chapter 3). If you do not specify a job order number, or if it is equal to zero, then this function can only be triggered by inputs on the process keyboard.

You cannot assign a job order number for value inputs, as a value input command must always be triggered via the process keyboard to avoid unwanted process status changes (e.g., turning on a motor).

If you try to change the job order number of a function that is already stored, COM requests an acknowledgment to the prompt **CHANGE JOB NO. -ALTHOUGH OLD AND NEW NO. DIFFERENT?** If you respond in the negative, the old job order number remains the same.

You can assign job orders the numbers 0 to 223, 254, and 255.

You can enter the name "HELP" in the FUNCTION NAME field, as long as other functions already exist under the specified program name.

Summary:
You can define the following types of functions as PC job orders. Each name must be a flag:

● Display output
● Date output
● Time output
● Pixel graphics background updating

While all other types of commands can also be triggered by the controller, a send direct job order is required. In this case, the name may not be a flag, and you must specify the job order number.

After you fill in the FUNCTION form, press the <F3 VALUE INPUT> function key in the first part menu to get to the VALUE INPUT COMMAND form.

```
-> USER DATA -> FUNCTION ->                              COROS / COM GRAPH
V A L U E   I N P U T   C O M M A N D                    DEVICE: CP526
---------------------------------------------------------------------------
                                       DRIVE: B   PROGRAM: EXAMPLE


             NAME: VALUE=          JOB No.:
NAME  to be stored: VALUE=         JOB No.:

Input is stored in CPU No. 1,  DB  20  from ADDRESS  DW    0.
The input is made in  16BIT FXP FORMAT  with the CHAR. POSITIONS   5.0.

The value actually stored is calculated from
VALUE=(INPUT - (0                        v    )) / (1v    )

Limit values: UPPER  LIMIT:  direct  10000or indirect
              LOWER  LIMIT:  direct  0or indirect


Indirectly specified limit values and conversion factors reside in the input DB,
indirect limit values use the same type of memory access as the input value.



   F 1        F 2        F 3        F 4        F 5        F 6       F 7        F 8

                                  DELETE                 STORE     HELP      EXIT
```

You can enter the name "HELP" in the NAME TO BE STORED field, as long as functions exist under the specified program name and in the ADDRESS and FORMAT fields.

For the NAME TO STORE field, the entry is taken from the FUNCTION NAME field in the FUNCTION form. If you want to store the function under a different name than the one entered in the FUNCTION form, you can change the name in this field. For names of commands or PC job orders, the same requirements apply.

In the JOB NO. field, you can also change the job order number (with the exception of a value input command). If a function already exists under the specified name with a different job order number, COM requests an acknowledgment to the prompt **CHANGE JOB NO. -ALTHOUGH OLD AND NEW NO. DIFFERENT?** before storing.

In the INPUT IS STORED IN CPU NO., DB, and ADDRESS fields, you determine in which CPU and controller data block the value will be stored.

Every input value can be converted and checked against limits before it reaches the process. You can specify the values for conversion or limit value check either directly or indirectly. With indirect specification, the control program takes over the responsibility of producing correct data (see also Dynamic Displays Value Input Field).

Pressing the < F6 STORE> function key in the VALUE INPUT COMMAND form stores the command VALUE = after the input data is checked. If COM finds no input errors, then the message **ELEMENT STORED** appears.

Use the < F4 DELETE> function key in this form to delete a function that is already stored on the data medium.

After pressing the <F4> function key, the prompt **DELETE ELEMENT?** appears.

If you respond affirmatively by pressing the <F1 YES> function key, then the value input command VALUE = is deleted from the data medium.

When you delete a function, you do not delete the form, so you only have to press the <F6 STORE> function key once more to regenerate the function VALUE = .

After exiting the form by pressing the <F8 EXIT> function key, you again enter the FUNCTION form.

5.4.6.3   Value Output Command

With the aid of this command the content of tank 1 is displayed in the message text of the command line.

<F4 VAL.OUTP. > in the first menu of the FUNCTION screen form takes you into the VAL.OUTP. screen form.

The tank content is fetched from CPU1, DB20, DF0, has five places and is on type 16 bit FPT.

An accompanying text to the tank content can be input in the TEXT column. This text is displayed to the right of the output value.

5.4.6.4   Time Input Command

This command allows you to input the time in the command line via the process keyboard.

Enter the name "TIME = " in the FUNCTION NAME field and press the < F2 INPUT TIME > function key in the third part menu of the FUNCTION form to branch to the TIME INPUT COMMAND form.
The order that you enter H, M, and S is random. You can enter them manually or by pressing the <F7 HELP> function key.

Since the fields of this form are already preassigned with "H M S," you can store the command immediately by pressing the <F6 STORE> function key.

You can select the separating characters for time and date in the CP 526 PARAM. ASSIGNMENT form (the U.S. standard is ":" for time and "/" for date).

The input on the process keyboard must then use the separating characters you have entered. If you give the time input command the format HH:MM:SS, then your time entry must be, for example, 22:59:30.

5.4.6.5   Set Bit and Reset Bit Commands

The following pages describe how to generate the set bit command SET1.

Pressing the <F4 S/R COMMAND> function key in the third part menu of the FUNCTION form brings you to the S/R BIT COMMAND form.

Set bit commands set a flag bit and reset bit commands reset a flag.

With this command, you can make changes in the process status (e.g., opening and closing valves.

Fill in the form as shown:

```
-> USER DATA -> FUNCTION ->                              COROS / COM GRAPH
S / R   B I T   C O M M A N D                                DEVICE: CP526
------------------------------------------------------------------------
                                          DRIVE: B   PROGRAM: EXAMPLE

                  NAME: SET1           JOB No.: 000
NAME  to be stored: SET1               JOB No.: 000



                             FLAG BIT :       20.0

                             SET / RESET :       S




    F 1        F 2        F 3        F 4        F 5        F 6        F 7        F 8

                                   DELETE             STORE      HELP       EXIT
```

You can make entries in the NAME TO BE STORED and SET/RESET fields with the <F7 HELP> function key.

For the NAME TO BE STORED or JOB NO. fields, the previously mentioned requirements apply.

In the FLAG BIT field, enter the number of the interprocessor communication flag bit to be set.

For the SET / RESET field, entering the letter "S" causes flag 20.0 to be set when you enter "SET1" in the command line.

After filling in the form, store the set bit command by pressing the <F6 STORE> function key. Now input the name of the associated reset command in the NAME TO BE STORED field, and change the inputs as shown in the following form:

```
-> USER DATA -> FUNCTION ->                           COROS / COM GRAPH
S / R   B I T   C O M M A N D                           DEVICE: CP526
-------------------------------------------------------------------------
                                        DRIVE: B   PROGRAM: EXAMPLE

              NAME: SET1              JOB No.: 000
NAME  to be stored: RESET1           JOB No.: 000




                       FLAG BIT :         20.0

                       SET / RESET :          R




    F 1        F 2        F 3        F 4        F 5        F 6      F 7      F 8

                                  DELETE              STORE    HELP    EXIT
```

After storing the reset command, generate the set bit and reset bit commands for flag 20.1 according to the same process. Enter for these the names "SET2" or "RESET2" in the NAME TO BE STORED fields.

Store the command and return to the FUNCTION form by pressing the <F8 EXIT> function key. The name SET1 remains in the FUNCTION NAME field. If you try to translate this command back and press any function key other than the one associated with its type, the error message ERROR 524: `Command is of type:` appears. In addition, the actual command type is shown (e.g., a set bit command).

Only when you press the <F4 S/R COMMAND> function key does the program branch to the S/R BIT COMMAND form.

5.4.6.6   Display Output Commands

Display Output Command START

In the FUNCTION NAME field in the FUNCTION form, enter the command name "START".
Then press the < F1 CMD DISP OUTPUT> function key to branch to the DISPLAY OUTPUT
COMMAND form.

A display output command helps you determine which static displays, dynamic displays,
or pixel graphics displays should make up a process display. Entering the display output
command on the process keyboard brings up the associated process display on the CP
screen (with the exception of the display output command START).

A function key menu is output for each process display, in the bottom lines on the CP
monitor. You can label each function key with a single line of text.

You may only use the command name START to designate one display output command.
This command always exists on the CP 526 and is automatically carried out when you start
up the CP 526 i.e. the display behind the START function is shown. The standard command
START corresponds to a blank display. This means that you do not have to enter this
command name on the process keyboard for the display to be brought up on the CP
screen.

You must always designate a static display with a display output command. You can
designate a dynamic or pixel graphics display, but this is not necessary.

If a display consists of only one pixel graphics and/or dynamic display, as in this case you
must still designate a blank static display (i.e., BACK0).

In addition, you can redefine the message and command lines and assign functions to the
function keys on the process keyboard.

Fill in the form as shown:

```
┌──────────────────────────────────────────────────────────────────────────┐
│                                                                            │
│  -> USER DATA -> FUNCTION ->                          COROS / COM GRAPH    │
│  D I S P L A Y   O U T P U T   C O M M A N D             DEVICE: CP526     │
│  --------------------------------------------------------------------------│
│                                         DRIVE: B   PROGRAM: EXAMPLE        │
│                                                                            │
│                  NAME: START          JOB No.: 000                         │
│  NAME   to be stored: START           JOB No.: 000                         │
│                                                                            │
│      NAME STAT.DISP: BACKO            NAME DYN.DISP:                        │
│      NAME OF PIXELGRAPHICS DISPLAY: PXG1                                    │
│                                                                            │
│      Only when re-specifying command and message line :                    │
│      MESSAGE LINE:                    COMMAND LINE:                         │
│                                                                            │
│                                                                            │
│      MESSAGE LINE:                                                          │
│      BACKGROUND COLOR:        FOREGROUND COLOR:        FLASHING:            │
│      COMMAND LINE:                                                          │
│      BACKGROUND COLOR:        FOREGROUND COLOR:        FLASHING:            │
│                                                                            │
│                                                                            │
│                                                                            │
│    F 1       F 2       F 3       F 4       F 5       F 6      F 7      F 8  │
│  DEFINE                                                                     │
│  FCT.KEY                        DELETE              STORE    HELP    EXIT   │
│                                                                            │
└──────────────────────────────────────────────────────────────────────────┘
```

In the NAME STAT.DISP, NAME DYN.DISP, and NAME OF PIXEL GRAPHICS DISPLAY fields, press the <F7 HELP> function key to enter the names of the elements that will make up the process display (if they are already present on the data medium). If you try to store element names that are not yet present under the designated program name on the data medium, COM displays the message `Store, although stat., dyn., pixel graph. disp. does not exist?`

If you need to redefine command and message lines, you must fill in all three fields for that line.
You can make entries in the FOREGROUND COLOR, BACKGROUND COLOR, and FLASHING fields with the <F7 HELP> function key.

If you need to assign command names to the function keys on the process operation keyboard, or assign cursor jumps to input fields, you must do this in the FCT.KEY COMMAND form (to get to this form, press the <F1 DEFINE FCT.KEY> function key).

You can define the <F1> to <F8> function keys in this form. To define the <F9> to <F16> function keys , press the <F1 PAGE FORWARDS> function key in the FCT. KEY COMMAND form to branch to the next form.

If you need more than 16 function keys, assign a display output command, which does not output any display but just defines function keys, to one of the function keys from the function key menu.

It is also possible to generate a function key command.

```
-> USER DATA -> FUNCTION -> (DISP CMD.->)             COROS / COM GRAPH
F C T.  K E Y   C O M M A N D                           DEVICE: CP526
---------------------------------------------------------------------------
                                               DRIVE: B   PROGRAM: EXAMPLE

              NAME: START              JOB No.: 000
NAME  to be stored: START              JOB No.: 000

FCT       COMMAND         TEXT      B-COLOR    F-COLOR    LINE    COLUMN     ACK
01:     DISP1           F1          WHITE      BLACK                          N
02:                                 WHITE      BLACK
03:                                 WHITE      BLACK
04:                                 WHITE      BLACK
05:                                 WHITE      BLACK
06:                                 WHITE      BLACK
07:                                 WHITE      BLACK
08:                                 WHITE      BLACK

Delete the function key assignment: Y



   F 1         F 2        F 3        F 4       F 5       F 6        F 7        F 8
   PAGE
   FORWARDS                        DELETE              STORE      HELP       EXIT
```

In this example, you assign names to the function keys. You do not fill in the LINE and COLUMN fields.

In the COMMAND field, you enter the name of the command to be carried out when you press the function key on the process keyboard. The function name may not be the same as a PC job order.

If this command already exists under the specified program name, then you can enter it by pressing the <F7 HELP> function key.

In the TEXT field, you enter the text that labels the function key in the process display. If there is no entry, then the command name is automatically used for labeling (however, only the first eight characters are used this way).

The B-COLOR and F-COLOR fields specify the background and foreground colors, respectively, of the function keys. If you want to output the function keys and their labels in a color different from the one specified in the CP parameters, then enter the new colors in these fields (i.e., by entering the text or by pressing the <F7 HELP> function key). These fields are preassigned with the colors that you specified for the function keys in the CP 526 PARAM. ASSIGNMENT form.

When you enter a "Y" in the ACK (acknowledgment) field (which may be done by pressing the <F7 HELP> key), then after pressing the function key on the process keyboard, you must also press the <RETURN> key.

If you enter "Y" in the DELETE THE FUNCTION KEY ASSIGNMENT field, the existing key assignments are deleted before new function key assignments are made. The existing function key assignments are deleted from the CP screen. If you enter "N", the existing function key assignments are only deleted with function key commands, but not with display output commands.

In the program EXAMPLE, you only need to program the <F1> function key (it is not necessary to branch to the next form to assign the <F9> to <F16> function keys). However, the label for the <F1> function key should have two lines.

To assign the second line, press the <F1 PAGE FORWARDS> function key in this form to get to the form for defining the <F9> to <F16> function keys.

```
-> USER DATA -> FUNCTION -> (DISP CMD.->)              COROS / COM GRAPH
F C T.   K E Y   C O M M A N D                         DEVICE: CP526
-----------------------------------------------------------------------
                                              DRIVE: B   PROGRAM: EXAMPLE

                NAME: START          JOB No.: 000
NAME  to be stored: START          JOB No.: 000

FCT      COMMAND         TEXT      B-COLOR    F-COLOR   LINE   COLUMN    ACK
09:      DUMMY           TANK      WHITE      BLACK                        N
10:                                WHITE      BLACK
11:                                WHITE      BLACK
12:                                WHITE      BLACK
13:                                WHITE      BLACK
14:                                WHITE      BLACK
15:                                WHITE      BLACK
16:                                WHITE      BLACK

Delete the function key assignment: Y



   F 1        F 2        F 3        F 4        F 5       F 6       F 7       F 8
 PAGE
 BACKWARDS                         DELETE             STORE     HELP      EXIT
```

In the COMMAND field, you enter a command name that does not yet exist on the data medium.

In the TEXT field, you enter the lower label line of the function key.

After completely filling in the form, press the <F6 STORE> function key. COM then checks the inputs in both of the function key definition forms.

At the same time, the program returns to the DISPLAY OUTPUT COMMAND form. Now store the display output command by pressing the <F6 STORE> function key. COM requests an acknowledgment to the prompt **STORE, ALTHOUGH CMD. ENTRY OR DIRECT VALUE ENTRY?**, since the specified command DUMMY does not yet exist under this program name. If you respond by pressing <F1 YES>, the display output command is stored.

If you initiate the function DUMMY, that does not yet exist on the CP 526, by pressing the
<F9> function key during operation, then the CP error message `Command does not`
`exist` appears in the command line.

Display Output Command DISP1

Now generate the display output command DISP1 according to the steps used to define
the display output command START.

The process display DISP1 should consist of the static background display BACK1, the
dynamic display DYN1, and the pixel graphics display PXG2.

```
-> USER DATA -> FUNCTION ->                              COROS / COM GRAPH
D I S P L A Y   O U T P U T   C O M M A N D                 DEVICE: CP526
------------------------------------------------------------------------
                                          DRIVE: B   PROGRAM: EXAMPLE

              NAME: DISP1              JOB No.: 000
NAME  to be stored: DISP1             JOB No.: 000

      NAME STAT.DISP: BACK1           NAME DYN.DISP: DYN1
      NAME OF PIXELGRAPHICS DISPLAY: PXG2

      Only when re-specifying command and message line :
      MESSAGE LINE:                   COMMAND LINE:

      MESSAGE LINE:
      BACKGROUND COLOR:        FOREGROUND COLOR:           FLASHING:
      COMMAND LINE:
      BACKGROUND COLOR:        FOREGROUND COLOR:           FLASHING:



  F 1       F 2       F 3       F 4       F 5       F 6     F 7     F 8
DEFINE
FCT.KEY                         DELETE              STORE   HELP    EXIT
```

The following form shows the assignments for the <F1> to <F8> function keys:

```
-> USER DATA -> FUNCTION -> (DISP CMD.->)              COROS / COM GRAPH
F C T.   K E Y   C O M M A N D                         DEVICE: CP526
-------------------------------------------------------------------------
                                             DRIVE: B  PROGRAM: EXAMPLE

             NAME: DISP1              JOB No.: 000
NAME  to be stored: DISP1            JOB No.: 000

FCT    COMMAND        TEXT     B-COLOR   F-COLOR   LINE  COLUMN      ACK
01:    RESET1         F1       WHITE     BLACK                        N
02:    SET1           F2       WHITE     BLACK                        N
03:    RESET2         F3       WHITE     BLACK                        N
04:    SET2           F4       WHITE     BLACK                        N
05:    VALUE=         F5       WHITE     BLACK                        Y
06:    TIME=          F6       WHITE     BLACK                        Y
07:    MESSAGE DISP   F7       WHITE     BLACK                        N
08:    START          F8       WHITE     BLACK                        N

Delete the function key assignment: Y



  F 1         F 2       F 3       F 4       F 5       F 6       F 7       F 8
  PAGE
  FORWARDS                        DELETE              STORE     HELP      EXIT
```

NOTE:
Take special care when assigning an input command to a function key.

When you enter "N" in the ACK field and then press the <F7> function key, the command is initiated immediately, without any pause to allow you to enter the value in the command line.

When you enter "Y" in the ACK field, the CP cursor appears to the right of the " = " symbol in the command line. You can input the desired value on the process keyboard and then press the <RETURN> key.

To initiate the value input command without an acknowledgment, you can enter the desired value directly after the " = " symbol in the COMMAND NAME field.

For example:

05:    "VALUE = 100   F7 WHITE BLACK Y"

You should also assign a two-line function key label in this process display.

Select the screen for assigning the <F9> to <F16> function keys by pressing the
<F1 PAGE FORWARDS> function key in this form.

```
-> USER DATA -> FUNCTION -> (DISP CMD.->)              COROS / COM GRAPH
F C T.  K E Y   C O M M A N D                           DEVICE: CP526
---------------------------------------------------------------------------
                                          DRIVE: B   PROGRAM: EXAMPLE

               NAME: START              JOB No.: 000
NAME    to be stored: START             JOB No.: 000

FCT     COMMAND        TEXT        B-COLOR    F-COLOR    LINE  COLUMN    ACK
09:     DUMMY          VENT1 C     WHITE      BLACK                       N
10:     DUMMY          VENT1 O     WHITE      BLACK                       N
11:     DUMMY          VENT2 C     WHITE      BLACK                       N
12:     DUMMY          VENT2 O     WHITE      BLACK                       N
13:     DUMMY          VALU INP    WHITE      BLACK                       N
14:     DUMMY          TIME INP    WHITE      BLACK                       N
15:     DUMMY          MESSAGE     WHITE      BLACK                       N
16:     DUMMY          EXIT                                               N

Delete the function key assignment: Y


    F 1        F 2        F 3        F 4        F 5        F 6        F 7        F 8
    PAGE
    FORWARDS                        DELETE                STORE      HELP       EXIT
```

It is also possible to assign cursor jumps to function keys.

Then, when you press the assigned function key on the process keyboard, the CP cursor
jumps to an input field in the dynamic display.

The LINE and COLUMN fields mark the position of the dynamic VALUE INPUT field on the CP screen.

The COMMAND field can remain blank.

With the ACK field you can control whether an entry made in the field is completed immediately after it is input (by entering "N"), or whether additional inputs can be made (by entering "Y").

5.4.6.7   Message Display Command

To define a message display command, you must enter the name "MESSAGE DISP" in the FUNCTION NAME field in the FUNCTION form. The name of the message display command must follow the general name conventions for command names. Then branch to the MESSAGES form by pressing the <F3 MESSAGES> function key in the second part menu of the FUNCTION form (for a definition of the term "messages", see Planning Guide, Appendix B).

The MESSAGES form has the following preassignments:

```
-> USER DATA -> FUNCTION ->                              COROS / COM GRAPH
M E S S A G E S                                             DEVICE: CP526
------------------------------------------------------------------------
                                          DRIVE: B   PROGRAM: EXAMPLE


Name of the message display command: MESSAGE DISP        Job number: 000

The command name should only be specified when defining a message display.
When programmming the message text, the message number can be entered in
the following screen form.









    F 1       .F 2       F 3       F 4       F 5       F 6       F 7       F 8
   PROGR.               PROGR.
   MSG.DISP             MESSAGE                                             EXIT
```

Press the < F1 PROGR. MSG. DISP > function key in this form to branch to the MESSAGE DISPLAY form.

The command MESSAGE DISPLAY displays all messages that have arrived from the controller. You can initiate this function by entering the command name on the process keyboard, by pressing the function key that is assigned this command name, or by PC job order directly from the controller (e.g., for important messages).

```
-> FUNCTION -> MESSAGES ->                              COROS / COM GRAPH
M E S S A G E    D I S P L A Y                          DEVICE: CP526
--------------------------------------------------------------------------
                                              DRIVE: B   PROGRAM: EXAMPLE
             NAME: MESSAGE DISP       JOB No.: 000
NAME  to be stored: MESSAGE DISP      JOB No.: 000
DATE   : M D Y    TIME   : H M S

GROUP                    TITLE of message display      B-COLOR F-COLOR  F
all    TITLE FOR GROUP 1                               BLACK   WHITE    N
1                                                      WHITE   RED      N
2                                                      BLACK   WHITE    N
3                                                      BLACK   WHITE    N
4                                                      BLACK   WHITE    N


ATTRIBUTES of the area of a message marked with * :
BACKGROUND   FOREGROUND  FLASHING  Acknowledgement status
   MAGENTA      WHITE        Y      Message arrived, but not acknowledged
   BLACK        WHITE        Y      Message arrived, gone and not acknowledged
   MAGENTA      WHITE        N      Message arrived, and acknowledged
   GREEN        WHITE        Y      Message arrived, acknowledged and gone


   F 1        F 2        F 3        F 4       F 5       F 6       F 7       F 8
   DEFINE                          DELETE              STORE
   FCT. KEY                        MSG.CMD             MSG.CMD   HELP      EXIT
```

The fields before the TITLE field are preassigned with the date and time formats; you can change these entries by pressing the <F7 HELP> function key. The order of date and time in the message heading is already determined. However, you can rearrange the M, D, and Y fields for the date, as well as the H, M, and S fields for the time.

In the TITLE field, depending on the message group, you can specify the text of the title for the message display.

In the B-COLOR, F-COLOR, and F fields, you can enter the attributes for each heading separately.

With the BACKGROUND, FOREGROUND, and FLASHING fields, you can assign one attribute for the entire message display, for each of the four acknowledgment conditions. The attribute changes with the individual messages according to the acknowledgment condition, at those locations that are marked above the text with random characters (no blank characters), in the PROGRAM MESSAGE form.

By pressing the <F1 DEFINE FCT. KEY> function key in the MESSAGE DISPLAY form, you branch to the DEFINE FUNCTION KEYS form for messages.

```
-> FUNCTIONS -> MESSAGE DISPLAY ->                        COROS / COM GRAPH
D E F I N E   F U N C T I O N   K E Y S                      DEVICE: CP526
-------------------------------------------------------------------------
                                                 DRIVE: B    PROGRAM: EXAMPLE
  No.      TEXT        B-COLOR      F-COLOR                 MEANING^
   1    ↑Cursor↑       WHITE        BLACK      Cursor up
   2       ↑           WHITE        BLACK      Scroll messages backward
   3     ↑ ↑ ↑         WHITE        BLACK      Page messages backward
   4      ↖            WHITE        BLACK      Display starting with first message
   5      Q+++         WHITE        BLACK      Acknowledge all displayed messages
   6    GROUP 1        WHITE        BLACK      Display messages from group 1
   7    GROUP 3        WHITE        BLACK      Display messages from group 3
   8    Gr 1....4      WHITE        BLACK      Display messages from all groups
   9    ↓Cursor↓       WHITE        BLACK      Cursor down
  10       ↓           WHITE        BLACK      Scroll messages forward
  11     ↓ ↓ ↓         WHITE        BLACK      Page messages forward
  12      ↙            WHITE        BLACK      Display last message
  13      Q+           WHITE        BLACK      Acknowledge message at cursor
  14    GROUP 2        WHITE        BLACK      Display messages from group 2
  15    GROUP 4        WHITE        BLACK      Display messages from group 4
  16      END          WHITE        BLACK      EXIT

    F 1        F 2        F 3        F 4        F 5        F 6        F 7        F 8

                                                          SAVE       HELP       EXIT
```

In the TEXT field, you can either input a text for labeling the function keys, or you can use the standard texts. You can assign the standard texts to any function key. Make sure that the meaning of the standard text and the entry in the MEANING field agree.

If you leave the field blank, then the standard label is assigned. The message display with the function key labels is displayed when you enter the message display command on the process keyboard, or when it is output by the controller via a send direct job order (i.e., for dangerous plant conditions).

You can enter the following texts by pressing the < F7 HELP > function key:

↑ CURSOR ↑	↓ CURSOR ↓
↑	↓
↑ ↑ ↑	↓ ↓ ↓
↖	↙
Q + + +	Q +
GROUP 1	GROUP 2
GROUP 3	GROUP 4
GR 1...4	END

In the B-COLOR and F-COLOR fields, you can specify the colors in which the message-display function keys should be represented (i.e., input by entering the text, or pressing the < F7 HELP > function key).

The following choices are available with the <F7 HELP> function key in the MEANING field :

- "CURSOR UP"
- "SCROLL MESSAGES BACKWARD"
- "PAGE MESSAGES BACKWARD"
- "DISPLAY STARTING WITH FIRST MESSAGE"
- "ACKNOWLEDGE ALL DISPLAYED MESSAGES"
- "DISPLAY MESSAGES FROM GROUP 1"
- "DISPLAY MESSAGES FROM GROUP 3"
- "DISPLAY MESSAGES FROM ALL GROUPS"
- "CURSOR DOWN"
- "SCROLL MESSAGES FORWARD"
- "PAGE MESSAGES FORWARD"
- "DISPLAY LAST MESSAGE"
- "ACKNOWLEDGE MESSAGE AT CURSOR POSITION"
- "DISPLAY MESSAGES FROM GROUP 2"
- "DISPLAY MESSAGES FROM GROUP 4"
- "EXIT"

NOTE:
When you press the <F7 HELP> function key with the cursor in the MEANING field, the associated standard text is displayed in the TEXT field.

When you press the <F6 SAVE> function key, the program checks the inputs and returns to the MESSAGE DISPLAY form.

When you press the <F6 STORE MSG. CMD> function key, the message display command is stored. Then press the <F8 EXIT> function key to return to the FUNCTION form.

To define messages, you do not need to input a command name in the FUNCTION form. Press the <F3 MESSAGES> function key in the second part menu of this form and then press the <F3 PROGR. MESSAGE> function key in the MESSAGES form to branch to the PROGRAM MESSAGES form.

You can assign message texts to process conditions or changes of such conditions (see Planning Guide COM GRAPH, Chapter 4).

In this form, you can generate message texts for each of the four message groups, whereby the number of messages is limited to 1000, the number, however, may be in the range from 1 through 1999.

```
-> FUNCTION -> MESSAGES ->                              COROS / COM GRAPH
P R O G R A M   M E S S A G E                           DEVICE: CP526
-------------------------------------------------------------------------
                                          DRIVE: B    PROGRAM: EXAMPLE


MESSAGE NUMBER:    M 0001                  Job number and DB number
GROUP NUMBER:      1                       must be specified in the
BACKGROUND COLOR: RED                      CP PARAMETER ASSIGNMENT
FOREGROUND COLOR: WHITE                    mask.
FLASHING:          N


     Mark ATTRIBUTE CHANGE with a character (except space) when changing
     the acknowledgement status.
     777777
TEXT: TEXT OF THE FIRST MESSAGE & THMS

     To insert date and time in a message text use &DMDY; and &THMS;.
     Not all parts need be specified.




     F 1        F 2        F 3        F 4        F 5        F 6      F 7       F 8
PREVIOUS    NEXT       FETCH      DELETE     BASIC      STORE
MESSAGE     MESSAGE    MESSAGE    MESSAGE    STATUS     MESSAGE    HELP      EXIT
```

With the MESSAGE NUMBER field, message numbers between 1 and 2000 are possible.

In the GROUP NUMBER field, there are four different message groups.

The BACKGROUND COLOR, FOREGROUND COLOR, and FLASHING fields determine how the message looks in the message display. If the acknowledgment status changes, then the parts of the message text marked with random characters in the line above the TEXT field are displayed with the attributes specified in the MESSAGE DISPLAY form.

The field above the TEXT field marks where an attribute change in the message text should occur if the acknowledgment status changes. You can make this marking with any character, with the exception of a blank.

When the message text is decompiled, these characters are represented as a series of "*" symbols.

With the TEXT field, you can incorporate the date and time areas into the message text in any form and at any location. If you use handling blocks for message processing, i.e. you do not use the software package "Signaling functions for standard CP" (see also subsection 3.3.6.3 "Message exchange between programmable controller and CP 526"), date and time can be read from the CP 526 using the programmable controller RECEIVE job 218.

Message texts are output in the message display from column 3 to column 71, inclusive.

The function keys have the following meanings:

●   <F1 PREVIOUS MESSAGE> - The message number entered is decremented by 1.

●   <F2 NEXT MESSAGE> - The message number entered is incremented
    by 1.

●   <F3 FETCH MESSAGE> - The message with the number entered is decompiled and
    the data is transferred to the form.

●   <F4 DELETE MESSAGE> - The message with the number entered is deleted after a
    positive acknowledgment.

●   <F5 BASIC STATUS> - The form returns to the basic status (i.e., it is overwritten with
    the standard defaults). All fields are deleted except the BACKGROUND - BLACK,
    FOREGROUND - WHITE, and FLASHING - N fields.

●   <F6 STORE MESSAGE> - The form is checked for correctness and the message is
    stored under the specified number. Form data is not lost. If several messages are
    similar to each other (e.g., "MOTOR 1 ON", "MOTOR 2 ON"), you do not need to
    reenter all the data each time.

If messages come up, the program checks whether the corresponding message text exists. If the message text is missing, the message is passed over. If the text does exist, then it is stored in the order of processing in a buffer, along with date and time.

When the buffer is full, the last message written to the buffer is overwritten (overflow message). The number of overflow messages is communicated to the CP and displayed flashing in the last line of the message display, starting at column 72. The control program printed for this example outputs messages M0181 to M0196. However, since these can only be displayed when they exist in the CP program, you should also enter the text for the remaining messages.

The messages, or acknowledgments, are exchanged with SEND DIRECT and RECEIVE ALL job orders. You must specify the job order number and data block number in the CP/OP PARAM. ASSIGNMENT form.

You can find further information on the exchange of acknowledgments and messages between the CP 526 and the controller in the CP description in Chapter 3.

5.5   PLC Program for Application Example for the S5-135U with CPU 928

For the application example to run successfully, there must be a program running in the controller that is coordinated with the CP program.

This program must perform the following functions:

- Control data communication between the CP 526 and controller

  For communication to occur between the controller and CP 526, you must synchronize each with the other for every cold restart (OB20) and every warm restart (OB21, OB22). Without synchronization, no data exchange is possible.

  In this program, synchronization of the controller and CP occurs in program blocks PB6 and PB7.

  You must set interface number 4 on the CP 526. The actual data exchange between the controller and CP 526 occurs via handling blocks FB126:SEND-A and FB127:REC-A. These are called in the cyclic part of the program (OB1) by program block PB1.

- Tank simulation

  A process with two tanks is represented in the TANK display. As the actual process does not exist, the control program must simulate the following functions:

  a) empty the tanks when the drainage valves are open

  The liquid level in the tanks is entered in value input fields or by a value input command, and stored by the CP 526 in data word 0 or data word 1 of data block 20. This is where the value output fields and the bar windows fetch their output values to display the liquid level in the tanks.

When you press the <F2 VALVE1 O> function key, interprocessor communication flag bit F 20.0 is set on the CP by the set bit command SET1. It is reset when you press the <F1 VALVE1 C> function key.

FB20:FILL 1 decreases the liquid level in tank 1 (DB20, DW0) when interprocessor communication flag bit F 20.0 sets. Function block FB21:FILL 2 does the same for tank 2. It decreases the liquid level in tank 2 (DB20, DW1) when drainage valve 2 is opened (i.e., when interprocessor communication flag F 20.1 has a value of 1).

For interprocessor communication flag bits F 20.0 and F 20.1 to be evaluated in the control program, you must enter flag byte FB20 of data block DB1 in the PERIPHERAL ALLOCATION form as the input interprocessor communication flag. This data block is set up with a cold restart (OB20).

b) generate the temperature curve for the curve window

The temperature curve of both tanks is represented in the pixel graphics curve window. Function blocks FB22:TEMP 1 and FB23:TEMP 2 generate these temperature curves. The temperature value for tank 1 is stored in flag word FW21, and the value for tank 2 is stored in flag word FW23.

c) organize old and new values for the CP 526

The temperature curves are displayed with old values. This means that when you call the display TANK, the CP 526 fetches the values of the last two minutes from the controller and displays them in the curve window.

Then the values are updated normally. The CP fetches old values as entered in the CURVE WINDOW form from data block DB21, starting with data word DW0. The structure of this data block is described in the operating instructions for the pixel graphics expansion board. The old and new temperature values are organized by FB10:CURVE and made available to the CP 526 in DB21.

This data block is set up with a cold restart (OB20) and preassigned zeros at that time. After a cold restart, it takes two minutes until the data block is filled with old values.

If the display TANK is called during this time, then impermissible old values (zeros) are still in the data block, and the CP 526 outputs the error message `Impermissible curve output value in the PC`.

d) trigger fault messages

For fault messages to be output in the message line and in the message display, they must be triggered by the control program. This function is performed by FB24:TRIGGER. It triggers messages M0181 through M0196 cyclically. These messages constantly change their attributes (message arrived, message gone) and can be acknowledged in the message display. However, only those messages are displayed that are programmed on the CP.

● Call the 'Signaling functions for standard CP'

To accommodate message processing in the controller, the software package 'Signaling functions for standard CP' is available (see Ordering Data).

For a cold restart of the signaling concept, program block PB6 is called in OB20. For a warm restart, PB8 is called in OB21 and OB22.

In OB1, the signaling concept is called by PB3. In OB13, a pulse is generated for a signaling concept timer location. A completely parameterized example for the CP 526 is included with the product. The control program is coordinated with this example.

You must set interface number 4 on the CP 526 module. Message triggering occurs when you set (message active) and erase (message passive) flag bits. In this program, messages M0001 to M0016 are triggered by cyclic setting and erasing of flag bits 0 to 15 in flag word FW25 by FB:TRIGGER. You can find details on parameterizing the "Standard Signaling Concept" in the user description that accompanies this package.

You must load the preparameterized example included with the product, and the blocks listed on the following pages, into the controller.

The program listed on the following pages also runs without the signaling concept. You must make the changes described in the listing to do this. Then no messages will be displayed by the CP.

However the following standard handling blocks are necessary:

> FB120:SEND, FB121:RECEIVE, FB122:FETCH, FB123:CONTROL,
> FB124:RESET, FB125:SYNCHRON, FB126:SEND-A and FB127:RECEIVE-A.

You must load these, and the blocks listed on the following pages, into the controller.

5.5.1   STEP 5 Program with Handling Block Parameterization for the S5-135U with CPU 928

OB1   LEN = 15   ABS
                                                                  PAGE 1

NETWORK 1

IF THE STANDARD SIGNALING CONCEPT IS NOT USED, THE PB 3 AND THE FB 130 CALL
MUST BE DELETED.

```
0000          :
0001          :JU     PB1              CALL OF HANDLING BLOCKS
0002          :                        SEND-ALL AND RECEIVE-ALL
0003          :
0004          :JU     PB2              SIMULATION OF TANK PROCESS
0005          :
0006          :C      DB6
0007          :JU     FB130            CALL OF STANDARD SIGNALING
0008 NAME :ANLAUF                      CONCEPT
0009          :
0010          :JU     PB3
0011          :
0012          :BE
```

OB13   LEN = 11   ABS
                                                                  PAGE 1

NETWORK 1

IF THE STANDARD SIGNALING CONCEPT IS NOT USED, THIS BLOCK SHOULD NOT BE
INPUT.

```
0000          :C      DB6              SECONDS PULSE FOR TIMER
0001          :AN     D 8.8            LOCATION OF STANDARD
0003          : =     D 8.8            SIGNALING CONCEPT
0005          :BE
```

OB20   LEN = 27   ABS
                                                  PAGE 1

NETWORK 1

FIRST GENERATE DATA BLOCK DB 20 FOR LIQUID LEVELS AND TRANSFER DATA BLOCK DB
21 FOR CURVE VALUES.
THE SIGNALING CONCEPT IS PARAMETERIZED IN DATA BLOCKS DB 6:SRK, DB 14:PARA-R,
DB 15:PARA-S AND DB 208:RANGE PARAMETERIZED. THESE DATA BLOCKS HAVE PRE-
ASSIGNED PARAMETERS AND ARE INCLUDED ON THE DISK PROVIDED. THIS EXAMPLE IS
COORDINATED WITH THIS PARAMETERIZATION, SO THAT THE SIGNALING PACKAGE
DOES NOT NEED TO BE PARAMETERIZED AT STARTUP. ONLY THE FLAG WORD TO
TRIGGER THE MESSAGES IS REPARAMETERIZED TO FW 25.
AFTER THIS REPARAMETERIZATION, PROGRAM BLOCK PB6 IS CALLED, FOR A COLD
RESTART OF THE SIGNALING CONCEPT.
IN THIS PROGRAM BLOCK, PC AND CP 526 ARE SYNCHRONIZED BY HANDLING BLOCK
FB:125 SYNCHRON.
INTERFACE NUMBER 4 MUST BE SET ON THE CP 526 MODULE.
IF THE STANDARD SIGNALING CONCEPT IS NOT USED, THEN FB:SYNCHRON ONLY NEEDS
TO BE CALLED TO SYNCHRONIZE CP 526 AND PC.


```
0000          :JU    FB2
0001   NAME:ANLAUF2
0002          :
0003          :
0004          :
0005          :                            REPARAMETERIZING OF THE
0006          :                            SIGNALING PACKAGE:
0007          :
0008          :C     DB208                 CALL DB:RANGE
0009          :
000A          :L     KY25,1                MESSAGE TRIGGER BY FLAG WORD
000B          :T     DW1                   FW 25
000C          :
```

```
000D        :JU   PB6              SYNCHRONIZATION
000E        :                      CP 526 <-> PC AND COLD
000F        :                      RESTART OF STANDARD
0010        :BE                    SIGNALING CONCEPT
```


OB21   LEN = 10   ABS
                                            PAGE 1

NETWORK 1

IF THE STANDARD SIGNALING CONCEPT IS NOT USED, THEN FB 125:SYNCHRON MUST BE
CALLED HERE INSTEAD OF PB 8 TO SYNCHRONIZE THE CP 526 AND THE PC.

```
0000        :
0001        :JU   PB8              SYNCHRONIZATION CP 526 <-> PC
0002        :                      AND WARM RESTART OF THE
0003        :                      STANDARD SIGNALING CONCEPT
0004        :BE
```


OB22   LEN = 10   ABS
                                            PAGE 1

NETWORK 1

IF THE STANDARD SIGNALING CONCEPT IS NOT USED, THEN FB 125:SYNCHRON MUST BE
CALLED HERE INSTEAD OF PB 8 TO SYNCHRONIZE THE CP 526 AND THE PC.

```
0000        :
0001        :JU   PB8              SYNCHRONIZATION CP 526 <-> PC
0002        :                      AND WARM RESTART OF THE
0003        :                      STANDARD SIGNALING CONCEPT
0004        :BE
```

PB1

NETWORK 1

   CALLING THE HANDLING BLOCKS:

FOR DATA EXCHANGE TO TAKE PLACE BETWEEN THE CP 526 AND THE PROGRAMMABLE
CONTROLLER, HANDLING BLOCKS FB 126:SEND-ALL AND FB 127:RECEIVE-ALL MUST BE
CALLED IN THE CYCLIC PART OF THE PROGRAM.
THE INTERFACE NUMBER 4 ENTERED HERE UNDER THE PARAMETER SSNR MUST ALSO BE
SET ON THE CP 526 MODULE!

```
0000          :
0001          :JU  FB126              CALL SEND-ALL:
0002 NAME  :SEND-A
0003 SSNR  :    KY0,4                 INTERFACE NUMBER = 4
0004 A-NR  :    KY0,0                 JOB ORDER NUMBER ZERO = ALL
0005 ANZW  :    FW4                   DISPLAY WORD
0006 PAFE  :    FB2                   PARAMETERIZATION ERROR BYTE
0007          :
0008          :JU  FB127              CALL RECEIVE-ALL:
0009 NAME  :REC-A
000A SSNR  :    KY0,4                 INTERFACE NUMBER = 4
000B A-NR  :    KY0,0                 JOB ORDER NUMBER ZERO = ALL
000C ANZW  :    FW8                   DISPLAY WORD
000D PAFE  :    FB3                   PARAMETERIZATION ERROR BYTE
000E          :
000F          :BE
```

PB2

NETWORK 1

TANK PROCESS SIMULATION AND OLD/NEW VALUE ORGANIZATION FOR THE CURVE
WINDOW:

THE DISPLAY OF THE PROCESS USING TANKS CONSISTS OF TWO HEATED TANKS. THE
TANKS ARE EMPTIED BY MEANS OF DRAINAGE VALVES AND FILLED BY MEANS OF
ENTRIES MADE IN VALUE INPUT FIELDS OR A VALUE INPUT COMMAND.
THE TANK TEMPERATURE TRENDS ARE REPRESENTED WITH OUTPUT OF OLD VALUES IN A
CURVE WINDOW.

AS THE ACTUAL PROCESS DOES NOT EXIST, THE PC PROGRAM MUST SIMULATE THE
TEMPERATURE TRENDS, THE EMPTYING OF THE TANKS, AND THE TRIGGERING OF FAULT
MESSAGES.
FOR THE CURVE WINDOW, THE PC MUST ORGANIZE OLD AND NEW VALUES.
THE INDIVIDUAL FUNCTIONS ARE DISTRIBUTED AMONG THE FOLLOWING FUNCTION
BLOCKS:

FB 20:FILL 1   EMPTY TANK 1, WHEN DRAINAGE VALVE 1 IS OPENED.

FB 21:FILL 2   EMPTY TANK 2, WHEN DRAINAGE VALVE 2 IS OPENED.

FB 22:TEMP 1   GENERATE TEMPERATURE CURVE FOR TANK 1.

FB 23:TEMP 2   GENERATE TEMPERATURE CURVE FOR TANK 2.

FB 24:TRIGGER   TRIGGER FAULT MESSAGES

FB 10:CURVE   ORGANIZATION OF OLD AND NEW VALUES FOR THE CURVE
WINDOW

TO PARAMETERIZE FB 10:CURVE:

DB   IN THE COM "CURVE WINDOW" FORM, DATA BLOCK
DB21 IS ENTERED IN THE "DB" FIELD AS THE SENDING DATA
BLOCK. -> DB21

N	CURVE NO. 1 AND CURVE NO. 2 HAVE BEEN CONFIGURED. THEY MUST BE ENABLED IN THE ENABLE WORD. -> KM00000000 00000011
AZ	THE CURVE RESOLUTION DEPENDS TO A GREAT EXTENT ON THE PARAMETERS OLD-VALUE NUMBER AZ AND F. ONE SECOND IS ENTERED AS THE UPDATE RATE UPD IN THE "CURVE WINDOW" FORM OF COM. THE TOTAL TIME SPAN TTSP OF TWO MINUTES MUST BE RECORDED IN THE OLD VALUES. THE NUMBER OF ALL OLD VALUES MAY NOT EXCEED 240. THERE ARE THEREFORE A MAXIMUM OF 120 OLD VALUES AVAILABLE PER CURVE. -> AZ < = 120 TO OBTAIN THE HIGHEST RESOLUTION POSSIBLE, F SHOULD BE SELECTED AS SMALL AS POSSIBLE (1 < = F < = 15). ACCORDING TO THE FORMULA TTSP = TA * F * AZ FOR THIS EXAMPLE YOU GET AN OPTIMUM PARAMETERIZATION OF: F = 1, AZ = 120 FURTHER POSSIBLE PARAMETERIZATIONS, BUT WITH LESS CLEAR RESOLUTION, COULD BE FOR EXAMPLE: F = 2, AZ = 60; F = 3, AZ = 40;.. HERE THE OPTIMUM PARAMETERIZATION HAS BEEN CHOSEN: - > AZ = 120, F = 1
W1	THE TEMPERATURE OF TANK 1, AS REPRESENTED BY CURVE 1, IS STORED BY FB:TEMP 1 IN FLAG WORD FW 21. -> KY0,21
W2	THE TEMPERATURE OF TANK 2, AS REPRESENTED BY CURVE 2, IS STORED BY FB:TEMP 2 IN FLAG WORD FW 23. -> KY0,23
W3 TO W6	THESE CURVES ARE NOT ENABLED. -> INPUT IRRELEVANT
TA	THE UPDATE RATE THAT WAS ENTERED IN THE COM "CURVE WINDOW" FORM MUST BE ENTERED HERE. -> KT100.0

PB2   LEN = 37   ABS
                                                 PAGE 1

F   SEE ABOVE -> KF + 1

TI   A FREE TIMER FOR UPDATING MUST BE ENTERED HERE.
                        -> T 6

Address			Comment
0000	:		
0001	:JU FB20		PROCESS LIQUID LEVEL TANK 1
0002 NAME	:FILL 1		
0003	:		
0004	:JU FB21		PROCESS LIQUID LEVEL TANK 2
0005 NAME	:FILL 2		
0006	:		
0007	:JU FB22		TEMPERATURE CURVE FOR TANK 1
0008 NAME	:TEMP 1		
0009	:		
000A	:JU FB23		TEMPERATURE CURVE FOR TANK 2
000B NAME	:TEMP 2		
000C	:		
000D	:JU FB24		TRIGGER FAULT MESSAGES
000E NAME	:TRIGGER		
000F	:		
0010	:JU FB10		ORGANIZATION OF OLD AND NEW
0011 NAME	:CURVE		VALUES
0012 DB	:	DB21	SEND DATA BLOCK
0013 N	:	KM00000000 00000011	ENABLE CURVES 1 AND 2
0014 AZ	:	KF + 120	120 OLD VALUES PER CURVE
0015 W1	:	KY0,21	TEMPERATURE TANK 1 FROM FW 21
0016 W2	:	KY0,23	TEMPERATURE TANK 2 FROM FW 23
0017 W3	:	KY0,0	CURVE NOT USED
0018 W4	:	KY0,0	CURVE NOT USED
0019 W5	:	KY0,0	CURVE NOT USED
001A W6	:	KY0,0	CURVE NOT USED
001B TA	:	KT100.0	WINDOW UPDATE RATE
001C F	:	KF + 1	EACH NEW VALUE BECOMES AN OLD VALUE
001D TI	:	T 6	TIMER FOR UPDATING
001E	:		
001F	:BE		

FB 2   LEN = 43
                                                          PAGE 1

NETWORK 1   0000

THIS FUNCTION BLOCK CHECKS THAT DATA BLOCKS DB 20 AND DB 21 EXIST (SEE DB 20).
IF A DB DOES NOT EXIST, IT IS GENERATED BY FB 2.

NAME:STARTUP2

```
0005          :L    KH DF00          SCAN TO DETERMINE IF DB 20
0007          :L    KF  + 20         ALREADY EXISTS IN PC.
0009          : + F
000A          :LIR  1
000B          :L    KH 0000
000D          :XF
000E          :JC = M001
000F          :
0010          :L    KF + 2           SET UP DB 20 WITH A LENGTH OF
0012          :E    DB 20            2 WORDS.
0014          :
0015 M001     :L    KH DF00          SCAN TO DETERMINE IF DB 21
0017          :L    KF + 21          ALREADY EXISTS IN PC.
0019          : + F
001A          :LIR  1
001B          :L    KH 0000
001D          :XF
001E          :BEC
001F          :
0020          :L    KF  + 256        SET UP DB 21 WITH A LENGTH OF
0022          :G    DB 21            256 WORDS.
0024          :
0025          :BE
```

FB20   LEN = 42   ABS
                                        PAGE 1

NETWORK 1

PROCESS LIQUID LEVEL TANK 1:

THIS FUNCTION BLOCK EMPTIES TANK 1 WHEN VALVE 1 IS OPENED. THE SET BIT
COMMAND "SET1" SETS FLAG BIT F 20.0 WHEN THE <F2 VALVE1 O> FUNCTION KEY IS
PRESSED IN THE "TANK" DISPLAY.
THE FLAG BIT IS RESET WHEN THE <F1 VALVE1 C> FUNCTION KEY IS PRESSED,
FOR THE STATUS OF THIS FLAG BIT TO BE EVALUATED IN THE PC PROGRAM, FLAG BYTE
FB 20 MUST BE DEFINED IN DATA BLOCK 1 AS AN INPUT INTERPROCESSOR
COMMUNICATION FLAG.
AS PROGRAMMED IN THE DYNAMIC DISPLAY "DYN1", THE CP 526 STORES THE INPUT
AND OUTPUT VALUE FOR THE LIQUID LEVEL OF TANK 1 IN, OR FETCHES IT FROM, DB 20,
DW 0.
WHEN F 20.0 IS SET, FB:FILL 1 DECREMENTS THIS VALUE BY 100 EVERY SECOND.
THIS CAUSES THE LIQUID LEVEL TO DECREASE BY 100 LITERS PER SECOND, WHEN THE
DRAINAGE VALVE IS OPENED.

NAME : FILL 1

0005	:AN F 20.0	VALVE 1 OPEN ?
0006	:BEC	NO -> END
0007	:	
0008	:A T 1	1 SECOND PASSED ?
0009	:BEC	NO -> END
000A	:	
000B	:L KT100.0	START TIMER
000D	:A T 1	
000E	:SI T 1	(PULSE TIMER)
000F	:AN T 1	
0010	:SI T 1	

```
0011        :
0012        :C      DB20            CALL I/O DATA BLOCK
0013        :
0014        :L      KF + 0
0016        :
0017        :L      DW0             LIQUID LEVEL TANK 1 (DB20, DW0)
0018        :ADD    KF-100          DECREMENT BY 100
001A        :
001B        :>F                     LIQUID LEVEL < 0 ?
001C        :JC = M001
001D        :
001E        :T      DW0             NO -> ENTER NEW LIQUID LEVEL
001F        :BEU
0020        :
0021  M001  :TAK                    YES -> ENTER LIQUID LEVEL ZERO
0022        :T      DW0
0023        :
0024        :BE
```

FB21   LEN = 41   ABS
                                        PAGE 1

NETWORK 1

PROCESS LIQUID LEVEL TANK 2:

THIS FUNCTION BLOCK EMPTIES TANK 2 WHEN VALVE 2 IS OPENED. SET BIT COMMAND
"SET2" SETS FLAG BIT F 20.1 WHEN THE < F4 VALVE2 O > FUNCTION KEY IS PRESSED IN
THE "TANK" DISPLAY.
THE FLAG BIT IS RESET WHEN THE < F3 VALVE2 C > FUNCTION KEY IS PRESSED,
FOR THE STATUS OF THIS FLAG BIT TO BE EVALUATED IN THE PC PROGRAM, FLAG BYTE
FB 20 MUST BE DEFINED IN DATA BLOCK 1 AS AN INPUT INTERPROCESSOR
COMMUNICATION FLAG.
AS PROGRAMMED IN THE DYNAMIC DISPLAY "DYN1", THE CP 526 STORES THE INPUT
AND OUTPUT VALUE FOR THE LIQUID LEVEL OF TANK 2 IN, OR FETCHES IT FROM, DB 20,
DW 1.
WHEN F 20.1 IS SET, FB:FILL 2 DECREMENTS THIS VALUE BY 100 EVERY SECOND.
THIS CAUSES THE LIQUID LEVEL TO DECREASE BY 100 LITERS PER SECOND, WHEN THE
DRAINAGE VALVE IS OPENED.

NAME : FILL 2

0005	:AN F 20.1	VALVE 2 OPEN ?
0006	:BEC	NO -> END (BLOCK END CONDITIONAL)
0007	:	
0008	:A T 2	1 SECOND PASSED ?
0009	:BEC	NO -> END
000A	:	
000B	:L KT100.0	START TIMER
000D	:A T 2	
000E	:SI T 2	
000F	:AN T 2	
0010	:SI T 2	
0011	:	

```
0012        :C    DB20              CALL I/O DATA BLOCK
0013        :
0014        :L    KF + 0
0016        :
0017        :L    DW1               LIQUID LEVEL TANK 2 (DB20, DW1)
0018        :ADD  KF-100            DECREMENT BY 100
001A        :
001B        :>F                     LIQUID LEVEL < 0 ?
001C        :JC = M001
001D        :
001E        :T    DW1               NO -> ENTER NEW LIQUID LEVEL
001F        :BEU
0020        :
0021  M001  :TAK                    YES -> ENTER LIQUID LEVEL ZERO
0022        :T    DW1
0023        :BE
```

FB22   LEN = 79   ABS
                                                  PAGE 1

NETWORK 1

SIMULATE TEMPERATURE TREND TANK 1:

THE TEMPERATURE TREND FOR TANK 1 CONSISTS OF FOUR LINEAR SEGMENTS. A TIMER
IS STARTED EVERY 120 SECONDS. DEPENDING ON THE TIME RANGE, THE TEMPERATURE
IS CALCULATED FROM ONE OF THE FOUR LINEAR EQUATIONS AND STORED IN FLAG
WORD FW 21.

TIME RANGE 1:   0 SEC. = < T < 31 SEC.   -> TEMP = 6 * T + 20
TIME RANGE 2:   31 SEC. = < T < 61 SEC.   -> TEMP = -3 * T + 290
TIME RANGE 3:   61 SEC. = < T < 91 SEC.   -> TEMP = 110
TIME RANGE 4:   91 SEC. = < T < 121 SEC. -> TEMP = -3 * T + 380

NAME :TEMP 1

```
0005          :A    T 3            TIMER RUN OUT ?
0006          :JC   = M001         JUMP IF NO
0007          :
0008          :L    KT120.2        START TIMER
000A          :A    T 3
000B          :SI   T 3
000C          :AN   T 3
000D          :SI   T 3
000E          :
000F  M001    :LD   T 3            TIME WITHIN TIME RANGE 1?
0010          :L    KT031.2        0 < = T < 31 ?
0012          :<F
0013          :JC   = M002         JUMP IF YES
0014          :
0015          :LD   T 3            TIME WITHIN TIME RANGE 2 ?
0016          :L    KT061.2        31 < = T < 61 ?
0018          :<F
0019          :JC   = M003         JUMP IF YES
001A          :
001B          :LD   T 3            TIME WITHIN TIME RANGE 3 ?
```

001C		:L KT091.2	61 < = T < 91 ?
001E		:<F	
001F		:JC = M004	JUMP IF YES
0020		:	
0021		:LD T 3	TIME WITHIN TIME RANGE 4 ?
0022		:L KT121.2	91 < = T < 121 ?
0024		:<F	
0025		:JC = M005	JUMP IF YES
0026		:	
0027	M002	:L T 3	TIME RANGE 1:
0028		:L KF + 6	
002A		:XF	TEMP = 6 * T + 20
002B		:L KF + 20	
002D		: + F	
002E		:T FW21	
002F		:BEU	
0030		:	
0031	M003	:L T 3	TIME RANGE 2:
0032		:L KF-3	
0034		:XF	TEMP = -3 * T + 290
0035		:L KF + 290	
0037		: + F	
0038		:T FW21	
0039		:BEU	
003A		:	
003B	M004	:L KF + 110	TIME RANGE 3:
003D		:T FW21	TEMP = 110
003E		:BEU	
003F		:	
0040	M005	:L T 3	TIME RANGE 4:
0041		:L KF-3	
0043		:XF	TEMP = -3 * T + 380
0044		:L KF + 380	
0046		: + F	
0047		:T FW21	
0048		:	
0049		:BE	

FB23   LEN = 67   ABS
                                              PAGE 1

NETWORK 1

SIMULATE TEMPERATURE TREND TANK 2:

THE TEMPERATURE TREND FOR TANK 2 CONSISTS OF THREE LINEAR SEGMENTS. A TIMER
IS STARTED EVERY 100 SECONDS. DEPENDING ON THE TIME RANGE, THE TEMPERATURE
IS CALCULATED FROM ONE OF THE THREE LINEAR EQUATIONS AND STORED IN FLAG
WORD FW 23.

TIME RANGE 1:   0 SEC. = < T < 51 SEC.   -> TEMP = 1 * T + 50
TIME RANGE 2:   51 SEC. = < T < 76 SEC.   -> TEMP = 4 * T - 100
TIME RANGE 3:   76 SEC. = < T < 101 SEC. -> TEMP = -6 * T + 650

NAME : TEMP 2

```
0005           :A    T 4                    TIMER RUN OUT ?
0006           :JC   = M001                 JUMP IF NO
0007           :
0008           :L    KT100.2                START TIMER
000A           :A    T 4
000B           :SI   T 4
000C           :AN   T 4
000D           :SI   T 4
000E           :
000F M001      :LD   T 4                    TIME WITHIN TIME RANGE 1 ?
0010           :L    KT051.2                0 < = T < 51 ?
0012           :<F
0013           :JC   = M002                 JUMP IF YES
0014           :
0015           :LD   T 4                    TIME WITHIN TIME RANGE 2 ?
0016           :L    KT076.2                51 < = T < 76 ?
0018           :<F
0019           :JC   = M003                 JUMP IF YES
001A           :
001B           :LD   T 4                    TIME WITHIN TIME RANGE 3 ?
```

© *Siemens AG C79000-B8576-C217-01*

```
001C           :L    KT101.2              76 < = T < 101 ?
001E           :<F
001F           :JC   = M004
0020  M002     :L    T 4                  TIME RANGE 1:
0021           :L    KF + 1
0023           :XF                        TEMP = 1 * T + 50
0024           :L    KF + 50
0026           : + F
0027           :T    FW23
0028           :BEU
0029           :
002A  M003     :L    T 4                  TIME RANGE 2:
002B           :L    KF + 4
002D           :XF                        TEMP = 4 * T - 100
002E           :L    KF-100
0030           : + F
0031           :T    FW23
0032           :BEU
0033           :
0034  M004     :L    T 4                  TIME RANGE 3:
0035           :L    KF-6
0037           :XF                        TEMP = -6 * T + 650
0038           :L    KF + 650
003A           : + F
003B           :T    FW23
003C           :
003D           :BE
```

FB24   LEN = 40   ABS
                                                  PAGE 1

NETWORK 1

TRIGGER MESSAGES:

IN OB 20, FLAG WORD FW 25 WAS PARAMETERIZED TO TRIGGER MESSAGES VIA THE SIGNALING PACKAGE.
BY SETTING BITS 0 THROUGH 15 IN THIS FLAG WORD, THE CP 526 ASSIGNS THE ATTRIBUTE "MESSAGE COMING" TO THE CORRESPONDING MESSAGES M0181 THROUGH M0196. BY DELETING BITS 0 THROUGH 15 IN THIS FLAG WORD, THE CP 526 ASSIGNS THE ATTRIBUTE "MESSAGE GOING" TO THE CORRESPONDING MESSAGES M0181 THROUGH M0196. FB 24:TRIGGER TRIGGERS THE OUTPUT OF MESSAGES EVERY 5 SECONDS. ONLY MESSAGES PROGRAMMED ON THE CP 526 ARE OUTPUT. THE MESSAGE THAT ARRIVED LAST IS OUTPUT IN THE MESSAGE LINE. ALL ACTIVE MESSAGES ARE OUTPUT IN THE MESSAGE DISPLAY WITH THE CORRESPONDING ATTRIBUTES AND CAN BE ACKNOWLEDGED.

NAME :TRIGGER

```
0005        :A   T 5                      5 SECONDS GONE BY ?
0006        :BEC                          NO -> END
0007        :
0008        :L   KT500.0                  YES -> START TIMER
000A        :A   T 5
000B        :SI  T 5
000C        :A   T 5
000D        :SI  T 5
000E        :
000F        :L   KM00000000 00000000      LOAD COMPARATOR
0011        :
0012        :L   FW25                     LOAD FLAG WORD TO TRIGGER
0013        :                             MESSAGES
0014        :
0015        :> <F                         ALL MESSAGES GONE?
0016        :                             (TRIGGER-BITS 0 THRU 16 = 0)?
0017        :JC  = M001                   NO -> TRIGGER NEW MESSAGES
0018        :
```



```
0019              :
001A              :L     KM00001100 10111001   YES -> START NEW MESSAGE
001C              :T     FW25                  CYCLE AND TRIGGER MESSAGES
001D              :BEU
001E              :
001F  M001        :SLW 1                       TRIGGER NEW MESSAGES
0020              :T     FW25
0021              :
0022              :BE
```

FB 10   LEN = 258   ABS
                                             PAGE 1

FUNCTION BLOCK TO ORGANIZE THE OLD AND NEW VALUES FOR PIXELGRAPHIC CURVE
DISPLAY WITH THE CP526/CP527/CP528 AND OP30B.

FB 10 TAKES OVER UPDATING AND ORGANIZING THE DATA BLOCK CONTAINING OLD
AND NEW VALUES FOR DISPLAYING TIME-DEPENDENT CURVE PLOTS WITH THE CP. IT
MUST BE CALLED IN THE CYCLIC PART OF THE PROGRAM. CALLING AND
PARAMETERIZING THE HANDLING BLOCKS NECESSARY FOR DATA EXCHANGE WITH THE
CP MUST BE DONE BY THE USER.

PARAMETERS:

DB   DB1 - DB255, FORMAT B
                      DATA BLOCK IN WHICH LIST OF OLD AND NEW VALUES SHOULD BE
                      STORED. THIS DATA BLOCK MUST BE SPECIFIED IN THE "CURVE
                      WINDOW" MENU AS THE DATA BLOCK FOR THE OUTPUT VALUES.
                      DW 0 MUST BE SPECIFIED AS THE DATA WORD. IT IS TO BE
                      ALLOCATED A LENGTH OF 255 DATA WORDS.

N   CONTROL-WORD, FORMAT KM
                      CURVES 1 - 6 ARE ENABLED BY SETTING THE CORRESPONDING BITS 0 -
                      5. UPDATING OF OLD AND NEW VALUE LISTS IS ONLY DONE FOR
                      ENABLED CURVES. BIT 6 DESIGNATES THE NUMERICAL FORMAT OF
                      THE VALUES. (1 = BYTE, 0 = 16-BIT FP). OLD- VALUE OUTPUT CAN BE
                      SUPPRESSED BY SETTING BIT 7, AS LONG AS NO NEW VALUES ARE
                      PRESENT, FOR EXAMPLE AFTER A COLD RESTART.

AZ
NUMBER OF OLD VALUES, FORMAT KF
NUMBER OF OLD VALUES PER CURVE.
1) 16-BIT FP REPRESENTATION
   (NUMBER OF OLD VALUES * NUMBER OF CURVES < = 240)
2) BYTE REPRESENTATION
   THE MAXIMUM NUMBER OF OLD VALUES AMOUNTS FOR

1 CURVE	TO	240 OLD VALUES
2 CURVES	TO	240 OLD VALUES
3 CURVES	TO	160 OLD VALUES
4 CURVES	TO	120 OLD VALUES
5 CURVES	TO	96 OLD VALUES
6 CURVES	TO	80 OLD VALUES

W1 - W6
CURVE VALUES, FORMAT KY
THE Y-VALUES TO BE DISPLAYED ARE SENT TO THE FB VIA FLAG
WORDS OR DATA WORDS, WHEREBY THE FOLLOWING APPLIES:
KY 0,Y   VALUE FROM FLAG WORD Y
KY X,Y   VALUE FROM DATA WORD Y
                DATA BLOCK X (X > 0)

TA
UPDATE TIME, FORMAT KT
THE AMOUNT OF TIME IN WHICH THE CP UPDATES THE VALUES IS TO
BE ENTERED HERE (1 SEC., 2 SEC., 5 SEC., 10 SEC., 30 SEC., 1 MIN., 2
MIN., 5 MIN.)

TI
TIMER FOR UPDATING, FORMAT T

F
IN ORDER TO SCAN EVEN LONGER TIME PERIODS WITH A LIMITED
NUMBER OF OLD VALUES, IT IS POSSIBLE TO ENTER A VALUE INTO
THE OLD-VALUE LIST AFTER MULTIPLE UPDATINGS, RATHER THAN
AFTER EACH UPDATING. IN THIS CASE, THE TIME INTERVAL BETWEEN
OLD VALUES IS F * TA. THE TOTAL TIME PERIOD COVERED BY THE
OLD VALUES AMOUNTS TO:
$$TTSP = AZ * F * TA.$$

NAME	:CURVE		
DES	:DB	I/Q/D/B/T/C: B	
DES	:N	I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KM
DES	:AZ	I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KF
DES	:W1	I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KY
DES	:W2	I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KY
DES	:W3	I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KY
DES	:W4	I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KY
DES	:W5	I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KY
DES	:W6	I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KY
DES	:TA	I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KT
DES	:F	I/Q/D/B/T/C: D	KM/KH/KY/KS/KF/KT/KC/KG: KF
DES	:TI	I/Q/D/B/T/C: T	

0029	:A	= TI	UPDATE TIME RUN OUT ?
002A	:BEC		
002B	:DO	= DB	CALL SENDING DATA BLOCK
002C	:LW	= TA	
002D	:A	= TI	
002E	:SP	= TI	START TIMER WITH UPDATE TIME
002F	:AN	= TI	
0030	:SP	= TI	
0031	:LW	= N	ENTER ENABLE WORD INTO DL0
0032	:T	DL0	
0033	:T	FB202	
0034	:LW	= AZ	ENTER NUMBER OLD VALUES IN DR0
0035	:T	DR0	
0036	:LW	= F	ENTER F IN DL9
0037	:T	DL9	
0038	:L	KB0	WORK BYTE FOR FORMAT "BYTE"
0039	:T	FB203	
003A	:***		

```
NETWORK 2                              PROCESS NEW VALUE CURVES 1 TO 6
003B            :L    KY0,202              BIT NO., BYTE NO. FOR IND. ADR.
003D            :T    FW204
003E            :L    KB1
003F            :T    FB206                COUNTER CURVES
0040            :T    FB207                JUMP LABEL POINTER
0041  LOOP      :DO   FW204
0042            :AN   F 0.0                ( = AN F 202.0 TO AN F 202.7)
0043            :JC   = WEI                ENABLE CURRENT CURVE
0044            :DO   FW206                (1, 3, 5, 7, 9, 11)
0045            :JU   = M001
0046  M001      :LW   = W1                 DEPENDING ON THE CURVE NUMBER, THE
0047            :JU   = RET                 CORRESPONDING PARAMETER IS LOADED
0048            :LW   = W2
0049            :JU   = RET
004A            :LW   = W3
004B            :JU   = RET                "JUMP ISLAND"
004C            :LW   = W4
004D            :JU   = RET


FB 10                                  A:FB10@@ST.S5D


004E            :LW   = W5
004F            :JU   = RET
0050            :LW   = W6
0051  RET       :T    FW200                STORE PARAMETERS W1 - W6
0052            :L    FB200
0053            :L    KB0                  CHECK IF NEW VALUE IN DB OR FW
0054            :>F
0055            :JC   = M002
0056            :DO   FW200                ENTER OUTPUT VALUE FROM FLAG INTO
0057            :L    FW0                  SENDING DATA BLOCK
0058            :DO   FW205                ENTRY POINTER
0059            :T    DW0
005A            :JU   = WEI
005B  M002      :DO   FW199                ENTER OUTPUT VALUE FROM DATA BLOCK
005C            :C    DB0                  INTO SENDING DATA BLOCK
005D            :DO   FW200
005E            :L    DW0
005F            :DO   = DB
```

0060		:DO	FW205	ENTRY POINTER
0061		:T	DW0	
0062	WEI	:L	KB6	
0063		:L	FB206	LAST STRING REACHED ?
0064		:<F		
0065		:JC	= SEGE	
0066		:I	1	INCREMENT STRING COUNTER
0067		:T	FB206	
0068		:L	FB207	
0069		:I	2	INCREMENT JUMP LABEL POINTER
006A		:T	FB207	
006B		:L	FB204	
006C		:I	1	INCREMENT STRING ENABLE BIT
006D		:T	FB204	
006E		:JU	= LOOP	RETURN TO THE BEGINNING OF THE LOOP
006F	SEGE	:***		

NETWORK 3				PROCESS OLD VALUE CURVES 1 TO 6
0070		:L	KY0,202	BIT NO., BYTE NO. FOR IND. ADR.
0072		:T	FW204	
0073		:L	KB1	
0074		:T	FB206	COUNTER CURVES
0075		:L	KB10	
0076		:T	FB207	BLOCK POINTER
0077		:L	DL9	
0078		:L	DL8	ENTER NEW VALUES IN OLD VALUE LIST ?
0079		:!=F		
007A		:JC	= M001	
007B		:I	1	INCREMENT CYCLE COUNTER
007C		:T	DL8	
007D		:BEU		
007E	M001	:L	KB1	RESET CYCLE COUNTER
007F		:T	DL8	
0080		:L	DR0	
0081		:L	DR8	
0082		:I	1	END OF POLLING BUFFER FOR OLD VALUES
0083		:>F		REACHED ?
0084		:JC	= M002	
0085		:L	KB0	RESET ENTRY COUNTER
0086	M002	:T	DR8	

```
0087              :T    FB208
0088              :AN   F202.6          IDENTIFIER BYTE
0089              :JC   = LOOP
008A              :SRW  1               :2


FB 10                                   A:FB10@@ST.S5D


008B              :T    FB208
008C              :SPZ  = LOOP
008D              :S    F203.0          IDENTIFIER DR
008E  LOOP        :L    FB208
008F              :L    FB207           COMPUTE ENTRY POINTER CURVE 1
0090              : + F
0091              :T    DR9             CURRENT  ENTRY POINTER
0092              :DO   FW204
0093              :AN   F0.0            ENABLE CURRENT CURVE ?
0094              :JC   = WEI
0095              :DO   FW205           STRING NUMBER
0096              :L    DW0
0097              :A    F202.6          IDENTIFIER BYTE
0098              :JC   = DL
0099              :DO   DW9             ENTER NEW VALUE CURVE X IN OLD-
009A              :T    DW0             VALUE LIST CURVE X
009B              :JU   = WEI
009C  DL          :A    F203.0          IDENTIFIER DR
009D              :JC   = DR
009E              :DO   DW9             ENTER NEW VALUE BYTE BY BYTE
009F              :T    DL0
00A0              :JU   = WEI
00A1  DR          :DO   DW9             ENTER NEW VALUE BYTE BY BYTE
00A2              :T    DR0
00A3  WEI         :L    KB6
00A4              :L    FB206           LAST STRING REACHED ?
00A5              :< = F
00A6              :JC   = SEGE
00A7              :I    1               INCREMENT STRING POINTER
00A8              :T    FB206
00A9              :L    DR0
00AA              :AN   F202.6          IDENTIFIER BYTE
00AB              :JC   = WEI1
```

```
00AC          :SRW  1              : 2
00AD  WEI1    :L    FB207
00AE          : + F               BLOCK POINTER TO NEXT BLOCK
00AF          :T    FB207
00B0          :L    FB204
00B1          :I    1             INCREMENT STRING ENABLE BIT
ßßB2          :T    FB204
00B3          :JU   = LOOP        RETURN TO THE BEGINNING OF THE LOOP
00B4  SEGE    :BE
```

Assignment of data block DB 1 for I/O allocation:

```
Digital inputs:        ,     ,     ,     ,     ,     ,     ,     ,     ,     ,     ,
Digital outputs:       ,     ,     ,     ,     ,     ,     ,     ,     ,     ,     ,
IPC flag inputs:     , 20 ,      ,     ,     ,     ,     ,     ,     ,     ,     ,
IPC flag outputs:    ;     ,     ,     ,     ,     ,     ,     ,     ,     ,     ,
Time block length:   ,     ,
```

Contents

Introduction

Ordering Data

Technical Description of the
CP 526 Module

Technical Description of the
CP Expansion Board

Example of an Application Program
for the CP 526

System Restrictions

6 System Restrictions

6.1 System Restrictions CP 526

Hardware	
Basic module color	Yes
Basic module monochrome	Yes
Expansion unit possible	Yes
Operation in programmable controller	For ambient temperature < 55°C with fan Operation with expansion board only with fan
Power supply	+ 5 V, tolerance 5% + 24 V, tolerance + 25/-15%
Power consumption	at 5 V; 2.2 A at 24 V; max. 240 mA (with process keyboard)
Slot width basic module	1 SPS (20.32 mm)
Slot width expansion board	2 SPS (40.64 mm)
Possible programmable controllers	115U/135U/150U/155U
Monitor connection to CP (color/monochrome)	R,G + S,B,S/composite video; max. 80/100 m
One serial interface for connection to programmer and keyboard	20 mA, TTY
Programmer	9600 bps
Keyboard	2400 bps; max, 200/150 m

Hardware	
Screen resolution, high resolution Screen resolution, low resolution Horizontal frequency, high resolution Horizontal frequency, low resolution Display format Pixel matrix high resolution (H * W) Pixel matrix normal resolution (H * W) Number of colors for foreground and background	640 * 468 640 * 234 30.84 kHz 15.44 kHz 26 * 80 (including 2 lines for function key labelling)) 18 * 8 9 * 8 8
User memory basic module Max. available user memory (with expansion board)	128 kbytes 256 kbytes
Internal hardware clock	Yes

Software	
Standard symbols/symbol set Editable symbols/symbol set Symbol sets per user file Lexicon data	128 Yes 20 63
Segments Static displays Segments per static display Dynamic displays Pixel graphics displays Functions Additional display of process windows	250 250 32 250 250 250 No
Message texts max. Internal message buffer for (max. messages) Variables in message texts Date/time of day in message text 4 message statuses 4 message groups	1000 100 No Yes Yes Yes
Pixel graphic text representation: [1] - Character path in degrees - Different increments - Increment of value output field	 0, 90, 180, 270, max. 16 max. 16
Curve representation: [1] - Curve windows/display - Curve windows/window - Colors - Line types - Fill-in patterns - Limit-dependent color range	 6 6 8 8 8 Yes

---

[1]   Can only be configured in conjunction with the CP expansion board.

Software	
One pixel graphic display can contain	50 bars, value output fields and curve windows per display
Bar representation - Bar width (number of symbols) - Limit-dependent color range [1]	 1 to 80 Yes
Configurable input/output fields Process variables per display	No approx. 150
Message printout Reports	No No
Programmable controller functions: - Scanning of cursor postitions in PLC - Monitoring/disabling/simulation of keyboard - Initiating display updating - Dynamic DB/DW offset - Shape output with variable color attributes - Readable job number/job type - Readable printer status	 No Simul. No No No No No

---

[1]　Can only be configured in conjunction with the CP expansion board.

General characteristics	
BEDI standard software for diagnostics - GRAPH 5 - CP 552 - Status/Force Variables	No No No
Configuration with Configuration on	COM GRAPH PG 635-II, 685, 695, 710-II, 730, 750, 770
Speed: Example: AUT-912 test display with 40 shapes 40 bars and 100 values Initial display generation Update	 8 sec 7 sec

6.2   System Restrictions Expansion Module

Hardware	
Operation in programmable controller Power supply Current consumption	only with fan + 5 V, tolerance 5% 2.3 A
User memory in combination with CP 526 basic board (color) CP 527 basic board	128 kbytes max. 256 kbytes max.

Software	
Text representation Character path in degrees Different increments Increments of value output fields	0, 90, 180, 270 max. 16 max. 16
Curve representation	6 curve windows / display 6 curves/window 8 colors 8 line types 8 filling patterns Configurable limit value and color change
Bar width	1 to 80 characters Configurable limit value and color change
One pixel graphic display can contain	50 bars, value output fields and curve windows per display

Contents

Communications Processor C79000-B8576-C217
CP 526
Contents

Appendix A: CP 526 Error Messages

When an error occurs, the CP 526 displays a message number in the second part of the command line. To display the corresponding text, you must store the message texts in the error message file. If you do not do this, and the standard error messages supplied (COMLIB) are not transferred with COM to the CP 526, only the message numbers (e.g., W02) are displayed. You can find the numbers and the corresponding text in the following pages.

W01:   `ERROR: command queue too long!`

The command buffer has an overflow; commands may have been lost, because they were sent faster than other commands were processed.

W02:   `Command does not exist!`

A command was not programmed, a command was not transferred, an incorrect name was entered, or an incorrect user card was inserted.

W03:   `Last input not enabled!`

This message is displayed when you press the < ENTER > function key, although you did not terminate the last entry in the entry field by pressing the < RETURN > function key.

W04:   `ERROR: Input field overwritten!`

When you tried to position the cursor at a specific entry field, the entry field was overlaid by an output field or a figure output. This indicates a planning error.

W05:   `ERROR in input processing   /5`

W06:   `ERROR in input processing   /6`

W07:   ERROR in input processing /7

W08:   ERROR in input processing

W09:   ERROR in command structure /9

W10:   Command incomplete in CP 526!

During selective transfer, parts of a display (i.e., segment, static display, dynamic display, symbol set) were not transferred. This indicates a planning error.

W11:   Trouble in data exchange with the controller!

At this time, the controller cannot deliver the data to the CP 526. If a controller/CP 526 synchronization should have occurred earlier, new values can be received only after a new synchronization.

W12:   ERROR in command structure!   /12

W13:   ERROR during system start-up!

W14:   Illegal output value in controller!

The controller value is illegal for the defined display format. Blanks are displayed for input/output fields.

W15:   Illegal input value!

The CP 526 cannot convert the value into the defined controller format. For example, letters are situated in a numeric field, or the number of places before the decimal point is greater than the number of character positions (COM planning).

W16:   Input value too small!

The limit value fell below the minimum.

W17:   Input value too great!

The limit value exceeded the maximum.

W18:   Input value required for command!

You did not specify an input value after the "-" character.

W19:   Illegal date/time format!

For example, a wrong separator, a wrong sequence, or illegal values were specified.

W20:   Gain factor 0 illegal!

The value 0 stands for the gain value in the data block. This results in a division by 0 during the input check.

W21:   Date/time cannot be shown at present!

Access is not possible because at the time of the display, the CP 526 was changing the time, or the controller was writing or reading the time.

W23:   No pixel graphics monitor connected!

An attempt was made to call pixel graphics displays, although the CP 526 parameters are set for a low-resolution monitor.

W24:   Illegal bar value in the controller!   /24

W25:   Expansion unit not present!

Commands have been programmed which require the expansion unit (pixel graphics) in order to be processed.

W26:   ERROR in command structure!   /26

W27: `Displayed process data invalid!`

If programs are deleted or overwritten in the RAM memory module of the CP 526 by COM, the process displayed is no longer updated.

W28: `CP 526 mode selector on STOP!`

In this operating mode, no values are accepted by or transmitted to the controller.

W29: `CP 526 not synchronized with the controller!`

There was no synchronization between the controller and CP 526.

W30: `Input value processed without error!`

The current values in the input fields were accepted by the controller free of error. This is an acknowledgement message and does not indicate an error.

W31: `Bar diagram incorrectly configured! /31`

This error occurs when you do not observe the following relations when defining the bar diagram:

*   Smallest scale value ≤ lower limit ≤ reference value ≤ upper limit ≤ largest output value

*   Largest output value = smallest output value + index of the length * step width
These values can also be transferred indirectly into a data block (DB).

W32: `Illegal bar value in controller! /32`

A measured value represented as a bar lies outside the defined range (i.e., smallest output value, largest output value.) When displayed, it is shown as the smallest or the largest output value.

W33:   Bar chart incorrectly configured!   /33

If message W31 appears, the bar chart cannot be updated. You must make changes in the bar chart before updating can take place.

W34:   Curve diagram incorrectly configured!   /34

This is the same error as message W31. In addition, the error can be caused by incorrect curve definitions.

The following relations apply for the curve window:

● Smallest scale value ≤ lower limit ≤ reference value ≤ upper limit ≤ largest output value

● Largest output value = smallest output value + X * step width of the y-axis, where X depends on the following window sizes:

window sizes 80 by 22 and 80 by 11 mean: X = 4
window sizes 40 by 22, 40 by 11 and 26 by 11 mean: X = 2

The following must hold for each curve:

Smallest output value < largest output value.

W35:   Illegal curve value in controller!

Like message W32, only the measured value is shown as a curve.

W36:   Curve diagram incorrectly configured!

Like message W34, only the measured value is shown as a curve.

W40:   Curves / bars not updated!

The planned updating time for curve windows cannot be adhered to.

W41:   Overflow in CP 526 message buffer!

Since the last acknowledgement, the controller has transmitted more then 100 messages, so that further messages are lost. Call the message display and acknowledge.

W42:   **ERROR:Illegal controller keyboard code!**

     Because of a SEND-DIRECT 219 function, key codes sent from the controller do not fit into the input field, or are not understood by the CP 526.

W60:   **Input command processed without error!**

     The value associated with the input command was entered by the controller without error. This is an acknowledgement message and does not indicate an error.

Messages W05, W06, W07 W08, W09, W12, W13, W24 and W26, indicate internal errors of the CP 526, or COM errors in the S5-DOS application programs. If any of these errors occur please contact the nearest Siemens office and inform them of the error. This will assist us in improving the CP 526/COM product.

Appendix B: Reference Literature

It is recommended that you have the following books that support the
CP 526. Order from your local Siemens representative unless otherwise indicated.

- *STEP 5 Language Primer*
  (Publication No. 05-019-03-1285)
  Order from: Siemens Energy & Automation, Inc.
                 Programmable Controls Division
                 10 Technology Dr.
                 Peabody, MA 01960 USA

- *Programming of Control Systems in STEP 5*
  Volume 1 - Basic Software
  (Order No. A19100-L531-F100-X-7600)

- *Programming of Control Systems in STEP 5*
  Volume 3 - Programming of Function Blocks
  (Order No. A19100-L531-F15-X-7600)
  Order Volumes 1 and 3 of *Programming of Control Systems in*
  STEP 5 from: Siemens AG
                   ZVW 85, Lager
                   8510 Fuerth-Bislohe
                   Federal Republic of Germany

For information about data handling between the communications processor (CP) and the programmable controller (PC), see the following books:

- *S5-135U Programmable Controller*
  Data Handling Blocks for the R Processor, Operating Instructions
  (Order No. C79000-B8576-C366)

- *S5-150U Programmable Controller*
  Data Handling Blocks, Operating Instructions
  (Order No. P72100-A0116-X-01-19)

- *S5-115U Manual*
  (Order No. 6ES5 998-OUF22)