SIEMENS

Wichtige Hinweise / Important Notes / Remarques importantes
Informationen / Information / Informations
Vorschläge/Korrekturen / Suggestions/Corrections /
Propositions/Corrections

Beschreibung	C79000-B8500-C365-04 C79000-B8500-C388-02	1
Description	C79000-B8576-C365-04 C79000-B8576-C388-02	2
Description	C79000-B8577-C365-04 C79000-B8577-C388-02	3

SIMATIC S5

Reglerstruktur R64 Standard-Funktionsbaustein und Programmieranleitung COMREG

R-Prozessor / CPU 928
Automatisierungsgerät S5-135U

Beschreibung

Controller Structure R64 Standard Function Block and Programming Guide COMREG

R-Processor / CPU 928
S5-135U Programmable Controller

Description

Structure de régulation R64 Bloc fonctionnel standard et guide de programmation COMREG

Processeur R / CPU 928
Automate programmable S5-135U

Description

C79000-G8563-C365-02

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 SIMATICcompatible 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."

SIEMENS

SIMATIC S5

Controller Structure R64
Standard Function Block
R-Processor/CPU 928
S5-135U Programmable Controller

Description Order No. C79000-B8576-C365-04

Contents Page

1 Introduction

Closed-loop control is a process in which the measured value of a controlled condition (*controlled variable, actual value*) is compared with a preset value (*reference variable, setpoint*) and corrected, depending on the difference.

The control action takes place in a *closed-loop* system comprising the *controlled system* or *plant* and the *controller*.

This generalized definition, which is based on DIN 19226, is illustrated in the following block diagram.

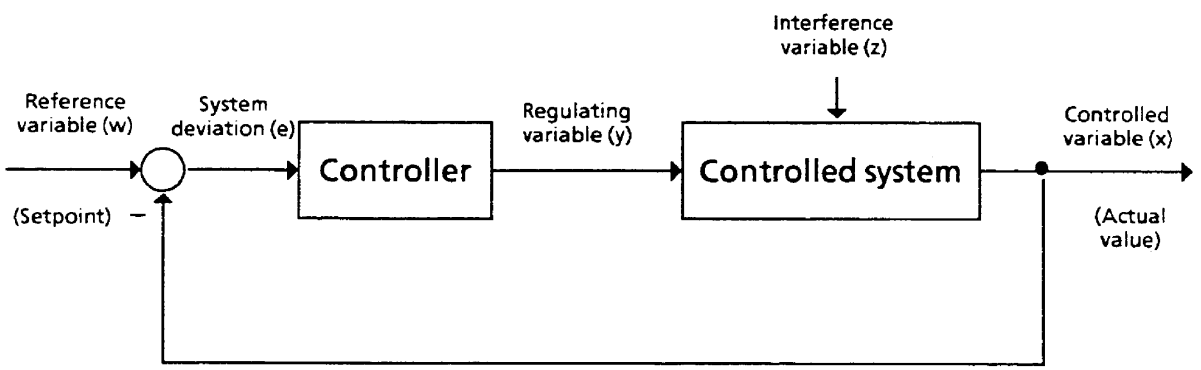


Figure 1-1 Block diagram of a control loop

The controlled system is normally affected by *interferences* that result in unwanted changes in the *controlled variable*.

The controller compares the *controlled variable* (actual value) with the preset *reference variable* (setpoint) and computes the *regulating variable* from the difference (*deviation*). The regulating variable corrects the deviation of the controlled variable from its setpoint caused by the interference.

1.1 The PID Controller

One controller type that has proved its worth is the PID controller. The PID function

$$y(t) = K_p \left\{ e(t) + \frac{1}{T_N} * \int_0^t e(v)\,dv + T_V * e(t) \right\}$$

is the basis on which the majority of industrial analog controllers operate: (y (t) = regulating variable, e(t) = deviation).

The proportional-action component (P component) determines the system deviation.

The integral-action component (I component) determines the average system deviation.

The derivative-action component (D component) determines the rate of system deviation.

The *proportional-action value* K_p, *integral-action time* T_N and *derivative-action time* T_V are the *controller parameters.*

Figure 1-2 shows the classical *control loop* with a PID controller and the symbols used (w = setpoint, x = actual value, e = w-x = deviation, y = regulating variable).

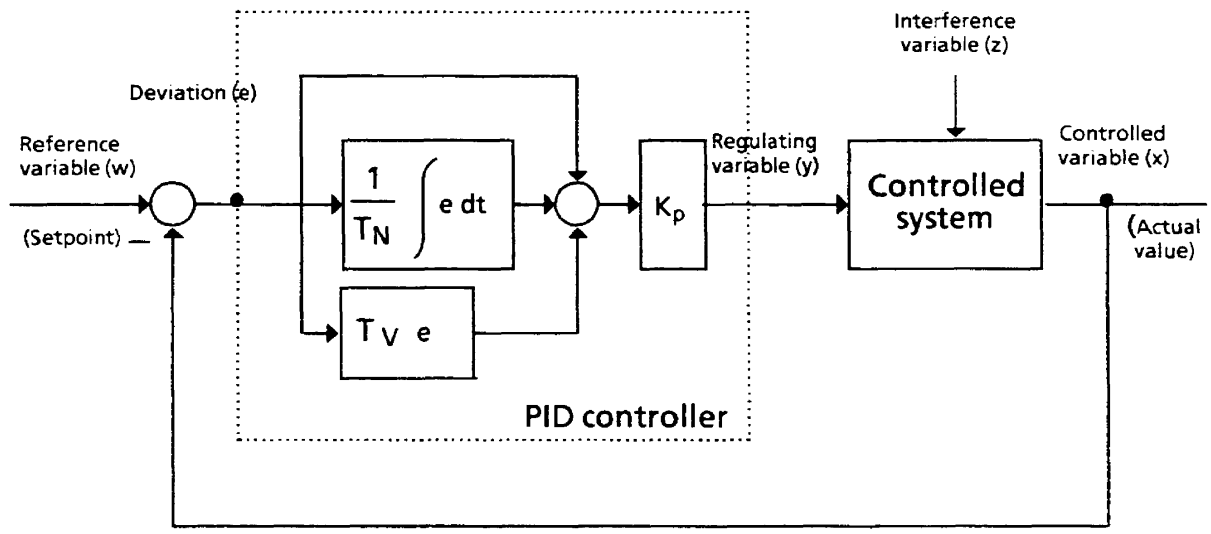


Figure 1-2: Classical PID control

Because PID structures have proven successful in a great variety of applications, and because practical design procedures are available for them, efforts are being made to combine the advantages of PID control with the enormous flexibility of *digital control systems.*

A PID controller based on digital techniques in the form of a program executing on a microcomputer must therefore closely approximate the dynamic performance of an analog controller.

What would this mean in real terms?

The setpoint and actual value of a control loop are sampled only within the program processing cycle of the controller, i.e., at *discrete instants*

$$t_k = k * T_A, \qquad k = 0, 1, 2, \ldots,$$

The PID operation is simulated by a program-based *PID algorithm*. Using stored historical process signal values, the algorithm computes a *differential equation* in each sampling step.

A controller that takes a string of input values

$$\{ e(t = t_k) := e_k \}, \qquad k = 0, 1, 2, \ldots\ldots,$$

and generates a string of regulating variables

$$\{ y(t = t_k) := y_k \}, \qquad k = 0, 1, 2, \ldots\ldots,$$

is called a *sampling controller*.

The interval between two such updating cycles is referred to as the scan time T_A.

The computed value of the regulating variable is converted from digital to analog form (actuator) and forwarded to the process via a *final control element*.

The *quasi-analog* action required of the digital PID controller is attained by sufficiently frequent sampling, i.e., by selecting a suitable scan time.

1.2 Selecting the Scan Time

1.2.1 Problem Definition

The *scan time* is a characteristic of digital control loops which has no counterpart in analog control systems. It represents the *interval* or period between two controller program cycles.

The scan time should be as long as possible to reduce program run time. The longer the scan time, the less often the control system need call upon the services of the processor services, so that the number of control loops the processor can handle increases.

The scan time must not be too long, however, as the controller is otherwise not able to react fast enough to a interference.

On the other hand, the scan time should not be below a certain minimum prescribed by the program execution times and the times the ADCs and DACs need for conversion. If, for example, the ADC (6ES5-465-4U module) were to provide a new measured value only every 400 ms, there would be no sense in having a scan time of less than 400 ms.

Another important aspect is the software controller, which should be able to be assigned parameters in the same manner as an analog controller, and still have the same performance characteristics. This requires that a software controller have approximately the same dynamic behavior as the analog controller.

Selection of a suitable scan time is therefore crucial.

1.2.2 Rule of Thumb for Selecting the Scan Time

The following must be taken into consideration before an attempt is made to formulate a rule of thumb for selecting the scan time:

- The scan time is generally a compromise between several requirements, many of which are contradictory. For this reason, a general formulation can be a *guideline* at best.

- Strictly speaking, the term *"time constant"* is defined only for first-order time-delay elements. In practice, however, the majority of systems have an order > 1, i.e., they have *more than one time constant.*

  In this case, it is therefore meaningful to define at least the *dominating time constant*, or possibly a substitute time constant T. This could be allocated to a system based on its settling time T_{95} in accordance with the approximate relationship $T_{95} = 3 * T$ for the TD1 component (!). (T_{95} is the time taken by a controlled system to reach 95% of the final steady value in response to a step change in the regulating variable.)

- The dynamic response of the control loop, and thus the selection of the scan time, is determined neither by the "smallest" nor the "largest" nor "the" system time constant, but rather by the *controlled system's time constants*; the latter result from the controller parameters that were set, and one of the controlled system's time constants is often a dominating time constant. Finally, the objective of a control system is usually a fast settling time, i.e., a reduction in the number of time constants effective within the control loop.

"Rule of thumb" for selecting the sampling time (for a quasi-analog controller):

> Experience has shown that a sampling time of approximately 1/10 of time constant T_{RK}, which determines the step-function response of the *closed control loop*, will result in a controller response comparable to that of an analog controller:
>
> $$\longrightarrow \quad T_A = \frac{1}{10} T_{RK}$$

1.3 Compact Closed-loop Control System - Characteristics and Objectives

The essential aspects of a *compact closed-loop control system* can best be illustrated by classifying the various possible control concepts.

One method of solving a control problem is the implementation of a *"tailor-made"* controller. This is the optimum solution, since it is adapted to suit specific requirements.

It is, however, a single-purpose controller, designed for a specific system, and must be redesigned for any other system. Any advantages in terms of functionality are outweighed by high development overhead.

Its counterpart is the *modular concept*. The modular concept enables the user to interconnect self-contained subfunctions in the form of pre-programmed software blocks to produce a controller structure of virtually unlimited complexity. This complexity presents great difficulties.

The compromise between these two extremes is the *compact closed-loop control system*, which is characterized by a restricted range of certain elementary functions. Structuring overhead is thus reduced to the omission of unnecessary subfunctions.

This concept is still relatively flexible, i.e., it is easily adaptable to changing process situations, requires only average know-how on the part of the user, and requires little time to produce a highly efficient control system.

It was this reasoning which led to the development of the *R64 controller structure*. This controller was developed for temperature control, drive control, volumetric flow control, level control and so forth. The controller provides the following advantages:

- In addition to the actual controller with PID algorithm, the compact controller provides functions for preprocessing of the setpoint and actual values and postprocessing of the computed regulating variable.

- It also provides display and monitoring functions.

- This basic function set can also be *configured to a limited extent*. Subfunctions or entire branches can be disabled via so-called *structuring switches*.

- *Parameters* need to be *assigned* only for the program sections remaining in the reduced structure.

- Implementation of a control system using the R64 controller structure, from structuring through initialization all the way up to its invocation by the operating system, requires no programming overhead, *nor* does it require knowledge of the *STEP 5 language*.

2 General Remarks

2.1 Standard Package

The R64 controller structure is delivered as a software *function block (FB102)* on a *PCP/M-86 diskette* suitable for a programmer. The function block is on the REGR64 ST.S5D file. This file also contains a DB1 whose use is described in detail in section 2.5.1.2. The diskette also includes the REGLERST.S5D file, which contains controller data blocks as sample applications (see chapter 6).

2.2 Performance Characteristics

The R64 controller structure's most important performance data are listed below. Note that the controller can be put into operation without programming experience, i.e., without a STEP 5 program. Parameters are used to interface the controller to the I/O module.

Scan times:	A maximum of eight different scan times is permissible. Shortest scan time T_A = 20 ms
Program run time:	< 2.5 ms
No. of controllers:	As many as 64, depending on the sampling times Max. 8 controllers when T_A = 20 ms Max. 64 controllers when T_A = 160 ms
Data memory:	One 1K byte controller DB per controller A 64K user RAM submodule is thus required for 64 controllers. The number of controllers is restricted to 22 (R processor) / 45 (928 module) when using an EPROM submodule.
Program memory:	The controller function block must be loaded only once into the module's user memory, regardless of the number of controllers used, and requires approximately 8K bytes.
Setpoint branch:	Input of a periodic sequence of setpoint values (setpoint string) Setpoint input over ADC Setpoint input from the programmer in the form of a constant Setpoint input via a STEP 5 program, e.g., for cascaded controllers Setpoint input through a setpoint generator (integrated in the ramp-function generator) Ramping of setpoint step changes (ramp-function generator) or filter
Actual value branch:	Actual value input over ADC Actual value input from the programmer in the form of a constant Actual value input and measured value conditioning via a STEP 5 program

	Elimination of mavericks from the conversion module (validity check) Actual value filtering (smoothing) Linearization of a non-linear characteristic (polygon curve)
Controller types:	Controllers with continuous output (analog) and actuating signal conditioner Controllers with mark space output (pulse duration modulation of continuous regulating variable) Point controllers Manual mode (open-loop mode) with precautionary measures for mode switching
Process visualization:	Display of status variables (setpoint, actual value, regulating variable) at specific measuring points by means of programmer (PG) or communications processor (CP) Output of these measured values to DAC Flagging of limit violations at these measuring points
Operator-process communication:	On-line modification of parameters via programmer, communications processor or STEP 5 program

2.3 System Components

The following *hardware and software components* are required for constructing a control loop based on the compact controller concept.

2.3.1 Hardware

A SIMATIC S5-135U programmable controller with R-processor (CPU 922 beginning release 6) or CPU 928 is required.

One of the following analog input modules is required:

6ES5-463-4U (coding time	20 ms for 4 channels)
6ES5-460-4U (coding time	0.48 sec for 8 channels)
6ES5-465-4U (coding time	0.48 sec for 8 channels, 0.96 sec for 16 channels)
Both modules need these plugin range cards:	+/- 20 mA +/- 50 mV, 500 mV, Pt 100 +/- 1 V, 5 V, 10 V 4 - 20 mA, see below

Both modules must be set for cyclic sampling (two's complement).

In the measuring range 4 to 20 mA, the value for 4 mA must be converted to zero % (= OH). This can be set by a switch on the 463 module. The 460 and 465 modules convert 4 mA to 25% (= 1000H) and 20 mA to 125%.
The controller must therefore adapt the format, which can be selected by setting D 24.6 to 1.

6ES5-243-1A (coding time 35s)
This module can be used only in conjunction with a STEP 5 program (→ 2.3.1.1).

One of the following analog output modules, where applicable, is required:

6ES5-470-4U

6ES5-243-1A with STEP 5 program only (→ 2.3.1.1)

One of the following digital I/O modules, where applicable, is required:

6ES5-420-4U to 6ES5-481-3B

A PG 635, PG 675, PG 685 or PG 695 programmer.
Process-specific peripherals (switches, displays, power supply, etc.).

2.3.1.1 Adapting the Module Format

The 243 analog I/O module can be used only when a STEP 5 program matches the input and output value to the controller's analog I/O format.

a) Analog input module, bipolar mode, +/- 5 V or +/- 10 V

In the example below, controller DB 50 is supplied with a setpoint and an actual value from a 243 analog module set to module address 192.

```
L    KH0000          Select channel 0
T    PY199           (PY = peripheral byte)
T    PY198           Start ADC
C    DB50            Select controller, e.g. DB50
L    PW198           ADC value from channel 0
L    KH8000          Match data format
-F
SRW 1
T    DW168           Setpoint in controller DB
L    KH0001
T    PY199           Select channel 1
T    PY198           Start ADC
L    PW198           ADC value from channel 1
L    KH8000          Match data format
-F
SRW 1
T    DW169           Actual value in controller DB
```

The SRW 1 command must be omitted when only half of the convertible voltage range is to be used, thus providing an overflow of 100% to 200%; 5 V is then converted to 100%.

b) Analog input module, unipolar mode, 0 - 10 V

In the following example, controller DB 50 is supplied with a setpoint and an actual value from the 243 analog module, which is set to module address 192.

```
L    KH0000
T    PY199          Select channel 0
                    (PY = peripheral byte)
T    PY198          Start ADC
C    DB50           Select controller, e.g. DB50
L    PW198          ADC value from channel 0
SRW 1               Match data format
T    DW168          Setpoint in controller DB

L    KH0001
T    PY199          Select channel 1
T    PY198          Start ADC
L    PW198          ADC value from channel 1
SRW 1               Match data format
T    DW169          Actual value in controller DB
```

The SRW 1 command must be omitted when only half of the convertible voltage range is to be used, thus providing an overflow of 100% to 200%; 5 V is then converted to 100%.

c) Analog output module

In the following example, the actuating signal at DAC 1 of the controller with DB number 50 is output to channel 1 of a 243 analog module.

```
C    DB50           Select controller DB, e.g. 50
L    DW208          Value of regulating variable from controller DB
L    KHFFFF         Match data format
XOW
SRW 3
A    D 208.15
T    DW210
=    D 210.15
L    DW210
T    PW192          Output to channel 1 of module
```

SRW 3 must be replaced by SRW 4 when only half of the convertible voltage range is to be used, thus providing an overflow of 100% to 200%; 5 V is then converted to 100%.

The dead time between the read-in of an exact actual value and output of the computed correcting signal can be calculated in the worst case (only when the value is supplied and read by a STEP 5 program) from the sum of:

- the ADC's coding time
- the scan time of the STEP 5 program for format-adjustment on input
- the controller's sampling time
- the scan time of the STEP 5 program for format-adjustment on output
- the DAC's conversion time

The dead time varies, as the ADC, the STEP 5 program and the time-controlled controller call are asynchronous.

2.3.2 Software

The following software is required:

Diskette for programmer with Personal CP/M-86 System

Diskette with S5-DOS basic package

Diskette with COM REG software package

Diskette with controller software

User diskette for storing the controller data blocks

Where applicable: PG 670 / PG 675 operating system for generating and modifying a STEP 5 control program

COM REG is the programmer software for the R64 controller structure, and supports the following:

- Input and modification of controller parameters through generation and modification of a controller DB
- Transfer of controller DBs and the function block to/from programmer (floppy, Winchester) /programmable controller
- Processor mode control (STOP, cold restart, warm restart) and setting the mode on individual controllers (STOP, Start)
- Visualization of status variables (setpoint, actual value, regulating variable) on individual controllers on line
- Modification of controller parameters and modes on-line

Note:

- *The controller DBs, as well as DB2, must be generated with COM REG. DX blocks cannot be used as controller data blocks.*
- *On 'Output' and ' Test', COM REG checks for a controller DB by viewing the version number (DW2).*
- *COM REG checks only a few parameters for validity.*
  For this reason, particular attention must be paid to the applicable value ranges when the parameters are assigned. Critical values are shown in bold type in the tables in chapter 3.

Figure 2-1 shows the interconnection of the hardware components and their interfaces to the process to be controlled; particular emphasis has been placed on the control loop structure.

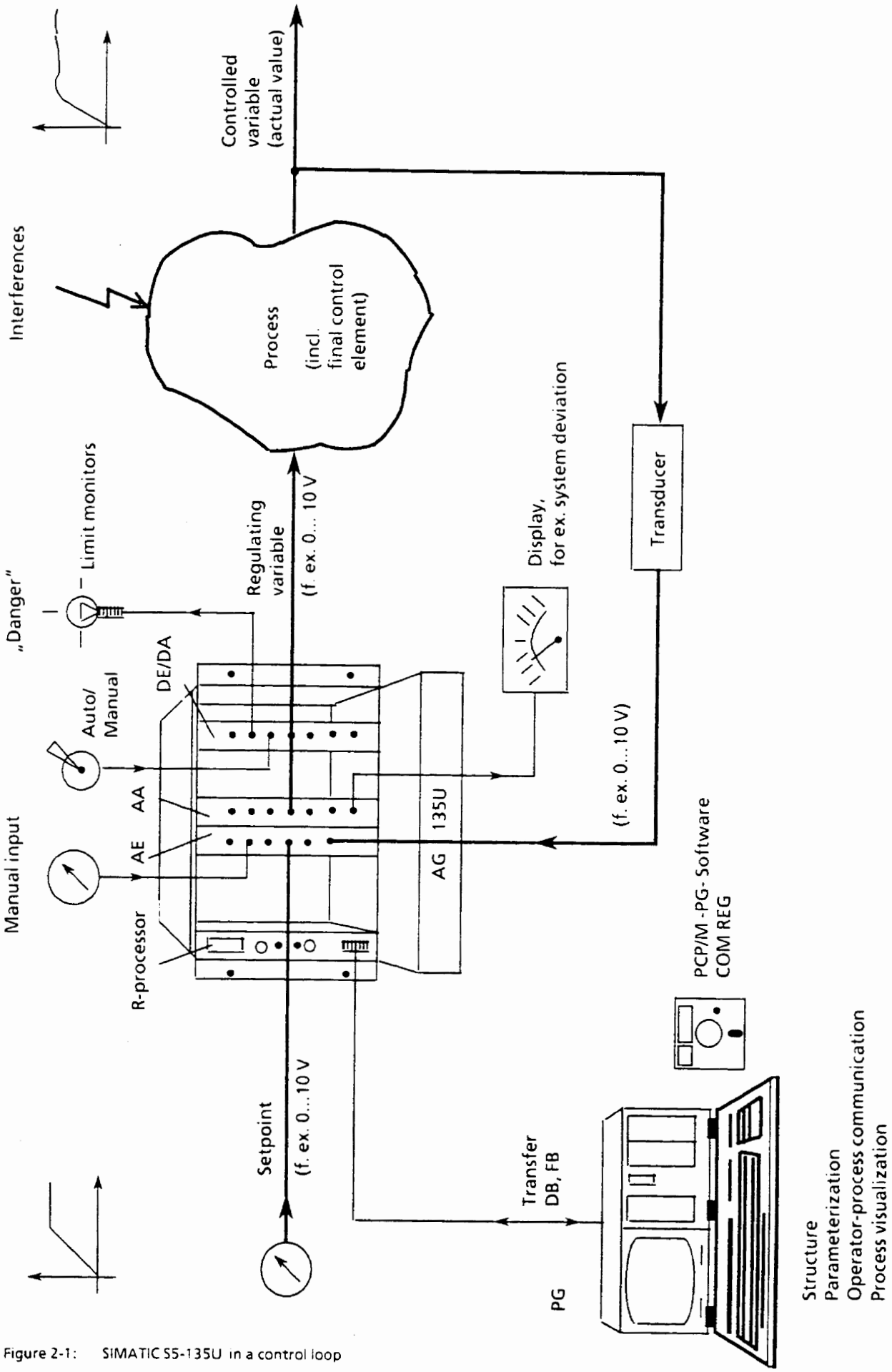


Figure 2-1: SIMATIC S5-135U in a control loop

2.4 Block Transfers between Diskettes with Different Formats

a) Function and data block handling with the S5-DOS basic package:

The programs in the *S5-DOS basic package* store blocks in files on diskette in the same format as COM REG. These software packages can thus access the same files.

b) Function and data block handling with the *PG 670 /PG 675 operating system* (STEP 5 software for the PG 675 which cannot execute under PCP/M or S5-DOS):

Data blocks can be transferred to a STEP 5 diskette via the module, which acts as a buffer. (The blocks are transferred to the programmable controller with COM REG and from there to the STEP 5 diskette).
Controller function blocks cannot be transferred with this software, and an attempt to do so could result in a system crash.
The following must be carefully observed:
Because the function block is written in assembly language, it must be stored at a paragraph address when transferred to the programmable controller. (The FB header must begin at address XXX0H or XXX8H, i.e., the starting address of the FB must be XXX5H or XXXDH). The STEP 5 software mentioned above cannot do this; the result is a system crash on execution of a cold restart.
This can be prevented as follows:
Transfer the controller FB twice consecutively to the programmable controller and invoke the 'COMPRESS' function.
The controller FB now begins at a paragraph address, thus enabling a cold restart.

As previously mentioned, the controller FB is written in assembly language and cannot be decompiled.

2.5 Invoking the Controller over the Operating System

2.5.1 Initializing the Call via DB 2

The controller PB need not be invoked with a JU FBxxx. A controller's call interface to the operating system is data block DB 2, referred to as the *controller list*. The numbers of the controller data blocks must be entered in DB2, and the number of the controller FB (which is delivered as FB102) in the controller DBs.
On a cold restart, the OS checks to see whether a DB2 is in the processor's memory, and puts the controller into operation in accordance with the sampling times selected for the DBs.
Thus, before a controller can be put into operation, the R processor's memory must contain a DB2, the controller DBs entered in the DB2's list, and the controller FB whose number was specified in these controller DBs.

Figure 2-2 shows how the system software organizes the various blocks.

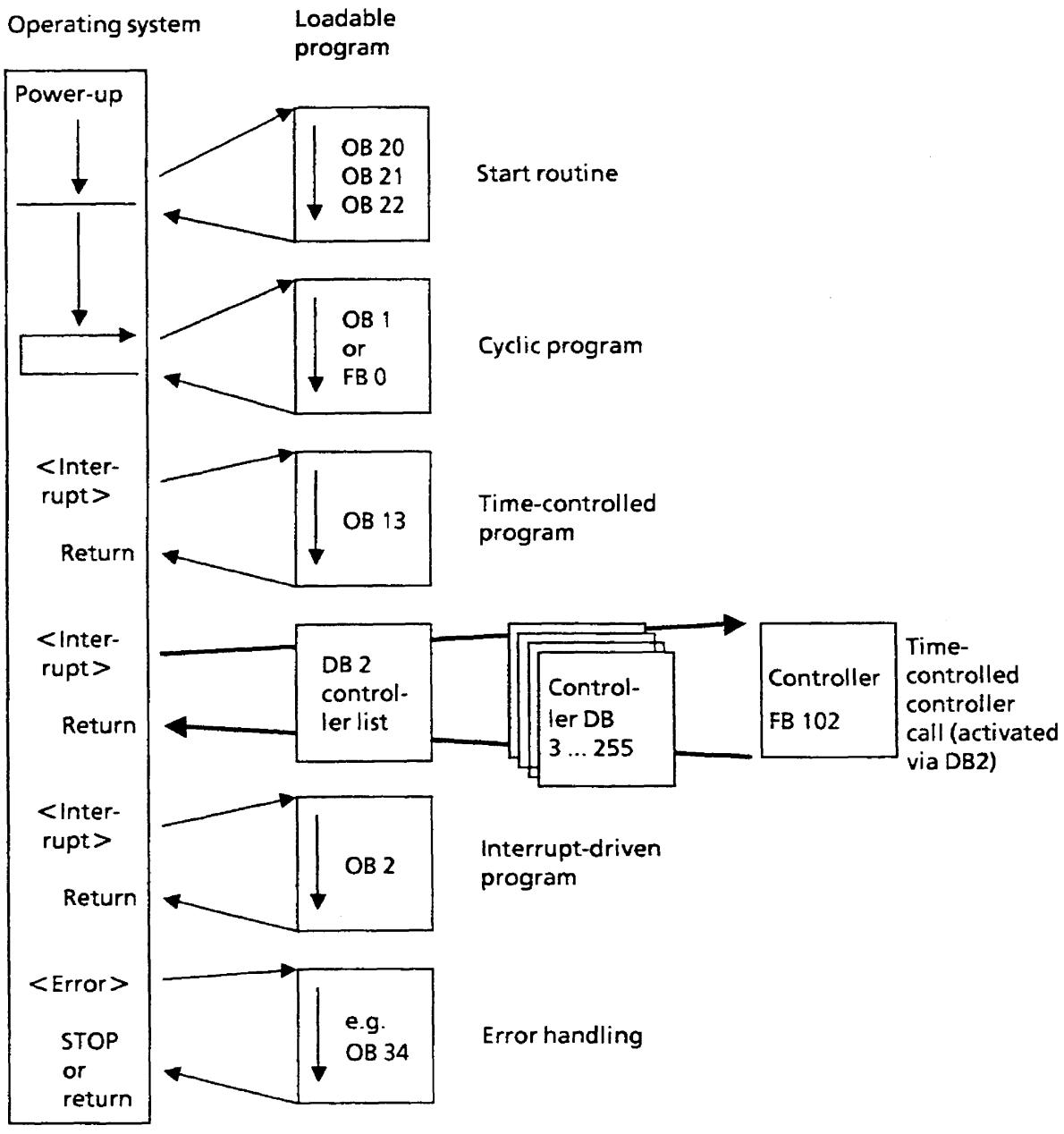


Figure 2-2: Organization of block processing

The priority of the program levels is evident, i.e. it increases in the progression. A higher-priority program can interrupt a lower-priority program at the end of a block (presetting) or statement. The latter must be programmed in the DX0 block (see the programming manual for the processor).

A special feature is the updating of the controller's process input image. The operating system maintains a process input image for each controller (→ 2.5.1.2). The input image is not generated or updated as a function of the controller, but rather at the precise sampling instant, independently of any program. The controller executes its functions as shown above.

Both DB2 and the controller DBs must be generated with the COM REG programmer software.

```
*****************************************************************************
*                                                                           *
*   Input                          Module      : R-Proc. Struct. :          *
*   Controller list                Source/Dest.: File    Block   : DB   002 *
*  ------------------------------------------------------------------------ *
*                                                                           *
*     Time base :     20 ms                                                 *
*                                                                           *
*     The Scan times of the columns are:                                    *
*                                                                           *
*   100 s  |  40 ms |  10 s  |  100 ms | 1.00 hm |   1 s   | 0.10 hm |  20 ms  *
*  ========+========+========+=========+=========+=========+=========+======== *
*    DB    |  DB    |  DB    |  DB     |  DB     |  DB     |  DB     |  DB     *
*  --------+--------+--------+---------+---------+---------+---------+-------- *
*     33   |   59   |  100   |   37    |   45    |   77    |  111    |    3    *
*          |        |        |   67    |         |         |         |   ---   *
*          |  ---   |        |         |         |         |         |   ---   *
*          |  ---   |        |         |         |         |         |   ---   *
*          |  ---   |        |         |         |         |         |   ---   *
*          |  ---   |        |   ---   |         |         |         |   ---   *
*          |  ---   |        |   ---   |         |         |         |   ---   *
*          |  ---   |        |   ---   |         |         |         |   ---   *
*                                                                           *
*                                                                           *
*    F 1   !  F 2  !  F 3  !  F 4  !  F 5  !  F 6  !  F 7  !  F 8          *
*          !       !       !       !       !       !       !                *
*  Enter DB !Remove DB!Time base!       !       !       ! Ready ! Break     *
*                                                                           *
*****************************************************************************
```

Figure 2-3: Example of a completed DB2 screen form (controller list)

Details on the scan times (→ 4.1):

100	ms	=	100 milliseconds
10	s	=	10 seconds
0.10	hm	=	0 hours, 10 minutes

--- indicates that fewer than eight controller DB entries may be made in one controller list column.

The first item in the controller list must be the *time base*. (The time base is discussed in detail in section 2.5.1.1). When selecting the time base, note that it must be a common divisor for all of the controllers' scan times.

Once the time base has been chosen, the controllers can be entered in the list by specifying their *DB numbers*. Observe the following when doing so:

- *Point controllers* and controllers with *mark space outputs* are entered in the controller list (DB2) with *minimum pulse duration* T_{min} rather than with the scan time.

- The minimum pulse duration is the shortest possible duration of a pulse. A pulse can be an integer multiple of the minimum pulse duration. These controllers must be called with the minimum pulse duration, as they can modify a pulse output only once each time they are invoked.

The minimum pulse duration thus corresponds to the pulse width. The controller DBs (which are initialized with either the required scan time or the minimum pulse duration) must already be present in the DB2's destination device (floppy disk, Winchester disk or programmable controller) when the controller list is generated.

If they are not, the programmer software cannot enter the controller in the list following entry of the DB number because it does not know its scan time (or the minimum pulse duration).
COM REG accesses the correct value (scan time or minimum pulse duration) when generating DB2, basing its choice on the type of controller involved.

Controllers with identical scan times are entered in the same column in data block DB2. The maximum number of controllers that can be entered in one column is ascertained by dividing the scan time by the time base, but may not exceed eight.

The processing time per controller call is < 2.5 ms. The maximum number of controllers which can be invoked (from 8 to 64) depends on the scan times.

2.5.1.1 Selecting the Time Base

The time base determines the time within which the controllers in one line of the controller list are invoked. Each controller is assigned an equal slice of the time base to ensure that the STEP 5 programs that are executing in parallel will do so as uniformly as possible.

Thus, the larger the time base, the better its distribution among the various controllers. If the time base is short (20 ms is the minimum), the controllers will execute consecutively in as little time as possible. This results in leftover time, dead time, in which no controller can be invoked (as all must wait for the next clock pulse for the first controller).

The interval between invocation of the controllers in one line is the time base divided by the number of columns in which entries have been made.
When the time specified by the time base has expired, the controllers that were entered on the next line of the list are invoked. The interval between the controllers in a column is thus identical to the time base. The interval between the last and the first controller in a column is computed as follows:
Interval = column scan time - (time base * (number of controllers - 1)), where the number of controllers is the number of controllers in the column.
The controllers in a cascade should be invoked in as short a time as possible. This is done by specifying the lowest possible value, which is 20 ms, as time base.

Two different times are of interest when a STEP 5 program is to execute in addition to the control system:

* The *average processor* load attributed to the control system is computed from the sum of the in-service times of the individual controllers, i.e. the sum of execution time/scan time for all controllers (execution time < 2.5 ms).

* The *maximum processor load* within a specific period of time is important, as it enables computation of the maximum amount of time by which the STEP 5 program cycle can be prolonged because of controller processing. The maximum processor load within the period prescribed by the time is computed with the formula:
  2.5 ms * number of columns occupied in the controller list / time base. The time base should define the longest possible time period in order to minimize this value.

2.5.1.2 Controller Process Image

Until recently, there was a process image for the cyclic STEP 5 program only. In a configuration with R processor and CPU 928, there is also a so-called *controller process image* for each controller executing under operating system control. The controller process image is located in the controller data block.
The process image is updated and read out via the I/O modules with the aid of the address lists (also located in the controller DB) and of the I/O addresses required for the controller process image. In this connection, not only the digital/analog modules and the IPC flag area on the coordinator, but also the processor's flag area, is regarded as an I/O area.
The controller process image is updated independently of the STEP 5 process image. When a controller references an I/O address lower than 128, that address need not be specified in DB 1, as is the case when referenced by a STEP 5 program.
A conflict ensues when a controller writes to an output address < 128 for which a STEP 5 process image is also to be generated. This can be avoided by initializing a DB1 in which this address is ***not*** specified. (This is described in more detail in section 5.7). Program file REGR64ST.S5D contains a "blank" DB1 for those who want to use a controller without the STEP 5 program. Transfer of this DB1 to the programmable controller prevents generation of a STEP 5 process image on initiation of a cold restart.

Two different procedures, both of which can be specified via COM REG, are available for updating the controller process image:

* *Common updating of inputs and outputs:*

  The presetting option is a common update of controller inputs and outputs immediately prior to controller processing.
  In this case, the dead time between process input and process output image is invariable, and is identical to the scan time. The dead time can be taken into account when the controller parameters are defined.

* *Separate updating of inputs and outputs:*

  Through selection of the controller behaviour:
  The controller outputs should be updated immediately following controller processing: **YES**

  D 2.6, PIQ = 1, see chapter 7), updating of the controller process image is carried out in two stages, i.e. the inputs are read before the controller is invoked, and the output values are output immediately prior to controller processing.
  This could result in a varying of the dead time between updating of the process input and process output images. The process input image is updated precisely in union with a controller's timing signal, but processing of the interrupted STEP 5 block is completed before the controller is invoked.
  The dead time thus varies between 2.5 ms and the execution time of the longest STEP 5 block. The variation in the dead time is scarcely noticeable when the scan time is appreciably longer than the longest STEP 5 block's execution time (which is frequently the case).
  In conjunction with a number of processors, a variation in the dead time can be intrusive and may result in instability. The option for common updating of inputs and outputs has been provided for this reason.
  Variations in the dead time between input and output can also be eliminated by enabling the interruptability of STEP 5 programs in block DX0 following completion of a statement, should this be feasible.

  Example of a DX0 block:

```
DW0 :   KS = MASKX0
DW3 :   KH = 0601;
DW4 :   KH = 1008;
DW5 :   KH = EEEE;
```

  When DX0 is input as shown in the example above, a STEP 5 program can be interrupted by a controller program or hardware interrupt following execution of a statement (→ programming manual for the relevant processor).

2.5.2 Calling a Controller in a STEP 5 Program

Beginning with release 4 of the S5-DOS basic package, a controller can also be called by a STEP 5 program. *No DB2* is used in this case.

NOTE: ***When a controller is invoked by a STEP 5 program, the number of the controller DB the controller is to work with must be entered in line 'DBR : '.***
Example: 'DR : DB25'

When a controller is called in a STEP 5 program, certain bits must be set in the controller DB prior to initiating a cold or warm restart. (For details on these bits → chapter 7). The controller process image is not updated, as is the case when a DB2 is used, i.e., the STEP 5 program must also supply the controller with process status information.

The following routine must be included in OB20 (the operating system calls this block on every processor cold restart) for each controller DB:

```
      :0   F 0.0
      :ON  F 0.0                SET RTL0
      :C   DBxxx                SELECT CONTROLLER DBxxx
      :S   D 2.0                SET BIT 2.0 IN THE DB (RIN)
      :JU  FB102                CALL CONTROLLER FB102
NAME  :FB102STA                 NAME OF THE CONTROLLER FB
DBR   :    DBxxx                ENTER CONTROLLER DB
```

The following program must be included in OB21 (the block that the OS calls on a manual processor warm restart) and in OB22 (the block that the OS calls on an automatic processor warm restart) for each controller DB:

```
      :0   F 0.0
      :ON  F 0.0                SET RLO
      :C   DBxxx                SELECT CONTROLLER DBxxx
      :S   D 2.3                SET BIT 2.3 IN THE DB (WAB)
      :JU  FB102                CALL CONTROLLER FB102
NAME  :FB102STA                 NAME OF THE CONTROLLER FB
DBR   :    DBxxx                ENTER CONTROLLER DB
```

Calling the controller in OB13 (which the OS processes every 100 ms) will ensure that the *scan time is observed*. Scan times of more than 100 ms can be implemented by programming a counting loop in OB13.
When using point controllers and controllers with mark space output, the interval between execution of two controller programs must correspond to the *minimum pulse duration*.

The scan time and, where applicable, the minimum pulse duration must be specified correctly when initializing the controller DB.

The controller can be called in the following manner:

```
      :C   DBxxx                SELECT CONTROLLER DBxxx
      :L   PW128                READ PERIPHERAL WORD 128
      :T   DW168                ENTER IN CONTROLLER DB AS SETPOINT
      :L   PW130                READ PERIPHERAL WORD 130
      :T   DW169                ENTER IN CONTROLLER DB AS ACTUAL VAL.
      :JU  FB102                CALL CONTROLLER FB102
NAME  :FB102STA                 NAME OF THE CONTROLLER FB
DBR   :    DBxxx                ENTER CONTROLLER DB
      :L   DW208                READ OUT MAN. VAR: FROM CONTROLLER DB
      :T   PW144                OUTPUT TO I/O MODULES
```

Please note the numerical formats of the I/O modules used; format conversions described in section 2.3.1.1 must also be carried out where necessary.

The processor load resulting from the call can be reduced by the programming of counting loops in OB13. For example, two controllers with sampling times of 200 ms can be invoked in such a way that only one of the two will execute every 100 ms. Unlike definition of the time base in DB2, OB13 does not support minimization of the processor load.

2.6 Structuring

It is recommended that compact controllers be structured as shown in Figure 2-4.

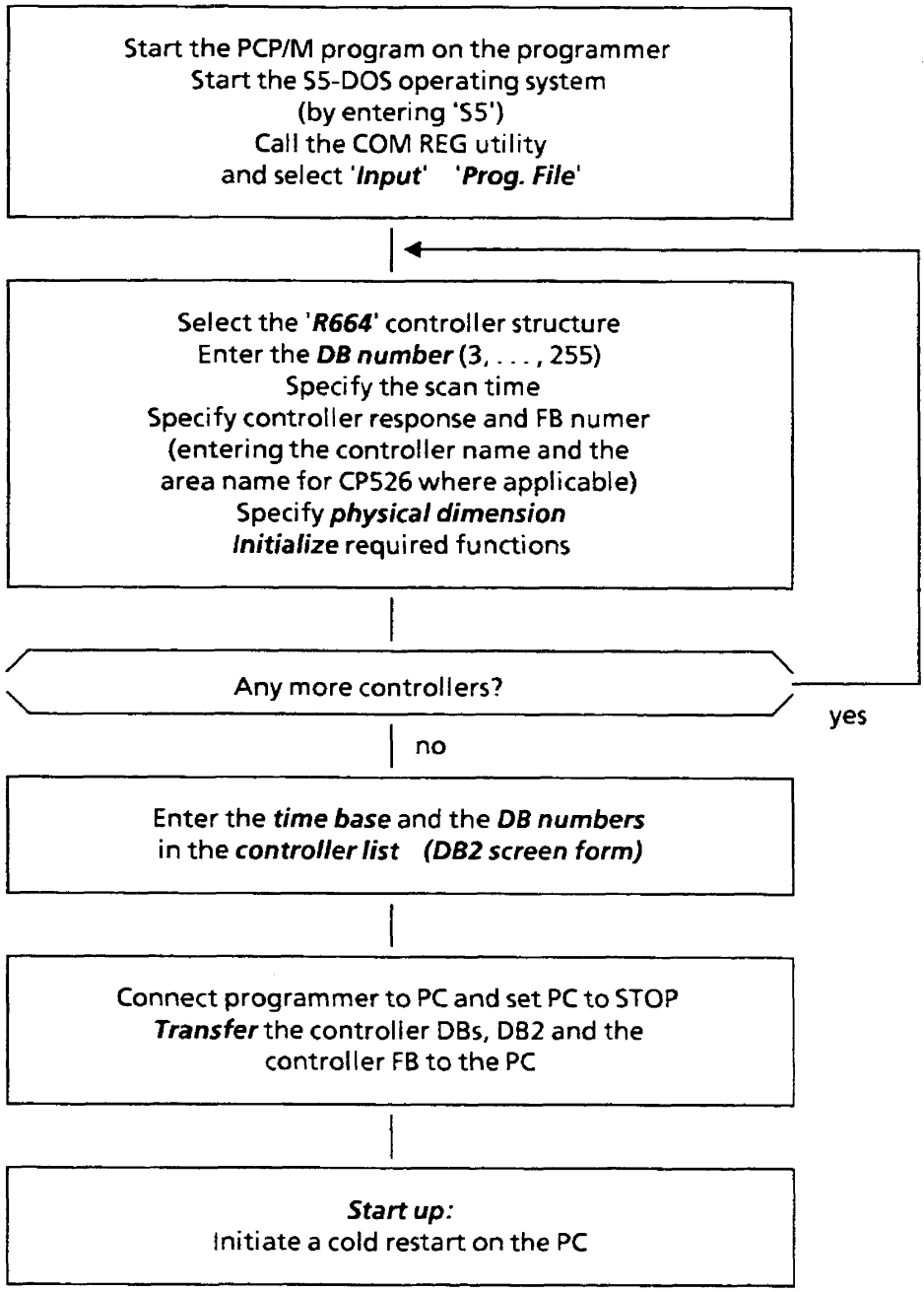


Figure 2-4: Configuring procedure

2.7 Entering and Modifying Input Variables

2.7.1 Parameters that Cannot be Modified during Operation

The *scan time* and *minimum pulse duration* parameters cannot be modified during operation.

Note: When the scan time or minimum pulse duration is changed, the associated controller must be deleted from DB2 and then reentered (and a cold restart subsequently initiated).

The *structuring switch* in data words 23 and 24 may not be changed either. The only exception is structuring switch S8 (ADC setpoint, presetting D 23.7), which can be modified during operation (via a STEP 5 program only; COM REG does not permit modification of the S8 switch).

2.7.2 Parameters that Can be Modified during Operation

The controller program converts the user-specified parameters (e.g. PID controller parameters K_p, T_N, T_V) into a numerical format which is not identical to the input format. In addition, arithmetic conversions are also required in some cases. For example, user parameters T_N and T_V are converted on the basis of scan time T_A into module parameters $T_I = T_A/T_N$ and $T_D = T_V/T_A$ when the controller in question is a PID controller.

These conversions are carried out on a cold restart.

The user must inform the controller of his intention to modify a parameter during operation by setting an *operator control bit*.

The modifiable parameters (which are located in DW25 to DW105 in the controller DB) are divided into two groups, each of which has its own operator control bit.

The time required for parameter conversion is divided between the two groups.

- Operator control bit *BED1* (D 7.8) must be set to modify a parameter located in DW25 - DW55.
- Operator control bit *BED2* (D 7.9) must be set to modify a parameter located in DW56 - DW105.

When a parameter is modified (forced) over COM REG, the associated control bit is set automatically.

The relevant operator control bit must be set when a parameter is modified via a STEP 5 program or the STEP 5 programmer software. When an operator control bit is set, the controller converts the parameters but does not compute the output values (e.g. regulating variable).

To ensure that continual setting of an operator control bit in a STEP 5 program will not suppress computation of the output values altogether, the controller must be called once with a JU FB102 (without supplying input or output values) when an operator control bit has been set. When called with JU FB102, the controller converts the parameters.

The controller then resets the operator control bit.

The operator control bits have different effects on different types of controllers:

a) *Controllers with continuous output signal*

Processing of the controller program is arrested for one cycle when only one of the two operator control bits (BED1 or BED2) is set, and the parameters allocated to that bit are converted instead.

When both BED1 and BED2 are set, the controller program executes once following conversion of the parameters allocated to BED1 before BED2 takes effect, thus preventing deferment of the program cycle twice in succession.

b) *Controllers with two-step or three-step output signal*

Point controllers and PID controllers with mark space output require no special provisions for a situation in which both BED1 and BED2 are set (= 1), the reason being that the minimum pulse duration (T_{min}) of the relevant pulse output module, rather than scan time T_A, is decisive for invoking them.

T_A should be defined as n * T_{min} (where n ≥ 1), as the regulating variable computed by the controller - converted into pulse duration - must be output in a smaller time grid. The complete controller program is thus actually executed in the scan time's (T_A) cycle.
Program calls between two sampling periods serve only to update the relevant pulse output. Since the pulse generation program needs only little time to execute, enough time is still left to convert the parameters allocated to one operator control bit.
If the entire controller program must execute, it is delayed by one controller call to enable conversion of the parameters allocated to BED1 or BED2.

2.7.3 Process Image Input Variables

The following mechanism is used for ADC inputs and relays (DW180):

The operating system automatically updates the process image. When an I/O address of an input is allocated an address value (→ chapter 4 for details on the numerical representation of addresses), a value is read from the relevant I/O module (or from a flag or interprocessor communication flag) and entered in the corresponding process image (a data word in the controller DB). The controller then evaluates this data word. The controller DB must therefore not be supplied with these values by a STEP 5 program.

A value can be written to a process image by a STEP 5 program or modified with the programmer using the 'Force' function (e.g. the relays using COM REG). In this case, however, the associated I/O address cannot be assigned a value. When using COM REG, blanks must be specified when the I/O address is entered (blanks, not a zero, as zero would be interpreted as I/O address 0). The operating system then no longer updates the process image in the controller DB.

The I/O addresses may be modified during operation (using a control program, not COM REG). It is therefore possible to disable and enable automatic process image updating.

NOTE When COM REG is used to disable an ADC input (structuring switches S8,S11, S13 and S12), the corresponding I/O address is not deleted, i.e. the I/O module will still be addressed.
If it is necessary to prevent this, the address must be deleted when the input is disabled (by the overwriting of it with spaces) before the structuring switch is modified.

When using an analog input module with an effective range of 4 to 20 mA, you can choose controller-initiated format adjustment when specifying the controller behaviour (D 24.6 = 1); 256 units are subtracted from the input value when you use a 463 module, 512 units when you use a 465 or 460 module. This adjustment operation is carried out at all ADC inputs. (Although the bit cannot be set with COM REG release A02, it can be set using the STEP 5 program software.)

2.8 Output Variables

2.8.1 Measured Values

COM REG's TEST function displays a controller's measured values (see the structure diagram). These can be read out and processed further using a STEP 5 program (DW232 to DW243). Attention must be paid to the numerical format (with dimension or as percentage → 4 and 7).

2.8.2 Process Image Output Variables

The same mechanism for automatic process image updating is used for the DAC outputs and output bits (DW220) as is used for the process input image:

The controller stores the output value in a data word in the controller DB. If the I/O address of an output is allocated an address value (for details on the numerical representation of addresses → chapter 4), the operating system reads the value from the relevant data word in the controller DB and outputs it to the I/O module (or flag or interprocessor communication flag).

If a value in the process image is to be evaluated by a STEP 5 program only (or not at all), the associated I/O address may not be assigned a value. When using COM REG, spaces must be entered when the I/O address is input (spaces, not a zero, as a zero would be interpreted as I/O address 0). The operating system then no longer addresses an I/O module for this output.

The I/O addresses may be modified during operation (using a control program; not COM REG). It is thus possible to enable and disable automatic process image updating.

NOTE: When COM REG is used to disable a DAC output (structuring switches S3, S15, S16, S19, S20) the corresponding I/O address is not deleted, i.e. the I/O modules will still be addressed.
If this is not acceptable, the address must be deleted (overwritten with spaces) when the output is disabled before the structuring switch is changed.

2.9 Interconnecting Inputs and Outputs

A controller's output value can also be the input value for another controller.

This is done by entering the value at a DAC output into a flag word (when the controller updates the process image). When another controller updates its process image, it can read in this value over an ADC.

A STEP 5 program can also establish this 'link' by reading a process output image and writing it to the required process input image. Format adjustment is unnecessary, as all values are in ADC format (→ chapter 4).

Measured values can be supplied to a controller as input values in one of two ways:

- The simplest way is to connect a test socket at the measuring point. The controller then converts the value into ADC format and enters it in the process image of either DAC 3 (DW218) or DAC 4 (DW219).

  A STEP 5 program can read it out from there (in the controller DB).

  By specifiying a flag word when initializing the test socket with COM REG, you can transfer the value to a flag word during process image updating and read it out from there.

- TAnother way is a direct readout of the measured value (DW232 to DW243). In this case attention must be paid to the data format when the value is to be evaluated and processed further (with dimension or as percentage, → chapter 4).

2.10 Effects of the Various Startup Modes

2.10.1 Processor Cold Restart

On a cold restart, the start routine first sets data words DW208 to DW511 in all controller DBs to zero, thus deleting all internal historical values of the controller, as well as the process output image and the measuring points.

The parameters are then converted.

The actual processing of the controller program then begins, including evaluation of the first process input image.

2.10.2 Processor Warm Restart

As far as controller program processing is concerned, there is no difference between a *manual* and an *automatic warm restart*. On the basis of deviation e = w - x, the following *restart criterion* is checked in both cases:

$$|e_{new} - e_{old}| < 0.25 * |w| .$$

If this condition is true, it is assumed that the interruption was of such short duration that actual value x has not yet deviated much from setpoint w, and that it is therefore possible to continue using the old historical values and the old regulating variable for subsequent computations.

If the condition is not fulfilled, all historical values are deleted and the control system is started as if a cold restart has been initiated; the regulating variable is zero.

2.10.3 Restarting a Controller (Cold Restart)

A single controller can be stopped using the *special* COM REG *function 'Controller processing'* (D 7.0 = 0 in the controller DB).

When the controller is switched back on (D 7.0 = 1), it is started in the same way as for a processor warm restart.

Bit D 7.0 can be used only to stop or start a controller that was entered in DB2 when the processor cold restart was initiated.

2.11 Transition from RUN --> STOP

When initializing the controllers, you must specify whether the I/O channels are to be set to 0 (D 2.7) when the PC goes from the RUN to the STOP mode, or when one controller is switched off (→ 2.10.3).

When using COM REG, you must specify this when defining the controller behaviour.

This procedure refers only to the controller outputs declared in the output address list in the controller DB.

If the function is disabled with 'NO' (D 2.7 = 0), the relevant I/O port retains its current state.

2.12 Operator-process Communication and Process Visualization Using Communications Processors

In addition to the COM REG utility, many other sophisticated programs are available for operator control and monitoring of the R64 controller structure using ***standard displays*** and communications processors (CPs). These programs can output graphic displays or printouts of essential values from one or more control loops. Data display terminals equipped with a process keyboard can be used to enter and display parameters and input variables.

The standard software blocks for closed-loop control and communication (SRK) provide the following facilities.

- *Local operator-process communication and process visualization*
  (Local Operator System, LOS)

- *Centralized operator-processor communication and process visualization*
  for up to eight programmable controllers (Master Operator System, MOS)

- A *message control system* for standard CPs.

2.12.1 Local Operator System

Together with the data handling blocks, the ***SRK-LOS*** software package serves as the interface between processor and communications processor. Only the first slot in the programmable controller is available for this purpose, i.e. each PC may be equipped with only one processor.
COM SRK is the comparable utility for the programmer. The following displays are available:

Loop Display

A loop display enables the user to control and monitor a single controller. The following functions have been implemented:

- Setpoint, actual value and regulating variable are displayed in the form of numbers and colored bars.

- The following limits are displayed when exceeded:

  - Upper danger limit
  - Upper warning limit
  - Lower danger limit
  - Lower warning limit
  - Upper setpoint limit
  - Lower setpoint limit
  - Upper limit for the regulating variable
  - Lower limit for the regulating variable

- The following parameters are displayed and may be modified:
  - Proportional value K_p
  - Proportional gain R
  - Integral-action time T_N
  - Derivative-action time T_V
  - Actuating time TM (point controller)
  - Upper threshold for the dead band (point controller)
  - Lower threshold for the dead band (point controller)

- The following modes are available:

  - Auto mode
  - Manual mode, value input from keyboard or ADC
  - Internal mode (setpoint input from keyboard, controller setpoint)
  - External mode (setpoint input via ADC)

- The controller can be switched off and on.

Group Display

A group display enables the user to monitor the setpoints, actual values and regulating variables, as well as the modes, of up to six controllers: all values can be displayed as numbers and/or bars.

The user may specifiy the mode, setpoint and manual input value for any controller in the group.

Plant Display

The plant display is the highest-level display. Proceeding from this display, the user may select as many as eight group displays. It is thus possible to control and monitor as many as 48 controllers (eight groups of six control loops each).

Required Hardware Configuration

The following components are required in addition to those necessary for putting the R64 controller structure into operation (i.e. programmer, programmable controller, I/O modules: → 2.3.1):

- A CP526 communications processor
- A color monitor
- A process keyboard
- Cable for connecting the process keyboard to the CP

Required Software Configuration

The following software packages are required in addition to those necessary for starting up the R64 controller structure (i.e., PCP/M, S5-DOS, COM REG, R64 controller structure: → 2.3.2):

- COM 526
- Data handling blocks for communications processors
- COM SRK utility (programmer software)
- SRK-LOS software package (PC software)

2.12.2 Master Operator System

Controllers in up to eight PCs can be serviced and monitored using the *SRK-MOS* software package. The PCs are interfaced over the SINEC H1 local area network in conjunction with a CP 535 communications processor.
The scope of the SRK-MOS software (loop, group and plant displays) is identical to that of the SRK-LOS software, but includes an option for showing as many as eight plant displays, i.e. up to 384 controllers in eight PCs, in a single overview display.

2.13 Programming EPROMs

COM REG cannot be used to write or blow blocks into *EPROM*.

The S5-DOS *EPROM/EEPROM* package, beginning with release 4, can be used for this purpose.

Note the following when using the PG 670 / PG 675 operating system to transfer blocks to EPROMs:

> Because the function block is written in assembly language, it must begin at a paragraph address (this means that the FB header must begin at address XXX0H or XXX8H, i.e. the FB itself starts at address XXX5H or XXXDH). The starting address of the EPROM (the address at which the first block transferred is stored) is a paragraph address.
>
> The controller FB (FB102) must be blown as first block to an empty EPROM.
>
> Failure to do so, i.e. failure to transfer the FB first to make sure that it starts at a paragraph address, will cause the R processor to have an undefined state following a cold restart.

On the first cold restart, controller DBs stored in **EPROM** must be copied to DB RAM following an Overall Reset. Because the controller DBs are initialized before the restart OB executes, they must be reinitialized after they have been transferred to RAM.

The following sample program (which is in OB20) copies controller DBs 3 and 4 to RAM and initializes them only when they are located in EPROM. The program can be easily modified for one or more controller DBs.

OB20

Network 1

```
0000       :C   DB3
0001       :L   DW210          Check to see if DB3 is in
0002       :L   KF+1           the EPROM
0004       :+F                 (by attempting to write and
0005       :T   DW210          then read out a word)
0006       :L   DW210
0007       :!=F
0008       :BEC                Block end when DB is in RAM

0009       :O   F 0.0
000A       :ON  F 0.0          Set RLO

000B       :L   KF+3
000D       :JU  OB255          Copy DB3 to RAM
000E       :C   DB3
000F       :S   D 2.0          Set initialization bits for
0011       :S   D 7.1          cold restart in DB3
0013       :JU  FB102          Initialize controller
0014 NAME  :FB102STA
0015 DBR   :    DB3

0016       :L   KF+4
0018       :JU  OB255          Copy DB4 to RAM
0019       :C   DB4
001A       :S   D 2.0          Set initialization bits for
001C       :S   D 7.1          cold restart in DB4
001E       :JU  FB102          Initialize controller
001F NAME  :FB102STA
0020 DBR   :    DB4
0021       :BE
```

Blocks DB1 and DB2 need not be copied to RAM.

3 Functional Description

When you read this section, place the *structure diagram* of the controller (→ 11) next to the text. Before going into the various controller branches, note the following in regard to input of a DB using COM REG:

> Before entering a controller DB, it is necessary to specify the positions of the structuring *switches* (DW23 and DW24). With the exception of S8, D23.7 these switches cannot be modified once a cold restart has been initiated.
>
> The *relays* (DW180) can be modified only by use of the COM REG test function.

As the switches are numbered in the structure diagram but referred to by name in COM REG (e.g. S5 = validity check), both designations have been given in the text.

A physical dimension may be specified when parameters are entered in the setpoint and actual value branches (→chapter 4, Numerical Formats). This consists of a unit (up to six characters, e.g. DEG_C) and allocation of the values read in on the ADC as a percentage of the conversion range. For example, the value 20 DEG_C can be allocated to 0% on the ADC, the value 1200 DEG_C to 100% on the ADC (= nominal range). The value assigned to 0% may be a negative quantity. The highest value that may be entered is 10000, and may have a decimal point at any location (e.g. 10.000 or 1000.0). The number of decimal places in the 0 percent and in the 100 percent value must be identical.

3.1 Setpoint Branch

The setpoint for measuring point MP 1 may be input as follows:

Over **ADC 1** (→2.7.3)	ADC/entry (S8, D 23.7)	= 0
Over the **control setpoint** This value may be entered from the programmer or via a STEP 5 program	ADC/entry (S8, D 23.7)	= 1
	Control setp./setpoint string (D 180.5 can be modified with COM REG in Test mode only)	= 0
Over the **setpoint string**	ADC/entry (S8, D 23.7)	= 1
	Control setp./setpoint string (D 180.5 can be modified with COM REG in Test mode only)	= 1

The setpoint at measuring point MP 1 can be processed by the **ramp-function generator** (S9, D 23.8 = 1) and the **filter** (S10, D 23.9 = 1).

The setpoint at measuring point MP 2 is monitored by a **limit monitor**. D 255.5 = 1 indicates that the upper setpoint limit has been exceeded, D 255.4 = 1 that the setpoint has violated the lower setpoint limit. The setpoint itself is not limited.

The setpoint used for computation of the deviation can be locked at zero percent (setpoint disable, D 180.6 = 1).

3.1.1 Setpoint String

A periodic, stepped or ramp-shaped setpoint profile can be generated from up to ten specifiable setpoint values.

Stepped Profile (D 24.4 = 0)

The output variable is stabilized at a setpoint for the duration of the interval, and goes to the next setpoint when the interval is up. The procedure begins again at the first setpoint when the interval for the last setpoint expires.

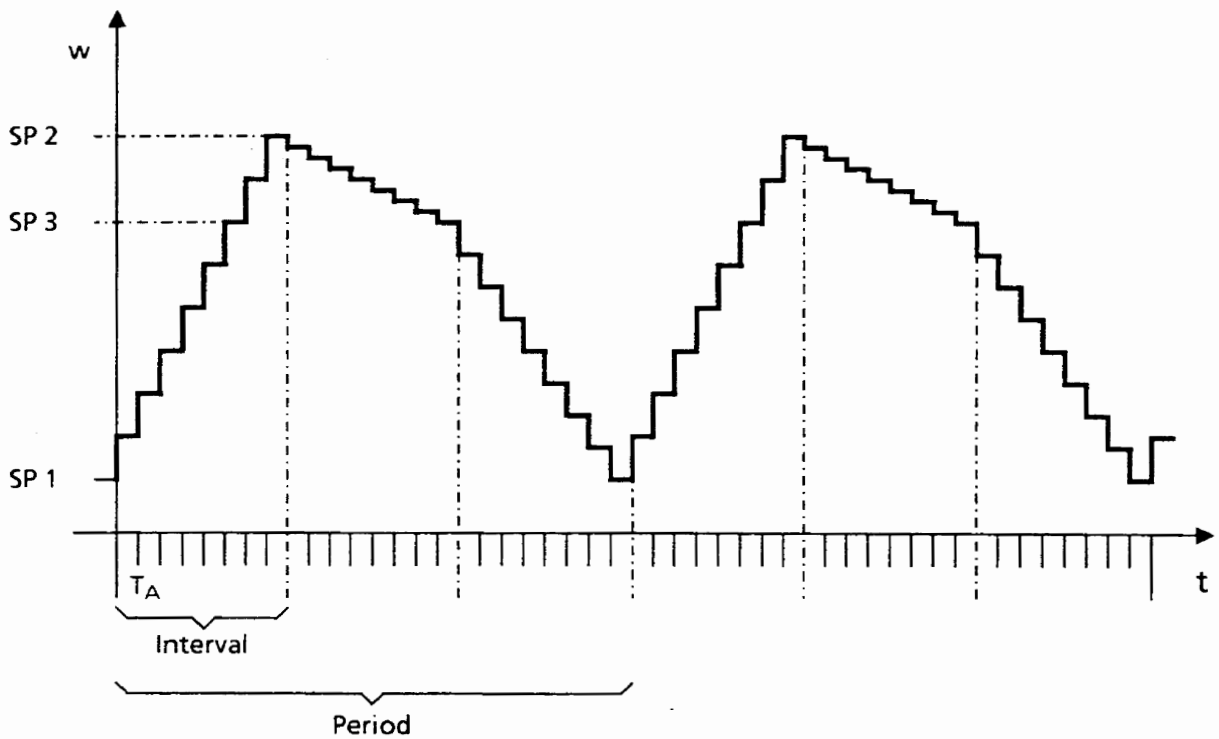


Figure 3-1: Stepped setpoint profile
Example with three setpoint values

Linear Interpolation (24.4 = 1)

A ramp-shaped setpoint profile is generated.
The slope is computed from:
(preceding setpoint - next setpoint) / interval.
When the last setpoint is reached, linear interpolation takes place between the last and the first setpoints and the period is subsequently resumed.

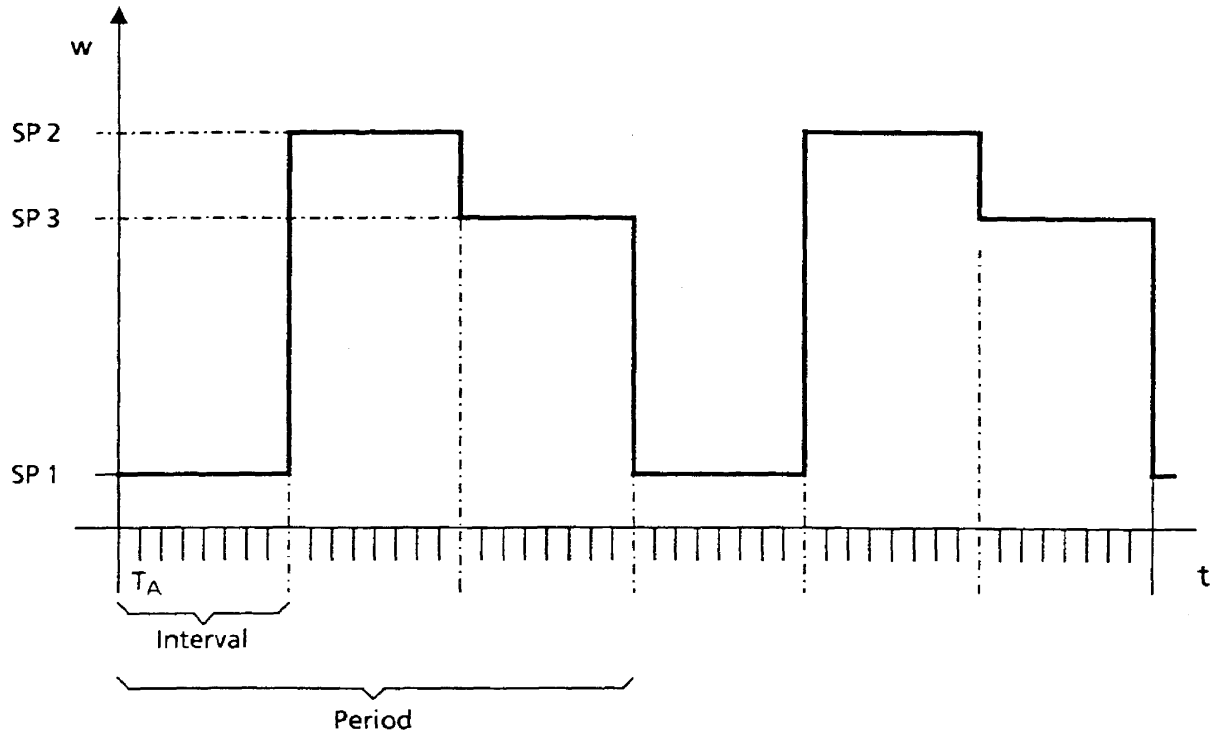


Figure 3-2: Linear interpolation
Example with three setpoints

A change from/to stepped/linear does not take effect until the next cold restart.

When you switch from control setpoint to setpoint string (D 180.5 = 1), the setpoint string begins at instant zero of the preceding period, as on a cold restart.

The setpoint string is independent of the controller mode, e.g., manual mode, controller disable.

3.1.2 Ramp-Function Generator

On a step change in the setpoint at the input, the ramp-function generator generates a *ramp-shaped output signal* which approaches the required setpoint.

A positive slope (away from zero) is computed with the formula:

Increase * *ramp-up time*

A negative slope (toward zero) is computed with the formula:

Increase * *ramp-down time*

Figure 3-3 shows the slope produced by the ramp-up time, ramp-down time and increase parameters.

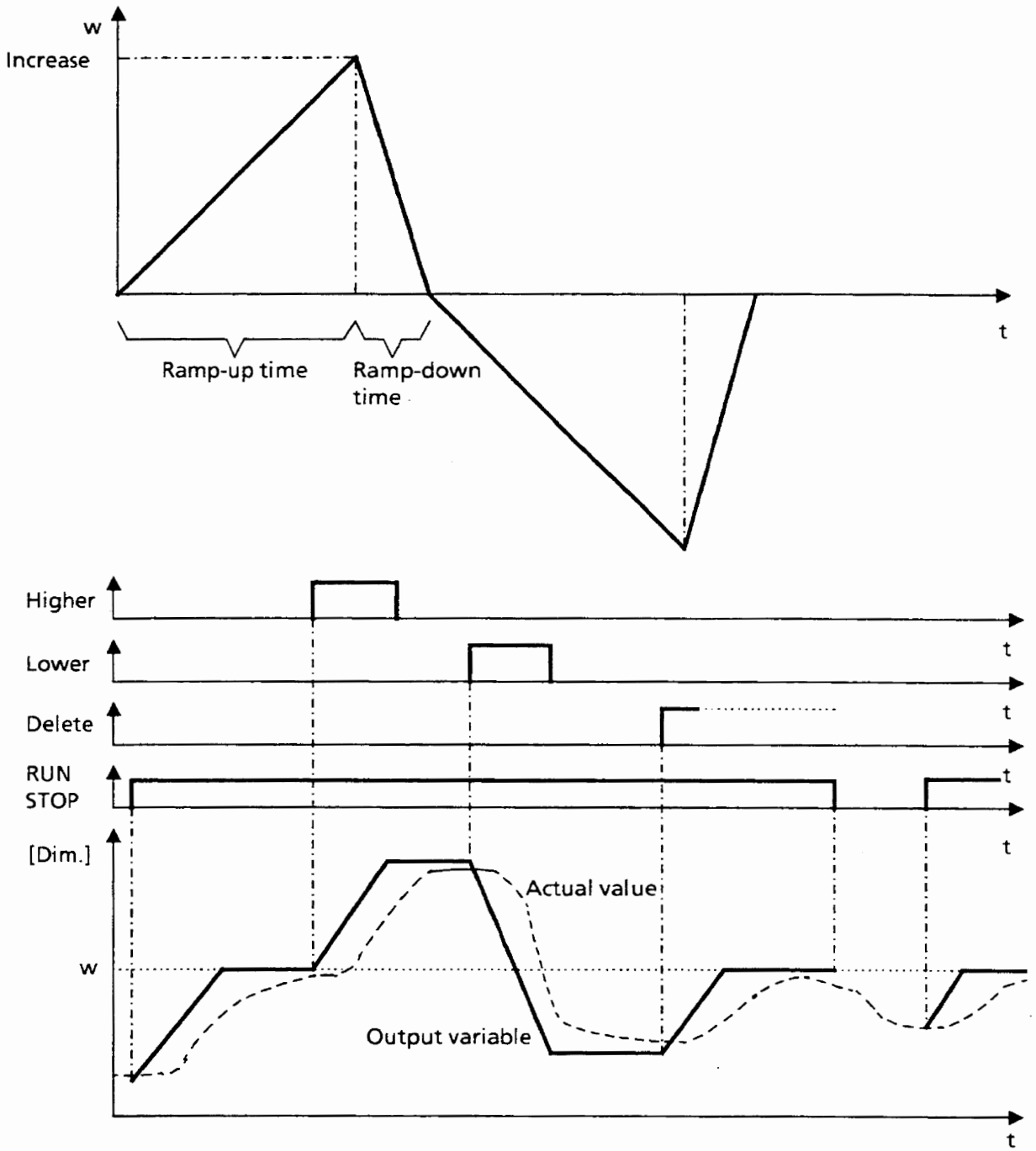


Figure 3-3: Slope generated by the ramp-function generator and response in setpoint Adjustment mode and on startup (STOP = = > RUN. See allowance for actual value)

Allowance for Actual Value

The ramp-function generator (and the filter or smoothing element in the setpoint branch) starts the output variable at the current actual value (i.e. at measuring point MP 9) on a cold restart, a warm restart, on a controller enable (special COM REG function "Controller processing") and when a switch is made to one of the following modes:

- Disable controller (D 180.9 = 1 for point controller, D 180.15 = 1 for PID controller)

- Constant regulating variable (D 180.13 = 1 for PID controller)
- Manual mode (D 180.11 = 1).

The controller then begins with the small deviation prescribed by the slope (and the filter time constant), thus ensuring a specifiable modification of the regulating variable.
The display at measuring point MP2 is somewhat higher (or lower) than the value at measuring point MP9 in this case. The difference corresponds to the increase of the value initialized by the slope or smoothing element time constant, one such increase being permissible per sampling period. The deviation with which the controller begins to control when switched to auto mode is thus shown at measuring point MP3.

Setpoint Adjustment

The output variable generated by the ramp-function generator can be increased in accordance with the slope (up to the maximum setpoint + 100%) by setting the *higher bit* (D 180.2).

Conversely, the output variable can be decremented in accordance with the slope (up to the minimum setpoint -100%) by setting the *lower bit* (D 180.1).

The current output value is retained when the higher or lower bit is reset or when both are set simultaneously.

The *delete bit* (D 180.0 = 1) resets the ramp-function generator's setpoint adjustment mode; the generator's output variable is adjusted to the input variable in accordance with the initialized slope.

The higher or lower bits have priority over the delete bit.

The ramp-function generator's setpoint adjustment mode is reset on a cold restart (as when the delete bit is set).

Allowance is made for the actual value in setpoint adjustment mode (see above). The ramp then rises to the preceding setpoint.

3.1.3 Smoothing in the Setpoint Branch

A first-order time-delay element (VZ1) is simulated with the transfer function

$$\frac{Xa(s)}{Xe(s)} = \frac{1}{1 + TG * s}$$

Filter time constant TG is specified in DD56.

The formula for computing the output variable is

$$Xa_k = \frac{2*TG - T_A}{2*TG + T_A} (Xa_{k-1} - Xe_k) + Xe_k$$

The same allowance is made for the actual value during smoothing in the setpoint branch as by the ramp-function generator (→ 3.1.2).

Input Variables in the Setpoint Branch

Designation	Address	Format	Setting range
ADC/entry	D 23.7	Bit	0/1
Control setp./setpoint string	D 180.5	Bit	0/1
Control setpoint	DW191	With dim.	-100% to + 100%
ADC 1 (address)	DW120	Address	
ADC 1 (value)	DW168	Analog	-100% to + 100%
No. of setpoints	DW38	Non-dim.	1 - 10
Setpoint 1	DW41	With dim.	-100% to + 100%
:	:	:	:
:	:	:	:
Setpoint 10	DW50	With dim.	-100% to + 100%
Interval	DD39	Timer	T_A - 59.59 hhmm n * scan time, n = 1, 2, ...32767
Stepped/linear	D 24.4	Bit	0/1
Ramp-function generator	D 23.8	Bit	0/1
Ramp-up time	DD51	Timer	T_A ... 59h 59min
Ramp-down time	DD53	Timer	T_A ... 59h 59min
Increase	DW55	With dim.	0... + 100%
Higher bit	D 180.2	Bit	0/1
Lower bit	D 180.1	Bit	0/1
Delete bit	D 180.0	Bit	0/1
Filter setpoint	D 23.9	Bit	0/1
Filter time constant	DD56	Timer	$T_A/2$ ... 59.59hhmm
Upper limit for setpoint	DW98	With dim.	-100% to + 100%
Lower limit for setpoint	DW99	With dim.	-100% to + 100%
Disable setpoint	DW 180.6	Bit	0/1

Output Variables in the Setpoint Branch

Designation	Address	Format	Output range
Measuring point MP 1	DW232	With dim.	-100% to + 100%
Measuring point MP 2	DW233	With dim.	-100% to + 100%
Upper setpoint limiting value violation	D 255.5	Bit	0/1
Lower setpoint limiting value violation	D 255.4	Bit	0/1

3.2 Actual Value Branch

The actual value for measuring point MP 8 is read in via **ADC 2** (→ 2.7.3). The actual value can be processed by the **validity check** (S5, D 23.4 = 1), the **filter** (S22, D 24.5 = 1) or the **polygon curve generator** (S7, D 23.6 = 1).

A switch can be made to the **initial actual value** prior to measuring point MP 9 (initial enable, D 180.7 = 1).

A **limit monitor** monitors the actual value at measuring point MP 9; D 255.3 is set to 1 to indicate an upper warning limit violation; D 255.2 is set to 1 to flag a lower warning limit violation; D 255.1 is set to 1 when the upper danger limit is exceeded, and D 255.0 is set to 1 when the value violates the lower danger limit. The value itself is not limited.

3.2.1 Validity Check

Interferences which cause the deviation between two consecutive sampling values to exceed a specifiable maximum value are suppressed.

Actual output variable Xa_k is computed as follows:

$$Xa_k = Xe_k \qquad \text{for} \mid Xe_k - Xa_{k-1} \mid \leq \text{max. perm. difference}$$

$$Xa_k = Xa_{k-1} + \frac{(Xa_{k-1} - Xa_{k-2}) + (Xa_{k-2} - Xa_{k-3})}{2} \qquad \text{for} \mid Xe_k - Xa_{k-1} \mid > \text{max. perm. difference}$$

If, however, a deviation exceeding the permissible maximum persists for more than three sampling periods, the input variable is regarded as being valid and the output variable is brought into line with the input value by applying the power function.

The output variable is set to the value of the input variable when the deviation between the two once again falls below the maximum permissible limit.

3.2.2 Filter in the Actual Value Branch

Filtering in the actual value branch is implemented in accordance with the same formula used in the setpoint branch. The filter time constant is specified in DD104.

Filtering or smoothing in the actual value branch is not possible when you are using COM REG release A02, but an averaging function can be enabled instead (D 23.5 = 1). The following special situation applies when bit D 23.5 is set to 1 (and D 24.5 to 0):

- The filter time constant (DD104) is not evaluated. Instead, filtering is implemented with time constant 8 * scan time.

No allowance is made for the actual value (as in the setpoint branch) in connection with the filter in the actual value branch.

3.2.3 Polygon Curve Generator

A function value Xa = f (Xe) is computed via *linear interpolation* for an input value (abscissa) Xe from interval X (1) ≤ Xe ≤ X (N), where N = 1, 2 ..., 10.

The function is defined by N = 1, ..., 10 value pairs { X (I), f (I) } (N = number of interpolation points). The N abscissa values X (1), ..., X (N) are specified by abscissa value X (1) of the first interpolation point (initial value) and distance DX between the interpolation points.

The abscissa values are *equidistant.*

The associated ordinate values must be entered in the order f (1), f (2), ..., f (N).

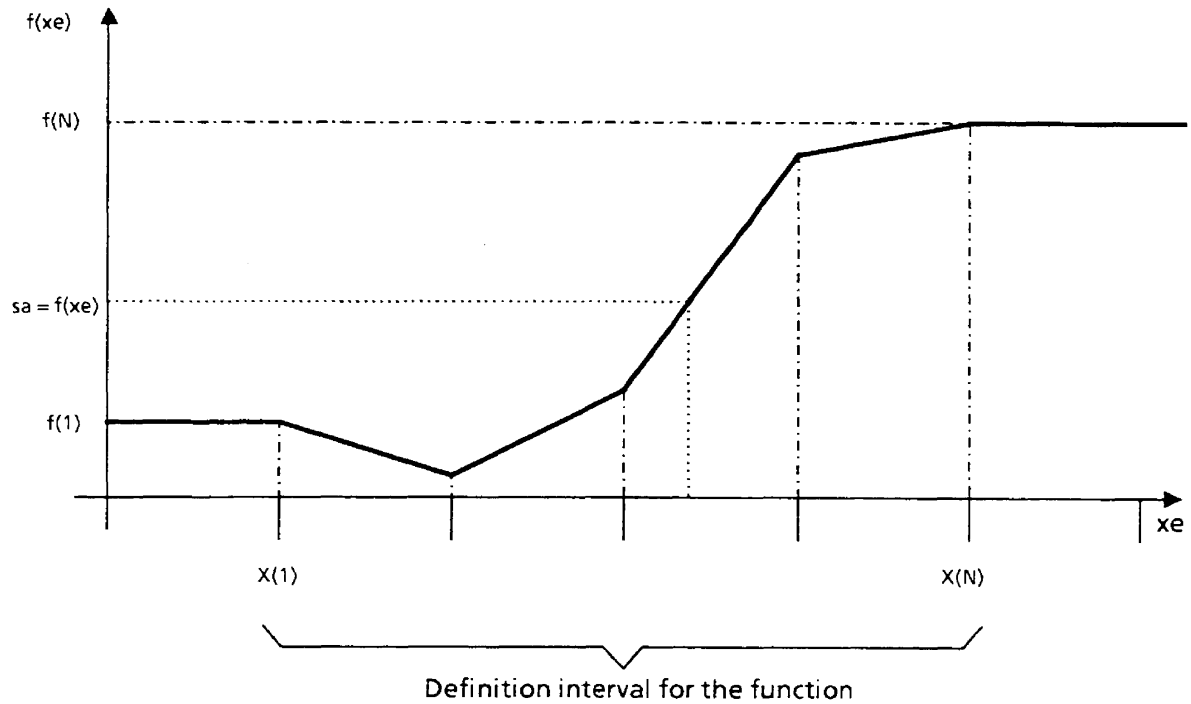


Figure 3-4: Function curve generated by the polygon generator (xe = input value, xa = output value)

The function value is defined outside the definition interval as follows:

Xa = f (1) for Xe < X (1)
Xa = f (N) for Xe > X (N)

Example of Generating a Polygon Curve

In this example, the characteristic of a temperature sensor is to be linearized without reference to a real characteristic with precise dimensional specifications.

The user knows the temperature sensor's resistance values at a specific temperature. A characteristic of the voltage versus temperature results when constant current is applied to the sensor. The injected current must be chosen so that it produces the input module's maximum voltage (e.g. 10V) at the maximum temperature (= at the maximum resistance). An analog input module converts this voltage value into a digital percentage value. The characteristic curve derived from these computed percentage values and the associated temperature values must then be plotted (Figure 3-5).

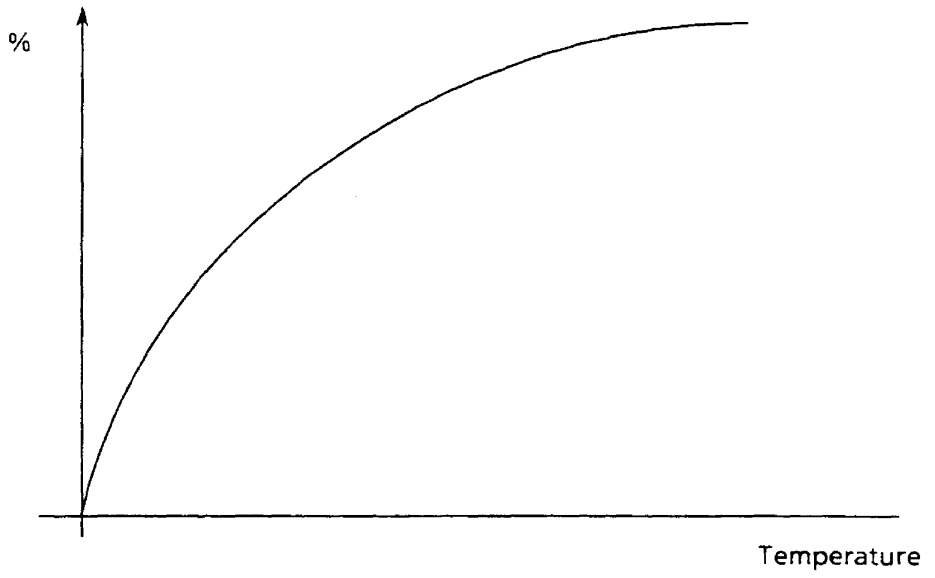


Figure 3-5: Sensor characteristic (percentage versus degrees)

The linearized characteristic resulting from the specification of zero percent and one hundred percent values via COM REG is plotted in Figure 3-6.

The initial value of the X (1) abscissa values must be specified next. The input range of the polygon curve is determined by specifying the number of interpolation points and the distance between these points (Figure 3-6).

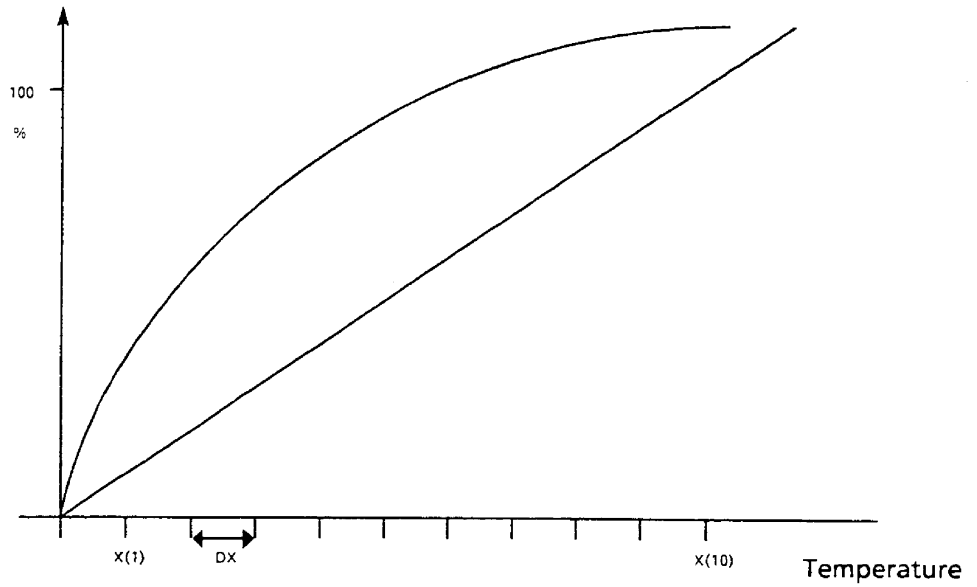


Figure 3-6: Sensor characteristic and linearized characteristic

The output value to be allocated to an input value is ascertained as follows:

Determine the percentage value associated with an input value on the linearized curve.
The temperature value belonging to this percentage value on the non-linearized curve is the corresponding output value of the polygon curve.

In Figure 3-7, output value f (9) is allocated to input value X (9).

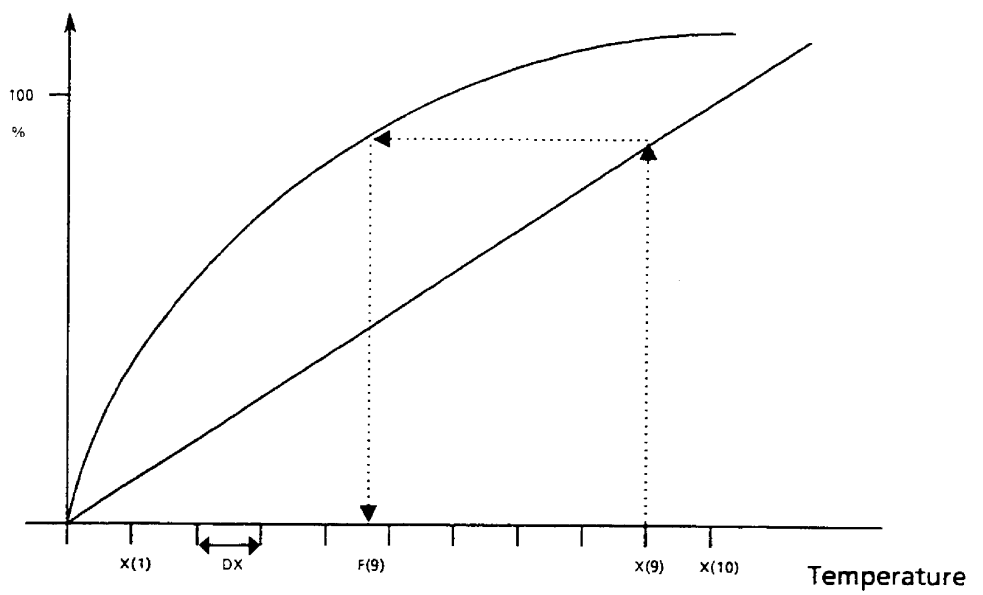


Figure 3-7: Allocating output values to input values

Input Variables in the Actual Value Branch

Designation	Address	Format	Output range
ADC 2 (address) ADC 2 (value)	DW121 DW169	Address Analog	 -200% to +200%
Validity check Max. perm. deviation	D 23.4 DW58	Bit With dim.	0/1 0 to +100%
Actual value filtering Filter time constant	D 24.5 DD104	Bit Time	0/1 $T_A/2$ to 32765 * T_A
Polygon curve No. of Interpolation points Initial value Distance between interpolation points Ordinate 01 : : Ordinate 10	D 23.6 DW25 DW26 DW27 DW28 : : DW37	Bit W/o dim. With dim. With dim. With dim. : : With dim.	0/1 1 to 10 -100 % to +100 % +0 % to +100 % -100 % to +100 % -100 % to +100 %
Start enable Initial actual value	D 180.7 DW94	Bit With dim.	0/1 -100 % to +100 %
Upper warning limit Lower warning limit Upper danger limit Lower danger limit	DW100 DW101 DW102 DW103	With dim. With dim. With dim. With dim.	-100% to +100% -100% to +100% -100% to +100% -100% to +100%

Output Variables in the Actual Value Branch

Designation	Address	Format	Output range
Measuring point MP 8 Measuring point MP 9	DW239 DW240	With dim. With dim.	-200% to +200% -200% to +200%
Upper warn. limit viol. Lower warn. limit viol. Upper danger limit viol. Lower danger limit viol.	D 255.3 D 255.2 D 255.1 D 255.0	Bit Bit Bit Bit	0/1 0/1 0/1 0/1

3.3 Controllers

The deviation (limited to ± 200 %) is shown at measuring point MP 3.

Both startup and transfer to auto mode are **bumpless**. A maximum increase in the regulating variable can be attained by using the ramp-function generator or the filter in the setpoint branch (→ 3.1.2, Allowance for Actual Value), or by limiting the increase. The purpose of these measures can prolong the service life of an actuator or adapt to an actuator's speed to prevent dynamic loss, i.e., to improve the control system.

The following controllers may be used:

- A controller with *continuous output signal* (S1, D 23.0 = 0, S3, D 23.2 = 0),
- A controller with *mark space output* signal (pulse duration modulation of the regulating variable, e.g. colling heating in a temperature control system, S1, D23.0 = 0, S3, D23.2 = 1)
- A *point controller* (e.g. for driving an actuator) (S1, D 23.0 = 1).

When a continuous controller or a controller with mark space output is used, the regulating variable is computed with one and the same program at measuring point MP 4.

3.3.1 PID Controller

The equation for a continuous PID controller in the time range

$$y = k_p \left(e + 1/T_N * \int_0^t e(v)\,dv + T_V * e\right),$$

where K_p = Proportional value (controller gain)
T_N = Integral-action time
T_V = Derivative-action time
y = Regulating variable
e = Deviation

is simulated by a PID velocity algorithm with trapezoidal integration.

Particularly when you are using a PI controller, trapezoidal integration enables precise simulation of an anlog controller's integration.

The P component is multiplied with auxiliary gain factor R, thus enabling the P component, for example, to be set to zero.

A time-delay element (delay of the first order) with time delay constant T_A (can be deactivated) is integrated in the D branch.

The regulating variable is computed according to the following formula:

$$y_k = \Sigma \left[K_p * \{ R^* (e_k - e_{k-1}) + T_A/(T_N*2)*(e_k + e_{k-1}) + D_k \} \right]$$

where

$$D_k\ 0.5 * \left[T_V/T_A * \{ (e_k - e_{k-1}) - (e_{k-1} - e_{k-2}) \} + D_{k-1} \right]$$

A controller branch can be deactivated by setting the relevant parameter (R, T_N, T_V) to zero. **T_N = 0** also *deactivates the I component*, a fact which is not obvious from the quotient T_A/T_N.

3.3.1.1 Standard Version of the PID Controller

The *standard controller version* (S2, D 23.1 = 0) has the following properties:

- Auxiliary gain R is not specifiable, and is set to 1.00.
- The controller can be set to *manual mode* (D 180.11 = 1). The regulating variable must then either be entered manually as constant (S13, D 23.12 = 0) or read in via ADC 5 (S13, D23.12 = 1).
- If entered manually from the programmer (S13, D23.12 = 0), this value is matched to the manipulated variable in auto mode (DW95 : DW235). The controller's output thus remains constant after a switch to manual mode. The manual entry made at the programmer can be modified only in manual mode, thus enabling the value to be specified via a manual value adjuster implemented in a STEP 5 program. Using I/O bits for control, the manual value can be ramped up or down in manual mode.

In the example below, input 0.0 is used as manual/auto switch. Input 0.1 = 1 raises the manual input value (and has priority); input 0.2 = 1 lowers it. A progressive incrementation function has been included in the example for modifying the manual value, i.e., the longer input 0.1 remains set, the faster the value increases (the same applies for lowering the value).

```
      :A    I 0.0
      :=    D 180.11        SET MANUAL/AUTO BIT
      :JC  =MANUAL
KLDE  :L    KF+100
      :T    DW253           SET INITIAL DELTA VALUE
      :     BEU             AUTO
HAND  :L    DW253
      :L    DW95
      :A    I 0.1
      :JC   = ON
      :A    I 0.2
      :JC2  =OFF
      :JU   =KLDA
ON    :+F                   "MAN.ON" ACTIVE
```

```
      :JU   =END
OFF   :-F                    "MAN OFF" ACTIVE
END   :T    DW95             LOAD PG MANUAL INPUT VALUE
      :O    I 0.0
      :ON   I 0.0
      :=    D 7.9            SET OF CONTROL BIT
      :JU   FB102            CALL FB102 FOR
NAME  :FB1  02STA            PARAMETER CONVERSION
DBR   :     DBxxx            xxx = CONTROLLER DB NUMBER
      :L    DW253
      :L    KF+10
      :+F                    INCREASE DELTA VALUE
      :T    DW253            (PROGRESSION)
```

If the value read in via the ADC is to serve as the manual input value (S13, D 23.12 = 1), it is not matched to the regulating variable. In this case, the regulating variable is matched to the manual input value as follows after switching from auto to manual mode:

$$A_k = 0.5 * (\text{manual input value} + y_{k-1})$$

The manual input value can be monitored at measuring point MP 10.

When the controller is **switched from manual to automatic mode**, the regulating variable reacts with a step change whose magnitude depends on the deviation applicable at that instant. The deviation is thus adjusted as for a step change in the setpoint.

If the deviation is zero at the instant at which the mode is switched, the controller's output remains unchanged. Transfer is bumpless.

Limiting of the increase in the regulating variable when switching from manual to automatic can be attained by suitable structuring of the ramp generator, the filter, or both, in the setpoint branch. The output values of both the ramp generator and the filter are set to the current actual value at the instant at which the mode change takes place, and rises to the setpoint read in in accordance with the module parameter. The system deviation is thus limited, and the rate at which the regulating variable is modified is correspondingly slower.

The regulating variable itself can be limited. When it reaches the prescribed upper limit, the I-action component is disabled to *prevent integral windup*, and is once again enabled when it contributes to a reduction in the regulating variable.

Overflow flags (D 255.7 and D 220.6 for the upper limit; D 255.6 and D 220.7 for the lower limit) are set when the regulating variable violates the prescribed limits.

The identifiers in data word 220 can be transferred to the I/O module area, flag area or IPC flag area, when the process image is updated (→ 2.8.2).

The controller disable bit (D 180.15 = 1) sets the regulating variable to zero regardless of the states of the remaining controller inputs.

The ramp generator or the filter in the setpoint branch can be used to prevent a step change in the regulating variable when the controller is enabled (→ 3.1.2).

The precision of the controller's integration is limited, and depends on deviation e, scan time T_A, regulating variable y and integral-action time T_N.
Integration is possible when

$$e * T_A / T_N > 3.0518 * 10^{-5} * y.$$

3.3.1.2 Enhanced Version of the PID Controller

In addition to the functions provided by the standard version, the enhanced version (S2, D 23.1 = 1) enables the following:

A *separate input variable* 'xz' (S11, D 23.10 = 1) can be chosen instead of deviation 'e' as input for the controller's D branch.
This variable is read in over ADC 3, and can be monitored at measuring point MP 11.
The D component is computed with the formula

$$D_k = 0.5 * [T_V/T_A * \{ (xz_k - xz_{k-1}) - (xz_{k-1} - xz_{k-2}) \} + D_{k-1}]$$

To counteract a measurable interference 'z' as early as possible, 'z' can be added to the computed regulating variable (S12, D 23.11 = 1 and D 180.14 = 1) along the lines of a feedforward injector of the interference variable. The measured interference must be injected as a negative value, as it is added to the regulating variable without sign inversion. The value is read in via ADC 4 and can be monitored at measuring socket MP 12.

During operation, a change in the regulating variable can be prevented by the interference variable (interference enable, D 180.14 = 0).

The D-action component can be computed with a *time delay* (delay time = scan time corresponding to a real PID controller, D 180.12 = 0) or without a time delay (corresponding to an ideal PID controller, D 180.12 = 1).
The real PID controller's regulating variable reacts to a transient interference in the actual value with smaller step changes than does the regulating variable of the ideal PID controller.

The regulating variable can be stabilized independently of the deviation (D 180.13 = 1). This is necessary, for example, when a cascade control loop is split up.

The *correcting rate*, i.e. the correction increase, may be limited, a feature which is useful for a slow final control element. A negative correcting rate limiting value must be specified as a negative value.

Note the following about the controller behaviour:

The I-action component is not disabled until the limiting value for the manipulated value has been reached. If the I component's increase is greater than the maximum increase for the regulating variable, as defined by the increase limiting value, a higher value may be used internally until the regulating variable reaches its maximum value. The controller might require more time to decrease this internally computed value, however, when sign inversion of the system deviation is involved, i.e. the regulating variable remains at its limiting value longer.

This type of response becomes apparent when the following condition remains true for a longer period of time (e = deviation):

$$e * K_p * T_A/T_N > \text{positive correction increase limiting value}$$

or

$$e * K_p * T_A/T_N < \text{negative correction increase limiting value}$$

Make sure that the value chosen for limiting the correction increase or integral-action time T_N is not too low (in accordance with the formulas shown above).

Figure 3-8 shows the PID controller's fundamental signal flow (enhanced PID controller). For the sake of clarity, the figure shows only the basics rather than a detailed diagram of the implemented structure. The figure does not show how to stabilize the regulating variable.

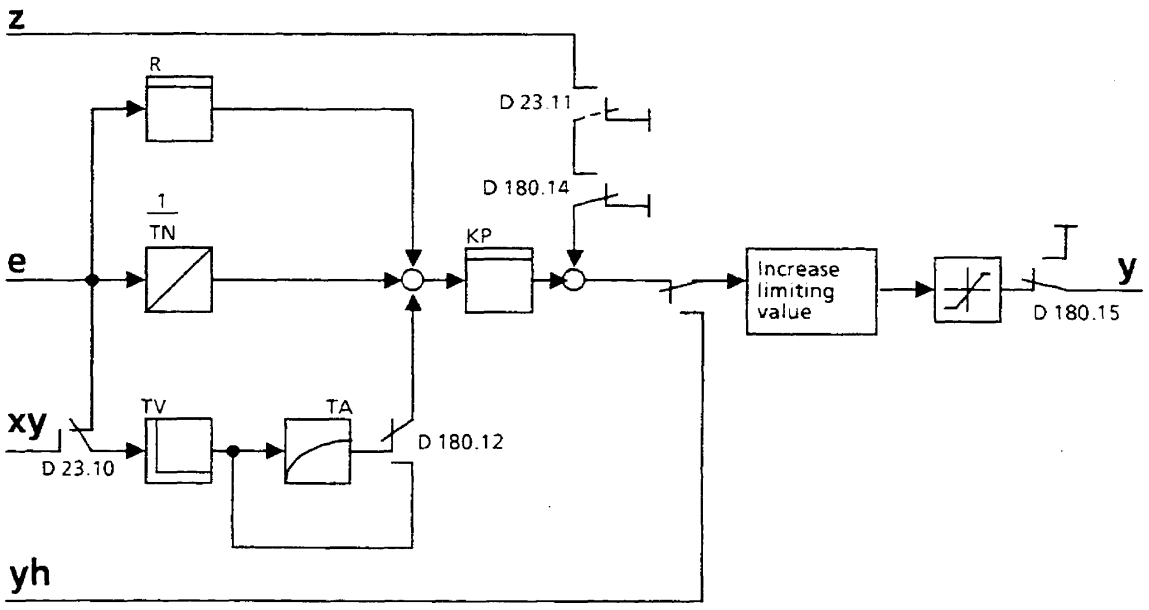


Figure 3-8: The PID controller's signal flow (enhanced version)

3.3.2 Controller with Continuous Output Signal

When final control element or actuator adjustment is required (S14, D 23.13 = 1), the controller's output value (according to the formula a * y + b) is multiplied by a gain factor, and an offset is added to the result.

The result can be monitored at measuring point MP 5, and is output via DAC 1 (→ 2.8.2).

Input Variables for the PID Controller with Continuous Output

Designation	Address	Format	Setting range
Continuous/step	D 23.0	Bit	0/1
Standard/enhanced	D 23.1	Bit	0/1
Sep. D input	D 23.10	Bit	0/1
ADC 3 (address)	DW122	Address	
ADC 3 (value)	DW170	Analog	-100% to + 100%
Interference input	D 23.11	Bit	0/1
ADC 4 (address)	DW123	Address	
ADC 4 (value)	DW171	Analog	-100% to + 100%
Interference enable	D 180.14	Bit	0/1
Manual input PG/ADC	D 23.12	Bit	0/1
ADC 5 (address)	DW124	Address	
ADC 5 (value)	DW172	Analog	-100% to + 100%
Auto/manual mode	D 180.11	Bit	0/1
Disable controller	D 180.15	Bit	0/1
Real/ideal PID controller	D 180.12	Bit	0/1
Mani. variable constant	D 180.13	Bit	0/1
Auxiliary gain	DW74	W/o dim.	-99.99 to + 99.99
Pos. limiting value for correction increase	DW81	%	**0%** to + 100%
Neg. limiting value for correction increase	DW82	%	-100% to **0%**
Upper controller limiting value	DW79	%	-100% to + 100%
Lower controller limiting value	DW80	%	-100% to + 100%
Proportional value	DW73	W/o dim.	-99.99 to + 99.99
Integral-action time	DD75	Time	0 to 59.59 hhmm
Derivative-action time	DD77	Time	0 to 59.59 hhmm
Cont./PDM output	D 23.2	Bit	0/1
Actuator adjustment	D 23.13	Bit	0/1
Gain	DW96	W/o dim.	-99.99 to + 99.99
Offset	DW97	%	-100% to + 100%
DAC 1 (address)	DW144	Address	

Output Variables of the PID Controller with Continuous Output

Designation	Address	Format	Output range
DAC 1 (value)	DW208	Analog	-200 % to + 200 %
Overflow flag for upper limit	D 255.7 D 220.6	Bit Bit	0/1 0/1
Overflow flag for lower limit	D 255.6 D 220.7	Bit Bit	0/1 0/1
Measuring point MP 3	DW234	With dim.	-200% to + 200%
Measuring point MP 4	DW235	%	-100% to + 100%
Measuring point MP 5	DW236	%	-200% to + 200%
Measuring point MP 10	DW241	%	-100% to + 100%
Measruing point MP 11	DW242	%	-100% to + 100%
Measuring point MP 12	DW243	%	-100% to + 100%

3.3.3 Mark Space Output

When the mark space output is used (S3, D 23.2 = 1), the controller's regulating variable is converted into pulses by a pulse duration modulator.

The pulse grid must be specified by the minimum pulse duration, where the scan time must be an integer multiple of the latter.

The pulse duration modulator functions as follows:

- The duration of a pulse per scan time is proportional to the regulating variable.
- A regulating variable of 30 % thus means:
  A positive pulse of 0.3 * scan time,
  no pulse for 0.7 * scan time.
- This pulse duty factor is recomputed at each sampling instant.

A negative regulating variable results in output of a negative pulse whose duration is proportional to the variable.

When the controller is to be used as a two-point controller, the output of negative pulses can be suppressed (D 23.14 = 0). Three-step output is selected with S15, D 23.14 = 1.

The pulses are output via a digital (S16, D 23.15 = 1) or analog output (S16, D 23.15 = 0).

When digital output is selected, the positive pulses are flagged in D 220.15, the negative pulses in D 220.14. Transfer of the bits to I/O modules by process image updating is possible only in word-serial mode (→ 2.8.2), i.e. one digital output word is required per controller to enable the best possible output timing accuracy.
The status of the pulse outputs can be output to individual I/O bits via a STEP 5 program. When the controller is invoked by initializing DB2, this STEP 5 program is not processed in synchronism with the controller call. This can be attained by invoking the controller per STEP 5 program (OB13) (→ 2.5.2).

When analog output is selected, a positive pulse is output as + 100 % via DAC 1, a negative pulse as + 100% over DAC 2.

Significance of the Response Threshold

The response threshold enables the minimum duration of a pulse or pulse interval to be limited to a specific value (≥ minimum pulse duration) to protect the actuator better.

As an example, a response threshold of 10 % means (y = regulating variable, T_A = scan time):

No pulse is output for -10 % < y < + 10 %.

A positive pulse of duration T_A is output for y > 100 % - 10 %.

A negative pulse is of duration T_A is output for y < -100 % + 10 %.

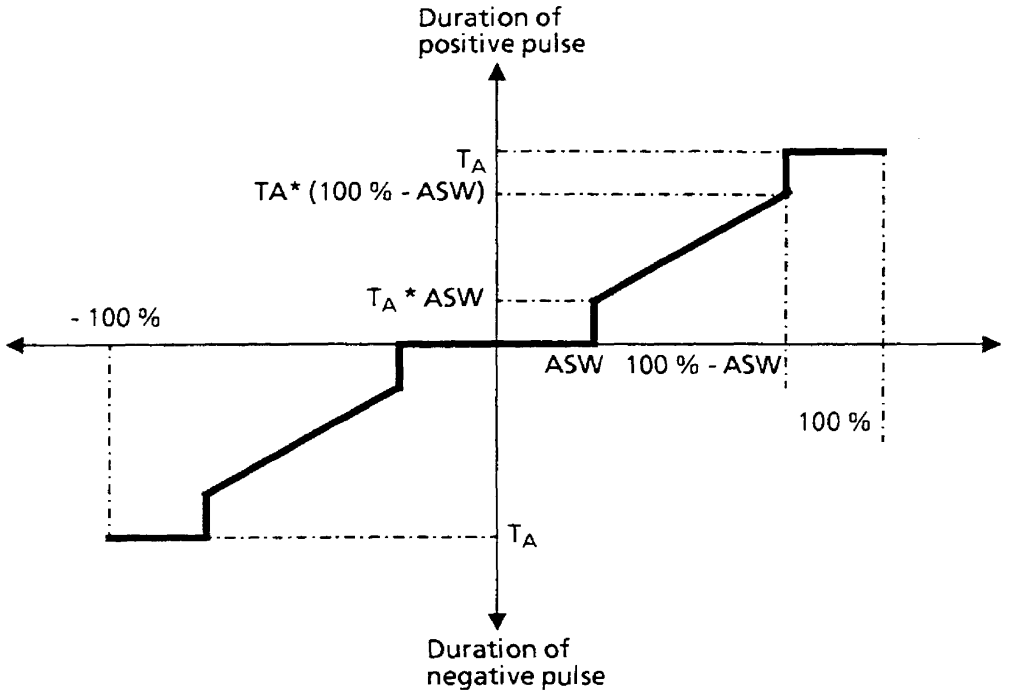


Figure 3-9: Example 1 (adjustment factor = 1.00)

Significance of the Adjustment Factor

The adjustment factor is used to modify the ratio of the duration of negative to positive pulses (e.g. when the heating unit heats faster than the cooling unit cools).

Adjustment factor = 1: The duration of positive and negative pulses is computed by multiplying the regulating variable by the scan time
Pulse duration = |y| * T_A.

Adjustment factor < 1: The pulse duration computed by multiplying the regulating variable by the scan time is shortened by the adjustment factor at the negative pulse output.
Positive pulse duration = |y| * T_A
Negative pulse duration = |y| * T_A * adjustment factor

Adjustment factor > 1: The pulse duration computed by multiplying the regulating variable by the scan time is shortened at the positive pulse output by 1/adjustment factor.
Positive pulse duration = |y| * T_A / adjustment factor
Negative pulse duration = |y| * T_A

The standard value for the adjustment factor is 1.00.

The response threshold is modified by the adjustment factor.

A adjustment factor < 1 means that the response threshold for negative pulses is multiplied by the adjustment factor.

As an example, a response threshold of 10% and a adjustment factor of 0.5 means:

- No pulse is output for -5% < y < + 10%.
- A positive pulse of duration T_A (scan time) is output for y > 100% - 10%.
- A negative pulse of duration 0.5 * T_A (scan time) is output for y < -100% + 5%.

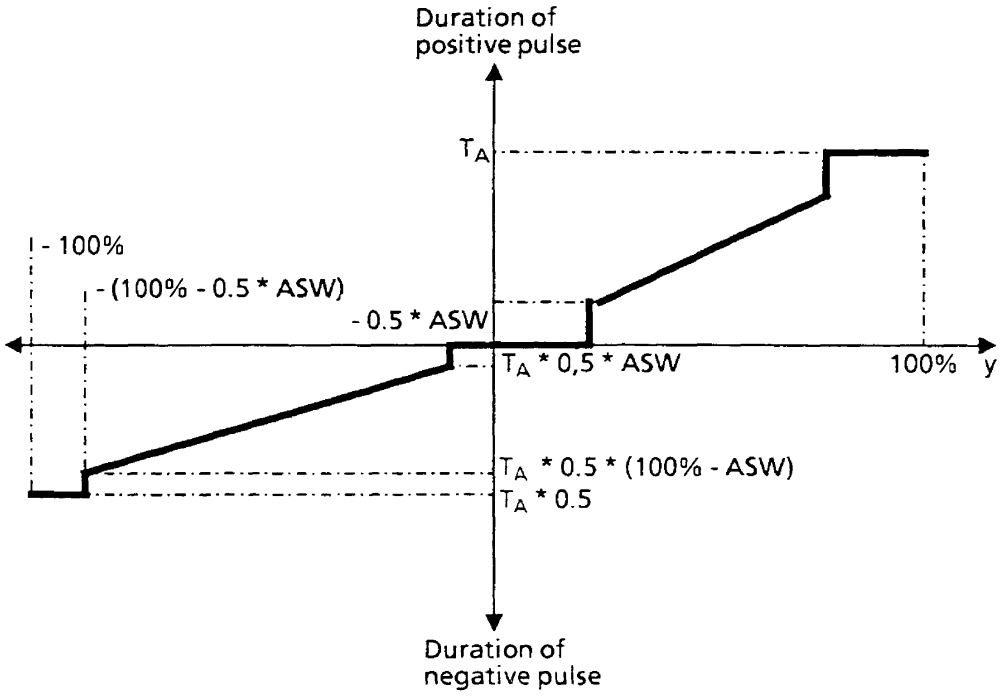


Figure 3-10: Example 2 (adjustment factor = 0.5)

A adjustment factor > 1 means that the response threshold for positive pulses is multiplied by 1/adjustment factor.

For example, a response threshold of 10% and a adjustment factor of 2.00 means:

- No pulse is output for -10% < y < +5%.
- A positive pulse of duration $T_A/2$ (scan time) is output for y > 100% - 5%.
- A negative pulse of duration T_A (scan time) is output for y < -100% + 10%.

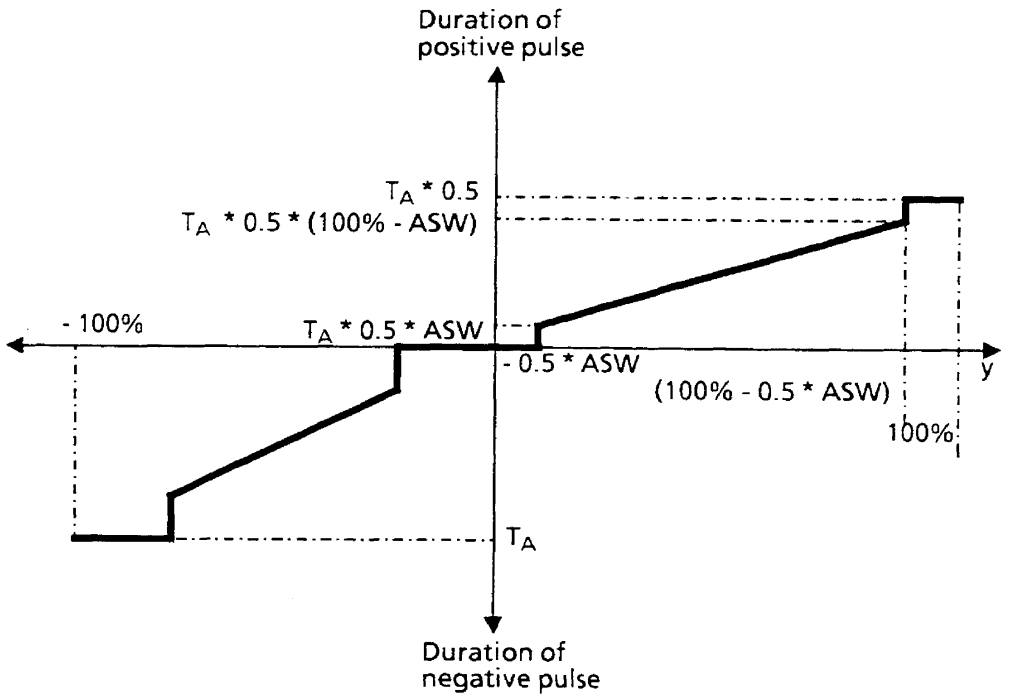


Figure 3-11: Example 3 (adjustment factor = 2.0)

Input Variables for the PID Controller with Mark Space Output

Designation	Address	Format	Setting range
Continuous/step Standard/enhanced	D 23.0 D 23.1	Bit Bit	0/1 0/1
Sep. D input ADC 3 (address) ADC 3 (value)	D 23.10 DW122 DW170	Bit Address Analog	0/1 -100% to + 100%
Interference input ADC 4 (address) ADC 4 (value) Interference enable	D 23.11 DW123 DW171 D 180.14	Bit Address Analog Bit	0/1 -100% to + 100% 0/1
Manual input PG/ADC ADC 5 (address) ADC 5 (value) Auto/manual mode	D 23.12 DW124 DW172 D 180.11	Bit Address Analog Bit	0/1 -100% to + 100% 0/1
Disable controller Real/ideal PID controller Mani. variable constant Auxiliary gain Pos. limiting value for correction increase Neg. limiting value for correction increase Upper controller limiting value Lower controller limiting value Proportional value Integral-action time Derivative-action time	D 180.15 D 180.12 D 180.13 DW74 DW81 DW82 DW79 DW80 DW73 DD75 DD77	Bit Bit Bit W/o dim. % % % % W/o dim. Time Time	0/1 0/1 0/1 -99.99 to + 99.99 **0%** to + 100% -100% to **0%** -100% to + 100% -100% to + 100% -99.99 to + 99.99 0 to 59.59 hhmm 0 to 59.59 hhmm
Cont./PDM Min. pulse duration T_{min} Response threshold Adjustment factor 2/3-step controller Analog-digital output	D 23.2 DD12 DD14 DW92 DW93 D 23.14 D 23.15	Bit *2.5 ms Time % W/o dim. Bit Bit	0/1 20 ms ... T_A T_A = scan time **0 %** to + 100 % **0** to + 99.99 0/1 0/1
DAC 1 (address) DAC 2 (address)	DW144 DW145	Address Address	

Output Variables of the PID Controller with Mark Space Output:

Designation	Address	Format	Output range
Overflow flag for	D 255.7	Bit	0/1
upper limit violation	D 220.6	Bit	0/1
Overflow flag for	D 255.6	Bit	0/1
lower limit violation	D 220.7	Bit	0/1
Positive pulse	D 220.15	Bit	0/1
Negative pulse	D 220.14	Bit	0/1
DAC 1 (value)	DW 208	Analog	-100% to + 100 %
DAC 2 (value)	DW 209	Analog	-100% to + 100%
Measuring point MP 3	DW234	With dim.	-200% to + 200%
Measuring point MP 4	DW235	%	-100% to + 100%
Measuring point MP 5	DW236	%	-100% to + 100%
Measuring point MP 6	DW237	%	-100% to + 100%
Measruing point MP 7	DW238	%	-100% to + 100%
Measuring point MP 10	DW241	%	-100% to + 100%
Measuring point MP 11	DW242	%	-100% to + 100 %
Measuring point MP 12	DW243	%	-100% to + 100 %

3.3.4 Point Controller

The point controller is used to drive a final control element with integrating function. The output pulses position the actuator in the positive and negative directions. No pulse means that the actuator retains its current position (one example would be the use of an actuator for valve adjustment).

The equation for the continuous PID controller in the time range,

$$y = k_p \left(e + 1/T_N * \int_0^t e(v)\, dv + T_V * e\right),$$

where
K_p = Proportional value (controller gain)
T_N = Integral-action time
T_V = Derivative-action time
y = Regulating variable
e = Deviation

is simulated by PID velocity algorithm with trapezoidal integration.

Particularly when a PI controller is used, trapezoidal integration enables precise simulation of an analog controller's integration.

A delay element (first order) with time delay constant T_A is integrated in the D branch.
Instead of the regulating variable, the point controller computes the number of pulses (x) using the formula:

$$x_k = \Sigma \left[K_p + TM/T_{min} * \{ (e_k - e_{k-1}) + T_A/(T_N*2) * (e_k + e_{k-1}) + D_k \} \right]$$

where

$$D_k = 0.5 * \left[T_V/T_A * \{ (e_k - e_{k-1}) - (e_{k-1} - e_{k-2}) \} + D_{k-1} \right].$$

TM = Actuator runtime
T_{min} = Minimum pulse duration

Specification of ***actuator runtime TM*** and ***minimum pulse duration T_{min}*** is therefore required in addition to the controller parameters. The minimum pulse duration is the shortest pulse possible for the actuator, i.e. depends on the actuator used.
The actuator runtime is the time which the actuator needs to go from end stop to end stop at maximum speed.
Quotient TM / T_{min} in the formula thus determines the number of pulses required to move the actuator so as to accord with the deviation.

A controller branch can be disabled by setting the relevant parameter (T_N, T_V) to zero. ***$T_N = 0$*** also disables the ***I component***, which is not obvious from the quotient T_A/T_N.

Before it is forwarded to the point controller, the deviation first passes through a ***dead band with hysteresis.***

The dead band has the following purpose:

The minimum pulse duration does not permit an exact approach to the setpoint, and the controlled variable will be either slightly too low or slightly too high.

To prevent hunting of both final control element and controlled variable, the controller's function is disabled when the absolute value of the deviation drops below the ***lower response***. No further pulses are computed, and the actuator or final control element remains in its current state.

The controller does not go into operation again until the deviation exceeds the absolute value of the ***upper response threshold***.

The deviation is thus maintained at a value lower than the upper response threshold.

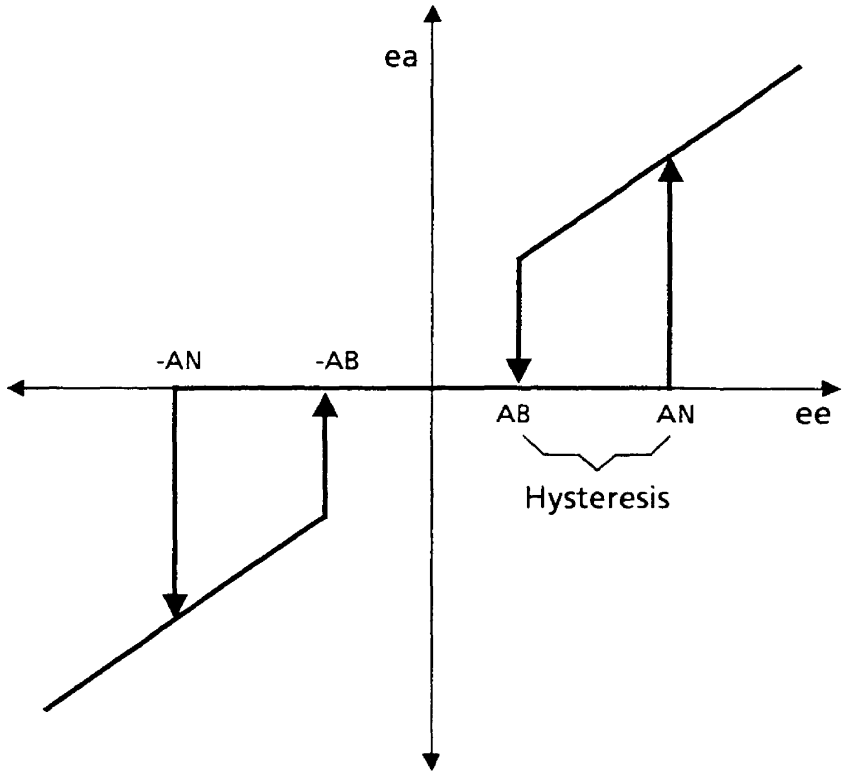


Figure 3-12: Dead band (ee = input value, ea = output value, AN = upper threshold, AB = lower response)

The algorithm is such that the change in the deviation when the dead band is entered or exited does not affect the pulses at the controller's output.

The dead band function can be disabled only for the I-component (D 180.10 = 1).

Point Controller in Manual Mode

The following are two different methods for adjusting the actuator or final control element when the point controller is set for manual mode (D 180.11 = 1):

1) A manual input value specifies the pulse duration per scan time. A negative value defines the duration for negative pulses.

   If bit D 180.8 is set (manual input value effective), the number of positive or negative pulses output per scan time is based on the manual value.
   No pulses are output when D 180.8 = 0.

   The manual input value may be entered as a constant by setting D 23.12 = 0 or read in via ADC 5 by setting bit D 23.12 to 1.

2) The following adjustment is possible only when D 180.8 = 0.

   D 180.12 = 1 means the actuator's maximum negative adjustment rate (close manual mode, yh_{neg}).
   D 180.13 = 1 means the actuator's maximum positive adjustment rate (open manual mode, yh_{pos}).

No further pulses are output when the controller is disabled (D 180.9 = 1).

No further positive pulses are output when the 'Final OPEN position reached' input is set (D 180.4 = 1, E_{open}).
No further negative pulses are output when the 'Final CLOSED position reached' input is set (D 180.3 = 1, E_{closed}).

The pulses are output over a digital (S16, D 23.15 = 1) or analog (S16, D 23.15 = 0) output.

When digital output is selected, the positive pulse is flagged in D 220.15, the negative pulse in D 220.14. The bits can be transferred to the I/Os during process image updating in word-serial mode only (→ 2.8.2), i.e. one digital output word is required per controller in order to attain the best possible timing accuracy.

The status of the pulse outputs can be output to individual I/Os via a STEP 5 program. When the controller is invoked by the initializing of DB2, this STEP 5 program is not executed in synchronism with the controller call. This is made possible by invoking the controller over a STEP 5 program (OB13) (→ 2.5.2).

When analog output is selected, a positive pulse is output as + 100 % via DAC 1; a negative pulse as -100 % via DAC 2.

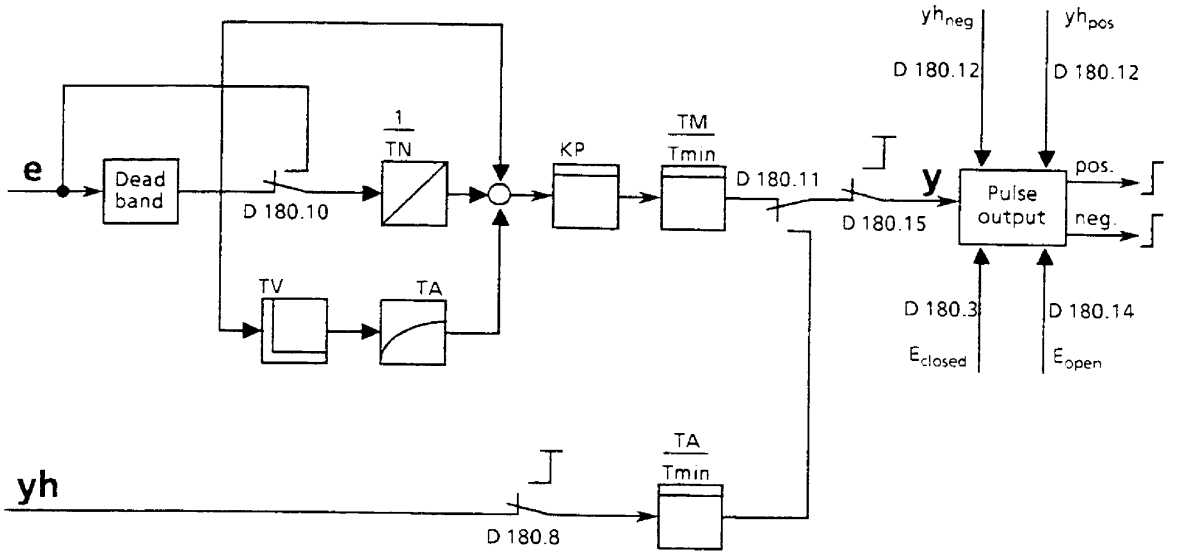


Figure 3-13: The point controller's signal flow

Input Variables for the Point Controller

Designation	Address	Format	Setting Range
Continuous/Step	D 23.0	Bit	0/1
Manual input val. PG/ADC	D 23.12	Bit	0/1
ADC 5 (address)	DW124	Address	
ADC 5 (value)	DW172	Analog	-100 % to + 100 %
Auto/manual mode	D 180,11	Bit	0/1
Manual input val. effective	D 180.8	Bit	0/1
Close (manual mode)	D 180.12	Bit	0/1
Open (manual mode)	D 180.13	Bit	0/1
Final OPEN pos. reached	D 180.4	Bit	0/1
Final CLOSED pos.reached	D 180.3	Bit	0/1
Disable controller	D 180.9	Bit	0/1
Dead band ineffective	D 180.10	Bit	0/1
Proportional value	DW83	w/o dim.	-99.99 to + 99.99
Integral-action time	DD84	Time	0 to 59.59 hhmm
Derivative-action time	DD86	Time	0 to 59.59 hhmm
Actuating time	DD88	Time	T_{min} to 59.59 hhmm
Min. pulse duration T_{min}	DD12	*2.5 ms	
	DD14	Time	20 ms to T_A
			T_A = scan time
Upper response threshold	DW 90	w/o dim.	**LR threshold** to 100 %
Lower response threshold	DW91	%	0 % to **UR threshold**
Analog-digital output	D 32.15	Bit	0/1
DAC 1 (address)	DW144	Address	
DAC 2 (address)	DW145	Address	

Output Variable from the Point Controller

Designation	Address	Format	Output Range
Final OPEN pos. reached	D 255.7	Bit	0/1
Final CLOSED pos.reached	D 255.6	Bit	0/1
Positive pulse	D 220.15	Bit	0/1
Negative pulse	D 220.14	Bit	0/1
DAC 1 (value)	DW208	Analog	-100 % to + 100%
DAC 2 (value)	DW209	Analog	-100 % to + 100 %
Measuring point MP 3	DW234	with dim.	-200% to + 200%
Measuring point MP 5	DW236	%	-100% to + 100%
Measuring point MP 6	DW237	%	-100% to + 100%
Measruing point MP 7	DW238	%	-100% to + 100%
Measuring point MP 10	DW241	%	-100% to + 100%

3.4 Controller Monitoring Functions

3.4.1 Test Sockets

The values at the controller's measuring points can be output to I/Os via two test sockets (S19, D24.2 = 1 or S20, D 24.3 = 1), thus enabling the measured values to be recorded. The test sockets can be connected to all active measuring points.

The test sockets also enable connection to other controllers, since you can specify output of a measured value to a flag word by setting the relevant parameter.

The test sockets can also be used for format conversion. The test socket stores the measured value in analog I/O format in the controller DB (DW218 or DW219). If format conversion only is required, the user must enter spaces instead of the I/O address when initializing the test socket with COM REG.

3.4.2 Limit Monitors

Two limit monitors can also be connected to the controller's measuring points (S17, D24.0 = 1 or S18, D24.1 = 1). The limit monitors can be connected to all active measuring points.

A bit is set in data word 220 to flag violation of a positive or negative limiting value (see table). The contents of data word 220 can be output to I/Os by entering address DA 01, or suppressed by entering spaces in place of the address. The status of data word 220 can also be evaluated via a STEP 5 program.

Input Variables for Test Sockets and Limit Monitors

Designation	Address	Format	Setting range
Test socket 1 Measuring point number Address DAC 03	D 24.2 DW116 DW154	Bit w/o dim. Address	0/1 1 to 12
Test socket 2 Measuring point number Address DAC 04	D 24.3 DW117 DW155	Bit w/o dim. Address	0/1 1 to 12
Limit monitor 1 No. of limiting values Measuring point number Limiting value 01 Limiting value 02 Limiting value 03 Limiting value 04 Limiting value 05 Limiting value 06	D 24.0 DW59 DW118 DW60 DW61 DW62 DW63 DW64 DW65	Bit w/o dim. w/o dim. with dim. or % % % %	0/1 1 to 6 1 to 12 -100% to + 100% -100% to + 100% -100% to + 100% -100% to + 100% -100% to + 100% -100% to + 100 %

Limit monitor 2	D 24.1	Bit	0/1
No. of limiting values	DW66	w/o dim.	1 to 6
Measuring point number	DW119	w/o dim.	1 to 12
Limiting value 01	DW67	with dim.	-100% to + 100%
Limiting value 02	DW68	or	-100% to + 100%
Limiting value 03	DW69	%	-100% to + 100%
Limiting value 04	DW70	%	-100% to + 100%
Limiting value 05	DW71	%	-100% to + 100%
Limiting value 06	DW72	%	-100% to + 100 %

Output Variables for Test Sockets and Limit Monitors

Designation	Address	Format	Output range
DAC 03 (value)	DW218	Analog	-200% to + 200%
DAC 04 (value)	DW219	Analog	-200% to + 200%
Limiting value flag 01	D 220.2	Bit	0/1
Limiting value flag 02	D 220.1	Bit	0/1
Limiting value flag 03	D 220.2	Bit	0/1
Limiting value flag 04	D 220.3	Bit	0/1
Limiting value flag 05	D 220.4	Bit	0/1
Limiting value flag 06	D 220.5	Bit	0/1
Limiting value flag 01	D 220.8	Bit	0/1
Limiting value flag 02	D 220.9	Bit	0/1
Limiting value flag 03	D 220.10	Bit	0/1
Limiting value flag 04	D 220.11	Bit	0/1
Limiting value flag 05	D 220.12	Bit	0/1
Limiting value flag 06	D 220.13	Bit	0/1

3.5 Miscellaneous Controller Parameters

The FB number entered in the controller DB determines which function block the operating system is to invoke. The R64 controller structure is stored as FB102 in the REGR64ST.S5D file.

Miscellaneous Controller Parameters

Designation	Address	Format	Setting Range
FB number	DR0	w/o dim.	1 to 255
Sscan time	DD8	*2.5 ms	
	DD10	Time	20 ms - 59.59 hhmm
Controller physical dimension	DW17-DW22	ASCII	
Controller and interface data area name for CP526	DW192 to DW199	ASCII	
Process output image (PIQ)	D 2.6	Bit	0/1
I/O reset bit (PLB)	D 2.7	Bit	0/1
Operator control bit 1 (BED1)	D 7.8	Bit	0/1
Operator control bit 2 (BED2)	D 7.9	Bit	0/1
Controller management bit (RVB)	D 7.0	Bit	0/1

Description of the bits in the table above:

PIQ (process output image, D 2.6) specifies when the process output image is to be updated. When PIQ = 0, the controller inputs and outputs are both updated prior to execution of the controller program. When PIQ = 1, the inputs are updated on a time-controlled basis before the controller program executes and the outputs immediately after it executes.

PLB (I/O reset bit, D 2.7) specifies whether the outputs are to be set to zero following a transition to the STOP mode, or upon selective disabling of controllers (in accordance with RVB. See below).
PLB = 1 = = = > Reset outputs
PLB = 0 = = = > Do not reset outputs

RVB (controller management bit, D 7.0) is used to disable and reenable a controller that has been entered in DB2. When enabled, the relevant controller responds as upon initiation of a cold restart.
RVB = 1 = = = > Enable controller
RVB = 0 = = = > Disable controller

BED1 (operator control bit, D 7.8) is the operator control bit for the parameters in DW25 - DW55

BED2 (operator control bit, D 7.9) is the operator control bit for the parameters in DW56 - DW105

4 Number Formats

The operator must enter and/or specify a number of different types of variables when initializing, servicing or monitoring the controller; these include

* Times
* Percentages
* Variables with physical dimension
* Variables without physical dimension
* Number formats for analog I/O modules
* Number formats for I/O addresses

The formats for all values in the controller DB are listed in chapter 7.
The sections dealing with these formats discuss the entry of values from the programmer using COM REG and their entry in the data block.

4.1 Times

Examples of times:
- PID controller parameters T_N and T_V
- Scan time T_A
- Minimum pulse duration T_{min}
- Filter time constant

Modes of representation on the programmer:

0.1 msec	to	9999 msec
0.001 sec	to	9999 sec
00.01 hh.mm	to	59.59 hh.mm (fixed-point)

Representation in the controller DB:
The programmer generates two 16-bit data words from the user's entry:

One **time** as 16-bit fixed-point number

One **format identifier** as 16-bit fixed-point number: this identifier is used for coding the time unit and locating the decimal point.

Synopsis of time formats:

Entry on programmer	Time in the DB in fixed-point format	Identifier
00.00 to 59.59 h min	0 to 5959	1
0000 to 9999 sec	0 to 9999	2
000.0 to 999.9 sec	0 to 9999	3
00.00 to 99.99 sec	0 to 9999	4
0.000 to 9.999 sec	0 to 9999	5
0000 to 9999 msec	0 to 9999	6
000.0 to 999.9 msec	0 to 9999	7

Example: PID controller's integral-action time T_N = 23.55 sec

= = > DW75 = KF + 2355 (DF = fixed-point constant)
DW76 = KF + 0004 (identifier 4)

4.2 Percentages

Examples of percentages:
- Limiting values for regulating variables
- Manual input value (programmer)
- Threshold of the mark space output
- Measuring points MP4 - MP7 and MP10

Representation on the programmer:

- For inputs: ± 0.01 % to ± 100.00 %
- For outputs: ± 0.01 % to ± 200.00 %

Extension to ± 200 % is necessary to enable representation of the deviation at a measuring point when setpoint w = 100 % and actual value x = -100 %.

Representation in the controller DB:

- Percentages are stored in the controller DB as 16-bit fixed-point numbers (100% = = > 10000 FP).

Example: Value entered on programmer = 34.15 %

= = > DW95 = KF + 3415 (KF = fixed-point constant)

4.3 Variables with Physical Dimension

Examples:
- Abscissas and ordinate values in a polygon curve
- Upper response threshold of the point controller's dead band
- Initial actual value
- Control setpoint
- Upper setpoint limiting value

Representation on the programmer:

± .0001 [dim] to ± 10000. [dim] .

The physical dimension specification allocates the variables with physical dimensions to the values read in via the analog-digital converters.
When entering a variable with physical dimensions on the programmer, the operator must first define the dimension and then allocate a value to the zero percent value and the 100 percent value on the ADC.
The number of decimal places in these two values must be identical, and is stored in DW22.
All values entered with physical dimensions after the dimension has been defined must have the same number of decimal places, as they are stored in the controller DB without a decimal point.
For example, a value of 32.55 DEG_C is stored in the controller DB as KF + 3255 (KF = fixed-point constant).

(0 percent on the ADC may correspond to 0 volts, 0 amperes or 4 mA).

Representation in the controller DB:

The specified physical dimension is stored in the controller DB in words DW17 to DW22.
A variable with physical dimension is stored as a word (without identification of the decimal point location).

Example:

```
PHYS. DIMENSION D1:        DEG_C
      0 PERCENT CORRESPONDS TO          20.0 DEG_C
    100 PERCENT CORRESPONDS TO         850.0 DEG_C

= = >   DW17 = KH 4752    (KH = hexadecimal constant, 4752 = ASCII "GR")
        DW18 = KH 4144                                (4144 = ASCII "AD")
        DW19 = KH 5F43                                (5F43 = ASCII "_C")
        DW20 = KF + 200   (KF = fixed-point constant,
                                200 = 0 percent value)
        DW21 = KF + 8500  (8500 = 100 percent value)
        DW22 = KF + 1     (1 means one decimal place)
```

Variables with physical dimension may be entered once the dimension has been defined. A single numerical value is represented as a 16-bit fixed-point number.

Example: Control setpoint = 155.0 DEG_C

= = > DW191 = KF + 1550 (KF = fixed-point constant)

NOTE: When the number of decimal places is changed after the dimension is defined, all variables with physical dimension entered previously will be incorrectly interpreted.
If the 0 percent value is changed to 20 DEG_C and the 100 percent value to 1200 DEG_C in the example above, the previously entered control setpoint is interpreted as 1550 DEG_C instead of as 155.0 DEG_C.

4.4 Variables without Physical Dimension

a) Integers

Examples:
- Number of limiting values for limit monitor 1
- Measuring point number for test socket 1

Representation on the programmer:

-32768 ,..., + 32767

Representation in the controller DB:

Stored as a word in KF format (fixed-point constant)

Example: Measuring point number for test socket 1 = 5

= = > DW116 = KF + 5 (KF = fixed-point constant)

b) Variables with decimal point

Examples: - Proportional value K_p
- Adjustment factor for the Mark space output

Representation on the programmer:

± 00.01 ... ± 100.00 (fixed-point)

Representation in the controller DB:

16-bit fixed-point notation (100 * value input)

Example: K_p = 11.40 (for PID controller)

= = = > DW73 = KF + 1140 (KF = fixed-point constant)

4.5 Analog I/O Format

The compact controller requires U-range analog modules as interfaces between process and controller (→ 2.3.1, Hardware).
These modules have the following characteristics:

- Unitized format for analog input and analog output:
  The format is uniform for all modules, thus making format conversion unnecessary.
  (Exceptions: 460 and 465 analog input modules with a measuring range of 4 to 20 mA. In this case, format conversion must be selected by setting D 24.6 to 1; → 2.3.1).

- Representation of the nominal range:
  The nominal range is linear, and is identical for all measuring ranges.

- Representation of values in two's complement format

- Overflow:
  An overflow is available in addition to the nominal range.

Significance of the bit positions in the data word:
(the numbers in the blocks indicate the percent of the nominal range)

15	14	13	12	11	10	9	8
(VZ)	100	50	25	12.5	6.25	3.125	1.5625

7	6	5	4	3	2	1	0
.78125	.39062	.195	.09750	.00488	x	x	x

Bits 0 to 2 or 0 to 3 are irrelevant, depending on the module's precision.

Examples:

100 % = KF + 16384 (= 4000 H = 0100 0000 0000 0000 B) .

0 % = KF + 0 (= 0000 H = 0000 0000 0000 0000 B) .

100 % = KF-16384 (= C000 H = 1100 0000 0000 0000 B) .

(KF = fixed-point constant)

The overflow in which the analog modules still convert correctly corresponds in size to the nominal range. The maximum value acceptable to an analog module is thus approximately + 199.9%.

4.6 Format of the I/O Address

As mentioned in sections 2.7.3 and 2.8.2, it is the entry of I/O addresses which causes the I/Os to forward input values to the controller and the controller to forward output values to the I/Os.

Entry using COM REG:

Addresses need not necessarily be real I/O addresses, but may also be flags and IPC flags (refer to the STEP 5 Programming Guide). Only even-numbered addresses are permissible, however, as process image updating is word-serial.

When spaces are entered (= presetting), there is no exchange of input/output values between controller and I/Os.

Representation in the controller DB:

Type of I/O		Stored in Controller DB
Digital I/Os	(0-126)	F000H-F07EH on read/write
Analog and digital I/Os	(128-254)	F080H-F0FEH on read/write
Flags	(0-254)	7E00H-7EFEH on read/write
IPC flags	(0-254)	F200H-F2FEH on read/write
Q-range I/Os	(0-255)	F100H-F200H on read/write
No I/O read/write:		000H

NOTE: IPC flags (i.e. interprocessor communication flags) are flags on the coordinator in the interface data RAM.

4.7 Table for Number Format Conversion

When using the controller, it is sometimes necessary to convert number formats. For example, a controller's output on DAC 1 can be interfaced to a secondary controller's setpoint input over the control setpoint only when the analog format has been converted to the secondary controller's physical dimension format. The purpose of the table below is to facilitate conversions of this type. Format conversion is not required to interface variables of like type, e.g. analog output on DAC 1 with the secondary controller's analog output on ADC 1.

Terms used in the table:

Q : The converted value required for further processing

I : The value supplied by the controller and which is to be converted into another format

0 % : The value specified via COM REG (e.g. 0 % = 1.00 kg)

100 % : The value specified via COM REG (e.g. 100 % = 80.00 kg)

The values to be converted (I, as in input) must be interpreted/entered as fixed-point numbers without decimal places, i.e. as integer fixed-point numbers. Accordingly, a percentage value of 54.62 % must be entered as KF + 5462, a value of 227.4 m/h with physical dimension as KF + 2274. The procedure for converted values (Q, as in output) is analogous to that for input values. Assuming that the 0% values were defined as 0.000 volts (i.e. with three decimal places), a converted value of KF + 2674 (with physical dimension) is interpreted as 2.674, a converted value of KF + 3434 (percentage) as 34.34 %.

		Values to be converted		
		Percentages	Values with phys. dimens.	Analog I/O format
Converted values	Percentages	$Q = I$	$Q = \frac{I - 0\%}{100\% - 0\%} * 10000$	$Q = \frac{10000}{16384} * I$
	Values with physical dimens.	$Q = \frac{100\% - 0\%}{10000} * I + 0\%$	$Q = I$	$Q = \frac{100\% - 0\%}{10000} * I + 0\%$
	Analog I/O format	$Q = \frac{16384}{10000} * I$	$Q = \frac{I - 0\%}{100\% - 0\%} * 16384$	$Q = I$

Example:

The percentage at measuring point MP 5 (data word 236) of a master controller is to be forwarded to the control setpoint of a slave controller (data word 191). The slave controller's 0 % value was defined as 0.000 volts, the 100 % value as 5.000 volts.

The value 20.50 % was read from measuring point MP 5. According to the table, the result is the following:

((100 % - 0 % / 10000) * 1 + 0 % =

((5000 - 0000 / 10000) * 2050 + 0000 = 1025

This result must be interpreted as 1.025 volts. It is forwarded as fixed-point number 1025 to data word 191 of the slave controller.

5 Error Handling

This chapter deals with the various errors which can occur in the control system. Sections 5.1 and 5.2 relate only to the controller call initialized via DB2 (controller list).

5.1 DB2 Errors

In all restart modes, a check is made to make sure that the controller DBs specified in DB2, as well as the controller FBs entered in these DBs, are loaded.

If a block is missing, the restart attempt is aborted and the type of error, the block type and the block number are entered in the PC's memory in system data words 3 and 4. Such an error is fatal and the programmable controller enters the STOP mode regardless of what is occurring at the user interface.

The following ***error identifiers*** are possible, depending on the processor release (cf. programming manuals for the relevant processors).

SD 3	SD 4	Cause of error
0421H	DBxxH	Controller data block not loaded (xx = number of the missing data block)
0422H	FByyH	Controller function block not loaded (yy = number of the missing function block)
0423H	FByyH	Controller function block not an assembler FB (yy = number of the bad function block)
0424H	FByyH	Controller function block not at a paragraph address in user memory (yy = number of the relevant function block)
0425H	DBxxH	Controller data block not 1034 bytes (xx = data block number)

The error information can be read out via the 'INFO ISTACK' when using the S5 programmer software and via the OS memory area (SD3: EA03H, SD4: EA04H) when 'INFO ADR' is used. ANL-ABB and error identifier DB2-FE are also flagged in the control bits.

5.2 Controller Errors

The term ***controller error*** is used to designate errors which occur during execution of the controller's program. Two examples of controller errors are exceeding of the scan time and a delayed acknowledgment from the I/Os (time-out). Errors of this type may have one of several effects:

- A controller program abort, which would set the PC to the STOP mode
- Invocation of the controller error OB (OB 34), which then determines subsequent system performance
- Continuation of the program without any reaction to the error

When a controller error is detected, the error-specific user information shown in the table below is entered in S5 registers ACCUM1 and ACCUM2.

ACCUM1	ACCUM2	Cause of error	Reaction
0801H	DBxxH	Scan time error (xx = DB no. of the controller)	Calls OB34 unless masked (D 4.0 = 1)
0802H	DBxxH	Controller data block not loaded (xx = DB number)	Calls OB34
0803H	FByyH	Controller function block not loaded (yy = Fb number)	Calls OB34
0804H	FByyH	Controller function block not an Asembler FB (yy = FB number)	Calls OB34
0805H	FByyH	Controller function block not at paragraph address (yy = FB number)	Calls OB34
0806H	DBxxH	Controller data block not 1034 bytes (xx = DB number)	Calls OB34

ACCUM1	ACCUM2	Cause of error	Reaction
0880H	zzzzH	Time-out occurred during process image updating (zzzz = time-out address)	Calls OB34

Whether some of the error codes listed above are implemented depends on the processor release (cf. programming manual for the relevant processor(s)).

On the programmer, error identifier REG-Fe is flagged in the control bits in all cases (when the PG 670/PG 675 operating system is used, the error identifier is shown in the free field in the bottom line of the CONTROL BITS screen form).
All further reactions are dependent on the controller-specific user interface, the controller error OB (OB34):

a) The entire PC enters the STOP mode if no OB34 has been loaded.
The cause of error can be ascertained by reading out the ISTACK (ACCUM1, ACCUM2).
Individual reactions to errors are not possible.

Note: A missing block error occurs only when the relevant block was deleted or declared invalid during the PC cycle.

b) If you have loaded OB34, you can initiate an error-dependent error recovery routine by evaluating ACCUM1 and ACCUM2 in a STEP 5 program. After the error recovery routine has executed, the program is resumed at the point of interruption unless OB34 contains a STOP command.
OB34 need comprise only a BE command if you want to ignore any controller errors.

c) A scan time error has special significance among the controller errors. A scan time error occurs when updating of the process image and execution of the controller's program get "out of step" because of higher-priority operating system activities.

A scan time error can be treated either as a normal controller error as described in a) and b), or can be suppressed by means of a mask.
Setting bit D 4.0 (which is not possible via COM REG) prevents the PC from entering the STOP mode (even when no OB34 has been loaded) when a scan time error occurs on one of the controllers.
Setting of this bit also informs the operating system that failure of this controller is acceptable.

When programming the error OB, note that its execution time increases the execution time required for the controller program and can therefore result in a timing error.

5.3 Bad Controller DB

A controller DB must be generated using the COM REG programmer software. It then has a size of 1034 bytes.

When a controller DB entered in DB 2 is overwritten by a DB with a length other than 1034 bytes, and the controller call is initialized over DB2, the result is the error described in section 5.2, i.e., "Controller data block not 1034 bytes".
(R processors older than revision level 6 do not check the length of controller DBs, and an error can therefore cause the entire system to assume an undefined state.)

When the controller FB is called via a STEP 5 program, the system is stopped with an STP command on detection of a DB comprising more or fewer than 1034 bytes. Error code 0806H is entered in ACCUM1 and DBxxH in ACCUM2 (xx is the number of the bad DB).

The length of a DB can be queried using the STEP 5 programmer software function "INFO DIR".
COM REG's "Display" and "Test" functions check the version number in data word 2 to ascertain whether the DB is a controller DB, aborting the relevant function with an error message should the check prove negative.

5.4 User Errors

When using the R64 controller structure, errors made at crucial points may result in the controller's not operating at all, in its failure to produce the relevant output variables, or in a reaction that the user does not expect. Errors of this type, as well as ways to remedy them, are discussed in the remainder of this section.

The initial state of the **bit variables in the Controller's process input image** (DW180) was chosen so that the controller can operate in a "normal" mode when these bits are zero. For this reason, note that setting the controller enable bit to 1 (D 180.9, D 180.15) will disable the controller. This arrangement was necessary to enable the controller to operate without a digital input module and without altering the preferred states of the relays.

When a previously configured *test socket* is disabled via COM REG, the latter does not permit the test socket to be reinitialized; the I/O address entered in the input address list, however, is not deleted. This may cause problems when the address is to be used elsewhere, e.g. by another controller, as it is assumed that the process image for this address will no longer be updated. To disable a test socket in the proper manner, its address must first be deleted (by entering blanks) and then the appropriate structuring switch set to zero.

5.5 The Controller Does Not Operate

One good way to test a controller is to configure a setpoint string (ADC/manual entry, D 23.7 = 1 and control setpoint/setpoint string, D 180.5 = 1).
By specifying two setpoint values and selecting the interpolation function, COM REG can be used to check whether anything is happening by checking measuring point MP 1. One of the following may be the cause of the controller's failure to respond:

a) **The PC is in the STOP mode**, in which case it is necessary to initiate a cold restart. If the processor fails to go into cyclic operation, it could be that the controller DBs entered in DB2 are not in the PC's memory. The missing data block(s) must then be transferred to the PC.
Another reason for the error may be a missing function block: either an FB was forgotten and not transferred to the PC, or the wrong FB number was entered in a data block during initialization. The relevant FB must be tranferred to the PC or the bad FB number corrected.
The processor will not enter the RUN mode when an I/O address which cannot be referenced was specified.
The I/O addresses specified when the data blocks were initialized can be printed out in a cross-reference list.
To correct the error, the I/O address must be corrected in the data block (if necessary by overwriting it with blanks) or an I/O module with this address plugged into the programmable controller.

b) **The PC is in cyclic operation** (the RUN LED is on). First check DB2. Has it been loaded into the PC, and has the controller been entered? Was a cold restart executed after transferring DB2 to the PC or after entering the controller DB? If so, it is possible that the controller management bit (RVB, D 7.0 in the controller DB) has not been set. This can be ascertained by invoking the COM REG function "Controller processing". The controller must function when this bit is set and all of the instructions given above have been followed.
(There is an error which can be remedied by specifying a DB1. This error is discussed in detail at the end of the section.)

Once the controller has been induced to respond by the modifying of a value at measuring point MP 1, you can proceed measuring point by measuring point until a regulating variable has been output.

5.6 Difficulties with I/O

For the control algorithm, the process input image (DW168 to DW191) is a "read-only" memory area, i.e., an area from which information can only be fetched.

There are different ways of entering information into this area. One way is to enter an address in the process input image's address list (DW120 to DW143). Each word in this list is allocated to a specific word in the value list. Prior to each call, the R processor's operating system supplies those values in the input value list for which an address other than zero has been specified.
Values may also be entered in the value list via a STEP 5 program, or the COM REG test function (e.g., for controlling relays) can be used. A **conflict** may result when both the operating system in the process of updating the process image and a control program attempt to enter input values.

It is pointless to attempt to control bits in the process input image when a program or the operating system also has write access to these bits. This possibility is presented only to make you aware of all the options at your disposal.
It is your responsibility to restrict yourself to the permissible modes of access, and therefore to avoid such problems.

Problems can also occur as regards the process output image. Again, it is possible to have the operating system output the value stored in the value list to I/Os by specifying a corresponding address in the address list. The value list can also be read out via a STEP 5 program. Care must be taken, however, that the value output to an I/O module is not overwritten from the other side. The regulating variables of two controllers cannot be output to the same address.

Inexperienced users can expect the most difficulties with the cyclic STEP 5 program's process image. The processor reads the current values at I/O addresses 0 - 127 and writes them to the process input image during each cycle (in accordance with the cyclic control program's execution time). The values in the process output image are also written to I/O addresses 0 - 127.

The addresses for which this STEP 5 process image is generated can be specified (byte by byte) in a DB1. If no DB1 has been loaded into the PC's memory, the processor (single-processor mode) checks for modules having addresses 0 - 127 on initiation of a cold restart, and generates the process image for the STEP 5 program accordingly. If, for example, a controller's digital output is applied to PW 0 and the processor is operating in single-processor mode without a DB1, both the controller and the STEP 5 program's process image facility will write to I/O addresses 0 and 1, thus jeopardizing the reliability of the information. To prevent such a conflict, a DB1 (in which, in this example, addresses 0 and 1 have not been specified as outputs) must be loaded into the PC.

If only controllers, but no control program, are used in the processor, and no output addresses lower than 128 are to be used, then the "empty" DB1 provided on the controller diskette can be loaded into the PC.

Changes in DB1 are transferrable following a cold restart only.

NOTE: The possibility of error when accessing I/Os is even greater in multiprocessor mode.

Additional information can be found in the programming manual for the relevant processor.

SIEMENS

COM REG
Programmer-Software

User's Guide — Order No. C79000-B8576-C388-02

TABLE OF CONTENTS

3 General Notes

3.1 Structure of the Masks

As in other SIMATIC S5 operator areas, the dialog between the user and COM REG is executed exclusively by means of masks. All the masks are structured basically the same:
- two headlines with information on the selected function and the block which is being processed
- 19 lines of working area
- three lines of function key menu for selecting functions and controlling the parameter input.

This structure is illustrated in the "Controller selection" mask. This mask appears immediately after selecting "COM REG" by means of the <F1> / <Package> key. All existing COM REG packages are displayed.

```
**********************************************************************************
*                                                                                *
*                                                                                *
*    Controller select:                                                          *
*  ------------------------------------------------------------------------------ *
*                                                                                *
*    The following operator routines are available:                              *
*    ==============================================                              *
*                                                                                *
*    COM REG for the R64 Controller Structure  ...   C:S5OECR2X.CMD              *
*                                                              C:S5OECI3X.CMD    *
*    COM REG for IP 252 Closed loop control module   C:S5OECI2X.CMD              *
*                                                              C:S5OECI3X.CMD    *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*   F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                         *
*   REG.!      !      !      !      !      !      !                               *
*   R64 ! IP252!      !      !      !      !      ! Break                         *
*                                                                                *
**********************************************************************************
```

In the "Controller selection" mask the user chooses the controller he wishes to work with using the COM software. For this purpose, the function keys offer the following options:

<F1> Controller structure R64
<F2> Closed-loop controller IP 252
<F8> Break

The assignment of the function keys <F1> to <F7> in the mask shown above depends on the existing packages.

After a module has been selected the preset mask is displayed.

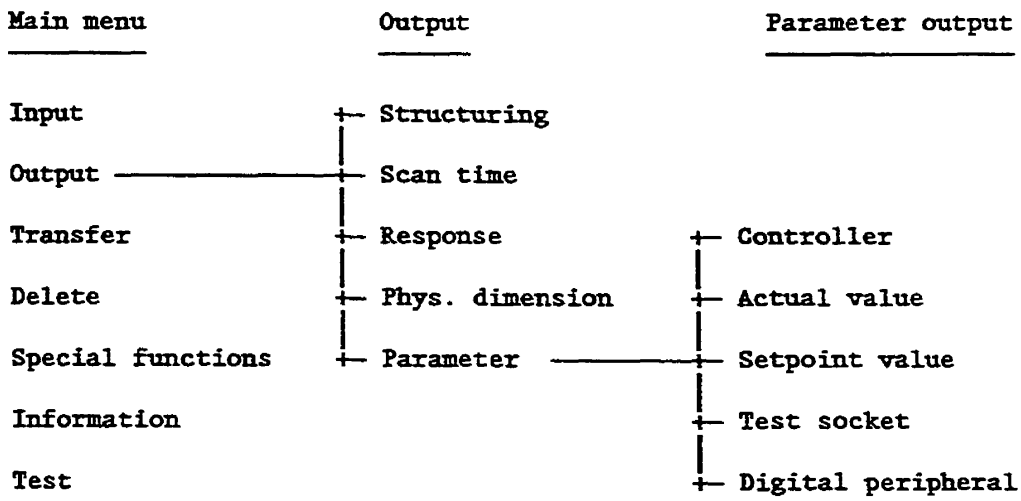
To exit COM REG press the break key <F8>. Then the command interpreter is loaded which will display again all available packages. To exit S5-DOS press the function key <F8>. This command must be confirmed by activating the <Enter> key.

3.2 Selecting Functions

COM REG is operated by means of the alphanumeric keyboard, the cursor keys, the eight function keys, the <Break> key, the <Enter> key, and the <Hard copy> key. All other STEP5 specific keys are ineffective.

When you press an ineffective key the message "BLOCKED KEY" is displayed; afterwards you can correct the input. Keyboard entries are closed by pressing the <Carriage return>. After that the user is prompted to enter further data or, the next menu is automatically displayed.

COM REG is structured hierarchically, i. e. there are different levels on which the program may run. The upper level is the main menu. When the user selects a function from the main menu (for example output) a new menu is displayed from which the data that is to be output may be selected. When the user selects, for example, "PARAMETER" the third level is entered. The following diagram shows this in detail:



Normally a function is closed by pressing the function key <F7> (Ready); then the program returns to the next higher level. The entries or changes made are stored. When a function is cancelled by pressing <F8> the program returns to the next higher level without storing the entries or the changes.
As with other operator areas the <Break> key and the <Carriage return> are effective under COM REG. Often the <Break> key functions the same as <F8> and the <Enter> key functions in the same way as <F7>.
When the hardcopy key is pressed the screen contents may be output at any time via a connected printed.

3.3 Entry Fields

You enter data into the entry fields by means of the alphanumeric keyboard and close the entries by pressing the <Enter> key. Before pressing this key you can move the cursor within the entry field and correct wrong entries.
Before you start to enter data into the entry field the cursor is positioned in the leftmost position of the field. When the entry field is blank, data can be entered left-justified. After you have pressed the <Enter> key COM REG displays the entry right-justified. When the entry field is not blank you can enter data in one of two ways:

- You enter the new value left-justified. Since the previous entry is still displayed, right-justified, it is not overwritten completely. A blank must follow the last character of the new entry so that COM REG can recognize it. After pressing the <Enter> key COM REG accepts the new value and displays it right-justified. In the following example the value "10.5" should be replaced by "9.5". (Blanks are represented by "_")

  Before the entry is made: "_10.5"
  Before pressing the <Enter> key: "9.5_5"
  After pressing the <Enter> key: "__9.5"

- You overwrite the old value with the new one in such a way that the position of the decimal point is maintained, i. e. the new value is entered right-justified. The advantage of this method is that only those characters that really have to be changed are entered. In the previous example "10" is overwritten with "_9".

  Before the entry is made: "_10.5"
  Before pressing the <Enter> key: "__9.5"
  After pressing the <Enter> key: "__9.5"

When you wish to enter parameters of different time units or data types COM REG offers possible time units such as ms, s, hm or different data types such as IFW, XW, PW, FW in the function key menu, during completion of the entry field. The default unit can be changed by means of the appropriate function key before pressing the <Enter> key.

3.4 Types of Parameters and their Input Formats

There are different types of parameters under COM REG and for each type of parameter only particular input formats are allowed. In the following table all types of parameter are listed:

Type of Parameter	Dimension	Sign	Decimal places	Range
Controller no.	–	no	0	1 – 8
DB no.	–	no	0	2 – 255
FB no.	–	no	0	0 – 255
Bit value	–	no	0	0/1
Address	–	no	0	0 – 254
Fixed point	–	yes	2	+/–100.00
Percent value	%	yes	2	+/–100.00
Times	ms	no	1 0	0.0 – 999.9 0 – 9999
	sec	no	3 2 1 0	0.000 – 9.999 0.00 – 99.99 0.0 – 999.9 0 – 9999
	hm	no	2	0.00 – 59.59
Determination of physical dimension	6 optional characters	yes	4 3 2 1 0	0.0000 – 1.0000 0.000 – 10.000 0.00 – 100.00 0.0 – 1000.0 0 – 10000
Dimension–dependent parameter	6 optional characters	yes	as determ. above	format determined above

Dimension–dependent parameters are parameters with the same unit as the values to be controlled, for example the setpoint value, the error signal, the setpoint value error limits, the processed actual value. For further information see chapter 4, section "Input/Entering a standard controller/Input characteristic".

3.5 Recommendations for the Procedure

You can project a controller by means of COM REG according to the steps described below:

- Enter the controller structure and parameters into a program file.
- Create the controller list (only for controller structure R64)
- Transfer the data blocks (and possibly the function blocks) from the program file to the module.
- Online test of the controller; optimize the parameters.
- Change the structure with "Output", if required, and test it again.
- Store the tested controller data block in the program file.
- Program the data blocks in EPROMs, if required.
- For documentation, print the controller structure and parameters.

3.5.1 Entering a Controller

In order to structure a controller the "INPUT"-function is selected from the main menu by pressing function key <F1>. In order not to lose any data, in the event of interference, it is advisable to initially enter controller data blocks into a file and not send them directly to the module. When you enter data COM REG guides you with masks so that you can enter the data in a suitable order:
- structuring
- determining the sampling time [1])
- determining the controller response
- determining the physical dimension
- parameterizing the branches and modules that were switched on during structuring

When a controller is being entered no values are preset by COM REG in order to avoid unintended controller functions being activated. For an appropriate parameterization of controller structure R64 it is necessary to enter values $>< 0$ for the following parameters:

- Scan time [1])
- Time base of the controller list
- Upper limit (of the controller)
- Lower limit (of the controller)
- Positive increment limit
- Negative increment limit
- Adjustment factor on a continuous controller with pulse/pause output
- Gain of the actuator adjustment
- Minimum pulse duration on a step or continuous controller with pulse/pause output
- Number of vertices of the polygon curve
- Distance between the vertices of the polygon curve
- Number of setpoints of the setpoint sequence
- Filter time constant
- Increase at the ramp-function generator
- Number of limit values of the limit monitors
- Measuring point number for limit monitors and test sockets

[1]) Here, 'sampling time' and 'scan time' have the same meaning

3.5.2 Creating the Controller List (only for R64)

The system program of the processor must be informed in the reserved data block DB2 in which sequence each controller is called. Therefore, after the creation of the data blocks, each of which contains a controller structure, the controller list must be created by COM REG.

3.5.3 Transfer to the Module

In order to be able to test the controller the data block should be transferred from the file to the module using the function <F4> "TRANSFER". When working with controller structure R64 the function block (FB102) and the controller list (DB2) must be transferred along with the data block in which the controller structure is stored.

3.5.4 Online Test of the Controller

After starting the PC the controller can be tested by means of the function <F8> "TEST". It is recommended to ensure that the controller receives the correct values from the input module by selecting a test point table where the different test point values are visible at a glance.

You can optimize the parameters online during the controller test by overwriting the current parameter using the function "FORCE".

3.5.5 Changing the Structure

Since only the parameters may be changed during test you have to exit the test and select the function <F2> "OUTPUT" in order to change the structure. It is reasonable to change the data blocks at the module (in STOP state) since it is there that the parameter changes, resulting from test, will be accepted, in contrary to the data block on file. After a structure has been changed the processor must be restarted. The user may now change the masks completed by "INPUT". In contrary to input, the user won't have to follow the complete sequence of masks, but may select directly the masks to be changed. The "Test" and "Change" steps are repeated until the desired controller characteristics are obtained.

3.5.6 Transferring to a File

After the test all data blocks that have been changed should be saved in the file using the function <F4> "TRANSFER".

3.5.7 Programming the EPROMs

When you work with the IP252, COM REG offers the possibility to program the created data blocks into EPROMs using the function <F4> "TRANSFER". When using the "controller structure R64" the programming of EPROMs is only possible using the STEP5 program package.

3.5.8 Documentation

After you have finished projecting the controller, the structures and parameters should be printed. To do this, select <F8> "PRINT OUTPUT" using the <F2> key. In addition to the parameters, a cross reference list with the input/output modules used by each controller can be printed.

3.6 Possible Sources of Errors

This section attempts to list all possible problems and sources of errors so that you may avoid them after having studied this section.

- If there are notes on the delivered program version at the beginning of this manual, please pay attention to the remarks given!

- A valid filename should be preset when blocks are to be read from or written onto floppy/hard disk. The valid form of a filename is: "X:YZZZZZST.S5D" where "X" is a valid drive name (e. g. "A"), "Y" is a capital letter and "Z" a capital letter or a digit.
  When you do not enter a filename or when you enter one incorrectly COM REG cancels the access to the floppy/hard disk and displays the message "ERROR EXTERNAL STORAGE". If this error occurs when you are writing a data block this block can be saved because the data is still in the "programmer" medium, although the input medium "FILE" was selected. You can save this data block from the programmer by transferring it into a valid file by means of the function "TRANSFER".

- When you work with the controller structure R64 and the IP252 you should store the data block for both devices in different files so that they are not mixed unintentionally!

- After disconnecting and reconnecting the cable between PG and PC, when a power failure on the PC occurs, or following a cold restart of the R-Processor a transfer error occurs that is detected during the next communication between PG and PC. COM REG then cancels the transfer and displays the message "PC <-> PG TRANSFER ERROR". When the cancelled function is called again the connection may be established.

- In order to increase data security each data block is almost always simultaneously sent to the PG storage as you process a data block (input, output, transfer). It is this PG storage that always contains the last data block processed, even if it

was input to the module or into a file.
When you cancel the input or output function unintentionally the PG storage retains the data block until you work on a different one; therefore the block can be saved by transferring it from the PG to the module or the file.

- The input function offers "PG" as the target for a data block. Because of the characteristics of the PG storage mentioned above this data block is overwritten without warning when you work on another DB. Therefore you are advised against selecting the PG as destination for input!

- When you cancel the function "OUTPUT" (except "OUTPUT/PARAMETER") not only the output of the mask but also output of the complete data block is cancelled with the result that all other changes (e. g. parameters) are lost. This block can only be saved by means of the PG storage mentioned above.

- The encoding switch of the printer PT88 should be set as it was when the printer was delivered, i. e. all switches "ON".

  Note: An incorrect printout indicates that the parity of the printer interface is not set correctly.

3.7 Checklist for Troubleshooting

The following list contains the most frequent causes of the most important possible user errors, from call of COM REG to controller test.

PCs are highly complex, high performance devices where many functions must run correctly at the same time, in particular in the multitasking mode, in order to realize the required action. The large number of available functions may lead to a large number of possible operator errors. It is therefore reasonable to restrict the troubleshooting of particularly stubborn and apparently inexplicable faults to the necessary modules and functions. Experience has shown that faults are not necessarily found in the function where they appear but in a different one.

COM REG cannot be started

- Does the floppy/hard disk contain all tools and drivers required?
- Were all tools and drivers taken from the same ZEFU package?
- Does the floppy/hard disk contain all COM REG files?
- Were all COM REG files taken from the same COM REG package?

You cannot exit a COM REG mask

- Is there still a wrong entry in the input field (e. g. entry of time base or determination of input characteristic)?
- Is it possible to leave the mask by pressing the <Break> key or the <Enter> key?
- Is the keyboard locked with the key switch?

Problems when working with the program file

- Is the name of the default program file in the correct format?
- Is there enough space on the floppy/hard disk?
- Is the attribute of the selected program file "Read only"?
- Is the selected program file stored in a different user area? (User areas can read files with the attribute "SYS" from user 0 but cannot write into them)
- Is the disk formatted?
- Is the disk drive closed?

Problem when accessing the module

- Is the default of COM REG "Online"?
- Is the PC supplied with power?
- Is the module plugged in?
- Is the cable connection between the PC and the PG correct?
- Is the connection cable servicable?
- Are the correct S5DOS drivers used?

The following module-specific notes refer to the controller structure R64:

Following cold restart, the processor does not enter the RUN mode

- For multiprocessing: Is there a valid DB1?
- For multiprocessing: Is the coordinator installed? Are all jumpers set correctly? Is the correct mode set?
- Is the controller function block missing?
- Is a data block that was entered in the controller list missing?
- Do all accessed input/output modules exist? (Print out cross reference list!)
- Is the correct base address set for all modules?
- Do all modules receive the enable and supply voltage required?
- Are the modules plugged in correctly?

  Chapter 5 in which the controller structure R64 is described gives information on how controller faults (and acknowledgement delay on controllers) can be identified and evaluated.

The processor is in the 'RUN' mode but the controller does not function correctly. Then refer to the following checklist:

- After any transfer/change on the module a restart must be carried out
- The controller must be entered in the controller list
- The controller must be enabled (see Special functions/Controller processing)
- If you entered an address in digital inputs: Was this entry supplied with meaningful values (by STEP5 programs, CP, or switch)? E.g., flagwords must be filled with values by the CP or the STEP5 program, special switches connected with I/O modules. Special attention is to be paid to the position of the bit values for "Inhibit controller", "Final position ON achieved", "Final position OFF achieved", "Manual operation", and "Maintain regulating variable constant".
- The bits mentioned above could have been set in test.
- Are all parameters preset with meaningful values? (see chapter "Entering a controller")
- The address of an ADC (DAC) may not be maintained in a switched off branch. (You should activate the appropriate branch using the structure switch to check it!)
- Do the input/output modules work correctly? Were they parameterized for the correct range?
- Do the input/output modules work with the correct data format? (Negative values must be represented in two's complement!)
- Does a STEP5 program describe an incorrect data word in the controller data block?

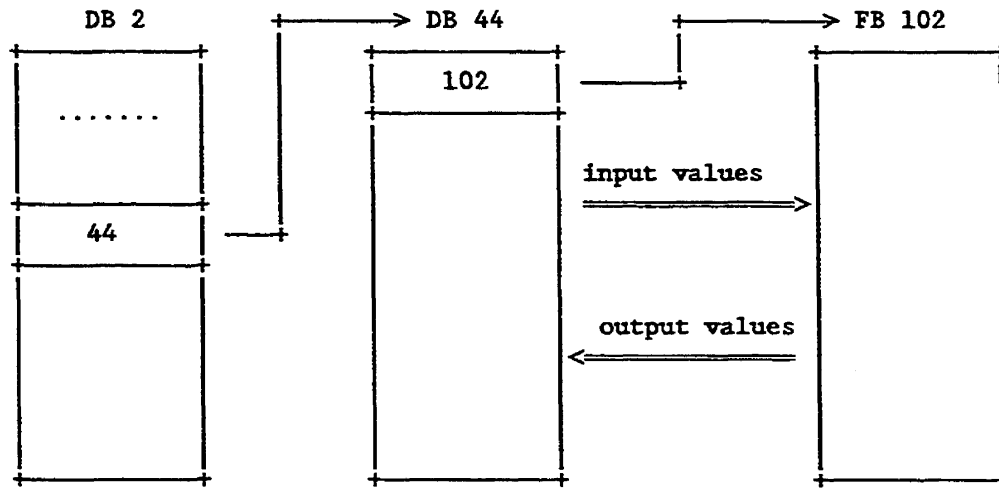
Values entered during the test are not accepted or overwritten

- Not all parameters may be changed during the test!
- Were the parameters entered in the correct format?
- Was an input made with the addressing for the digital inputs? Then the digital inputs cannot be controlled during the test!

4 Projecting the Controller Structure R64

4.1 Processing the Controller FB

The processor executing the control process is a word processing control processor that recognizes the common STEP5 functions and organization blocks (OB1, OB13, etc.).
Only when data block DB2 (controller list) containing usable data is available, following restart, will the time-controlled call-up cycle of the controllers be started by the system program.
Each controller is represented by a data block. The system program receives information from the controller list as to which controller is called with which scan time. When calling, the appropriate data block is selected. In data word 0 the number of the function block, which contains the control algorithm (e.g. 102 for the controller structure R64) is entered. This function block which is called by the system program processes the projected controller structure using the data in the data block. During this procedure all input values (setpoint, actual value etc.) can be taken from the data block. The computed output values (e.g. regulating variable) are stored in the data block. The process image (takes input values from the peripheral to the data block and output values from the data block to the peripheral) is produced by the system program, if there is an address given in the appropriate parameters. The input/output values can also be supplied by a STEP5 program. In this case no addresses may be allocated with the respective parameters so that the system program does not execute a process image for these parameters. The function block that contains the controller struc-ture must exist only once for all controllers because all controller specific data is stored in the data block.



Those times not marked for controller processing are available to the STEP5 program. During these times, optional STEP5 programs may run to process input and output values of controllers. Thus multi-loop control systems may be created.

The higher the processor load, i. e. the more controllers that are entered in the controller list, the longer the cycle time of the STEP5 program. The STEP5 programs run asynchronically compared with the controllers.

The following blocks must be available in order to enable a control procedure:

- DB 2 controller list
- DB x a controller data block ($3 \leq x \leq 255$)
- FB 102 function block with the control algorithm

Since a STEP5 program always runs in the background (idle cycle if no program was input) the operating system executes the process image for the digital peripherals (PB 0 ... PB 127). When the control procedure uses input/output devices whose addresses are less than 128, the process image of the STEP5 program overwrites their process images. In order to avoid this a DB 1 that does **not** assign input or output to the STEP5 program should be entered in the processor. In order to execute a control procedure without STEP5 software and without any knowledge of STEP5, a "dummy DB1" which prevents the cyclic STEP5 program from accessing peripheral addresses < 128 is delivered together with the controller structure R64.

Note: - The file "REGR64ST.S5D" contains the control algorithm as function block FB102. When you already use an FB102 on your system you can assign another block number to the delivered one using COM REG or "LAD CSF STL". You should enter this new name in the controller data block.

- **All changes to the structure and the controller list are transferred correctly only when the processor is restarted. Controllers must not be changed using the function "OUTPUT", during processing! Only those changes that the function "TEST" offers are permitted. Before changing the structure the processor must be brought to the "STOP" mode in order to accept the changes afterwards by means of the function "RESTART".**

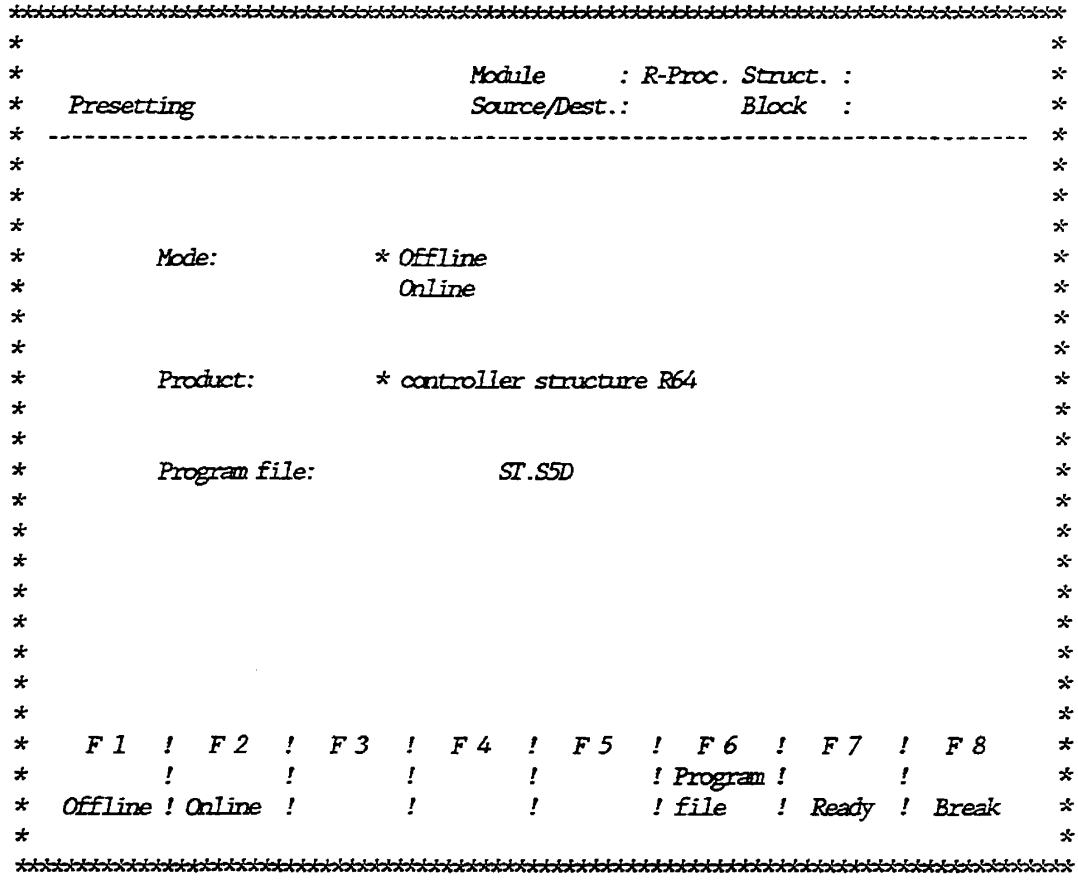
- In the following sections the processor is also called "module" following the conventions of COM REG.

You can find more detailed information on the processor and its system program in the User's Guide for the processor and in the description of the controller structure R64.

Please pay particular attention to the application example given in the controller structure R64 description and to the explanation of all terms used by COM REG!

4.2 Preset

After S5DOS is started and COM REG is selected (see section "Starting COM REG") COM REG displays the preset mask.



Select the operating mode and the program file from the Preset mask.

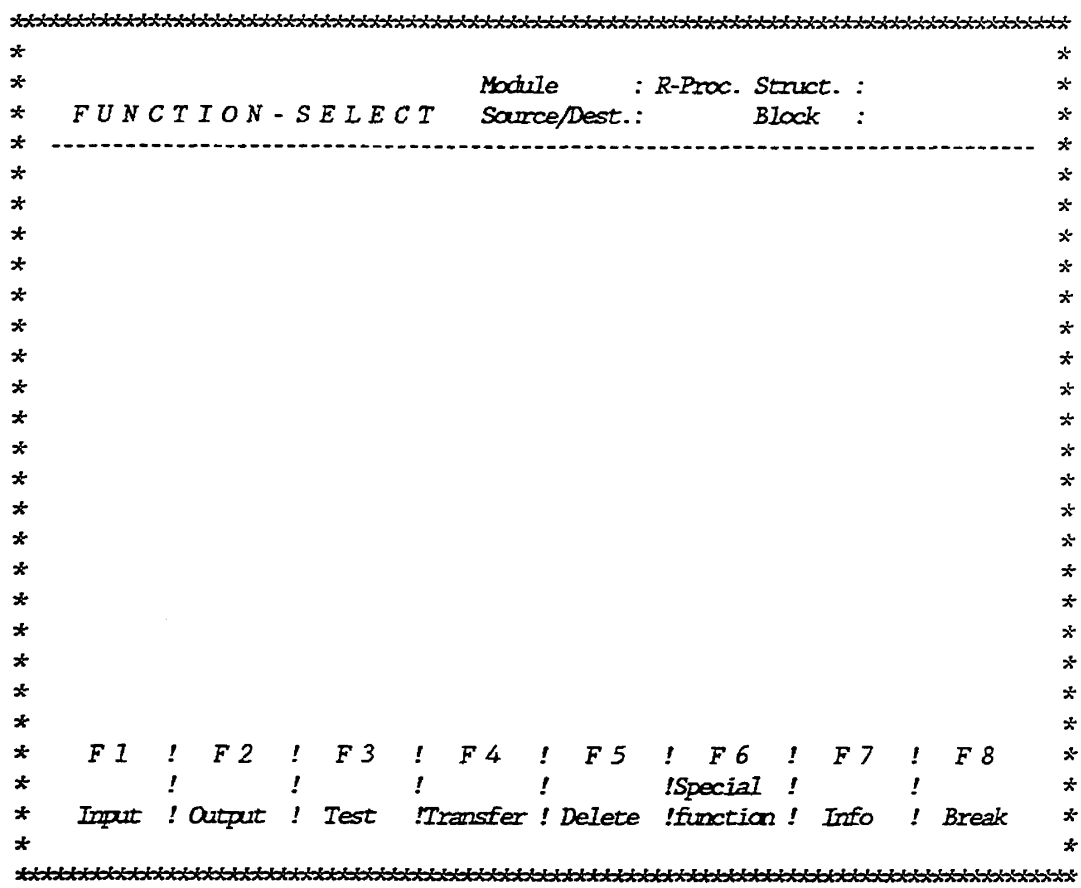
- "ONLINE" means that the PG can communicate directly with the module. You can select this operating mode only when the module is ready to operate and when it is connected with the PG. In the online mode, controllers can be projected directly to the module.
  If no PC is provided, the controller data blocks can be created using COM REG and stored in a program file. In this case, you should select the operating mode "OFFLINE".

- COM REG can store data and function blocks in a program file. All following accesses to external storage (floppy or hard disk) refer to the blocks in the specified program file. You can enter the program file name after you have pressed the <F6> key. The format of the filename is "X:YZZZZZST.S5D" where "X" is the drive name (e. g. "A:", "B:", or "C:"), "Y" is an upper case and "Z" an upper case letter or a digit. The name should have the number of characters specified above, otherwise it is filled with "@". The entry is completed with <CR>.

Example: "B:KILN02ST.STD"
The name of the program file should refer to the project. In the specified program file, STEP5 DB, FB and controller DB and FB can be stored together.

By pressing the function key <F7> you can exit the preset mask and enter the main menu. To return to the module selection press <F8>.

4.3 Main Menu



The main menu is the highest mask level of COM REG. From there you can select the following functions:

INPUT	- enter the controller data blocks - enter the controller list
OUTPUT	- change the existing controller data blocks - change the existing controller list - print the controller structure - print the cross reference list
TEST	- test controllers online: starting-up, operating, monitoring and changing the parameters
TRANSFER	- transfer the data blocks and function blocks between the module (programmable controller), program file and block memory of the PG
DELETE	- delete data and function blocks in the module and program file - delete the program file and all blocks within
SPECIAL FUNCTIONS	- cold restart of the processor - warm restart of the processor - stop the processor - compress the data blocks in the processor

- change the presetting
- controller processing

INFO
- list of contents of all control loops on the module, program file or submodule
- SYSID module
- SYSID submodule (frame number, version)

4.4 Input

Using the function "INPUT" you can project controllers by creating a controller data block. Additionally a controller list (DB2) can be created using the function "INPUT".

After selection of the function "INPUT" by pressing the function key <F1> the main menu offers the possible destinations where entries may be stored automatically:

<F1> program file
<F4> module
<F5> programmer (PG)

Note: - It is advisable, even in online mode, that the entries are stored in the program file first and then transferred to the module so that in the case of disturbances at the PC (e. g. unintentional removal of the module etc.), the data is still available.

- You should never enter the data directly into the PG because its data block storage can only hold one block, as described by STEP5. The block stored in the PG is lost without warning when you work on the next one (e. g. input, output, transfer).

In the following mask input of a controller data block and a controller list is offered:

<F1> controller data block
<F8> controller list

It is only realistic to create a controller list after the creation of all controller data blocks. Following the input of a controller data block or the controller list, described below, the mask is displayed again so that you can make further entries.

The function "INPUT" is terminated by pressing the Break key!

Note: If COM REG realizes that an entered data block already exists on the specified destination medium the following question is displayed: "OVERWRITE EXISTING DATA BLOCK? (Y/N)". If you do not wish to overwrite the existing data block, you can save the newly created one. It still exists in the PG memory even though you specified the program file or the module as destination. In order to save the new data block enter "N" and copy the data block from the PG to the program file or to the module by means of the function "Transfer". The data block number may also be changed during this procedure.

4.4.1 Entering a Controller DB

The first step on entering a controller DB consists of specifying a data block number between 3 and 255. A controller DB entry is generally carried out in a linear way, i. e. all inputs are requested one after the other, without branching, in the form of masks. Entries are offered in the following order:

- entry of structure
- entry of scan time
- entry of controller behaviour
- entry of function block number
- entry of controller and area name for CP526
- definition of dimensions and numerical range
- parameterization of selected branches

Note: Parameterization offers only those branches and modules that have been selected during the preceding structuring operation. If you realize during parameterization that you have forgotten a module when structuring the controller, exit the input and change the structure using "OUTPUT", in order to be able to parameterize the desired module.

4.4.1.1 Controller Structuring

After entering the data block number the structuring mask is displayed. By means of the structuring function, you can establish an individually required controller structure by setting the structure-switches. Based on the given primary structure you can activate specific controller branches.

The structure-switches have the functions of either on/off or change-over switches. The switches can only be set by entering "0" or "1". In case of on/off switches (e. g. "ramp-function generator"), entry of "1" causes the module to be activated. When entering "0" the module remains switched off. Changeover switches are marked in the text by a slash "/" (e. g. "continuous/step"), where one of two selections is possible: "0" for the alternative preceeding the slash or "1" for the alternative following the slash.

```
*****************************************************************************
*                                                                           *
*   Input                          Module     : R-Proc. Struct. : Control.  *
*   S T R U C T U R I N G          Source/Dest.: File    Block   : DB  003     *
*  ------------------------------------------------------------------------ *
*                                                                           *
*                                                                           *
*      1: Controller          1          2: Actual value          1        *
*       Continuous/Step       0           Validity check          0        *
*         Manual value PG/ADC 0           Filter actual value     0        *
*         Analog/Digital output 0         Polygon curve           0        *
*         Standard/Upgraded   0                                            *
*           Separate D-input  0                                            *
*           Interference input 0                                           *
*           Manual value PG/ADC 0                                          *
*           Manual value PG/ADC 0                                          *
*         Cont./mark-space    0                                            *
*           2-/3 Point controller 0                                        *
*             Anal./Dig. output 0                                          *
*             Anal./Dig. output 0                                          *
*           Actuator adjustment 0                                          *
*                                                                           *
*                                                                           *
*                                                                           *
*    F 1  !  F 2  !  F 3  !  F 4  !  F 5  !  F 6  !  F 7  !  F 8          *
*         !       !       ! Scroll ! Scroll !       !       !              *
*         !       !       ! up     ! down   ! Help  ! Ready ! Break        *
*                                                                           *
*****************************************************************************
```

After each entry of "0" or "1" the cursor shows the next **logical** switch, i. e. only switches of branches that have been activated can be moved. Thus, you are spared unnecessary and confusing entries. If you select, for example, the continuous controller "standard" by '0' the cursor will never point to the item 'interference input' since the latter is optional only for the upgraded controller.

The cursor may also be moved to the respective switches using the four cursor keys. Since the list of switches is longer than the available space, the list can be scrolled up or down using the function keys <F4> and <F5>.

Notes: - If the switch indicated by the cursor position is preset to "0" and you do not wish to change it, you are nevertheless recommended to enter "0". As a result, the cursor is positioned to the next logical switch by the program. Entering the digits one or zero exclusively prevents you from actuating a switch several times or not at all. If you move the cursor using the cursor keys you do this without prompting from the program!

- The change-over switches "Manual Value PG/ADC" and "Analog/Digital Output" are given several times in the list for reasons of program structure. However, COM REG sets the cursor to the respective switch only once, depending on the specified structure.

- The branches "Controller", "Actual Value" and "Digital Addresses" cannot be switched off, thus they are present in each controller.

- When structuring you can only set structure-switches, digital inputs may not be set. Relays can be controlled during online test only.

In order to show the relationships between the structuring mask and the switches of the controller block diagram, a list with the corresponding assignments is provided in the description of the controller structure R64 (chapter 3, Tables).

Structuring operations are terminated by pressing the function key <F7> after which the next entry mask is displayed. Entry can be cancelled by pressing <F8>; all preceding entries are lost.

Note: If the entry is not cancelled during structuring but at a later point in time, COM REG attempts to store the data block along with previous entries at the destination medium! Only if a data block with the same number exists the question "Block x overwrite (Y/N)" is displayed.

4.4.1.2 Scan Time

The "scan time" mask enables you to enter individual scan times for each controller.
The scan time is the period of time in which the input values (actual value, setpoint, etc.) are read in once and the output values (e. g. manipulated variable, limit) are evaluated and output to the process.

```
*****************************************************************************
*                                                                           *
*  Input                          Module     : R-Proc. Struct. : Control.   *
*  S C A N - T I M E              Source/Dest.: File    Block   : DB  003    *
*  ------------------------------------------------------------------------ *
*                                                                           *
*                                                                           *
*                                                                           *
*        Comment :   For the quasi-continuous controller design the         *
*                    recommended scanning time value is 10% of the          *
*                    dominating time constant of the controlled             *
*                    system.                                                *
*                    The scanning time for the R-Proc. is a multiple        *
*                    of 10 ms and may not be less than 20 ms.               *
*                                                                           *
*                                                                           *
*        Scan time:          100 ms                                         *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*   F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                    *
*       !      !      !      !      !      !      !                          *
*       ! Input !     !      !      !      ! Ready ! Break                  *
*                                                                           *
*****************************************************************************
```

The scan time may be entered after pressing the function key <F2>. The notes on the sequence of keys during entry, given in the section "Input fields" should be observed. The following points are important for the numerical value:

- In the quasi-continuous controller outline, the recommended scan time value amounts to 10% of the dominating time constant of the system to be controlled.

- The scan time must be an integer multiple of 10 ms, i.e. each at least 20 ms.

- For step controllers and continuous controllers with pulse/ pause output, the scan time must be an integer multiple of the minimum puls duration.

- A processor can process a maximum of eight different scan times.

- The smaller the scan time the higher the processor load. Therefore, fewer controllers can be processed at the same time. In other words the cycle time of a STEP5 program in the background is becoming longer.

- In the controller list which is to be entered later, the maximum common divisor of the scan times of all continuous controllers must be entered as time base. If an unusual value is specified for the scan time it should either be corrected later, or the time base should be selected to be unnecessarily

small causing an increased processor load and STEP5 programs to be processed more slowly.
(It is not possible, for example, to enter a controller with a scan time of 30 ms and another with a scan time of 50 ms into the controller list at the same time; the maximum common divisor is 10 ms while the minimum time base must be 20 ms.)

- After changing the scan time the controller DB in DB2 must be switched off, then on again. Then, a cold restart is to be executed for the processor.
(Further information can be found in the description of the controller list and the controller structure R64.)

Use function key <F7> to terminate entry of scan time and step to the next entry mask. Entry may be cancelled by pressing <F8>; the data block is then created.

4.4.1.3 Controller Behaviour and FB Number

This mask requires four entries. The cursor may be moved by use of the keys <cursor up> and <cursor down>. Yes/no entries are to be made by means of the function keys <F1> and <F3>. The function block number should be entered via the keyboard and terminated by pressing the <Carriage Return>.

```
*************************************************************************************
*                                                                                   *
*  Input                          Module      : R-Proc. Struct. : Control.          *
*                                 Source/Dest.: File    Block   : DB   003          *
*  ------------------------------------------------------------------------------   *
*                                                                                   *
*                                                                                   *
*    Cntl.-Behaviour:                                                               *
*    ----------------                                                               *
*        If the cntl. is not operational                                            *
*        the outputs are set on zero:                 Yes                           *
*                                                                                   *
*        The updating of the controller output                                      *
*        follows immed. the controller operation:     No                            *
*                                                                                   *
*        Match format to measuring range 4 - 20 mA                                  *
*        (for modules without matching):              No                            *
*                                                                                   *
*    Input of FB-Number:                                                            *
*    -------------------                                                            *
*        The controller structure R64 is in FB:        0                            *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*   F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                            *
*       !      !      !      !      !      !      !                                  *
*   Yes !      !  No !      !      !      !Ready !Break                             *
*                                                                                   *
*************************************************************************************
```

With the PG you can switch on and off individual controllers during operation without influencing the other controllers (Special functions/controller enable). For this status and for the "STOP" status of the processor you can specify, with the first entry, if in case of a controller stop all outputs of the controller should be set to zero or if they are to remain in the same states as before the controller stop.

By means of the second entry, you may specify whether the controller output is to be updated immediately following the controller processing, or time controlled, i. e. together with the controller input.
In the first case, dead time is short but not constant due to a variable program execution time. In the second case, dead time is constant but of the same length as the scan time. This entry is relevant for those controlled systems where the time constant is about the same duration as the scan time. You then have to decide which entry to select, depending on the controlled system.

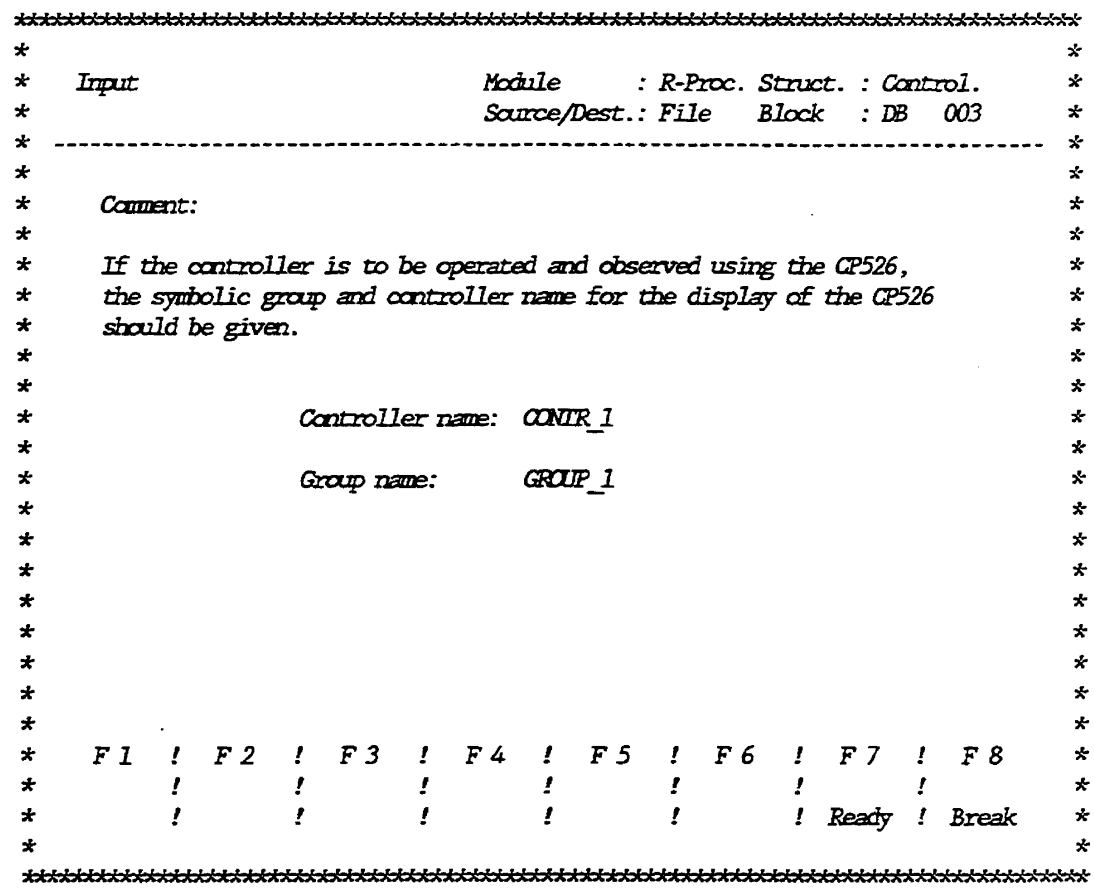
With the analog input modules 6ES5-460-4U and 6ES5-465-4U, the

measuring range 4 ... 20 mA is not mapped to 0 ... 100%, but to 25 (1000H) ... 125% (5000H).
You may select a suitable adaptation via the controller. 25% are then subtracted from all ADC input values of this controller.

Entering the function block number informs the processor which function block contains the control algorithm. Therefore this entry is mandatory. (The controller structure R64 is supplied with the function block number 102.)

Press the function key <F7> to terminate this entry and step on to the next entry mask. Entry can be cancelled by use of <F8>; the data block is then created.

4.4.1.4 CP526 Adaptation



This mask should be completed only if you wish to operate and monitor the controller using CP526.
Entries for controller and area name are used for display building of CP526 and they are stored in the data words 192 .. 199 of the data block. The entry comprises of eight characters maximum; letters, digits, spaces and underscores ("_") are allowed.

Press the function key <F7> to conclude the entry and go on to the next entry mask. The entry may be cancelled by pressing <F8>; the data block is then created.

4.4.1.5 Input Characteristic

Analog input/output modules work with normalized signals, e. g. with 4..20 mA for the analog part and with the range of values of 0000h...4000h on the digital part. With a temperature of 300 ° C a transducer provides a current of 12 mA. The analog input module transforms this current into a hexadecimal value, e. g. 2000h. This is the value that the controller structure R64 receives from the analog input. In order to work with COM REG using the actual numerical values and units you should tell the controller by means of this mask how to interpret the hexadecimal value of 2000h, i. e. the characteristic of the input range should be determined.

The controller structure R64 assumes that the input/output modules represent the positive range of values of 0..100% as digital value of 0000h..4000h. Negative values are expected in two's complement (for 4...20 mA = 1000U...5000U see chapter 4.4.1.3).

Three entries are required in order to specify the characteristic:

- physical dimension (6 characters maximum)
- numerical value of the controlled condition to be provided 0 % (=0000h) by the controller in the data format
- numerical value of the controlled condition to be provided 100 % (=4000h) by the controller in the data format

Notes: Many analog input/output modules have a resolution of 2048 units. On the digital part the modules work with 16-bit-words where the three least significant bits are not evaluated or set, i. e. the units that correspond to the analog value are shifted 3 bits to the left (multiplied by 8). Therefore, for a resolution of 2048 units, on the digital part is represented by 16384, corresponding to 4000h.

Example 1:

range of actual value:		-50 .. 150 Degr_C
transducer provides		4 .. 20 mA
analog input provides		0000h .. 4000h
		0 .. 2048 units
controller interprets		0 % .. 100 %

During the work with COM REG, in the initial position the following entries (in italics) apply:

Dimension D1 : *Degr_C*

0 Percent corresp.	*-50.0*	Degr_C
100 Percent corresp.	*150.0*	Degr_C

```
*****************************************************************************
*                                                                           *
*  Input                          Module      : R-Proc. Struct. : Control.  *
*  P A R A M E T E R              Source/Dest.: File    Block   : DB   003    *
*  ---------------------------------------------------------------------    *
*                                                                           *
*                                                                           *
*    Input of desired physical dimension:                                   *
*    ====================================                                   *
*                                                                           *
*                                                                           *
*      Dimension D1:   DEGR_C                                               *
*           0 Percent corresp.     -50.0 DEGR_C                             *
*         100 Percent corresp.     150.0 DEGR_C                             *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*   F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                    *
*       !      !      !      !      !      !      !                          *
*       !      !      !      !      !      ! Ready ! Break                  *
*                                                                           *
*****************************************************************************
```

Both input values that specify the characteristic of the input range determine the number of decimal places with which all dimension-dependent parameters are input and output. The following values apply for the valid input range:

0.0000 - 1.0000
0.000 - 10.000
0.00 - 100.00
0.0 - 1000.0
0 - 10000

The five formats, shown above, have the maximum resolution, i. e. the greatest possible number of decimal places. The user may ignore a part of the resolution by specifying less decimal places than permitted for this format. For example 1, the entry of -50..150 Degr_C would also be correct. Since most of the analog input modules can also process negative current and voltages the input range in example 1 is -250..150 Degr_C.

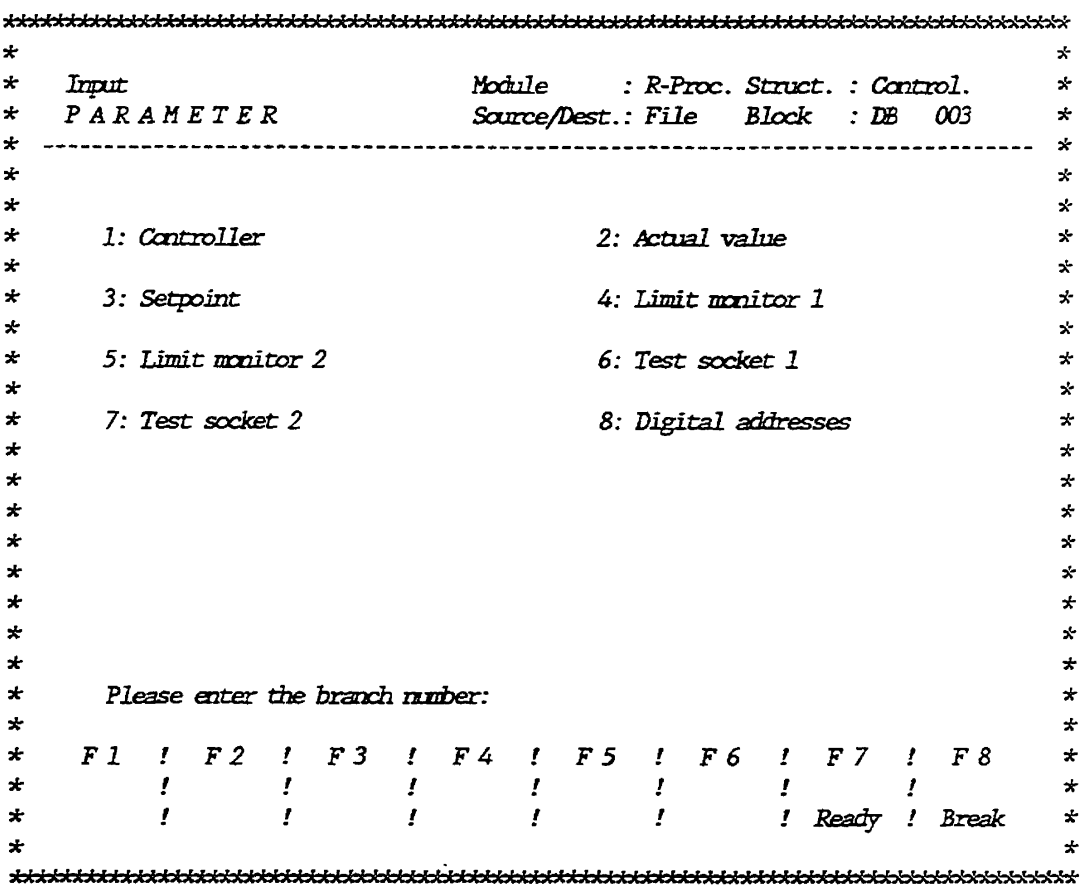
Note: - The algorithm specified in the input characteristic, with which the controller is to process the input/output, is valid for all dimension dependent parameters. Therefore, all input/output modules should have the same data format.

- When a digit whose format is not mentioned in the list above, is entered for the 0 % or 100 % values, the entry cannot be completed by <CR>, <F7>, <F8>, the <Enter> key or the <Break> key. Only after correction of the format is further processing possible.

- **For all further entries of dimension dependent parameters the format described must be kept. On the other hand, the format or the numbers of decimal places cannot be modified after the entry of the parameters without modifying all dimension dependent parameters!**

By pressing the function key <F7> the entry is completed and the next input mask is displayed. You can break the input procedure by pressing <F8>; then the data block is filed.

4.4.1.6 Parameter Input



The first mask of the parameter input contains a list of all active branches and their numbers. Since you can select the branch you wish to parameterize first, the linearity of the input in the function "Parameter input" is interrupted. This is realized by the independence of the input for the parameterizing of the controller branches, i. e. the order of input is unimportant. This program structure enables the user to modify a branch that is already parameterized, without leaving the function "Parameter input".
The controller branch to be parameterized is selected exceptionally by entering its number by means of the alphanumeric keyboard and then pressing the <Carriage Return>. This procedure enables COM REG to also process controllers with more than eight (= number or function keys) branches.

You can exit the branch selection menu by pressing the function key <F7>. The data block entered is stored at the specified destination medium.

For parameter input, note the following:

- When you make the first parameter input you always should select branch 1 first, COM REG then guides you (function key <F7> must be activated) through all branches. Only when you wish to enter modifications should you directly select the branch to be modified.

- For several parameters no input may be required (e. g. the danger limit of the actual value need not be entered when the corresponding bit is not evaluated). Under particular circumstances complete controller branches can be jumped without input (e. g. the branch digital address when neither digital input or digital output is intended).

- In addition to the input fields the masks also contain output fields that show the selections made when structuring. The cursor can be moved to the input fields only by using the keys <cursor up> and <cursor down>. When parameters are entered the procedure recommended in the section "Input fields" is to be considered.

- Since several parameter lists need more space than provided on the screen, these lists should be scrolled up or down (automatically) while the user completes them. Pressing the function keys <F5> "Scroll down" and <F6> "Scroll up" has the same effect. In order to show the end of a list the message "List end is reached !" is displayed when the last entry is transferred. This message must not be interpreted as an error message referring to the last entry!

- When an entry is made in the address of an ADC, DAC, DO or DI (i. e. input/output module) the corresponding parameter (e. g. actual value, setpoint, digital input, etc.) is described when the controller is operated by the system program creating the process profile. **When input values are not supplied by the input modules but by the CP or the PG to be tested (e. g. digital inputs) no entry must be made in the corresponding address of the input/output module otherwise the specified value is overwritten by the process profile.** (The default provides "PW" for each address. An entry is only accepted if this abbreviation (resp. "IFW", "FW", "XW") is followed by a number. Entries may be deleted by overwriting this number with three blanks.

In the following sections the peculiarities of each controller branch are briefly discussed.

4.4.1.7 Controller Branch

```
*************************************************************************************
*                                                                                   *
*   Input                          Module      : R-Proc. Struct. : Control.       *
*   P A R A M E T E R              Source/Dest.: File    Block   : DB   003       *
*  ---------------------------------------------------------------------------- *
*                                                                                   *
*     Branch   1: Controller                                                        *
*                                                                                   *
*                                                                                   *
*     Continuous/Step                    0                                          *
*       Standard/Upgraded                0                                          *
*         Manual value PG/ADC            0                                          *
*           Constant man. value       30.00 %                                       *
*                                                                                   *
*         High limit                  100.00 %                                      *
*         Lower limit                -100.00 %                                      *
*         Proportional value            1.23                                        *
*         Integral action time          2.123 sec                                   *
*         Derivative action time            0 sec                                   *
*                                                                                   *
*       Cont./mark-space                    0                                       *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*    F 1   !  F 2   !  F 3   !  F 4   !  F 5   !  F 6   !  F 7   !  F 8          *
*  Milli-  !        ! Hours   !        ! Scroll ! Scroll ! Next   ! Branch        *
*  seconds ! Seconds ! Minutes !        !  down  !   up   ! Branch ! Selection     *
*                                                                                   *
*************************************************************************************
```

In order to put a continuous controller into operation, the values entered for parameters "High limit" and "Lower limit" must not equal zero, otherwise the regulating variable is limited to zero.

The same is valid for the parameters "POS increment limit" and "NEG increment limit" of the upgraded controller, i. e. the values should not be zero.

If an actuator adjustment is projected the gain may not be zero.

4.4.1.8 Actual Value Branch

```
*********************************************************************************
*                                                                               *
*  Input                          Module      : R-Proc. Struct. : Control.      *
*  P A R A M E T E R              Source/Dest.: File   Block    : DB   003       *
*  ---------------------------------------------------------------------------  *
*                                                                               *
*    Branch  2: Actual value                                                    *
*                                                                               *
*                                                                               *
*    Address ADC 02                 PW  128                                     *
*    High warning limit               130.0 DEGR_C                             *
*    Lower warning limit              -20.0 DEGR_C                             *
*    High danger limit                140.0 DEGR_C                             *
*    Lower danger limit               -30.0 DEGR_C                             *
*                                                                               *
*    Start-up actual value            111.2 DEGR_C                             *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                  End of list is reached !                     *
*   F 1  !  F 2  !  F 3  !  F 4  !  F 5  !  F 6  !  F 7  !  F 8                *
*        !       !       !       ! Scroll ! Scroll ! Next  ! Branch            *
*        !       !       !       !  down  !   up   ! Branch ! Selection        *
*                                                                               *
*********************************************************************************
```

The parameters for the warning limit and the danger limit need not be entered if the corresponding bits in the digital output word are not evaluated.

If a polygon curve was entered during structuring the number of vertices in the mask for the parameterizing of the actual value branch is preset with zero. Only when a number between one and ten is entered will the corresponding number of lines be displayed, which is required for parameterizing the ordinal values of the polygon curve. Although COM REG would accept the default number of zero vertices, this number is illogical; it must be set to a value between one and ten.

4.4.1.9 Setpoint Branch

```
*********************************************************************************
*                                                                               *
*   Input                          Module      : R-Proc. Struct. : Control.     *
*   P A R A M E T E R              Source/Dest.: File    Block   : DB   003     *
*  ------------------------------------------------------------------------     *
*                                                                               *
*     Branch   3: Setpoint                                                      *
*                                                                               *
*                                                                               *
*     ADC/Entry                          0                                      *
*       Address ADC 01                PW   130                                  *
*                                                                               *
*     Setpoint high limit              125.0 DEGR_C                             *
*     Setpoint lower limit              80.0 DEGR_C                             *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                  End of list is reached !                     *
*    F 1  !  F 2  !  F 3  !  F 4  !  F 5  !  F 6  !  F 7  !  F 8               *
*         !       !       !       ! Scroll ! Scroll ! Next  ! Branch            *
*         !       !       !       !  down  !   up   ! Branch ! Selection       *
*                                                                               *
*********************************************************************************
```

The parameters "Setpoint high limit" and "Setpoint lower limit" do not limit the setpoint but a bit is set in the digital output word when these limits are exceeded. When these bits are not evaluated these parameters need not be entered.

When a setpoint sequence was entered during structuring the default number of setpoints in the mask for the parameterizing of the setpoint branch is zero. Only when a number between one and ten is entered will the corresponding number of lines be displayed, which is required for parameterizing the setpoints of the setpoint sequence. Although COM REG would accept the default number of zero setpoints, this number is illogical; it must be set to a value between one and ten.

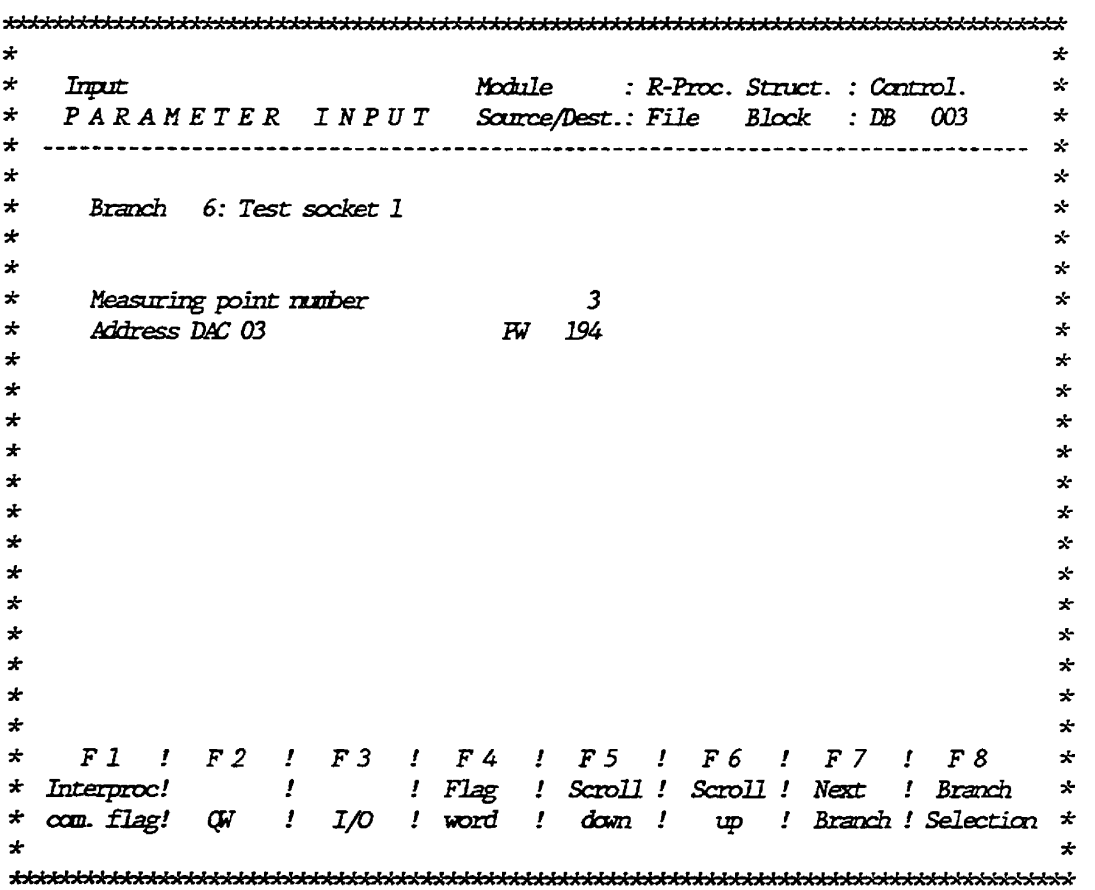
Note: When the structure key S8 is set to "1", i. e. when the setpoint is specified by a STEP5 program or the setpoint sequence, and a module address was entered in the address of ADC1, the specified input module should exist, although the values are not used.

4.4.1.10 Limit Monitor Branch

```
*****************************************************************************
*                                                                           *
*   Input                          Module      : R-Proc. Struct. : Control. *
*   P A R A M E T E R              Source/Dest.: File    Block   : DB  003    *
*  ------------------------------------------------------------------------ *
*                                                                           *
*     Branch  4: Limit monitor 1                                            *
*                                                                           *
*                                                                           *
*     Number of limit values              6                                 *
*     Measuring point number              8                                 *
*     Limit 01                          105.0 DEGR_C                        *
*     Limit 02                          110.0 DEGR_C                        *
*     Limit 03                          115.0 DEGR_C                        *
*     Limit 04                          120.0 DEGR_C                        *
*     Limit 05                          125.0 DEGR_C                        *
*     Limit 06                          130.0 DEGR_C                        *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                   End of list is reached !                *
*    F 1  !  F 2  !  F 3  !  F 4  !  F 5  !  F 6  !  F 7  !  F 8          *
*         !       !       !       ! Scroll ! Scroll ! Next   ! Branch      *
*         !       !       !       !  down  !   up   ! Branch ! Selection   *
*                                                                           *
*****************************************************************************
```

The number of limit values is preset to zero; it may be increased to 6 maximum. All limit values of the limit monitor branch refer to the same measuring point. When a positive or negative limit value is violated, the corresponding bit of the digital output word is set. The limit monitors can only refer to active measuring points.

4.4.1.11 Test Socket



The test sockets enable the user to output internal controller values to external recording instruments. By means of the simultaneous output of the read setpoint and the processed setpoint on a recorder, the functions of all modules of the setpoint branch may be checked (the same refers to the actual value branch).

In contrary to the ADC address, the measuring point number can be modified online during the test.

Note: Before a test socket can be switched off using the structure key, the address of the ADC should be overwritten with blanks; otherwise, the system program attempts to output the contents of the measuring point to the ADC. This means that the ADC should exist physically, else the processor passes into the STOP state signalling a controller fault (see chapter 5, description of controller structure R64.

4.4.1.12 Digital Addresses

```
*************************************************************************************
*                                                                                   *
*   Input                          Module      : R-Proc. Struct. : Control.       *
*   P A R A M E T E R              Source/Dest.: File    Block   : DB   003         *
*  ---------------------------------------------------------------------------      *
*                                                                                   *
*     Branch  8: Digital addresses                                                  *
*                                                                                   *
*                                                                                   *
*     Addr. of digital output DO 01    PW     2                                     *
*     Addr. of digital input  DI 01    PW                                           *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*    F 1  !  F 2  !  F 3  !  F 4  !  F 5  !  F 6  !  F 7  !  F 8                  *
*  Interproc!      !      ! Flag  ! Scroll ! Scroll !  Next  ! Branch               *
*  com. flag!  QW  !  I/O ! word  !  down  !   up   ! Branch ! Selection            *
*                                                                                   *
*************************************************************************************
```

The description of controller structure R64 shows that the controller has a digital output word (DO) and a digital input word (DI).

The output word output out the statuses of the limit monitor and in particular the status of the step controller outputs ("Open" and "Close") and the continuous controller with Pulse/Pause output. When the regulating variable is to be output digitally the address of a digital output module should be entered into the mask.

Use one of the following options to control the digital inputs, for instance, the signals "End position open reached", "End position closed reached", "Inhibit controller", "Dead zone effective", "Upper", "Lower", "Delete" etc.:

- When the address of a digital input module is entered for the input "Address DI 01" the digital input word is read in automatically by the operating system and stored in the particular data word.

- When a (interprocessor communication) flag is entered for the input "Address DI 01" the input word may be specified by a STEP5 program or by a CP where the defined flag is described.

- The digital input word can also be specified directly by a STEP5 program where the data word 180 in the controller data block is described.

- The bit values of the digital input word may be controlled by the PG during test.

In the last two cases data should not be entered under any circumstances for the input "Address DI 01" since the digital input word is overwritten by the operating system when the process image is created.

Important Note: - **The digital input word contains many important bit values. When one of these bit values is in an incorrect status the controller cannot continue running (e. g. the bit "Inhibit controller").**

- **When the user makes an entry in the address for the digital input the specified address, respectively the specified flag should supply a defined value.**

- **When the user makes an entry in the address for the digital input the relays cannot be controlled during test!**

4.4.2 Input of the Controller List

```
*****************************************************************************
*                                                                           *
*  Input                          Module     : R-Proc. Struct. :           *
*  Controller list                Source/Dest.: File   Block   : DB  002    *
*  ------------------------------------------------------------------------ *
*                                                                           *
*    Time base :    50 ms                                                  *
*                                                                           *
*    The Scan times of the columns are:                                     *
*                                                                           *
*  100 ms | 150 ms |  0 s  |  0 s  |  0 s  |  0 s  |  0 s  |  0 s         *
*  =======+=======+=======+=======+=======+=======+=======+=======         *
*   DB    |  DB   |  DB   |  DB   |  DB   |  DB   |  DB   |  DB            *
*  -------+-------+-------+-------+-------+-------+-------+-------         *
*    3    |   4   |       |       |       |       |       |                *
*         |       |       |       |       |       |       |                *
*    ---  |       |       |       |       |       |       |                *
*    ---  |  ---  |       |       |       |       |       |                *
*    ---  |  ---  |       |       |       |       |       |                *
*    ---  |  ---  |       |       |       |       |       |                *
*    ---  |  ---  |       |       |       |       |       |                *
*    ---  |  ---  |       |       |       |       |       |                *
*                                                                           *
*                                                                           *
*   F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                   *
*       !      !      !      !      !      !      !                         *
* Enter DB !Remove DB!Time base!      !      !      ! Ready ! Break        *
*                                                                           *
*****************************************************************************
```

The entries in the controller list determine the calling sequence of the controllers.

The controller list shows clearly the method of working and the loading of the processor:

The processor is able to manage a maximum of 8 different scan times, corresponding to the eight columns of the controller list. The processor processes the columns one after the other selecting only one controller per column, then stepping on to the next column. The time which passes after one controller of a column is processed until the next one in the same column is processed, is called time base which is to be entered into the head of the mask. When the time base, for example, is 100 ms the processor should process a controller every 25 ms when there are 4 lists; when there are 8 lists, a controller of a list should be processed every 12.5 ms. Since the processing time of a controller is 2.5 ms maximum, 22.5 ms or 10 ms respectively, are available for the processor to execute a STEP5 program between two controller processes, in the example above.

When, for this example, a list is processed every 100 ms it may contain a controller with a scan time of 100 ms. When the scan time of the controller of this column is 200 ms it should be called in every second column. Therefore, this column may contain two controllers with a scan time of 200 ms each. The number of

controllers per column is derived from the quotients of the controller scan time of the column and the time base. The number of controllers, however, is limited to 8 since storage, not time, is restricted. All scan times (minimum pulse duration) of the projected controllers (see description below) should be an integer multiple of the time base so that the quotient is an integer. The maximum common divisor of all scan times (minimum pulse duration) should therefore be entered for the time base.

If a column does not contain any controllers or if the maximum number of controllers per column is not reached the operating system detects this and may possibly execute an existing STEP5 program instead of a controller.

For step and continuous controllers with a pulse/pause output the regulating variable is converted into the number of pulses. In order to output a pulse with the length of the parameterized minimum pulse duration the controller structure R64 should be called at least once during the minimum pulse duration to enable the bit of the digital controller output to be set or reset. This is why the minimum pulse duration is decisive for entry of step and continuous controllers with pulse/pause outputs into the controller list.

When the controller list is processed the time base is to be entered first. The following two conditions should be met:

- It should be a multiple of 10 ms, at least 20 ms.
- It should be the maximum common divisor of the scan times (respectively the minimum pulse duration) of the controller to be processed. When a smaller scan time is selected the processor loading increases because of unnecessary administration work.

It is only possible to leave the entry field when a valid time base was entered. The function keys are assigned the following:

<F1> Enter DB
<F2> Remove DB
<F3> Time base
<F7> Ready
<F8> Break

In order to enter the controllers into the controller list, press the function key <F1>, enter the data block number, and conclude the input by pressing <Carriage Return>. In order to simplify the entry COM REG does not return to the previous menu but requires the entry of a new data block number. This input loop may be broken by pressing the <Break> key.

After the data block number is entered the program reads the scan time (minimum pulse duration) from the particular data block of the selected destination medium and attempts to enter the controller into the controller list. During this procedure the existing columns with the same scan time are filled before a new column is opened. If the scan time (minimum pulse duration) is not a multiple of the time base the controller cannot be entered and the time base or the scan time (minimum pulse duration) must be modified. The removal of individual controllers from the controller list is also carried out in form of a loop that is to be terminated by pressing the <Break> key.

After the function key <F3> is pressed the time base may be modified. However, the new value is only accepted if it is an integer divisor of all controllers contained in the list. The value is not deleted before a valid value is entered. If there is no further value, which meets all conditions for the time base, the old value should be entered again so that the user can leave the input field. If the modification was successful the controller list is updated.

The input of the controller list may be terminated by pressing <F7>; the DB2 that was created is stored. By pressing <F8> the input is terminated; the entries are lost.

4.5 Output

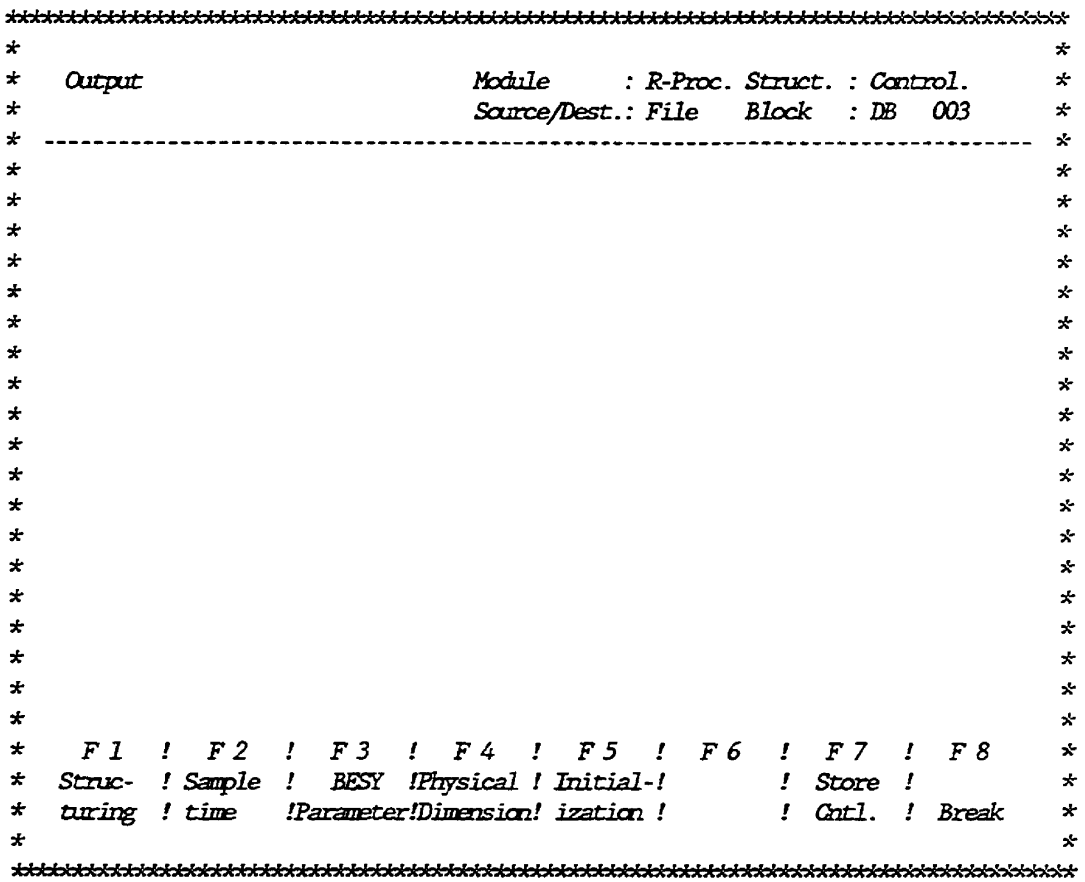
By means of the function "Output" an already existing controller data block or the controller list may be checked, amended or corrected. Additionally all controller data and a cross reference list may be printed out using a print function. The only difference between the functions "Input" and "Output" is that, in contrary to the rigid operator prompting in the input function, the mask the user wishes to complete may be directly selected in the output function.

After "Output" is selected from the main menu the user may select the medium on which the data block to be processed is stored. The print function may be branched to from this menu.

<F1> Program file
<F4> Module
<F5> Programmer (PG)
<F8> Print

After the medium is selected and the data block number is entered the mask selection menu is displayed. When data block two is selected the controller list may be processed. The controller list is output the same way as it was input.

The following description applies to controller data blocks, not to the controller list.



When the user presses function key <F1>, <F2>, or <F4>, the associated mask is displayed and may be directly completed; however, when function key <F3> is pressed, another menu is displayed, from which the user may select the masks, controller response/FB number or CP526 adaptation. When function key <F5> is pressed, the menu displayed for the controller branch selection is the same as for input displayed.

When the masks are terminated by pressing <F7> "Ready" COM REG returns to the mask shown above. When a mask is terminated by pressing <F8> "Break" the user leaves not only the processing of the mask but also the complete output of the data block. The program returns to that level where the user is requested to enter the data block number. In order to save changes already made, the data block which is in PG block memory may be transferred to the module or program file by means of the function "TRANSFER".

When the user leaves the output menu by pressing <F7> "Store controller", COM REG overwrites the old data block with the new one once the prompt "Block x owerwrite (Y/N) ?" is answered. If this prompt is answered with "N" or if the output menu is left by pressing <F8> "Break" the modified data block can still be saved from the PG memory.

Note: - When a structure switch is modified by means of the function "OUTPUT", COM REG changes this switch only, not the parameters that become invalid by this procedure. For example, when the polygon curve is switched off with the

corresponding structure switch the parameters for the polygon curve are maintained in the data block so that it need not be entered again when the polygon curve is reactivated. Another consequence is that the ADC address is still stored in the data block after the test socket was switched off with the structure switch or after the setpoint was transferred from the ADC to the setpoint sequence. As already described, this existing address input requests the system program to read cyclically the particular variable of the input module. This means that the input module must exist although the variable read is not evaluated. **Therefore the address of each ADC or DAC should be overwritten with blanks before the branch with the ADC (DAC) is switched off by means of a structure switch! After the address was switched off with the structure switch it cannot be deleted because it is no longer offered for parameterizing!**

- When the format or the number of decimal places is modified during entry of the input characteristic, all dimension dependent parameters are falsified by the decimal power that corresponds to the modification to the number of decimal places, i. e. all dimension-dependent parameters should be entered again.

  Example:

0% corresponds to	0.00 V
100% corresponds to	10.00 V
Setpoint upper limit	1.23 V

  After the format of the input characteristic was modified to one decimal place the upper limit of the setpoint is falsified.

0 % corresponds to	0.0 V
100 % corresponds to	10.0 V
Setpoint high limit	12.3 V

As described above, the function "OUTPUT" also contains the option to print out the controller data entered. After the function "Print" is selected by pressing function key <F8> of the first output menu the following functions are offered:

<F2> Print a controller data block
<F5> Print all
<F6> Print cross reference list
<F8> Break

After being selected all three functions request information on which medium (program file/module) the data to be printed is to be read. When only one controller data block is to be printed the block number should be entered. Of course, a printer ready to run should be connected to the module. When the "SIEMENS PT88" printer is used, all DIL switches should be set to "ON".

The cross reference list looks like this:

```
*****************************************************************************
*                                                                           *
*   Analog Inputs:                                                          *
*                                                                           *
*                    PW  128     DB 003      ADC 2                          *
*                    PW  130     DB 003      ADC 1                          *
*                    PW  132     DB 004      ADC 2                          *
*                    PW  134     DB 004      ADC 1                          *
*                                                                           *
*                                                                           *
*   Analog Outputs:                                                         *
*                                                                           *
*                                DB 004      DAC 1                          *
*                    PW  192     DB 003      DAC 1                          *
*                                                                           *
*                                                                           *
*   Digital Inputs:                                                         *
*                                                                           *
*                    PW    4     DB 003      DI  1                          *
*                    PW    6     DB 004      DI  1                          *
*                                                                           *
*                                                                           *
*                                                                           *
*   Digital Outputs:                                                        *
*                                                                           *
*                    PW    0     DB 003      DO  1                          *
*                    PW    2     DB 004      DO  1                          *
*                                                                           *
*****************************************************************************
```

The cross reference list shows the controllers and the input/output module types and the required addresses, whether addresses are accessed twice, and the channel of modules not yet assigned.

When the function "Print all" is selected the controller list and all controllers entered are printed. The representation of the controller data corresponds to the representation of the controller input.

When the function "Print a controller data block" is selected one single controller may be printed.

4.6 Transfer

By means of the function "Transfer" any (i. e. also STEP5) data and function blocks may be transferred between the media "program file", "module", and "PG".

```
*************************************************************************************
*                                                                                   *
*                                  Module     : R-Proc. Struct. :                   *
*   Transfer                       Source/Dest.:       Block   :                   *
*  ---------------------------------------------------------------------------      *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*    From source    : File              To destination: Module                      *
*         File name : B:CONTROST.S5D       File name  : -------                    *
*         Block     : DB                   Block      : DB                         *
*         Block no. :  3                   Block no.  :  3                         *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*    F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                          *
*        !      !Change !      !      !      !      !                              *
*        !      !block no !    ! Correct !    !Transfer ! Break                     *
*                                                                                   *
*************************************************************************************
```

In order to transfer data the source medium should be selected by pressing the corresponding function key. When the program file is selected as source the name of the preset program file is displayed automatically. Afterwards the user specifies the block type (DB or FB) to be transferred by pressing function key <F1> or <F2>. If all blocks of the selected type are to be transferred, function key <F1> should be pressed in place of the entry of a block number; otherwise the block number must be entered. The destination medium may also be selected by means of the function keys. When the program file was selected as destination, the name of the destination file may be entered according to the conventions described in section "Presetting". The destination file may be identical to the source file. This is only useful when the block number is modified during the transfer procedure. When the user wishes to change the block number, function key <F3> "Change block no." should be pressed and the new number entered.
When all entries are correct the function key <F7> "Transfer" may be pressed. The entries may only be corrected after <F5> is pressed. The cursor is moved over each input line which may then be corrected or acknowledged by means of <Carriage Return>.
The input may be terminated at any time by pressing function key <F8>.

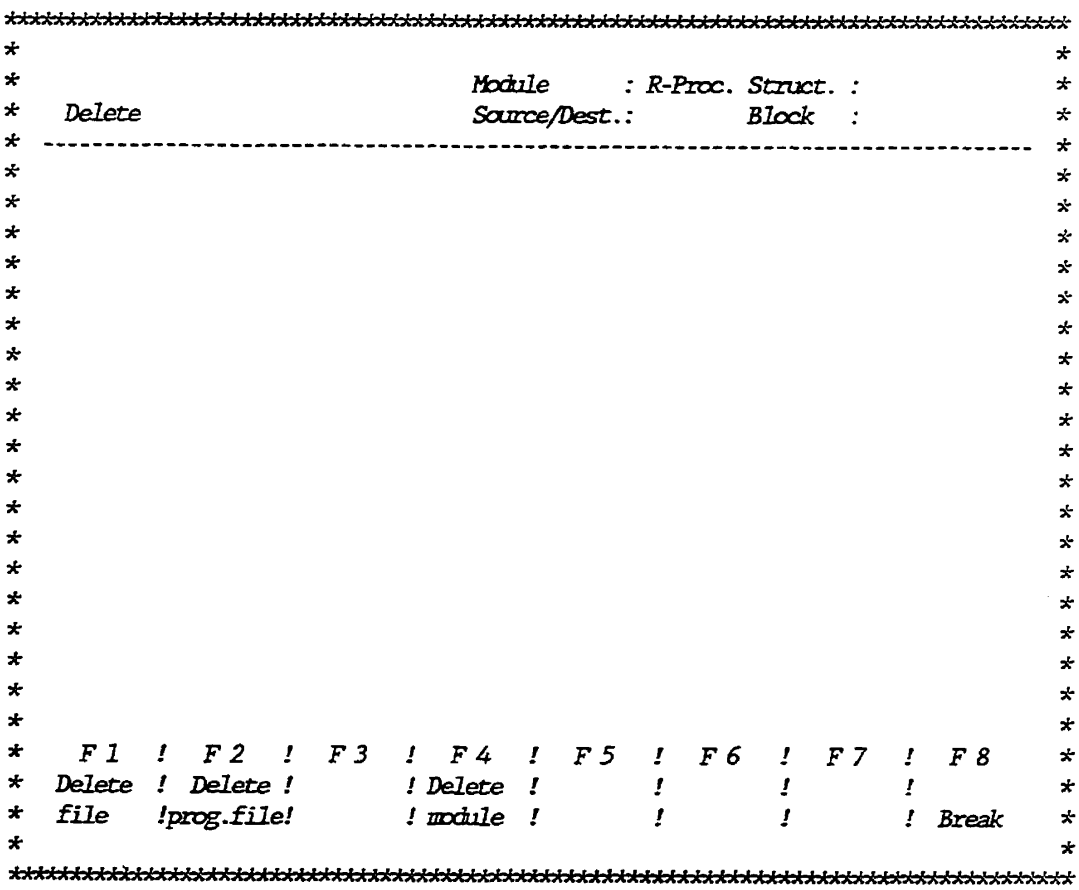
Notes: - "PG" is the term for the block memory of the programmer. It may contain only one block and is overwritten each time a new block is created.

- When blocks are transferred from the program file to the module they are initially stored in the plug-in RAM module of the processor. When this storage is occupied other data blocks, except function blocks, may be stored in the data block storage of the processor. Therefore, it is advisable to transfer the function block(s) first to the module.

- When no more blocks can be transferred to the module because there is not enough free memory the function "Compress PC" may possibly provide the memory required.

4.7 Delete

By means of the function "Delete" single data and function blocks, or the complete program file may be deleted. After the function is called the function key menu offers the following:



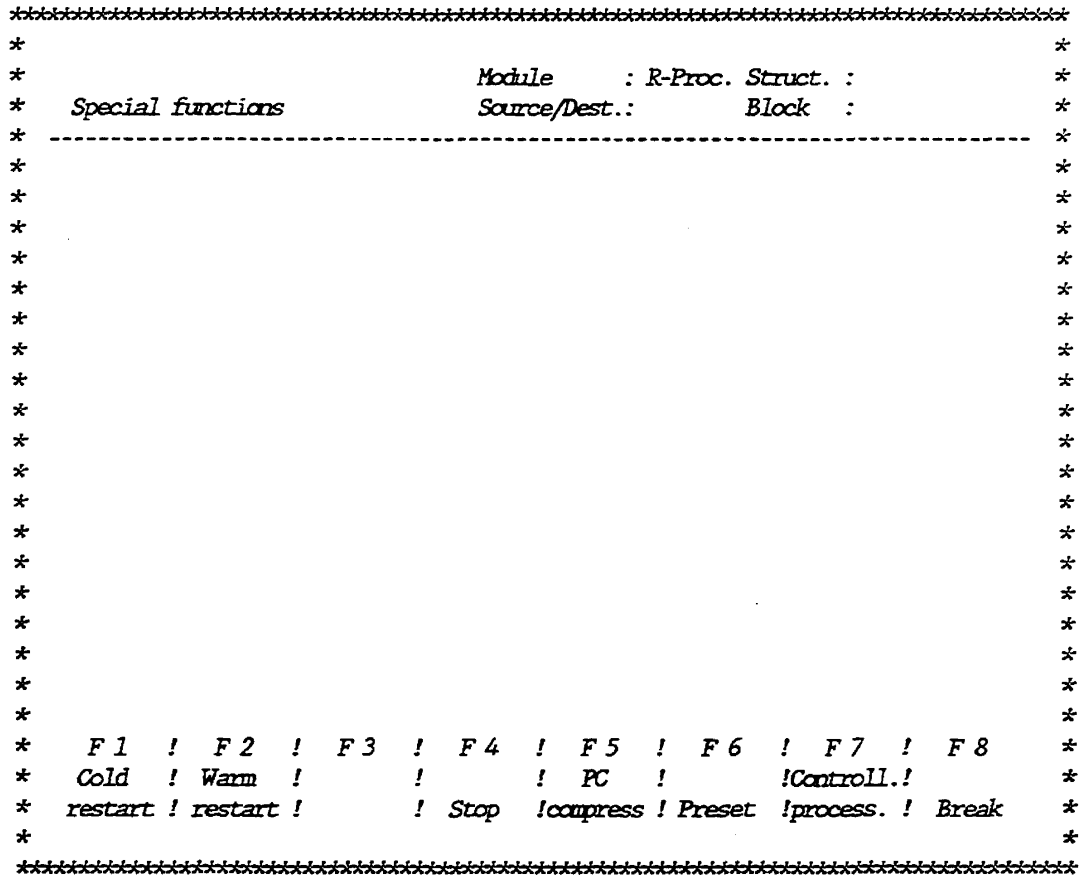
<F1> Delete single blocks from the program file
<F2> Delete the complete program file
<F4> Delete single blocks from the module
<F8> Break

When single blocks are to be deleted, the block type (DB or FB) and the block number should be selected by pressing function key <F1> or <F2>. When a program file is to be deleted, the name of this file should be entered. By means of this function not only the preset program file may be deleted but also each program file in the current user area of the user's disk (hard disk).

The function "Delete" may be terminated at any time by pressing the <Break> key.

4.8 Special Functions

After the "Special functions" key is pressed the following menu is displayed:



Each function, except "Preset" may be called if an operational processor is connected to the PG and if "Online" was selected when COM REG was preset.

- "Cold restart" has the same function as the manual restart of the R-Processor. "Cold restart" causes a new interpretation of the controller list and the acceptance of the modifications of the controller structure or modifications of the number of controllers. The cold restart deletes all internal controller variables (e. g. a large integral-action componenent which was created on the basis of a long-range controller deviation). The cold restart of the PG can be executed only if the operating mode switch of the processor is set to "RUN".

- The "Warm restart" causes the processor to continue the cycle with all previous values after it was stopped by means of "Stop". A warm restart may be executed only if the operating mode switch of the processor is set to "RUN".

- "STOP" will interrupt the running processor. The processor sets the digital output modules to zero by using the "command output inhibit" ('BASP') signal.

- When blocks are deleted from the processor the storage contains gaps that cannot be used because new blocks transferred are always stored after the already allocated storage. The function "Compress PC" compresses the blocks in the processor so that the gaps disappear and the continuous free storage increases. "Compress PC" may be called during operation.

- <F6> calls the presetting mask described in the chapter "Presetting". By means of this function the program filename may be changed for example.

- The function "Controller processing" enables or disables each controller of the module. <F1> (Yes) enables, <F2> (No) disables the controller. The cursor may be moved to each controller by means of the cursor keys. The response of each disabled controller regarding the outputs may be determined for each controller when the mask "Controller response" is completed.

  Note: Only those controllers that were entered in the controller list may be enabled or disabled. The function "Controller processing" may be called only if the processor contains a correct controller list.

The user may exit the menu "Special functions" without an entry being made by pressing the <Break> key.

4.9 Information

When the function "Info" is called, the list of contents referring to data and function blocks of the preset program file or the module is displayed. For purchased function blocks the library number is displayed in addition to the block number.

After "Info" is called the user selects - after pressing <F2> - between program file by pressing <F1> and module by pressing <F4>. The block type is then to be specified by pressing the function keys <F1> and <F2>.

```
*************************************************************************************
*                                                                                   *
*                                  Module      : R-Proc. Struct. :                  *
*   List of contents               Source/Dest.: Modul   Block   :                  *
* ----------------------------------------------------------------------------------*
*                                                                                   *
*   Data block number                                                               *
*   -----------------                                                               *
*         000                                                                       *
*         001                                                                       *
*         002                                                                       *
*         003                                                                       *
*         004                                                                       *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*    F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                           *
*        !      !      !      !      !      !      !  Main                          *
*        !      !      !      !      !      !      !  menu                          *
*                                                                                   *
*************************************************************************************
```

The function "Info" may be exited at any time by pressing the <Break> key.

Note: Not only the blocks created using COM REG are displayed but all blocks contained on the selected medium.

4.10 Test

The controller test enables the user to test operating and monitoring of the controllers in online mode (operating mode of the module: "RUN"). This function enables the user to

- start-up the system
- control the digital inputs
- change and optimize the parameters
- obtain an overview of all important controller statuses.

In addition to the correctable parameters the different masks also display the structure switch position, the input values, the averaging (measuring points), and the output values.

Each value is requested several times per second from the module and updated on the screen.

After the function "Test" is selected and the data block number of the controller to be tested is entered, the branch selection mask is displayed. This mask contains only the active branches as in the parameter input. In addition, a start-up branch and a measuring point table are available.

After the branch number is entered all data of the selected controller branch is displayed and permanently updated on the screen:

Note: The measuring points of the limit monitors can also be changed in test operation. If you want to change measuring points of different dimensions (example: MP4 regulating variable given as %, MP1 actual value dimension dependent), it may be necessary to re-enter the limit values in order to obtain a correct indication of the limit monitor bits.

```
*********************************************************************************
*                                                                               *
*                                Module      : R-Proc. Struct. : Control.       *
*   C O N T R O L L E R  T E S T Source/Dest.: Modul  Block    : DB  003         *
* ----------------------------------------------------------------------------- *
*                                                                               *
*     Branch  1: Controller                                                     *
*                                                                               *
*                                                                               *
*     MP 03: Controller deviation          0.0 DEGR_C                           *
*                                                                               *
*     Continuous/Step                        0                                  *
*       Standard/Upgraded                    0                                  *
*         Manual value PG/ADC                0                                  *
*           Constant man. value            30.00 %                            *
*                                                                               *
*         Automatic/Manual                   0                                  *
*         MP 10: Manual value              0.00 %                               *
*         Controller disable                 0                                  *
*         High overflow ID                   0                                  *
*         Lower overflow ID                  0                                  *
*                                                                               *
*                                                                               *
*                                                                               *
*    F 1  !  F 2  !  F 3  !  F 4  !  F 5  !  F 6  !  F 7  !  F 8              *
*         !       !       ! Scroll ! Scroll !       ! Next   ! Branch           *
*         ! Force !       ! up     ! down   !       ! Branch ! selection      *
*                                                                               *
*********************************************************************************
```

During the status display the function keys are assigned the following:

<F2> Force
<F4> Scroll up
<F5> Scroll down
<F7> Next branch
<F8> Branch selection

When the list requires more space on the screen than is available the display may be scrolled up (<F4>) or scrolled down (<F5>). The next branch is selected by pressing <F7>; when the user presses <F8> he is led back to the menu for branch selection.

After the function key <F2> is pressed the cyclical status display is frozen on the screen and the cursor may be moved by means of the cursor keys to the input field required. As soon as the new parameter value is enterd and the <Carriage Return> is pressed the new value is sent to the module by the PG and shown in the new cyclical status display. Since the value displayed comes from the module, it is confirmed that the module has accepted the value.
The following variables may not be controlled and thus must be changed by means of the function "Output":

- structure switch
- addresses
- number of limits
- number of vertices of the polygon curve
- number of setpoints for the setpoint sequence

The measuring point table is the most important aid for providing an overview of all important controller data. It supplies the following information:

- Do the input modules supply the correct values?

- Do the output modules output the correct values?

- Does the controller run? (the modified setpoint or actual value should be displayed at latest after the scan time)

- By comparing the fed and processed actual value (setpoint) the function of the processing modules may be tested in the corresponding branches.

- Are the controller parameters correct, does the controller output the expected regulating variable?

```
**********************************************************************************
*                                                                                *
*                                    Module     : R-Proc. Struct. : Control.     *
*  C O N T R O L L E R  T E S T      Source/Dest.: Modul  Block   : DB  003        *
*  ------------------------------------------------------------------------------ *
*                                                                                *
*     Branch  9: Measuring point table                                           *
*                                                                                *
*                                                                                *
*     MP 01: Setpoint input              0.0 DEGR_C                              *
*     MP 02: Processed setpoint          0.0 DEGR_C                              *
*     MP 03: Controller deviation        0.0 DEGR_C                              *
*     MP 04: Controller output          0.00 %                                   *
*     MP 05: Regulating variable        0.00 %                                   *
*     MP 08: Actual value input          0.0 DEGR_C                              *
*     MP 09: Processed actual value      0.0 DEGR_C                              *
*     MP 10: Manual value               0.00 %                                   *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*   F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                         *
*       !      !      ! Scroll ! Scroll !      ! Next   ! Branch                  *
*       ! Force !     ! up     ! down   !      ! Branch ! selection               *
*                                                                                *
**********************************************************************************
```

Note: When a data block is transferred from the module to the program file after test, the positions of the digital inputs are also transferred.

5 Projecting Controller Structures for IP252

The previous chapters (COM REG) described how COM-software is loaded and how the user steps from loading to the point where he may decide whether to use the COM REG software to operate either the R64 controller of the PC S5-135U or the controller device IP252. The controller selection menu is now displayed.

```
*************************************************************************************
*                                                                                   *
*                                                                                   *
*     Controller select:                                                            *
*  ---------------------------------------------------------------------------      *
*                                                                                   *
*     The following operator routines are available:                                *
*     ==============================================                                *
*                                                                                   *
*     COM REG for the R64 Controller Structure ....    B:S5OECI2X.CMD               *
*                                                                 B:S5OECR2X.CMD    *
*     COM REG 115U for the S5-115U ................    B:S5OECI2X.CMD               *
*                                                                 B:S5OECS3X.CMD    *
*     COM REG for IP 252 Closed loop control module    B:S5OECI2X.CMD               *
*                                                                 B:S5OECI3X.CMD    *
*     COM REG GRAPHICS for IP 252 Closed loop cntl.    B:S5ODCI2X.CMD               *
*                                                                 B:S5ODCG3X.CMD    *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*    F1   !  F2   !  F3   !  F4   !  F5   !  F6   !  F7   !  F8   .                *
*    Cntl ! Cntl  !       ! IP252  !       !       !       !                        *
*    R64  ! S5-115U ! IP252 ! GRAPHICS!      !       !       ! Break                  *
*                                                                                   *
*************************************************************************************
```

Fig. 5.1 Controller selection

In the controller selection the user specifies the module he wishes to operate using the COM software. The function keys offer the following options:

<F1> : Controller structure R64
<F2> : Control on CPU S5-115U
<F3> : Closed-loop control module IP252

<F4> : Closed-loop control module IP252 with graphic parameterization software
<F8> : Break

The assignment of the function keys <F1> to <F7> of the mask shown above depends on the existing packages.

After the controller selection the presetting menu is displayed.

```
*************************************************************************************
*                                                                                   *
*                                    Module      : IP 252 Struct. :                 *
*   Default                          Source/Dest.:       Block   :                  *
*  ---------------------------------------------------------------------------      *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*         Op. mode   : * OFFLINE                                                    *
*                        ONLINE                                                     *
*                                                                                   *
*                                                                                   *
*         Product    :     IP 252 clsd-lp cntl module                               *
*                      * W/out b/plane bus access on S5 115U, 135U, 150U            *
*                        With b/plane access to analog I/Os on S5-115U              *
*                                                                                   *
*                      * Mem. subm.  AR / SR (6ES5 374-0AA11)                       *
*                        Mem. subm.  ARS/ SR (6ES5 374-0AB11)                       *
*                                                                                   *
*                                                                                   *
*         Program file :  B:IP252AST.S5D                                            *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*    F1   !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                          *
*         !      !      !      !      !Program !      !                             *
*  OFFLINE ! ONLINE !      !Selection!      ! file   !Finished ! Break               *
*                                                                                   *
*************************************************************************************
```

Fig. 5.2 Presetting menu

This menu is structured the same as all other menus of this COM package and is an example of the principle structure of menus. The structure consists of a header area, foot area, and inbetween the particular input and output field.

In the left part of the header area, the function to be executed is displayed (e. g. presetting, input, etc.) The right part of this area contains information about entries already made. This information is: "Module:", "Source/Dest.:", "Struct.", and "Block:". "Module:" gives information about whether the user has selected the controller structure R64 or the IP252.
"Source/Dest." shows from where the output data was read or where the input data is stored. The following entries are possible: program file, submodule, module or PG. In the foot area the function keys are assigned the corresponding functions; the particular inputs and outputs are made in the area between header and foot area.

The presetting has the following defaults:

Operating mode: Offline
Module: : IP252 in S5-115U, -135U, -150U without access to backplane bus
AR/SR memory submodule

In the presetting the user specifies how to operate the module with this COM software. The function keys offer the following options:

<F1> : OFFLINE
<F2> : ONLINE
<F4> : Selection
<F6> : Program file
<F7> : Finished
<F8> : Break

The operating mode depends on whether the user wishes to work offline first, i. e. programming into the PG, the submodule or the disk, or whether he wishes to program directly to the module (ONLINE).

The functions IP252 with or without bus access refer to the direct bus access. The direct bus access means that the IP252 is not restricted to its eight analog inputs/outputs but that it can additionally use the analog peripherals of the central controller. In this case the IP252 behaves as a co-processor and leaves the bus administration to the CPU. However, the bus can be accessed only in the S5-115U programmable controller.

The user submodule which contains the structures "ARS/SR" supports the connection of control loops 1 to 8 with *master and servo controllers*. Wherever you find **ADC n** in the documentation on projecting (chapters 8.2 und 8.3, manual IP252) you can enter the following parameters:

e.g. ADC 6 =	0 ... 7	internal ADC channels
	128 ... 254	backplane bus addresses of the PC S5 analog peripherals: for PC S5-115U only
	C-no./MP-no.	**C-no.** here means the controller numbers 1 to 8 and **MP-no.** the measuring point number of this controller structure

It is thus possible to assign, for example, the processed actual value from controller no. 2 to controller no. 3 as a setpoint value. The following value is entered via the PG into the entry field of ADC 6 (of branch 8 of controller no.3):

ADC 6 PW 2.12

Use the <F4> function key to carry out these presettings.

For the description of the following masks a standard controller with memory submodule AR/SR is taken as an example.

When the <F7> key is pressed the main menu is displayed.

5.1 Main Menu

The main menu is the starting mask for all functions; the user can reach it from each position of the program by pressing the Break key once or several times.

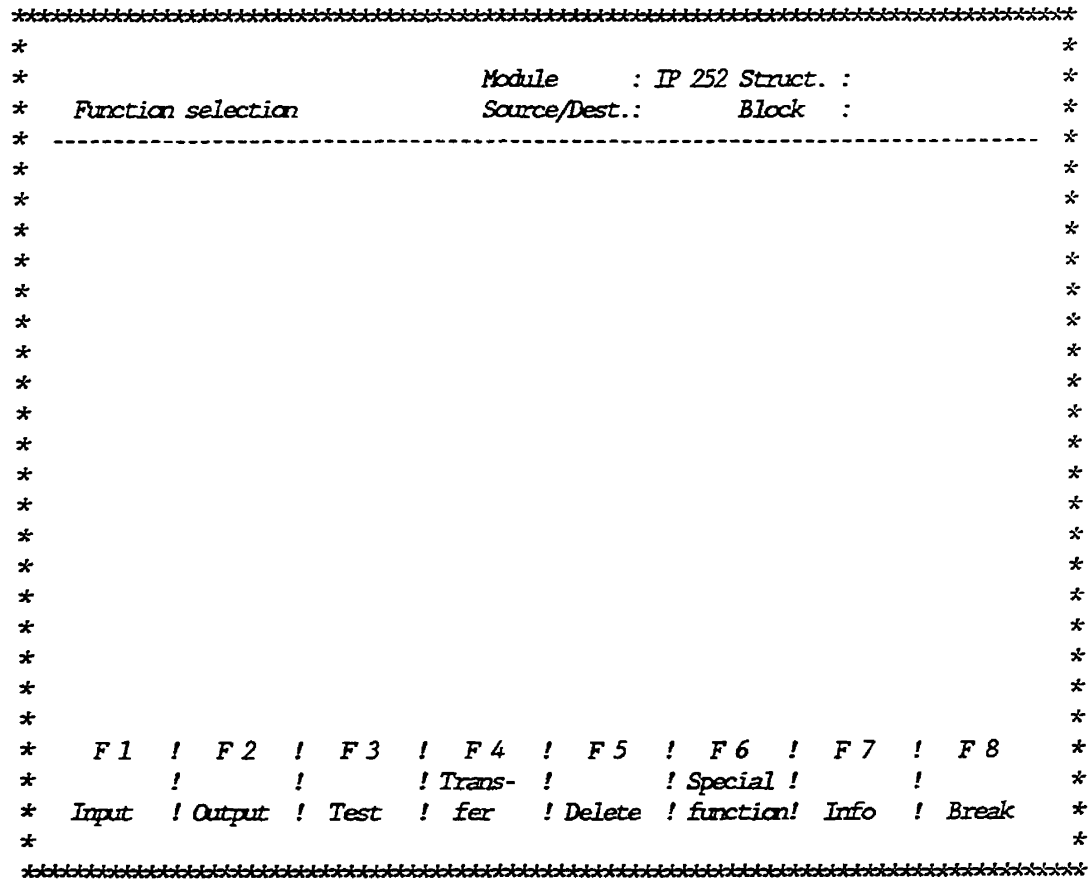


Fig. 5.3 Main menu

Starting from this main menu, the user steps to the sub-menus by pressing the respective function key. The selectable functions are:

<F1>	: Input	(see chapter 5.2)
<F2>	: Output	(see chapter 5.3)
<F3>	: Test	(see chapter 5.8)
<F4>	: Transfer	(see chapter 5.4)
<F5>	: Delete	(see chapter 5.5)
<F6>	: Special functions	(see chapter 5.6)
<F7>	: Info	(see chapter 5.7)
<F8>	: Break	

5.2 Input

The term "Input" describes each step for specifying the control loop. In order no step is omitted during the initial input the PG leads the user in a linear manner, i. e. without branches, through all input functions.

Input sequence:

Select the destination medium
I
I
Enter the control loop number
I
I
Select the structure
I
I
Structure
I
I
Enter the scan time
I
I
Specify the Stop behaviour
I
I
Specify the dimension
I
I
Parameterize the branches

After the input function is selected the PG replies by displaying the menu in figure 5.3.

This menu requests the input of the medium where the data to be input should be stored. This information is entered into the field "Source/Dest.:" in the header area. The following entries are permitted:

<F1> : Program file
<F2> : ----
<F3> : Submodule
<F4> : Module
<F5> : Programmer
<F6> : ----
<F7> : ----
<F8> : Break

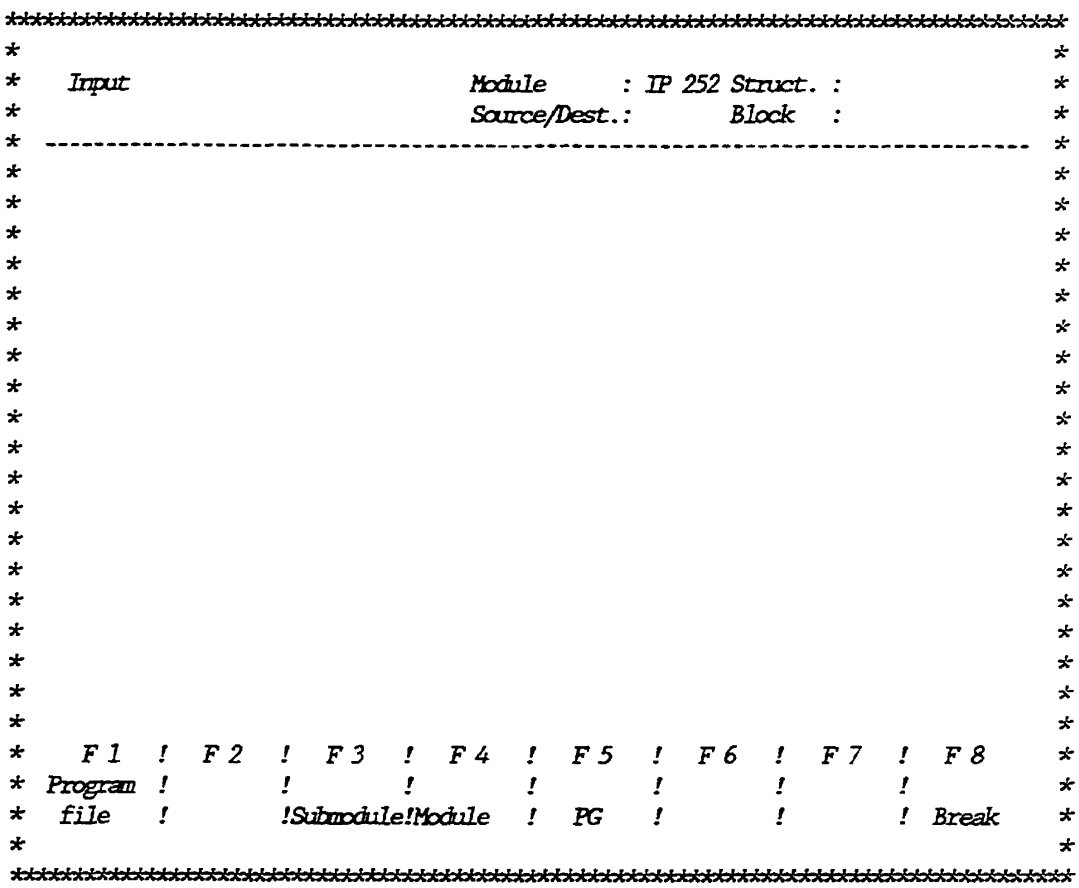


Fig. 5.4 Menu for destination input

After the destination is selected the controller structure is selected in the same way. The function keys are assigned the following:

<F1> Standard controller
<F2> Drive controller
<F8> Break

At this stage the PG requests the control loop number. Since IP252 is able to process up to eight control loops, the entry 1..8 is permitted. Thus the data entered in the following procedure is allocated to a specific control loop.

Each entry made appears in the right part of the header area. When the user has entered the control loop number and transferred it by pressing the Return key, the structuring mask (figure 5.5) is displayed.

5.2.1 Structuring

Structuring means that the software switches that exist between the permanently assigned branches, are set according to the application. That means that the switches of the branches required are closed (=1) and those of the branches not required are open (=0).

The structuring switches may function as on-off switches as well as changeover switches. For on-off switches "0" means No and "1" means Yes. Changeover switches are marked with a slash "/" in the text and are used to select one of two sub-branches. "0" is assigned to the first and "1" to the second alternative.

The structuring switches are subordinated hierarchically and assigned to corresponding levels. These divisions are indicated by the indentations of the particular subordinate structure switch terms in the structuring mask.

```
**********************************************************************************
*                                                                                *
*  Input                          Module     : IP252  Struct. : Standard        *
*  S T R U C T U R I N G          Source/Dest.: FILE   Block   : Cntl  003        *
*  ------------------------------------------------------------------------------ *
*                                                                                *
*                                                                                *
*    1: Controller            1        2: Actual value          1               *
*     Continuous/ Step        0         ADC/Pulse               0               *
*       Manuel input PG/ADC   0         Validity check          0               *
*       Standard/Upgraded     0         Averaging               0               *
*         Separate D-Input    0         Polygon curve           0               *
*         Interference input  0                                                 *
*         Manual input PG/ADC 0                                                 *
*         Manual input PG/ADC 0                                                 *
*       Cont./mark-space      0                                                 *
*         2-/3 Point controller 0                                               *
*         Actuator adjustment 0                                                 *
*                                                                               *
*    3: Setpoint              1        4: Limit monitor 1       0               *
*     ADC/Entry               0                                                 *
*     Ramp-function generator 0                                                 *
*                                                                               *
*                                                                               *
*  F1 ! F2 ! F3 ! F4 ! F5 ! F6 ! F7 ! F8                                        *
*     !    !    ! Scroll ! Scroll !    !    !                                   *
*     !    !    !   up   !  down  ! Help ! Ready ! Break                        *
*                                                                               *
**********************************************************************************
```

Fig. 5.4 Structuring mask

Since not all the structure switches of a controller structure can be displayed at a time, the contents of the screen may be scrolled up or down.
These functions can be executed by pressing the function keys <F4> or <F5>. The function key <F5> is assigned the function

Help; by pressing <F7> the entries made are transferred, by means of <F8> the processing of the mask is terminated and the main menu is displayed.

<F4> : Scroll up
<F5> : Scroll down
<F6> : Help
<F7> : Ready
<F8> : Break

The cursor is moved on the screen by means of the cursor keys. The user structures the branches and subbranches by entering a "1". The branches not selected are suppressed.

As soon as the branches are set the user has defined the final control loop structure. When he is satisfied with it he may terminate the projection by pressing the function key <F7> (Ready).

5.2.2 Entering the Scan Time

After the controller structure is defined in the last step, the scan time is entered. The user should note that the scan time may only be a two's square value between 4 ms and 32 sec.

In the mask shown in figure 5.5, a rough formula is offered for selecting the scan time; the possible range is indicated. Below, the set or the default scan time is displayed (default is always TA = 4ms). In the bottom line the PG provides information on the processor loading according to the selected scan time, for the structure selected for the control loop. The unit is percent (%), 100 % indicates complete loading of the processor.

```
*****************************************************************************
*                                                                           *
*   Input                          Module     : IP252  Struct. : Standard   *
*   S C A N - T I M E              Source/Dest.: FILE   Block   : Cntl  003  *
*  ------------------------------------------------------------------------ *
*                                                                           *
*                                                                           *
*                                                                           *
*         Comment :    For the quasi-continuous controller design the       *
*                      recommended scanning time value is 10% of the        *
*                      dominating time constant of the controlled           *
*                      system.                                              *
*                      The following scan times are permitted for IP252:    *
*                           4, 8, 16, 32, 64, 128, 256, 512  milliseconds   *
*                           1, 2,  4,  8,  16, 32  seconds                  *
*                                                                           *
*         Scan time:          8 ms                                          *
*                                                                           *
*         IP252 loading:      27 %                                          *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*                                                                           *
*    F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                   *
*        !      !      !      !      !      !      !                         *
*        ! Input !      !      !      !      ! Ready ! Break                 *
*                                                                           *
*****************************************************************************
```

Fig. 5.6 Scan time menu

The menu offers the following function keys:

<F2> : Input
<F7> : Ready
<F8> : Break

By pressing function key <F2> the default scan time can be modified. The keys <F7> and <F8> are assigned the same functions as in the previous menus.

To which extent the controller works stably with the scan time selected is the responsibility of the user. He is also responsible for the adaptation of controller parameters to the modified scan time.
When function key <F7> is pressed the mask for setting the controller behaviour (see figure 5.7) is displayed.

5.2.3 Controller Behaviour

In this menu the user specifies the controller behaviour in the case of a controller stop (e. g. when power failure occurs). For such a case three entries are necessary respectively three questions have to be answered. The default is "Yes" in all cases.

a) Should the controller outputs be set to "0" when the control loop is not executed?
b) Should the system restart, automatically, with resumption of power supply?
c) Is a restart condition to be observed when the system is started automatically (for a detailed description of this condition see sections 3.1 and 4.7)?

```
**********************************************************************************
*                                                                                *
*  Input                          Module      : IP252   Struct. : Standard      *
*                                 Source/Dest.: FILE    Block   : Cntl  003     *
*  ------------------------------------------------------------------------      *
*                                                                                *
*                                                                                *
*    Cntl.-Behaviour:                                                            *
*    ----------------                                                            *
*        If the cntl. is not operational                                         *
*        the outputs are set on zero:              Yes                           *
*                                                                                *
*        Automatic restart after power-on:         Yes                           *
*        The restart condition is valid:           Yes                           *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*  F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                          *
*      !      !      !      !      !      !      !                                *
*  Yes !      !  NO !      !      !      ! Ready ! Break                         *
*                                                                                *
**********************************************************************************
```

Fig. 5.7 Setting the controller behaviour

The displayed responses may be changed by pressing function key <F1> or <F3>. The function keys are assigned the following:

<F1> : Yes
<F3> : No
<F7> : Ready
<F8> : Break

5.2.4 Specifying the Controller Name and Area Name

After specifying the controller behaviour and pressing the "Ready" key <F7> each controller (No. 1 to 8) of the IP252 can be assigned a **controller name** and **area name** comprising each not more than 8 ASCII characters. Input in this mask, however, is useful only if controllers of the IP252 are to be operated and monitored via a CP526 or a similar device.

```
**********************************************************************************
*                                                                                *
*  Input                          Module      : IP252   Struct. : Standard      *
*                                 Source/Dest.: FIFE    Block   : Cntl  003     *
*  ------------------------------------------------------------------------      *
*                                                                                *
*    Comment:                                                                    *
*                                                                                *
*    If the controller is to be operated and observed using the CP526,           *
*    the symbolic group and controller name for the display of the CP526         *
*    should be given.                                                            *
*                                                                                *
*                                                                                *
*                  Controller name:  TEMP_001                                    *
*                                                                                *
*                  Group name:       TEST_A01                                    *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*                                                                                *
*  F1 ! F2 ! F3 ! F4 ! F5 ! F6 ! F7 ! F8                                         *
*     !    !    !    !    !    !    !                                            *
*     !    !    !    !    !    ! Ready ! Break                                   *
*                                                                                *
**********************************************************************************
```

Fig. 5.8 Adaptation to CP526

At this stage, note that there is a difference between standard controllers and drive controllers. With the drive controller, activating the <F7> key will lead directly to the parameter assignment, whereas the standard controller still requests an intermediate step.
In this intermediate step, the dimension and characteristic is specified.

5.2.5 Specifying the Dimension

When the standard structure contains values with physical dimensions the user may specify an ASCII string and a characteristic of the dimension before parameterization. The default ASCII string with 6 characters contains "%"; the 0% value contains "0", and the 100% value contains "100".
The specification of the dimension consists of entering
- the ASCII string,
- the characteristic.
The menu for these entries is shown in figure 5.9.

```
*************************************************************************************
*                                                                                   *
*  Input                          Module      : IP252   Struct. : Standard        *
*  P A R A M E T E R              Source/Dest.: FILE    Block   : Cntl   003       *
*  ------------------------------------------------------------------------------- *
*                                                                                   *
*                                                                                   *
*    Input of desired physical dimension:                                           *
*    ====================================                                           *
*                                                                                   *
*                                                                                   *
*      Dimension D1:   GRAD_C                                                       *
*           0 Percent corresp.      15.0 GRAD_C                                     *
*         100 Percent corresp.      50.0 GRAD_C                                     *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*                                                                                   *
*  F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                             *
*      !      !      !      !      !      !      !                                   *
*      !      !      !      !      !      ! Ready ! Break                           *
*                                                                                   *
*************************************************************************************
```

Fig. 5.9 Input of the physical dimension

In the first line of the menu "Dimension D1" the user may enter the dimension necessary for the control loop, with an ASCII string up to 6 characters long if the default "%" does not match. Afterwards the cursor jumps to the next line and the % sign in the characteristic lines is replaced by the dimension entered. Then the "0 %" value and the "100 %" value are assigned a numerical value.

When these values are entered the PG checks whether the decimal places are identical and whether the characteristic entered is positive, i. e. whether the 0% value is less than the 100% value (see section 4.4.3).

5.2.6 Parameter Input

The last step of the input is to enter the parameters of the last structured controller. Here the PG offers only those subfunctions (branches) activated in the previous structuring mode, for parameterizing.

The parameterizing starts with the PG displaying a list of the connected branches where each branch is assigned a selection number, which is identical to the branch number (see figure 5.10). By means of the selection number the user may call the parameter list of the subfunctions to be parameterized. After the selection number is entered the branch selection is terminated by pressing the <Enter> key.

```
*********************************************************************************
*                                                                               *
*  Input                          Module     : IP252  Struct. : Standard       *
*  P A R A M E T E R              Source/Dest.: FILE   Block  : Cntl  003       *
*  ---------------------------------------------------------------------------  *
*                                                                               *
*                                                                               *
*     1: Controller                        2: Actual value                      *
*                                                                               *
*     3: Setpoint                          4: Limit monitor 1                   *
*                                                                               *
*     6: Test socket 1                                                          *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*     Please enter the branch number:                                           *
*                                                                               *
*   F1  !  F2  !  F3  !  F4  !  F5  !  F6  !  F7  !  F8                        *
*       !      !      !      !      !      !      !                              *
*       !      !      !      !      !      ! Ready ! Break                      *
*                                                                               *
*********************************************************************************
```

Fig. 5.10 Parameter input/Branch overview

After a branch is selected the corresponding parameter list is displayed. Only the parameters of the selected subbranches are displayed.

After the branches are parameterized each entry should be terminated by pressing <Carriage Return>, whereas the complete parameter list of each branch is terminated by pressing the

<Enter> key. Then the parameter list of the next branch is displayed until all branches are parameterized. Figure 5.11 shows the parameters of branch 3 of the standard controller when the memory submodule ARS/SR is used.

```
*********************************************************************************
*                                                                               *
*  Input                          Module     : IP252  Struct. : Standard       *
*  P A R A M E T E R              Source/Dest.: FILE   Block   : Cntl  003       *
*  ----------------------------------------------------------------------       *
*                                                                               *
*    Branch  3: Setpoint                                                        *
*                                                                               *
*                                                                               *
*    ADC/Entry                          0                                       *
*      Address ADC 01               PW                                         *
*                                                                               *
*    Setpoint high limit              0.0 GRAD_C                               *
*    Setpoint lower limit             0.0 GRAD_C                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*                                                                               *
*  F1 ! F2 ! F3 ! F4 ! F5 ! F6 ! F7 ! F8                                        *
*     !    !    !    ! Scroll ! Scroll ! Next  ! Branch                        *
*     !    !    !    !  down  !   up   ! Branch !selection                      *
*                                                                               *
*********************************************************************************
```

Fig. 5.11 Parameter input

Following this procedure the PG redisplays the list of all activated branches. This enables the user to correct the entries later. When all the values are modified the <F7> key is pressed and the data, entered so far, is transferred. When the destination device already contains a data set with the same controller number the programmer displays a menu which prompts the user whether the data set stored in the PG may be deleted.

The user must answer the prompt "Controller x overwrite (Y/N)". He does this by pressing the function key "Yes" or "No". When "Yes" is pressed the data set just entered is stored and the data set in the destination medium is overwritten; when "No" is pressed the new data set is lost.

5.3 Output

When the user selects the function "Output" in the main menu by pressing function key <F2> the output menu as shown in figure 5.12 is displayed after entry of the destination medium and the control loop number. By means of this function an already existing control block may be checked, amended or corrected.

Contrary to the linear prompting during input, the desired masks may directly be selected in the output function. The screen masks "STRUCTURING", "SCAN TIME", "CONTROLLER BEHAVIOUR", "PHYSICAL DIMENSION", and "PARAMETERIZING" are handled in the output function as described in the previous section.
After each step the program returns to the output menu. If the <Break> key or the <Enter> key is pressed this menu is left and the saving dialog started.

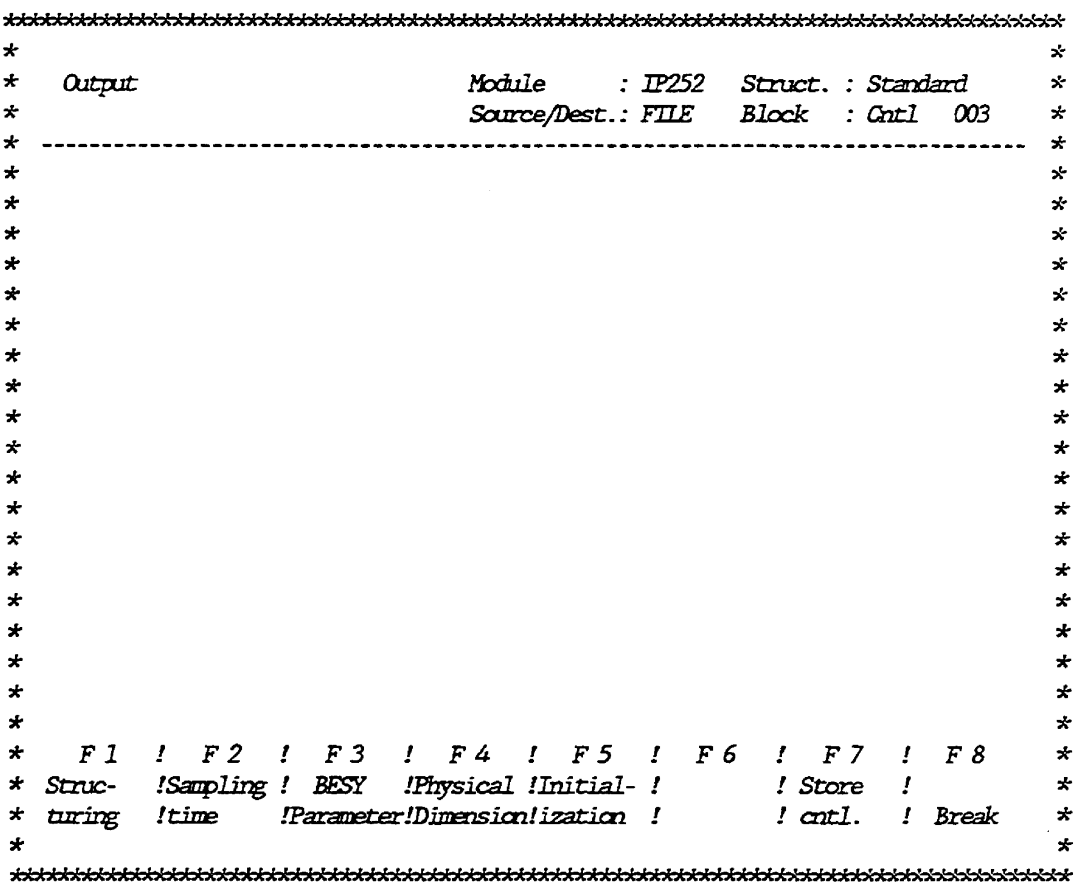


Fig. 5.12 Output menu

In the submenu to

<F3> BESY parameter

the softkeys

<F3> Behaviour
<F4> Adaptation to CP526

are displayed (see also figures 5.7 and 5.8).

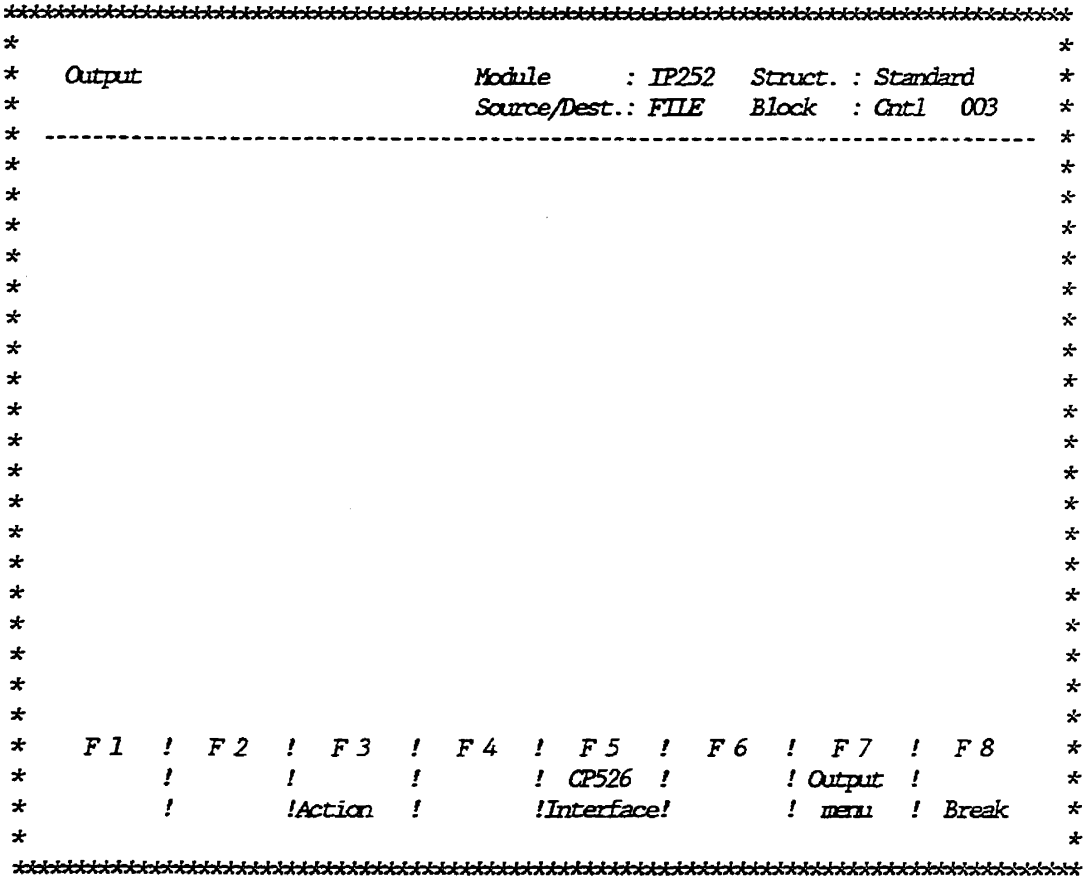


Fig. 5.13 Submenu to function "BESY parameter"

CAUTION:
When the controller is parameterized the set dimension is, if necessary, displayed initially and may be modified in the output function. The user should note that the modification of the characteristic does not cause another check of the values previously entered. After the characterstic is modified the new values may exceed the input range and lead to undefined states.

The user is completely responsible for the modification of the dimension characteristic in the output!

When the user presses function key <F2> in the main menu the output mode is called. When at this stage the "Print" function key is pressed, the "Print" menu shown in figure 5.14 is displayed.

In this menu the user may select the following printouts:

<F2> : Print block
<F5> : Print all
<F6> : Cross reference list
<F8> : Break

The cross reference list contains a list of all inputs and outputs used for each control loop. This documents the interconnection and multiple allocations may be avoided.

With the function "Print block" the controller structures, scan time and all parameters are printed for one selected control loop.

With the function "Print all" the cross reference list, controller structure, scan time and all parameters of all control loops are printed.

After the program is selected the input of the source medium is requested; when the function "Print block" is selected, the control loop number should also be entered.

When the <Break> key is pressed the user returns to the main menu.

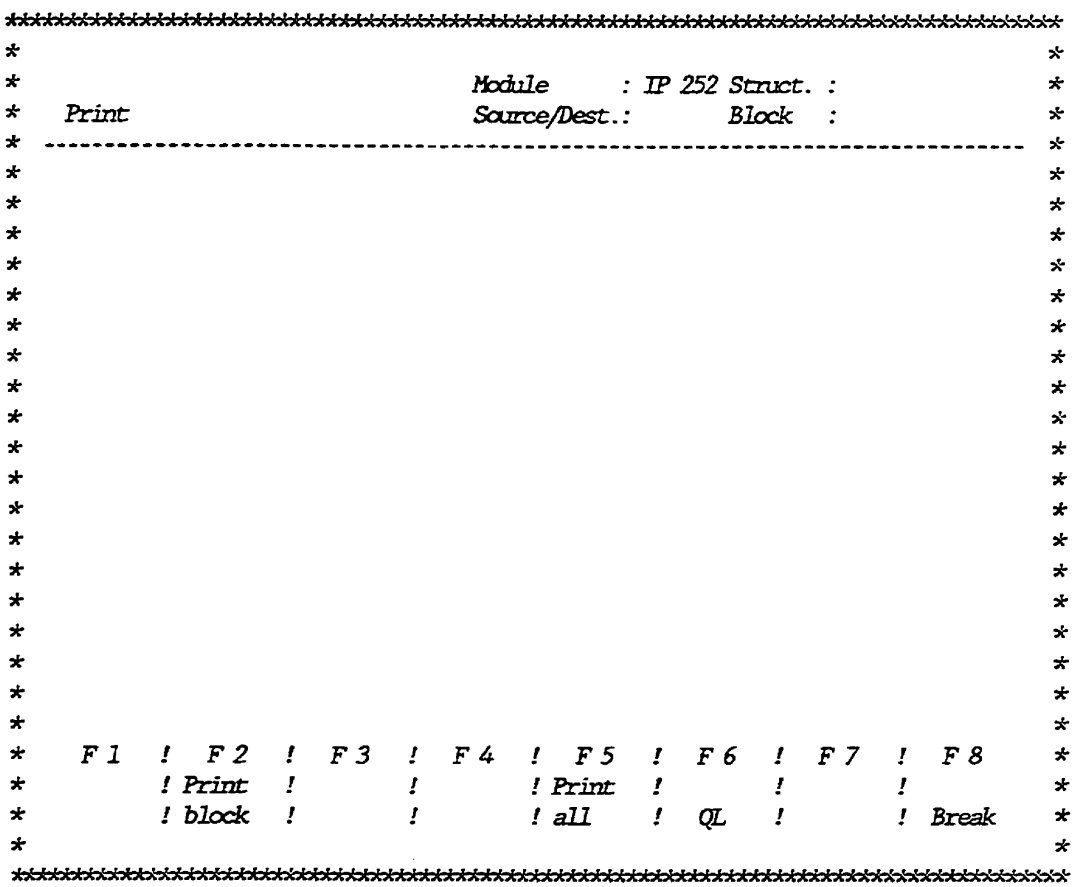


Fig. 5.14 Print menu

5.4 Transfer

With the function "Transfer" the blocks are transferred to the individual media; however, only one block can be transferred at a time.
When the function key <F4> is pressed in the main menu, the user is prompted to enter the source and the destination medium. The function keys are assigned the following:

<F1> : Program file
<F3> : Submodule
<F4> : Module
<F5> : PG
<F8> : Break

The user should additionally enter the block (parameter set) that is to be transferred. When an asterisk (*) is entered all blocks are transferred.
Afterwards the PG displays the menu shown in figure 5.15 where the function keys are assigned the following:

<F3> : Change block number
<F5> : Correct
<F7> : Transfer
<F8> : Break

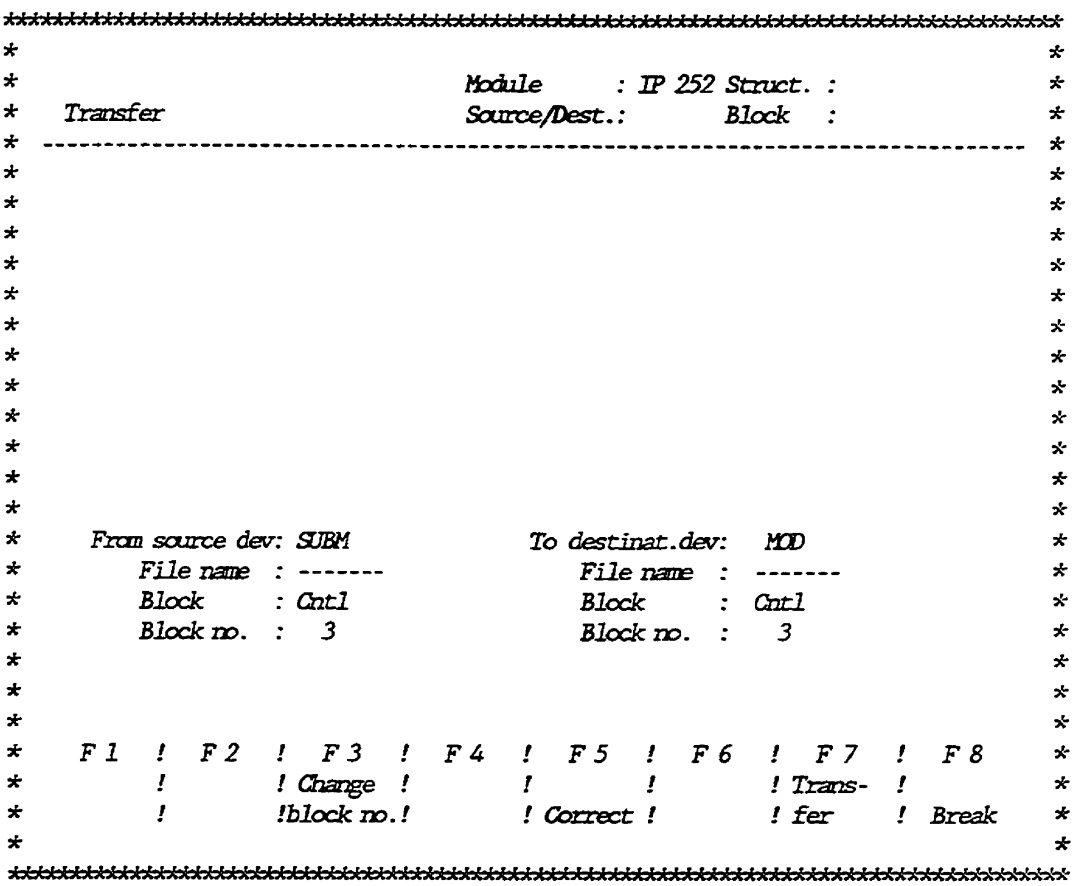


Fig. 5.15 Menu for block transfer

<F3> Change block number — This function enables the user to change the block numbers. The input of the new numbers is to be terminated by pressing <Carriage Return>.

<F5> Correct — This function enables the user to correct incorrect entries. When for example the data set is to be transferred not to the selected device (medium) but to another one the user may change this entry by means of this function.

<F7> Transfer — This function enables the transfer in the selected manner. If a block with the same number is already stored in the destination device (medium) the user is prompted "Overwrite module? (Y/N)".

<F8> Break — This function terminates the function without transferring the data.

5.5 Delete

The user may call the Delete function by pressing function key <F5> in the main menu. Then the PG displays the Delete menu shown in figure 5.16.

The following delete functions can also be called up by function keys:

<F1> Delete block from file
An individual block of the activated program file may be deleted.

<F3> Delete block from submodule
Individual controller data sets may be deleted from the memory module. When an asterisk (*) is entered instead of a controller number all controllers of this memory module are deleted.

<F4> Delete block from module
Individual controller data sets may be deleted from the module. When an asterisk (*) is entered instead of a controller number, all controllers are deleted from the module.

<F8> Delete complete prog. file
A program file is deleted completely from the disk or the Winchester. Thus all blocks are deleted from this file.

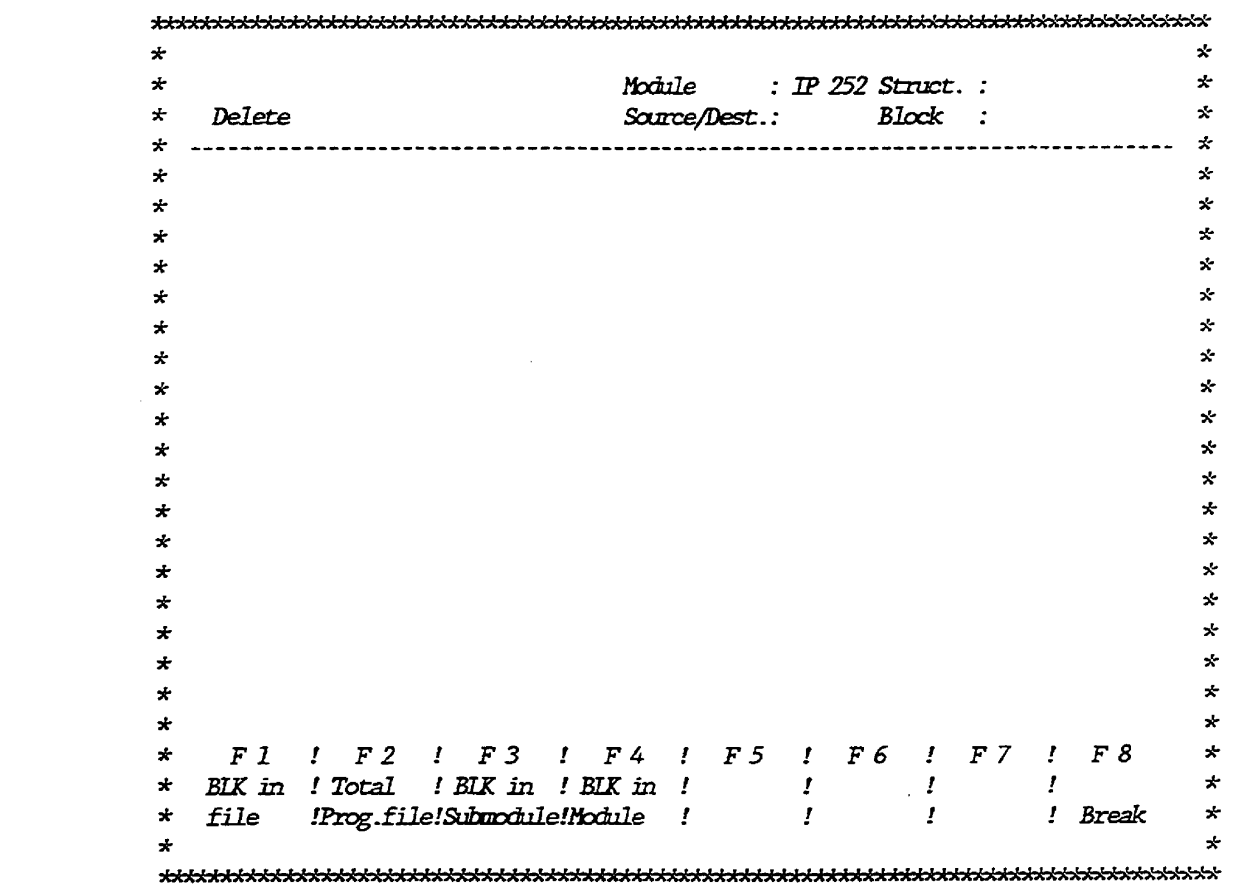


Fig. 5.16 Delete menu

5.6 Special Functions

The special functions which the user selects from the main menu by pressing <F6> offer the following:

<F1> Start
This function that only runs online, sets the operating mode "Run" for the module.

<F4> Stop
This function that only runs online, sets the operating mode "Stop" for the module.

<F6> Preset
From this menu that is shown in figure 5.2, the operating mode and the program file may be selected. When the operating mode is prompted the user should enter, offline, if programmer and programmable controller are not connected.
The user is offered two alternatives when operating the IP252:
- "IP252 without bus access" (<F3>)
- "IP252 with bus access" (<F4>).

The term "with bus access" respectively "without bus access" means that the IP252 may behave like a co-processor; if "with bus access" is selected the IP252 may access the peripheral, plugged into the same central processor, without increasing the CPU loading. This direct bus access is described in detail in section 4 and is restricted to the IP252 in the S5-115U.
The controller blocks may be stored together in a program file on an external storage medium. The file name is optional; the user should enter it into this presetting mask, to inform the program. The filename consists of 6 characters maximum; when it is smaller, the remaining places are filled with "@" signs. When no drive is selected the default drive is assumed.

<F7> Controller processing
Using this function which runs only online, the user is able to individually enable or disable the control loops of the module. This is shown in the menu in figure 5.17.

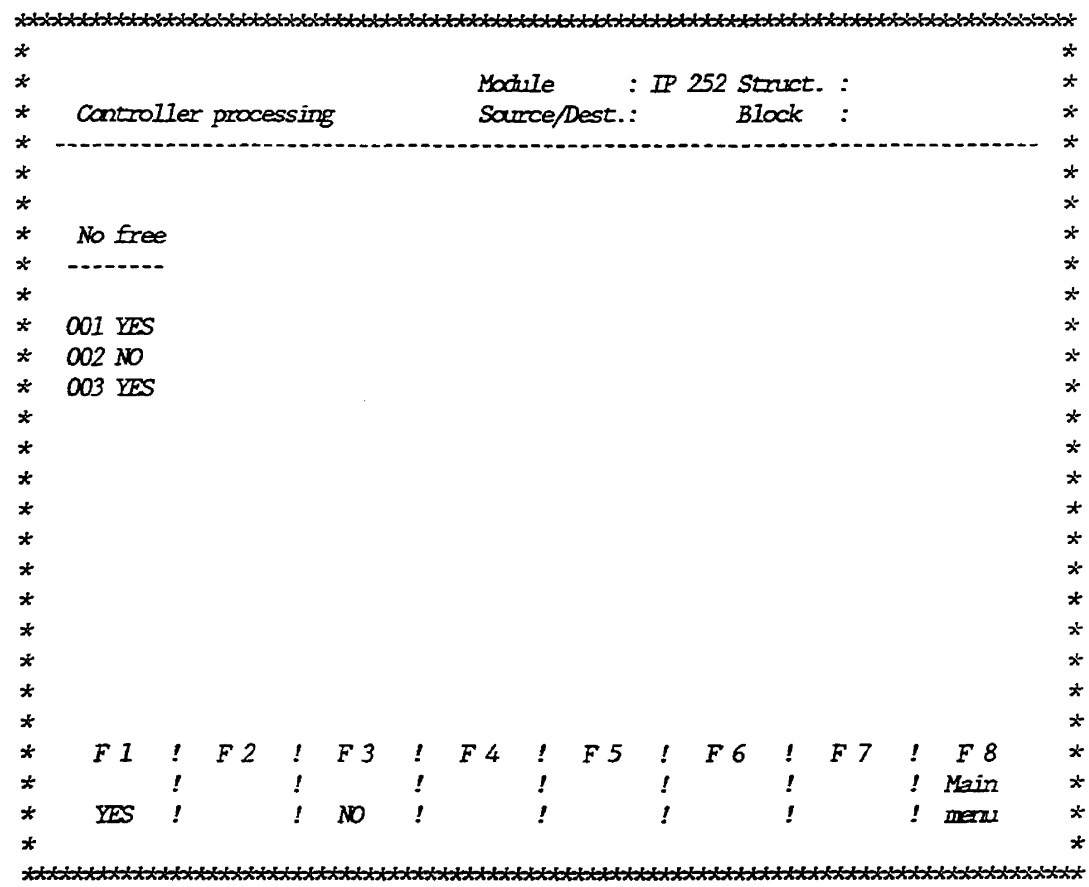


Fig. 5.17 Controller processing menu

The cursor may be moved to the individual control loops in this menu. The function keys are assigned the following:

<F1> Yes
The control loop on which the cursor is positioned is enabled.

<F3> No
The control loop on which the cursor is positioned is disabled.

<F8> Main menu
When this function key is pressed the main menu is displayed.

5.7 Information Functions on the IP252

When the user presses the function key <F7> in the main menu the menu "Information" is displayed. The function keys are assigned the following:

<F2> Directory:
After the source device is selected, the programmer supplies a list of contents of all control loops of this medium. In addition to the control loop number the structure, the version number, and the processor loading are displayed.

<F4> SYSID Module:
<F5> SYSID Submodule:
These functions that lead to the menu shown in figure 5.18 inform the user about the plant to be controlled, the module, the submodule, the version number of the firmware, and the bus addresses.
Input is possible in the fields
"Plant from : date" and
"Frame no.: x" (x = 0 to 254).
These functions were introduced in order to enable the CPU of a programmable controller to operate its IPs and CPs and in order to read the data described above for diagnostic purposes.

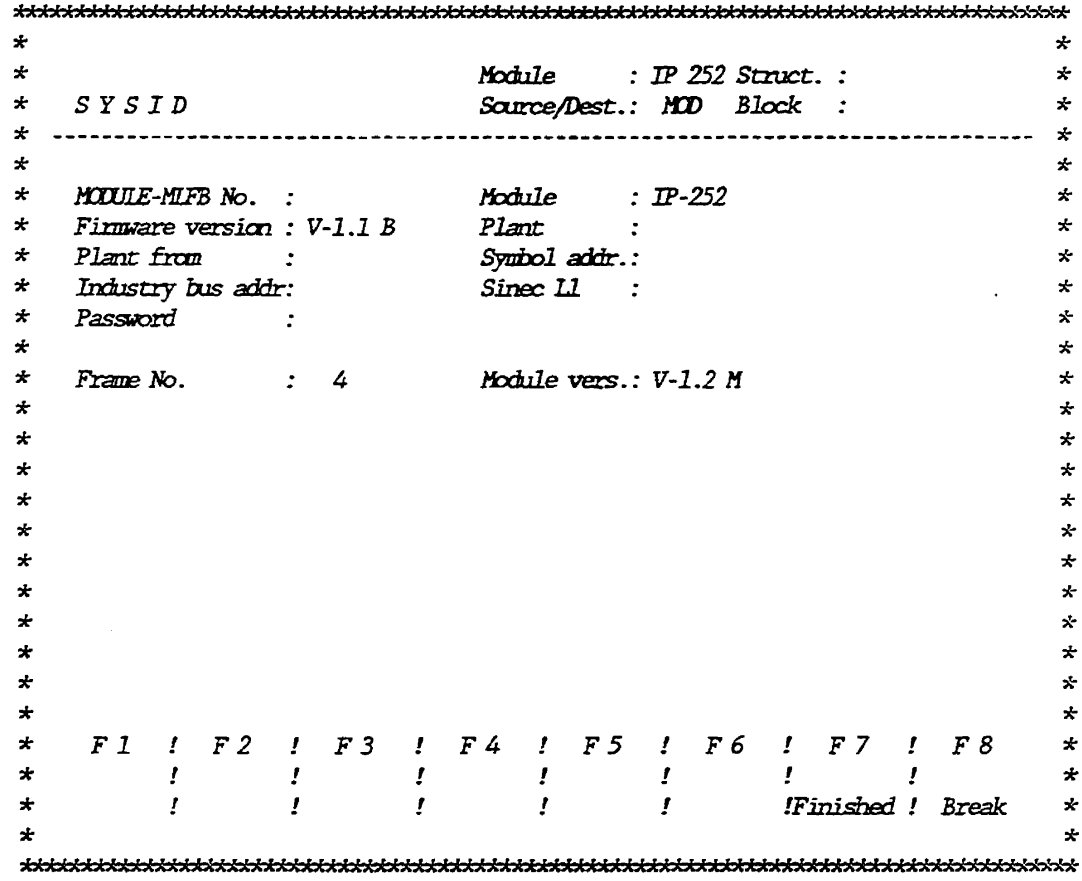


Fig. 5.18 SYSID menu

IP252 - Error diagnosis

<F6> IP252 fault
When a fault occurs in the IP252 this function displays an error message containing error number and description. Only one error message is displayed at a time, even if several errors should have occured. See figure 5.19 for the error messages.

When the "Info" function in the submenu "Error" is activated IP252 outputs the following messages for error diagnosis. The preceding fault numbers are stored as error code in the dual-port RAM (see also chapter 5, manual IP252).

Error code (decimal)	Text displayed at the PG	Error description	Reaction of IP
00.	No error	Normal state: "No error in IP252"	-
11.	Hardware	Timeout (except analog module)	"STOP"
12.	Hardware	Ckecksum of EPROMs is not valid	"STOP"
13.	Hardware	Offset check: deviation of a DAC >7LSB	"STOP"
14.	Hardware	Error in hardware test program: RAM	"STOP"
15.	Hardware	Error in hardware test program: MUART	"STOP"
20.	Watchdog	Monitoring time elapsed	"STOP"
21.	Dir. bus access	S5 bus is not enabled by S5 CPU	"STOP"
22.	Wire break at digital input	Open circuit at digital input (digital tacho)	"STOP"
23.	Error in analog sec.	Voltage supply of analog sec. has failed	"STOP"
30.	PC STOP	Inhibit command output (BASP) is active	"STOP"
31.	Subm. error	Wrong/no submodule in IP252	"STOP"
50.	Error in analog module	Timeout or open circuit in analog module	"STOP"
51.	Overload	IP 252 overloaded (time conflict)	"STOP" LED "F" flashing
70.	STOP switch	STOP switch of IP 252 in STOP position	"STOP"
71.	Software STOP	Stop of IP 252 (caused by PG or CPU)	"STOP"

The following messages only apply for the **ARS** structure

75.	Prepare self-setting		none
76.	Self-setting active		none
77.	Self-setting successfully terminated		none
78.	Structuring/ initializing error		none
79.	Invalid cntl number	Invalid controller no. (no. 1 or 2 only)	none
80.	Samp. time too long	Sampling time too long (TA=4 or 8 ms only)	none
81.	Load torque too high		none
83.	Unsuitable	Illegal procedure	none
84.	Optimization failed	Parameters could not been calculated	none
85.	Break by PG/PC	Break caused by programmer	none
86.	S5 communication error	S5 communication error with IP 240	none
87.	S5 wire break	Open circuit in IP 240 module	none

Fig. 5.19 Error messages of IP252 for error diagnosis

All messages listed above are recognized by the operating system of the IP252 and can also be fetched by the CPU from a specific RAM area of the IP252 (dual-port RAM) by means of RECEIVE 200. A fault entry will automatically be deleted when the module passes from "STOP" to "RUN".

Normally only the message which has been signalled first is entered. Message numbers 51 and 75 to 85 are an exception to this rule: These messages are overwritten by each following one!

<F7> Processor loading:
After the source device is entered this function supplies a list of all control loops together with the version number, a controller structure, the description list, and the processor loading of each control loop (IP252 loading). The total load of the individual modules must not exceed the 100% limit. See figure 5.20 for processor loading.

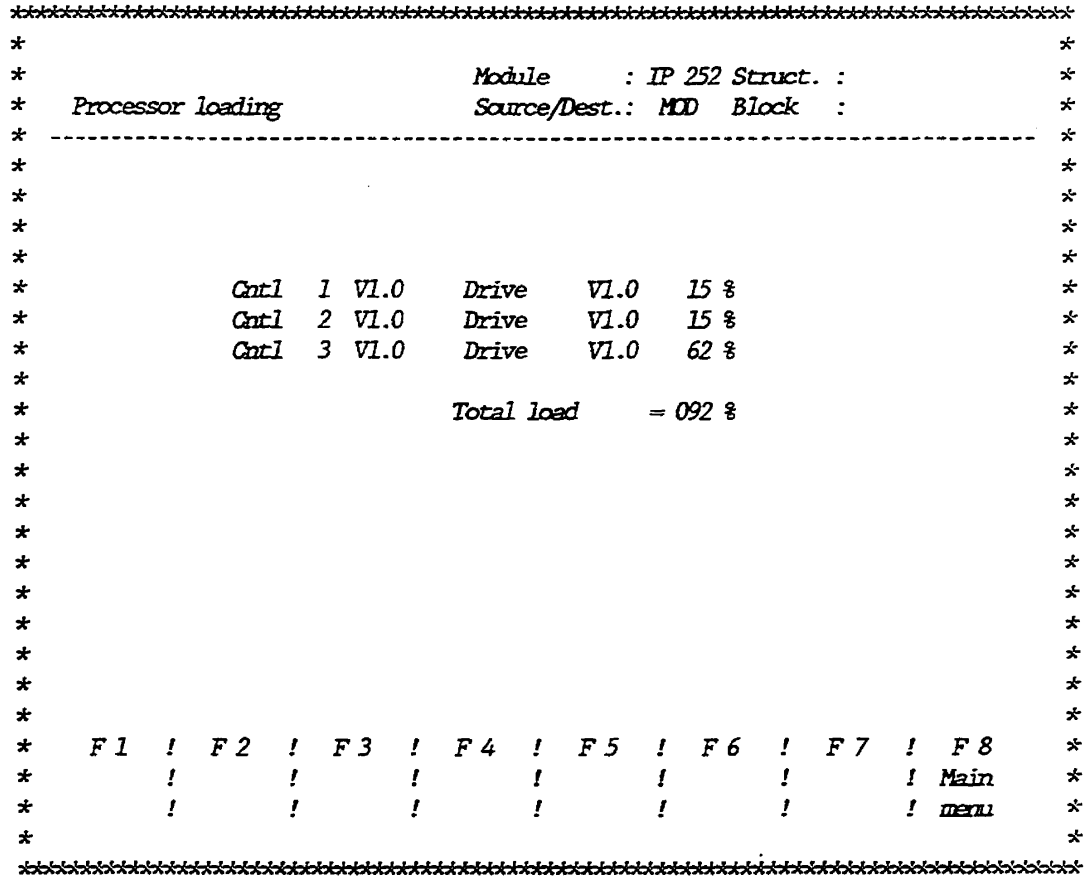


Fig. 5.20 Processor loading

5.8 Controller Test

The controller test function which the user calls by pressing function key <F3> in the main menu, enables the user to operate and monitor the controller in online mode; in addition to the input and the display of parameters (in control mode) the controller test shows actual bit values (e. g. enabling branches or limit value identification), input values (e. g. PC setpoints) or intermediate results of the algorithm (so-called measuring points). All values are requested and updated by the IP252 several times per second.

This function is suitable, in particular, for the starting-up procedure, error diagnosis and optimization of control loops.

After the user has selected the function "Test" (<F3>) and entered the control loop number, the branch selection menu is displayed which is identical to the branch overview shown in figure 5.10. In addition to the branches offered for parameterization, there are the two functions "Measuring point table" and "Starting-up" in the control test. Within the individual branches, bit values and switch positions are displayed in addition to the data displayed for parameterization. After the user has selected a branch, a mask is displayed as shown in figure 5.21, for example.

Note: The measuring points of the limit monitors can also be changed in test operation. If you wish to change measuring points of different dimensions (example: MP4 regulating variable given as %, MP1 actual value is dimension-dependent), it is necessary to re-enter the limit values in order to obtain a correct indication of the limit monitor bits.

```
*********************************************************************************
*                                                                               *
*                                Module     : IP252  Struct. : Standard       *
*   C O N T R O L L E R  T E S T Source/Dest.: MOD    Block   : Cntl  003       *
*  ---------------------------------------------------------------------------- *
*                                                                               *
*     Branch  1: Controller                                                     *
*                                                                               *
*                                                                               *
*     MP 03: Controller deviation      -  15.0 GRAD_C                           *
*                                                                               *
*     Continuous/ Step                    0                                     *
*       Standard/Upgraded                 0                                     *
*         Manuel input PG/ADC             0                                     *
*           Constant man. value         0.00 %                                  *
*                                                                               *
*         Automatic/Manual                0                                     *
*         MP 10: Manual value           0.00 %                                  *
*         Controller enable               1        <--                          *
*         High overflow ID                0                                     *
*         Lower overflow ID               0                                     *
*                                                                               *
*                                                                               *
*                                                                               *
*    F 1  !  F 2  !  F 3  !  F 4  !  F 5  !  F 6  !  F 7  !  F 8              *
*         !       !       ! Scroll ! Scroll !       ! Next   ! Branch           *
*         ! Force !       !   up   !  down  !       ! Branch !selection         *
*                                                                               *
*********************************************************************************
```

Fig. 5.21 Controller test menu

During this controller test the user may change each value except the structure switches. These changes should be made by means of the "Force" mode.

Controlling in the controller test

During the normal controller test the cursor is not visible. If the user wishes to modify a parameter during RUN he may initiate it by pressing the "Force" key. The result is that the cyclical status request is terminated and the last display is frozen. Then the cursor appears on the first field that can be accessed with the "Force" key respectively on the field that was accessed last in this section. In the "Force" mode, the cursor control is the same as in parameterization mode, except that the cursor can only be moved to "accessible" fields.

When a parameter, on which the cursor is to be moved, should be changed the existing value is overwritten by the new one and the entry is terminated by pressing the Return key. The PG transfers this new value immediately to the IP252, cancels the control mode and returns to the cyclical status output.

Non-controllable values: addresses
measuring points
number of limit values
number of vertices/setpoints

The following parameters may cause conflicts in the overlayed control of the IP252 since these parameters also may access the same lines via dual-port-RAM operation:

PC setpoints
PC enabling

6.4 Using the Controller in Multiple Control Loops

The R64 controller structure was developed to provide a SIMATIC control algorithm for single control loops such as those required for temperature, level and volumetric flow control. The algorithm had to be able to be put into use with little structuring overhead while still providing the functions for processing setpoints, actual values and regulating variables so often required for complex applications in the field of process engineering. The controller's application range can be expanded by conforming to certain rules. It can, for instance, be used without excessive difficulty for implementing multiple control loops. Indeed, it is even possible to implement systems for blending and ratio control with the help of STEP 5 programs.

Since interconnection of two or more controllers over DACs and ADCs converters would serve no practical purpose, a master controller's output might be applied to a flag (or possibly an interprocessor communication flag). The secondary controller could then read in its input value (e.g. setpoint) from this flag. Both flags and interprocessor communication flags are permissible as inputs and outputs for the controller. In a controller cascade, this would mean, for example, that the master controller's regulating variable output could be connected to the slave controller's setpoint input.

The next question is how to invoke the controllers via DB2. In a multiple control loop, it is often necessary to invoke the controllers successively in a predetermined order without a delay. The interval between execution of the various controller programs is often decisive for the stability of the control loop, particularly in critical control systems.

At this point, it should be noted that invoking controllers in a non-critical controller cascade present no problems. The interval between calls is of virtually no relevance when the master controller's scan time is much longer than the slave controller's (e.g. a ratio of 10:1). The following procedure is recommended as regards entry of the controllers in the DB2 list, however, when the controllers have scan times of approximately equal length.

The order in which controllers are invoked is determined by the order in which there are entered (controller DBs) in DB2 (controller list). On a cold restart, the operating system ascertains which controllers are in the list. Depending on the time base, the number of columns in which controllers have been entered, and the scan time for each column, the order in which the controllers are invoked and the interval between calls are determined as follows:

- First, the column interval is determined by dividing the time base by the number of columns in which controllers have been entered.

- Then one controller is invoked each time a column is processed. The first controller is invoked and then the next and so on until each controller in the column has been invoked once. The first controller in the column, however, is not invoked until the column scan time has expired. Depending on the number of controllers in a column, a column could be processed without a controller being invoked.

This is illustrated by the following example.

```
Input                              Module      : R-Proc. Struct. :
Controller list                    Source/Dest.: File    Block   : DB   002
-------------------------------------------------------------------------------

   Time base :    100 ms

   The Scan times of the columns are:

 100 ms | 500 ms |  1 s  |  5 s  |  0 s  |  0 s  |  0 s  |  0 s
========+========+=======+=======+=======+=======+=======+=======
   DB   |   DB   |  DB   |  DB   |  DB   |  DB   |  DB   |  DB
--------+--------+-------+-------+-------+-------+-------+-------
   10   |   11   |  13   |  15   |       |       |       |
   ---  |   12   |  14   |       |       |       |       |
   ---  |        |       |       |       |       |       |
   ---  |        |       |       |       |       |       |
   ---  |        |       |       |       |       |       |
   ---  |   ---  |       |       |       |       |       |
   ---  |   ---  |       |       |       |       |       |
   ---  |   ---  |       |       |       |       |       |
   Enter Contr. DB-No.:

 F 1  !  F 2  !  F 3  !  F 4  !  F 5  !  F 6  !  F 7  !  F 8
      !       !       !       !       !       !       !
      !       !       !       !       !       !       !  Break
```

Figure 6-3: Controller list with six controllers

Figure 6.3 shows four columns with a total of six controllers. The time base is 100 ms, i.e. one of the columns is due for processing every 25 ms (time base / number of columns = 100 ms/4 = 25 ms). The columns are processed beginning with the first.

The controllers in a column are invoked from top to bottom, i.e. beginning with the uppermost line. This means that the first time column two is processed, controller 11 is called first and controller 12 100 ms later (time base). No controller in the second column is due to be called in the next three column processing cycles (at intervals of 100 ms). Controller 11 is reinvoked in the sixth column processing cycle; precisely 500 ms (column scan time) have passed in the interim.

Figure 6-4 shows the call sequence for the example over a period of 1 second.

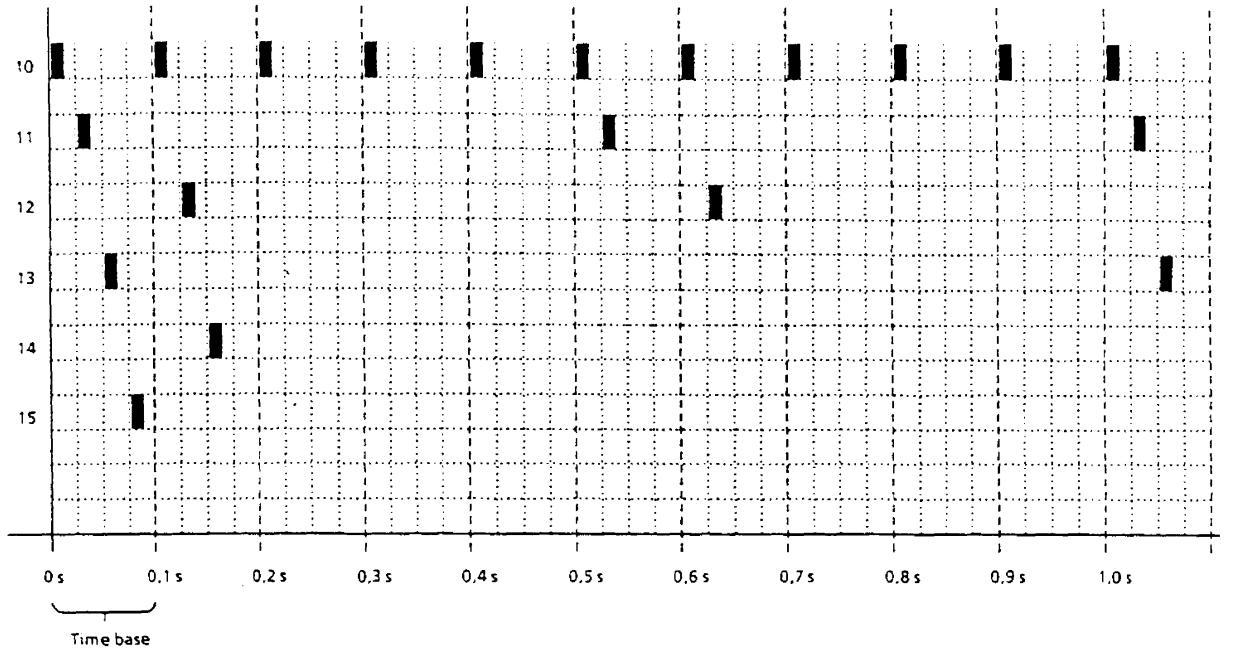


Figure 6-4: Controller call sequence

The times not marked for processing the controller programs are available to the cyclic STEP 5 background task and for time-controlled processing. This means that any STEP 5 program can execute during this time, including one which establishes a connection between multiple loop controllers. Note, however, that the STEP 5 programs and the controller programs execute asynchronously.

6.4.1 Cascading Two Controllers

In the next example, controllers 11 and 12 are to be cascaded. This requires as brief an interval as possible between controller calls. This can be achieved by entering the controllers on the same line in two contiguous columns and setting the time base to 20 ms (the lowest possible value).
Since COM REG does not permit the user to enter a controller at a specific list location, it is necessary to employ a "trick." A controller can be entered more than once in a column. If controller 11 is entered 11 times in the column, the next entry will automatically place controller 12 in the third column (→ Figure 6-5).

```
Input                              Module      : R-Proc. Struct. :
Controller list                    Source/Dest.: File    Block   : DB   002
-------------------------------------------------------------------------------

   Time base :     20 ms

   The Scan times of the columns are:

 100 ms | 500 ms | 500 ms |  0 s  |  0 s  |  0 s  |  0 s  |  0 s
========+========+========+=======+=======+=======+=======+=======
   DB   |   DB   |   DB   |  DB   |  DB   |  DB   |  DB   |  DB
--------+--------+--------+-------+-------+-------+-------+-------
   10   |   11   |   12   |       |       |       |       |
  ---   |   11   |        |       |       |       |       |
  ---   |   11   |        |       |       |       |       |
  ---   |   11   |        |       |       |       |       |
  ---   |   11   |        |       |       |       |       |
  ---   |   11   |        |       |       |       |       |
  ---   |   11   |        |       |       |       |       |
  ---   |   11   |        |       |       |       |       |

   F 1  !  F 2  !  F 3  !  F 4  !  F 5  !  F 6  !  F 7  !  F 8
        !       !       !       !       !       !       !
Enter DB !Remove DB!Time base!       !       !       ! Ready ! Break
```

Figure 6-5: Multiple entries in the controller list

Controller 11 must now be removed from the list seven times, and the remaining controllers entered. Figure 6-6 shows the finished list.

Controllers 11 and 12 will now be invoked successively with the least possible delay.

It is also possible to cascade controllers that have different scan times. Normally, the master controller has the longest scan time and is invoked first. Even in this case, however, both controllers must be entered on the same line in the controller list.

```
Input                              Module      : R-Proc. Struct. :
Controller list                    Source/Dest.: File    Block   : DB   002
-------------------------------------------------------------------------------

   Time base :     20 ms

   The Scan times of the columns are:

 100 ms | 500 ms | 500 ms |  1 s  |  5 s  |  0 s  |  0 s  |  0 s
========+========+========+=======+=======+=======+=======+=======
   DB   |   DB   |   DB   |  DB   |  DB   |  DB   |  DB   |  DB
--------+--------+--------+-------+-------+-------+-------+-------
   10   |   11   |   12   |  13   |  15   |       |       |
  ---   |        |        |  14   |       |       |       |
  ---   |        |        |       |       |       |       |
  ---   |        |        |       |       |       |       |
  ---   |        |        |       |       |       |       |
  ---   |        |        |       |       |       |       |
  ---   |        |        |       |       |       |       |
  ---   |        |        |       |       |       |       |

   F 1  !  F 2  !  F 3  !  F 4  !  F 5  !  F 6  !  F 7  !  F 8
        !       !       !       !       !       !       !
Enter DB !Remove DB!Time base!    !       !       ! Ready ! Break
```

Figure 6-6: Controllers 11 and 12 in cascade

When initializing the two controllers, make sure that they are interfaced over a flag word or interprocessor communication flag word in accordance with the control loop structure. The master controller enters its regulating variable in a flag word, and the slave controller reads this value as its setpoint. Figure 6-7 shows an example of a controller cascade. Figure 6-7 shows an example of a controller cascade.

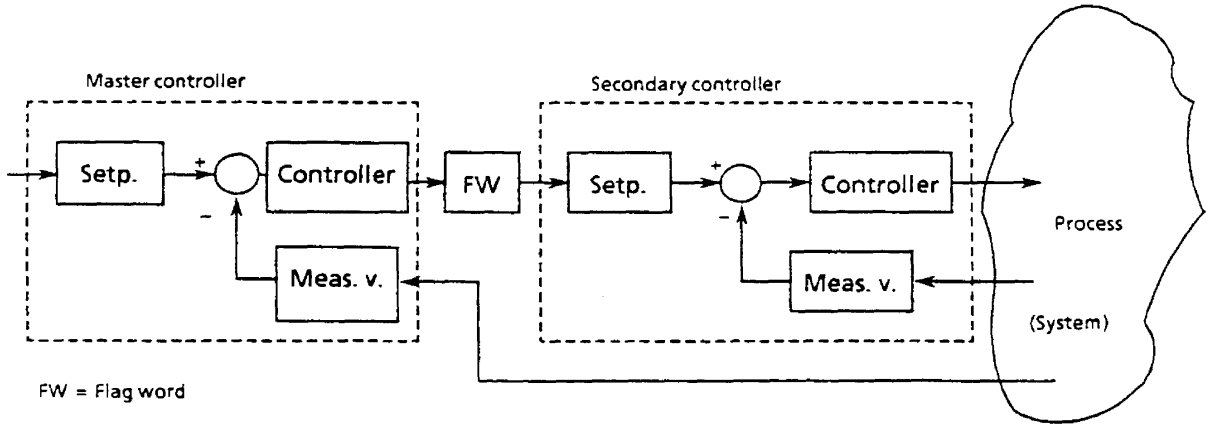


Figure 6-7: Principle of a controller cascade

It is also possible to cascade more than two controllers: the procedure is basically the same. Suitable scan times must be chosen for the controllers to enable synchronous operation. For example, two controllers with scan times of 500 and 700 ms cannot be "in step."

6.4.2 Implementing a Blending Control System

This section discusses how to implement a blending control system using the R64 controller structure. The illustration shows the difference between blending control and cascaded control.

The control system illustrated below is to be used to regulate the ratio between a primary component and an additive. A master controller is to regulate the total amount (the diagram is the same in principle, regardless of the number of additives).

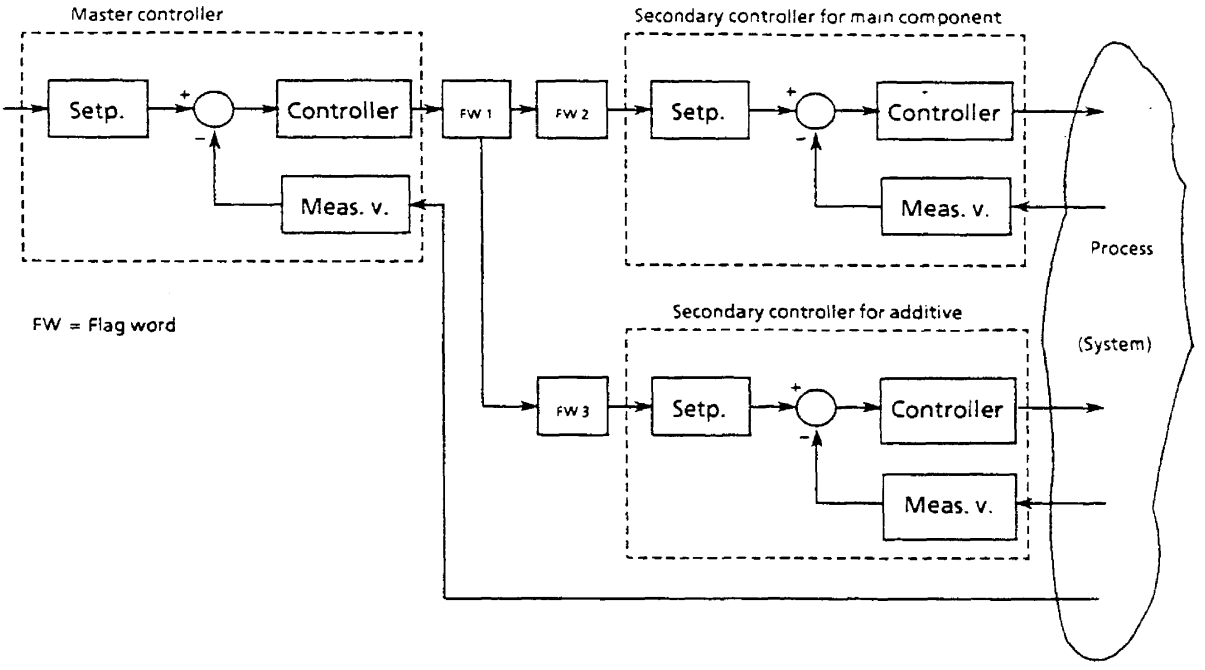


Figure 6-8: Principle of a mixing control system

The percentage of each component is computed by multiplying the setpoint of the component by a ratio factor.

As in a cascaded control system, flag words are used to interchange the slave controller's setpoint and the master controller's regulating variable. The only difference is that each component's setpoint must be multiplied by a ratio factor.

For this reason, a different flag word (FW 1 in the example) is used as the master controller's regulating variable output than for the secondary controller's setpoint inputs (FW 2 and FW 3 in the example). If a control program is used, the regulating variable in flag word FW 1 can be multiplied with a ratio factor and then transferred to the flag word which the relevant secondary controller uses as its setpoint input.

The setpoint for the secondary controller used for regulating the primary component is now multiplied with a ratio factor of 0.9 (= 90 %).
The STEP 5 command sequence required is as follows:

: L FW1	Read master controller's regulating variable (in U-range format)
: SSW 4	Convert into SIMATIC fixed-point format (by shifting the value four bits to the right, the equivalent of dividing it by 16)
: JU OB220	Convert into double-word
: FDG	Convert into SIMATIC floating-point format
: L KG + 1440000 + 02	Multiply with 14.4 = 0.9 * 16; 0.9 = ratio factor. A division by 16 is reversed by multiplication by 16.
: GFD	Convert into fixed-point number in U-range format, since it is already multiplied by 16
: T FW2	Transfer to flag word 2, i.e. the flag word which the secondary controller uses to read its setpoint

This routine could, for example, be incorporated into OB1. The routine must be called cyclically and the cycle time must be shorter than the master controller's scan time.
The ratio factor for the additive can be programmed in the same way. To do so, flag word 2 can be replaced by flag word 3, and a different factor can be used in the sample program shown on the preceding page.

6.4.3 Implementing a Ratio Control System

In a ratio control system, the master controller's actual value, multiplied with a ratio factor, serves as setpoint for the secondary controller. In contrast to a mixing control system, there is no controller for regulating the total amount.

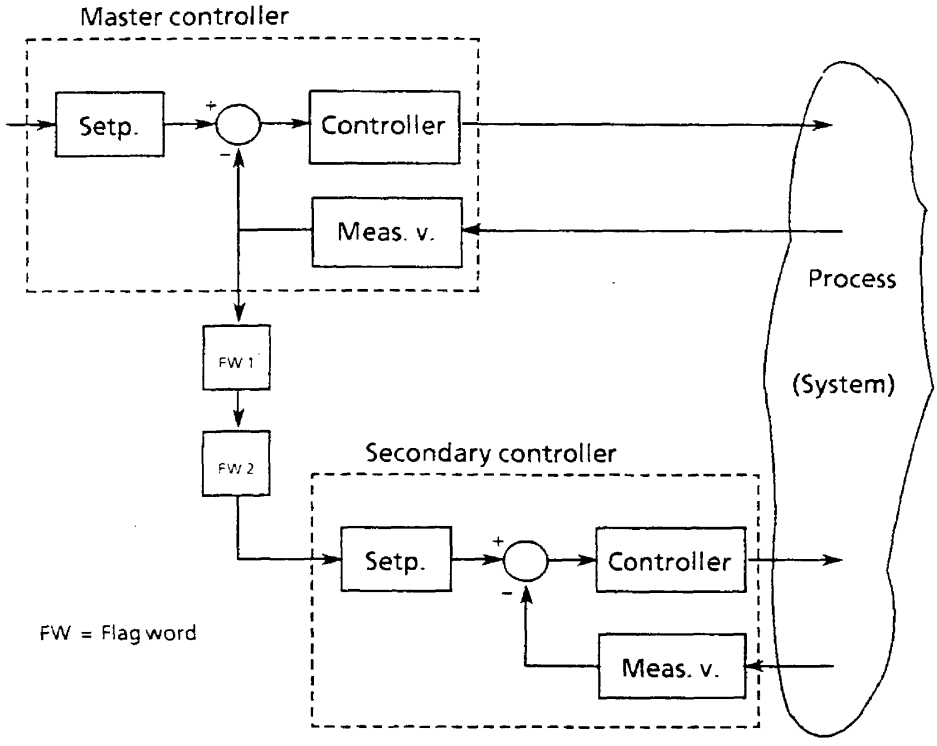


Figure 6-9: Principle of a ratio control system

The same STEP 5 program (i.e. the one shown on page 6-46) can be used to interface the master controller's actual value and the secondary controller's setpoint. The master controller's actual value must be transferred to a flag word through a test socket. Either the actual value that was read in (measuring point 8) or the processed actual value (measuring point 9) may be used.

6.5 Adaptive Parameter Entry

When adapting the controller parameters to suit the requirements of the controlled system, you may need to modify the characteristic controller values KP, TN and TV as a function of certain conditions. For example, it might become necessary to set the controller parameters so as to respond to setpoint changes on the one hand, and to optimize them to respond to interferences on the other. This can be done using a STEP 5 program.

For example, the parameters could be set to respond to setpoint changes when the deviation exceeds 20%, and then to respond to interferences when the deviation falls below 10%. This hysteresis helps to avoid a continual back-and-forth effect.

The STEP 5 program shown below might help to solve similar problems.

Physical dimensions: 0% = 0.000 volts
100 % = 10.000 volts

```
        : C   DB50                  Select controller DB, e.g. DB50
        : L   DW 234                Deviation at MP 3
        : L   KF + 2000             Load 20 %
        : >F
        : JC  = FUER                Jump if value > 20 %
        :
        : SRW 1                     Convert 20 % into 10 % in ACCUM1
        : >F
        : BEC                       End of block if value > 10 %
        :
        : L   DW106                 DW106 as flag: 000 means:
        : L   KH0000                Parameter set 1 entered
        : ><F
        : BEC                       End of block if < > 0000
        :                           Enter parameter set 2
        : L   KF + 0150
        : T   DW73                  Parameter KP = 1.50
        : L   KF + 1033
        : T   DW75
        : L   KF + 04
        : T   DW76                  Integral-action time TN = 1.33
        :                           (133 with identifier 4)
        : L   KHFFFF
        : T   DW106                 DW106 = FFFF and means:
        :                           Parameter set 2 entered
        : O   F 0.0
        : ON  F 0.0                 Set RLO
        : S   D 7.9                 Set operator control bit 2
        : JU  FB102                 Invoke controller for
NAME    : FB102STA                  parameter conversion
DBR     :     DBxxx                 Controller DB xxx
        : BEU
        :
FUER    : L   DW106                 DW106 as flag: 0000 means:
        : L   KH0000                Parameter set 1 entered
        : = F
        : BEC                       End of block if = 0000
        :
        :                           Enter parameter set 1:
        : L   KF + 0080
        : T   DW73                  Parameter KP = 0.80
        : L   KF + 0254
        : T   DW75
        : L   KF + 04
        : T   DW76                  Integral-action time TN = 2.54
        :                           (254 with identifier 4)
        : L   KH0000
        : T   DW106                 DW106 = 0000, which means:
        :                           Parameter set 1 entered
        : O   F 0.0
        : ON  F 0.0                 Set RLO
        : S   D 7.9                 Set operator control bit 2
        : JU  FB102                 Invoke controller for
NAME    : FB102STA                  parameter conversion
DBR     :     DBxxx                 Controller DB xxx
        : BE
```

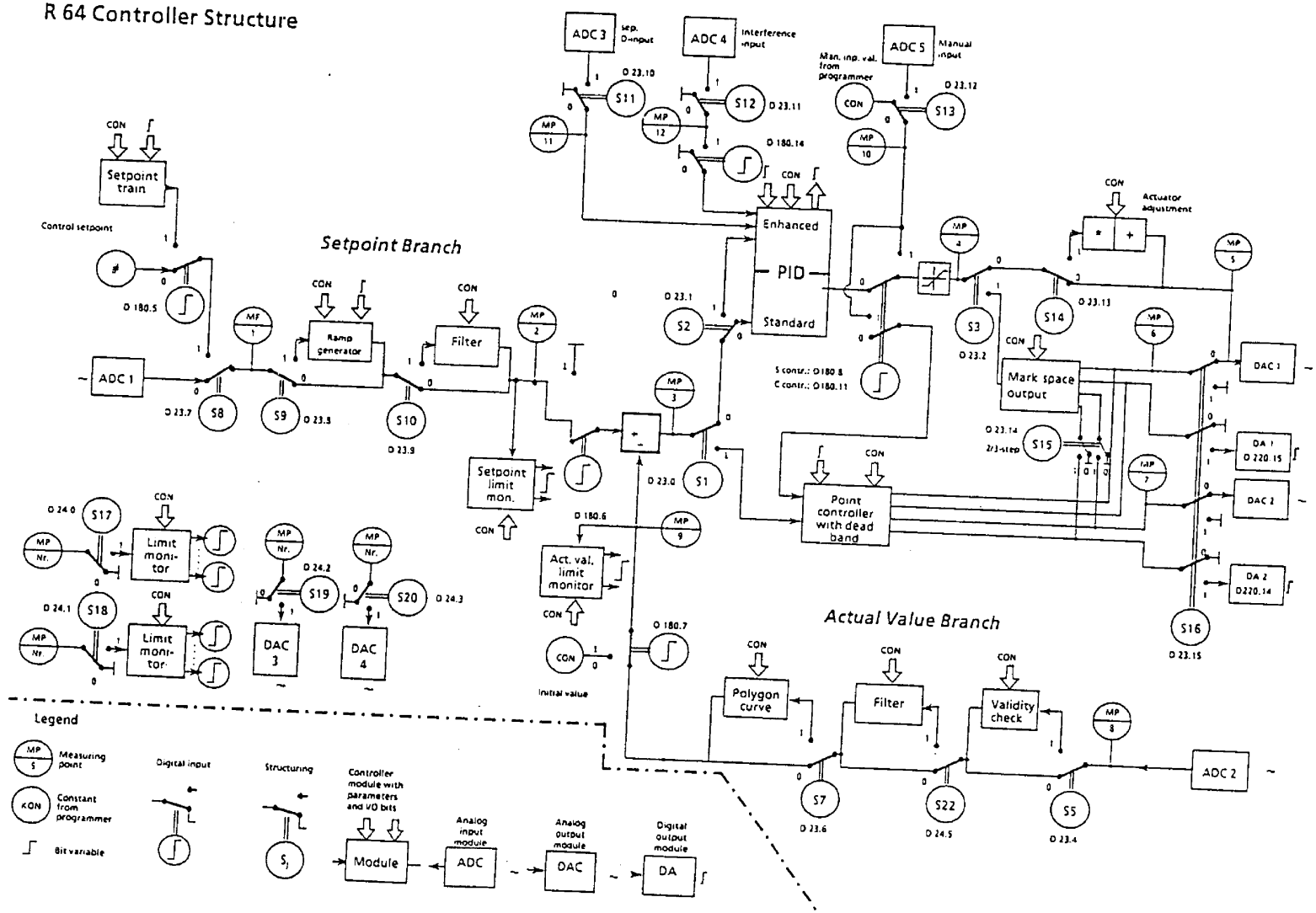
7 Structure of the Data Block

The tables on the next few pages show the structure of the controller data block from DW0 to DW255. The input and output bits which appear in the data block are described in detail at the end of the tables.
DW256 to DW511 are used for parameters and historical values to which only the controller program has access.

Note: DW208 to DW255 are set to zero on a cold restart.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Auxiliary identifier (for programmer)									FB number							DW 0
Version number (is entered by COM REG)																DW 1
RINa	Status word						RFB	PLB	PAA			WAB	RIB	RAB	RIN	DW 2
Error word															AZF	DW 3
Error mask word															MAZF	DW 4
(free)																DW 5
(free)																DW 6
						BED2	BED1						RNEU	RVBa	RVB	DW 7
Scan time TA in 2.5 ms units High Word																DW 8
Low Word (32-bit fixed-point)																DW 9
Scan time TA (Time format)																DW 10
Time format identifier for TA																DW 11
Minimum pulse duration Tmin in 2.5 ms units High Word																DW 12
Low Word (32-bit fixed-point)																DW 13
Minimum pulse duration Tmin (Time format)																DW 14
Time format identifier for Tmin																DW 15
(free)																DW 16
6-byte ASCII string (ASCII)																DW 17
specifying the controller's (ASCII)																DW 18
physical dimension (ASCII)																DW 19
Minimum range (= 0% value) (with dimension)																DW 20
Maximum range (= 100% value) (with dimension)																DW 21
Decimal point identifier (= no. of places after the ".") (w/o dimension)																DW 22
S16	S15	S14	S13	S12	S11	S10	S9	S8	S7	S6	S5	S4	S3	S2	S1	DW 23
Structuring switches									S23	S22	S21	S20	S19	Ss18	S17	DW 24
No. N (= 1,...,10) of equidistant interpolation points in polygon curve (w/o dimension)																DW 25
Abscissa value of 1st interpolation point in polygon curve (with dimension)																DW 26

R 64 Controller Structure



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Distance between adjacent interpolation points in the polygon curve														(w. dimension)		DW 27
Ordinate value of the 1st interpolation point in the polygon curve														(w. dimension)		DW 28
Ordinate value of the 2nd interpolation point in the polygon curve														(w. dimension		DW 29
Ordinate value of the 3rd interpolation point in the polygon curve														(w. dimension)		DW 30
Ordinate value of the 4th interpolation point in the polygon curve														(w. dimension)		DW 31
Ordinate value of the 5th interpolation point in the polygon curve														(w. dimension)		DW 32
Ordinate value of the 6th interpolation point in the polygon curve														(w dimension)		DW 33
Ordinate value of the 7th interpolation point in the polygon curve														(w. dimension)		DW 34
Ordinate value of the 8th interpolation point in the polygon curve														(w. dimension)		DW 35
Ordinate value of the 9th interpolation point in the polygon curve														(w. dimension)		DW 36
Ordinate value of the 10th interpolation point in the polygon curve														(w. dimension)		DW 37
No. of setpoint values in the setpoint train														(w/o dimension)		DW 38
Interval for setpoint train														(Time format)		DW 39
Time format identifier for interval																DW 40
1st setpoint in setpoint train														(w. dimension)		DW 41
2nd setpoint in setpoint train														(w. dimension)		DW 42
3rd setpoint in setpoint train														(w. dimension)		DW 43
4th setpoint in setpoint train														(w. dimension)		DW 44
5th setpoint in setpoint train														(w. dimension)		DW 45
6th setpoint in setpoint train														(w. dimension)		DW 46
7th setpoint in setpoint train														(w. dimension)		DW 47
8th setpoint in setpoint train														(w. dimension)		DW 48
9th setpoint in setpoint train														(w. dimension		DW 49
10th setpoint in setpoint train														(w. dimension)		DW 50
Ramp generator's ramp-up time														(Time format)		DW 51
Time format identifier for ramp generator																DW 52
Ramp generator's ramp-down time														(Time format)		DW 53

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Time format identifier for ramp-down time																DW 54
Increment for ramp generator														(w. dimension)		DW 55
Time constant for filter in setpoint branch														(Time format)		DW 56
Time format identifier for filter time constant																DW 57
Max. perm. deviation for validity check														(w. dimension)		DW 58
No. of limiting values for 1st limit monitor														(w/o dimension)		DW 59
1st limiting value for 1st limit monitor														(dimension or %)		DW 60
2nd limiting value for 1st limit monitor														(dimension or %)		DW 61
3rd limiting value for 1st limit monitor														(dimension or %)		DW 62
4th limiting value for 1st limit monitor														(dimension or %)		DW 63
5th limiting value for 1st limit monitor														(dimension or %)		DW 64
6th limiting value for 1st limit monitor														(dimension or %)		DW 65
No. of limiting values for 2nd limit monitor														(w/o dimension)		DW 66
1st limiting value for 2nd limit monitor														(dimension or %)		DW 67
2nd limiting value for 2nd limit monitor														(dimension or %)		DW 68
3rd limiting value for 2nd limit monitor														(dimension or %)		DW 69
4th limiting value for 2nd limit monitor														(dimension or %)		DW 70
5th limiting value for 2nd limit monitor														(dimension or %)		DW 71
6th limiting value for 2nd limit monitor														(dimension or %)		DW 72
Proportional value KP for PID controller (point controller, see DW 83)														(w/o dimension)		DW 73
Sep. gain R of the PID controller's P component (enhanced version)														(w/o dimension)		DW 74
Integral-action time TN for PID controller (point controller, see DW 84)														(Time format)		DW 75
Time format identifier for TN																DW 76
Derivative-action time for PID controller (point controller, see DW 86)														(Time format)		DW 77
Time format identifier for TV																DW 78
Upper limit of PID controller's regulating variable														(%)		DW 79
Lower limit of PID controller's regulating variable														(%)		DW 80

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Upper actuator correcting rate limiting value for PID controller (%)																DW 81
Lower actuator correcting rate limiting value for PID controller (%)																DW 82
Proportional value KP for point controller (w/o dimension)																DW 83
Integral-action time TN of point controller (Time format)																DW 84
Time format identifier for TN																DW 85
Derivative-action time TV of point controller (Time format)																DW 86
Time format identifier for TV																DW 87
Actuator runtime TM (point controller) (Time format)																DW 88
Time format identifier for TN																DW 89
Upper response threshold of dead band for point controller (w. dimension)																DW 90
Lower response threshold of point controller (w. dimension)																DW 91
Threshold value of the mark space output (%)																DW 92
Adjustment factor of the mark space output (w/o dimension)																DW 93
Initial actual value for actual value branch (w. dimension)																DW 94
Manual programmer input value for PID of point controller (%)																DW 95
Factor for actuator adjustment																DW 96
Additive constant for actuator adjustment (%)																DW 97
Upper setpoint limiting value (limit monitor in setpoint branch) (w. dimension)																DW 98
Lower setpoint limiting value (limit monitor in setpoint branch) (w. dimension)																DW 99
Upper warning limit for actual value (w. dimension)																DW 100
Lower warning limit for actual value (w. dimension)																DW 101
Upper danger limit for actual value (w. dimension)																DW 102
Lower danger limit for actual value (w. dimension)																DW 103
Time constant for filter in actual value branch (Time base)																DW 104
Time format of the filter time constant																DW 105
(unassigned)																DW 106
(uanssigned)																DW 107

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
(unassigned)																DW 108
(unassigned)																DW 109
(unassigned)																DW 110
(unassigned)																DW 111
(unassigned)																DW 112
(unassigned)																DW 113
(unassigned)																DW 114
(unassigned)																DW 115
Measuring point no. for test socket 1													(w/o dimension)			DW 116
Measuring point no. for test socket 2													(w/o dimension)			DW 117
Measuring point no. for limit monitor 1													(w/o dimension)			DW 118
Measuring point no. for limit monitor 2													(w/o dimension)			DW 119
Address of setpoint input's process image (ADC 1)													(I/O addr.)			DW 120
Address of actual value input's process image (ADC 2)													(I/O addr.)			DW 121
Address of sep. D input's process image (ADC 3)													(I/O addr.)			DW 122
Address of disturbance input's process image (ADC 4)													(I/O addr.)			DW 123
Address of manual input's process image (ADC 5)													(I/O addr.)			DW 124
Address list for process input image																DW 125
Address list for process input image																DW 126
Address list for process input image																DW 127
Address list for process input image																DW 128
Address list for process input image																DW 129
Address list for process input image																DW 130
Address list for process input image																DW 131
Address list for process input image													(I/O addr.)			DW 132
Address list for process input image																DW 133
Address list for process input image																DW 134

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Address list for process input image																DW 135
Address list for process input image																DW 136
Address list for process input image																DW 137
Address list for process input image																DW 138
Address list for process input image																DW 139
Address list for process input image																DW 140
Address list for process input image																DW 141
Address list for process input image																DW 142
Address list for process input image																DW 143
Address of process image for correction output DAC 1															(I/O addr.)	DW 144
Address of process image for correction output DAC 2															(I/O addr.)	DW 145
Address list for process output image																DW 146
Address list for process output image																DW 147
Address list for process output image																DW 148
Address list for process output image																DW 149
Address list for process output image																DW 150
Address list for process output image																DW 151
Address list for process output image																DW 152
Address list for process output image																DW 153
Address of process image for 1st test socket (DAC 3)															(I/O addr.)	DW 154
Address of process image for 2nd test socket (DAC 4)															(I/O addr.)	DW 155
Address of process image of output bits															(I/O addr.)	DW 156
Address list for process output image																DW 157
Address list for process output image																DW 158
Address list for process output image																DW 159
Address list for process output image																DW 160
Address list for process output image																DW 161

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Address list for process output image																DW 162
Address list for process output image																DW 163
Address list for process output image																DW 164
Address list for process output image																DW 165
Address list for process output image																DW 166
Address list for process output image																DW 167
Process image of the setpoint input (ADC 1)														(Analog I/O format)		DW 168
Process image of the actual value input (ADC 2)														(Analog I/O format)		DW 169
Process image of the sep. D input (ADC 3)														(Analog I/O format)		DW 170
Process image of the disturbance input (ADC 4)														(Analog I/O format)		DW 171
Process image of the manual input (ADC 5)														(Analog I/O format)		DW 172
Value list for process input image																DW 173
Value list for process input image																DW 174
Value list for process input image																DW 175
Value list for process input image																DW 176
Value list for process input image																DW 177
Value list for process input image																DW 178
Value list for process input image																DW 179
Process image of the relays															(Binary)	DW 180
Value list for process input image																DW 181
Value list for process input image																DW 182
Value list for process input image																DW 183
Value list for process input image																DW 184
Value list for process input image																DW 185
Value list for process input image																DW 186
Value list for process input image																DW 187
Value list for process input image																DW 188

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Value list for process input image																DW 189
Value list for process input image																DW 190
Control setpoint														(w. dimension)		DW 191
Controller														(ASCII)		DW 192
name														(ASCII)		DW 193
for														(ASCII)		DW 194
CP526 interface module (4 words)														(ASCII)		DW 195
Interface data area														(ASCII)		DW 196
name														(ASCII)		DW 197
for														(ASCII)		DW 198
CP526 interface module (4 words)														(ASCII)		DW 199
(unassigned)																DW 200
(unassigned)																DW 201
(unassigned)																DW 202
(unassigned)																DW 203
(unassigned)																DW 204
(unassigned)																DW 205
(unassigned)																DW 206
(unassigned)																DW 207
Process image of control output DAC 1														(Analog I/O format)		DW 208
Process image of control output DAC 2														(Analog I/O format)		DW 209
Value list for process output image																DW 210
Value list for process output image																DW 211
Value list for process output image																DW 212
Value list for process output image																DW 213
Value list for process output image																DW 214
Value list for process output image																DW 215

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Value list for process output image																DW 216
Value list for process output image																DW 217
Process image of test socket 1 (DAC 3)													(Analog I/O format)			DW 218
Process image of test socket 2 (DAC 4)													(Analog I/O format)			DW 219
Process image of the output bits															(Binary)	DW 220
Value list for process output image																DW 221
Value list for process output image																DW 222
Value list for process output image																DW 223
Value list for process output image																DW 224
Value list for process output image																DW 225
Value list for process output image																DW 226
Value list for process output image																DW 227
Value list for process output image																DW 228
Value list for process output image																DW 229
Value list for process output image																DW 230
Value list for process output image																DW 231
Value of measuring point MP 1															(w. dimension)	DW 232
Value of measuring point MP 2															(w. dimension)	DW 233
Value of measuring point MP 3															(w. dimension)	DW 234
Value of measuring point MP 4															(%)	DW 235
Value of measuring point MP 5															(%)	DW 236
Value of measuring point MP 6															(%)	DW 237
Value of measuring point MP 7															(%)	DW 238
Value of measuring point MP 8															(w. dimension)	DW 239
Value of measuring point MP 9															(w. dimension)	DW 240
Value of measuring point MP 10															(%)	DW 241
Value of measuring point MP 11															(w. dimension)	DW 242

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Value of measuring point MP 12															(%)	DW 243
(free)																DW 244
(free)																DW 245
(free)																DW 246
(free)																DW 247
(free)																DW 248
(free)																DW 249
(free)																DW 250
(free)																DW 251
(free)																DW 252
(free)																DW 253
Reserved for operator-process comm. and process visualization																DW 254
Status word								YOG	YUG	SOG	SUG	OW	UW	OG	UG	DW 255

7.1 Description of the Status Word (DW255)

YOG : = 1: Flag: Upper limiting value of regulating variable exceeded (= D 220.6) (PID controller with continuous output signal)
: = 1: Flag: Final OPEN position reached (= D 180.4) (point controller)

YUG : = 1: Flag: Lower limiting value of regulating variable exceeded (= D 220.7) (PID controller with continuous output signal)
: = 1: Flag: Final CLOSED position reached (= D 180.3) (point controller)

SOG : = 1: Flag: Setpoint has exceeded upper limiting value

SUG : = 1: Flag: Setpoint has violated lower limiting value

OW : = 1: Flag: Actual value has exceeded upper warning limit

UW : = 1: Flag: Actual value has violated upper warning limit

OG : = 1: Flag: Actual value has exceeded upper danger limit

UG : = 1: Flag: Actual value has violated lower danger limit

7.2 Description of the Operating System Bits (DW2 to DW7)

RIN (D 2.0) Controller initialization bit:

- Is set by operating system to identify startup mode "Cold restart" (also RINa = 0, WAB = 0), and is reset by the controller program at start of second program cycle

- Is set by the controller when restart condition is not fulfilled (RINa is also set). This informs the operating system that it is to update only the process input image (and NOT the process output image) before the controller program goes into its next cycle because the historical values are not correct. After evaluating them, the controller program resets both bits.

RINa (D 2.15) Previous controller initialization bit:
Is used only by the operating system and the controller program; also see RIN

RAB (D 2.1) Controller request bit:
Is used by the operating system only

RIB (D 2.2) Controller program in progress:
Is used by the operating system only

WAB (D 2.3) Restart bit:
Is set by the operating system in the event of a manual or an automatic restart, and is reset by the controller program after it has been evaluated.

PAA (D 2.6) Process image update mode:
PAA = 0 (presetting) = = > Common updating of controller inputs and outputs prior to execution of the controller program.
PAA = 1 = = > Time-controlled updating of the inputs prior to execution of the controller program and updating of the output bits following its execution.

PLB (D 2.7) I/O reset bit:
Is set by the user to obtain a defined state of the outputs on a transition to STOP, or after selective disabling of controllers:
PLB = 1 = = > Reset outputs
PLB = 0 = = > Do not reset outputs

RFB (D 2.8) Controller error bit:
The operating system always sets this bit when the error word contains a value other than zero.

AZF (D 3.0) Scan time error:
A scan time error occurs when the controller program cannot complete its execution cycle in the time available between two process image updating cycles and the MAZF mask (D 4.0) is not set, or the user did not load OB 34. The bit is reset on restart.

MAZF (D 4.0) "Scan time error" mask:
A transition to the STOP state caused by a sampling error can be prevented by setting MAZF to 1.

RVB (D 7.0) Controller management bit:
The RVB bit enables the user to disable and then reenable a controller specified in DB 2:
RVB = 1 = = > Controller enabled
RVB = 0 = = > Controller disabled

RVBa (D 7.1) Previous controller management bit:

Enables the operating system to ascertain when a temporarily disabled controller is to be reenabled

RNEU (D 7.2) Controller cold restart bit:
By setting RNEU to 1, the operating system informs the controller program that the controller is to go into operation again following a temporary shutdown.

BED1 (D 7.8) Operator control bit for parameter set 1 (DW25-DW55)

BED2 (D 7.9) Operator control bit for parameter set 2 (DW56-DW103)

7.3 Description of the Structuring Switches (DW23 to DW24)

Structuring switches are binary variables which are defined during structuring in the STOP mode and may not be modified in RUN mode (exception: S8).

The following structuring switches are not shown in the controller's structure diagram:

S2 (D 23.1) : PID controller bit:
- 0 = Standard version (R = 1, D 180.12 = 0, D 180.13 = 0, D 180.14 = 0, D 23.10 = 0, upper limiting value for the correction increase = 100%, lower limiting value for the correction increase = - 100%)
- 1 = Enhanced version

S6 (D 23.5): Filter with time constant 8 * T_A in the actual value branch
- 0 = Ineffective
- 1 = Effective when S22 (D 24.5) = 0

S11 (D23.10): PID controller bit:
- 0 = Separate D input not active
- 1 = Separate D input active

S15 (D 23.14): Mark space output bit:
- 0 = 2-step output
- 1 = 3-step output

S21 (D 24.4): Setpoint train bit:
- 0 = Step form
- 1 = Linear interpolation (ramp form)

S23 (D 24.6): Format conversion for 4 - 20 mA analog input (→ 2.3.1)
- 0 = No format conversion
- 1 = Subtract 1000H from input value

7.4 Description of the Relays in the Process Input Image (DW180)

D 180.0: Ramp generator's disable bit:
- 0 = Enable manual mode with higher/lower key
- 1 = Disable manual mode, i.e. the output tracks the input value with the ramp function

D 180.1: Ramp generator's lower bit:
- 0 = Ineffective
- 1 = Ramp with negative slope

D 180.2: Ramp generator's higher bit:
- 0 = Ineffective
- 1 = Ramp with positive slope

D 180.3: Point controller's check-back bit:
- 0 = Final CLOSED position not reached
- 1 = Final CLOSED position reached

D 180.4: Point controller's check-back bit:
- 0 = Final OPEN position not reached
- 1 = Final OPEN position reached

D 180.5: Setpoint select bit:
- 0 = Control setpoint active
- 1 = "Setpoint train" output active

D 180.6: Setpoint branch disable bit:
- 0 = Setpoint branch enabled
- 1 = Setpoint branch disabled

D 180.7: Actual value branch enable bit:
- 0 = Actual value branch enabled
- 1 = Actual value branch disabled, initial actual value effective

D 180.8: "Manual input value" enabled (point controller):
- 0 = Manual input value not effective, i.e., no pulse output (actuator at standstill)
- 1 = Manual input value effective, i.e., frequency of pulse output according to manual input value

D 180.9: Controller disable bit (point controller):
- 0 = Controller enabled, i.e., computed value is output
- 1 = Controller disabled, i.e., zero is output

D 180.10: Input variable select switch for 1 component (point controller):
- 0 = Dead band for 1 component effective
- 1 = Dead band for 1 component not effective

D 180.11: Auto/manual mode (PID or point controller):
- 0 = Auto mode
- 1 = Manual mode

D 180.12:
- PID controller (D 23.0 = 0):
  Ideal/real *PID controller:
  0 = Real
  1 = Ideal PID

- Point controller (D 23.0 = 1):
  Manual mode when D 180.11 = 1 and D 180.0 = 0
  0 = No negative pulse
  1 = Negative pulse

D 180.13: · PID controller (D 23.0 = 0):

Bit for stabilization of the regulating variable (PID controller):
0 = No effect
1 = Regulating variable constant

- Point controller (D 23.0 = 1):

Manual mode when D 180.11 = 1 and D 180.8 = 0
0 = No positive pulse
1 = Positive pulse

D 180.14: Feedforward injection of interference variable (PID controller):
0 = Changes in interference variable are disregarded
1 = Effective, i.e. interference variable is added to the computed regulating variable

D 180.15: Controller disable bit (PID controller):
0 = Controller enabled, i.e. computed value is output
1 = Controller disabled, i.e. zero is output
(historical values are set to zero and no value is computed for the regulating variable)

7.5 Bits in the Process Output Image (DW220)

D 220.0 - D 220.5 : Limit bits for limit monitor 1
0 = Associated limiting value not exceeded
1 = Associated limiting value exceeded

D 220.6: Bit for regulating variable's upper limiting value (PID controller):
0 = No limit violation
1 = Limit violation

D 220.7: Bit for regulating variable's lower limiting value (PID controller):
0 = No limit violation
1 = Limit violation

D 220.8 - D 220.13: Limit bits for limit monitor 2:
0 = Associated limiting value not exceeded
1 = Associated limiting value exceeded

D 220.14: Point controller's binary "close" output
or
Mark space output's binary "negative pulse" output

D 220.15: Point controller's binary "open" output
or
Mark space output's binary "positive pulse" output

8 Abbreviations

The following abbreviations are used in this manual:

ADC	=	Analog-digital converter
AI	=	Analog input module
AO	=	Analog output module
COM REG	=	Programmer software to initialize, start-up and test the R64 controller structure
CP	=	Communications processor
D	=	Identifies a bit in the data block, e.g., D 2.0 means bit zero in data word 2
DAC	=	Digital-analog converter
DB	=	Data block
DI	=	Digital input module
DL	=	Left-hand byte
DO	=	Digital output module
DR	=	Right-hand byte
DW	=	Data word
FB	=	Function block
FD	=	Floppy disk (diskette)
FP	=	Fixed-point (notation)
KF	=	Fixed-point constant
KH	=	Hexadecimal constant
MP	=	Measuring point
PCP/M	=	Personal Control Program for Microprocessors (a registered trademark of Digital Research)
PG	=	Programmer
PI	=	Process image
PLC	=	Programmable controller
S5	=	Designator for a programmable controller, e.g., S5-135U
S5-DOS	=	The S5-DOS basic package contains programs for the input of data and program blocks for the programmable controller. It comprises several programs, a number of which are required for running COM REG. (for details, → COM REG manual).

9 Alphabetical Index

10 Structuring Forms

R64 Controller Structure	Structuring Form 1
Controller Inputs/Outputs	
DB - No.	

Function	Converter	Address in Controller DB	Measuring Point	Value/ Address
Setpoint	ADC 1	DW 168	1	
Control setpoint	---------	DW 191	1	
Actual value	ADC 2	DW 169	8	
Sep. D-input	ADC 3	DW 170	11	
Interference input	ADC 4	DW 171	12	
Manual input	ADC 5	DW 172	10	
Analog output	DAC 1	DW 208	5	
Analog output	DAC 2	DW 209	7	
Digital output	DA 1	D 220.15		
Digital output	DA 2	D 220.14		
Test socket 1	DAC 3	218		
Test socket 2	DAC 4	219		

R64 Controller Structure	Structuring Form 2
Digital Inputs	
DB - No.	

Description	Bit in Controller DB	Value/Address
Ramp-function generator: Delete bit	180.0	
Ramp-function generator: Lower bit	180.1	
Ramp-function generator: Higher bit	180.2	
Final CLOSED position	180.3	
Final OPEN position	180.4	
Setpoint selection	180.5	
Setpoint branch disable	180.6	
Initial actual value	180.7	
Manual input value in force	180.8	
Point controller disable	180.9	
Dead band disable	180.10	
Auto/manual	180.11	
PID: Ideal controller Point controller: Manual val.neg.	180.12	
PID: Man. var. constant Point controller: Manual val.pos.	180.13	
Feedforward inj. of dist. var.	180.14	
PID controller disable	180.15	

R64 Controller Structure	Structuring Form 3
Digital Outputs	
DB - No.	

Description	Bit in Conroller DB	Value/Address
Limit monitor 1	220.0	
	220.1	
	220.2	
	220.3	
	220.4	
	220.5	
Upper limit violation	220.6	
Lower limit violation	220.7	
Limit monitor 2	220.8	
	220.9	
	220.10	
	220.11	
	220.12	
	220.13	
Negative pulse / close	220.14	
Positive pulse / open		220.15

R64 Controller Structure	Structuring Form 4
Controller Parameters	Continuous PID Controller
DB - No.	

Description	Data Word in Controller DB	Value/Address Standard	Enhanced
Proportional value KP	73		
Auxiliary gain R	74	----------	
Integral-action time TN	75		
Derivative-action time TV	77		
Upper controller limit	79		
Lower controller limit	80		
Pos. corr. increase limit	81	----------	
Neg. corr. increase limit	82	----------	
Gain (actuator adjustment)	96		
Offset (actuator adjustment)	97		

R64 Controller Structure	Structuring Form 5
Controller Parameters	PID-Controller with Mark Space Output
DB - No.	

Description	Data Word in Controller DB	Value / Address	
		Standard	Enhanced
Proportional value KP	73		
Auxiliary gain R	74	----------	
Integral-action time TN	75		
Derivative-action time TV	77		
Upper controller limit	79		
Lower controller limit	80		
Pos. corr. increase limit	81	----------	
Neg. corr. increase limit	82	----------	
Response threshold	92		
Adjustment factor	93		
Min. pulse duration TMIN	14		

R64 Controller Structure	Structuring Form 6
Controller Parameters	**Point Controller**
DB - No.	

Description	Data Word in Controller DB	Value/Address
Proportional value KP	83	
Integral-action time TN	84	
Derivative-action time TV	86	
Min. pulse duration TMIN	14	
Actuating time TM	88	
Upper resp. threshold	90	
Lower resp. threshold	91	

R64 Controller Structure	Structuring Form 7
Controller List	
DB - No. 2	

Time Base							
Scan time/Minimum Pulse Duration :							
DB	DB	DB	DB	DB	DB	DB	DB

Reglerstruktur R 64