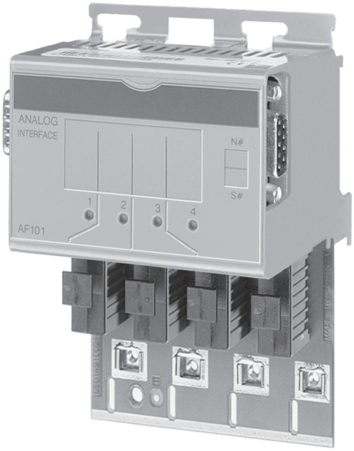
6.4 AF101

6.4.1 Technical Data



Module ID	**AF101**
General Information	
Model Number	7AF101.7
Short Description	2003 Adapter Module 4 slots for screw-in modules
C-UL-US Listed	Yes
B&R ID Code	$C0
Amount	
CP430, EX270	2
CP470, CP770 CP474, CP476, CP774 EX470, EX770 EX477, EX777	4
Adapter Interface	4 slots for screw-in modules
Electrical Isolation	No
Status LEDs	For each channel
Power Consumption	Max. 0.3 W
Mechanical Properties	
Dimensions	B&R 2003 single width

Always operate the adapter module directly to the right of the CPU!

Adapter modules are only allowed to be used in the first four module slots on the B&R 2003 system.

6.4.2 Status LEDs

The green status LEDs on the adapter modules indicate the operational status of each slot:

LED Status	Meaning
Not lit	Screw-in module defective or not inserted
Blinking slowly	Communication error with screw-in module
Blinking quickly	Screw-in module is new or has been exchanged with another module type
Lit	Screw-in module is ready for operation

6.4.3 Legend Sheets

The legend sheets supplied with the module are slid into the module front from the top of the module. These sheets can be used for labelling the screw in modules.

6.4.4 Accessing the Screw-in Modules

Every module description contains a "variable declaration" section. This section outlines the method for declaring variables in PG2000.
The channel number to be accessed is given in the VD Channel column with the data or configuration words. When accessing configuration words using I/O function blocks, an offset value must be added to the VD channel, according to which slot the screw-in module occupies on the AF101.

Data Word

When data is accessed using a data word, an offset does not need be added to the VD channel number, since the slot number is given in the variable declaration.

Configuration Word

When accessing configuration words using I/O function blocks, an offset value must be added to the VD channel, according to which slot the screw-in module occupies on the AF101.
The internal offset used to address the required configuration word, is calculated using the following formula:

Internal offset = VD channel + (sl - 1) x 32

Internal offset .... Internal offset used to access the required configuration word

VD channel .... The channel entered in the variable declaration

sl .... Slot number (1 - 4) of the screw-in module on the AF101

Refer to the example in Chapter 4, "Module Addressing".

6.4.5 Commands - AF101

Controller

The commands are available to the following controllers/access types:

Controller / Access Type	Command available
CPU - PCC 2003	Yes
Remote Slaves	No
CAN Slaves	No
Access using CAN Identifier	No

Command Execution

I/O function blocks are used to execute commands on the AF101. A shovel instruction must be generated for reading configuration word 14 (VD channel 28) of a screw-in module.
The variable used to store the command response is initialized with the command word. After successful execution of the shovel instruction, the command response is saved in the variable.

The AF101 commands *Automatic mode on/off* are available from revision 02.00 of the AF101 software. The software version of the AF101 module can be determined from the last two digits of the version number shown on the serial number label (Rev. xx.11 -> SW revision 01.10, Rev. xx.20 -> SW revision 02.00).

1) Read B&R ID code for the screw-in module

The variable must be initialized with 0 in order read the B&R ID code. Don't forget to enter the channel offset for the respective slot!

Command Word	$0000	
Command Response	$xxyy	xx .... B&R ID Code yy .... Reserved

2) Read B&R ID code for the AF101 module

Command Word	$8000	
Command Response	$C000	B&R ID Code AF101

3) Automatic mode

Starting with software revision 02.00 of the AF101, the data word will be written by default to an automatically generated I/O image (automatic mode). Read commands are executed by accessing this I/O image.
Special applications of screw-in modules sometimes require the automatic I/O image generation to be switched off (e.g. for the AI261 this makes certain that the scaled value or calibrated raw value is only read once per data output cycle).
When the I/O image generator is switched off, the commands for editing the data words are written directly to the screw-in modules instead.
Automatic mode is not available on AF101 modules with a software revision <02.00. These modules behave in the same way as modules with a software revision ≥02.00 where automatic mode has been switched off.

3.1) Switching off automatic mode

This command is only available to AF101 modules with a SW revision ≥ 02.00 (AF101 modules with SW revision < 02.00 behave in the same way as modules with a SW revision ≥02.00 where automatic mode has been switched off).

Command Word	$8400	
Command Response	$F400	Command executed - AF101 SW Rev. ≥02.00
	$C000	AF101 SW Rev. <02.00

3.2) Switching on automatic mode

This command is only available to AF101 modules with a SW revision ≥02.00. Automatic mode is not available on AF101 modules with a software revision <02.00.

Command Word	$8500	
Command Response	$F500	Command executed - AF101 SW Rev. ≥02.00
	$C000	AF101 SW Rev. <02.00

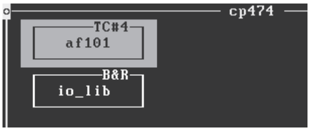
6.4.6 Program Example

In this program, the shovel instructions for the following commands are defined and executed in the INIT-SP :

- Read B&R ID code for the screw-in module
- Read B&R ID code for the AF101 module
- Switching off automatic mode

Task Overview

The application program is divided into the following tasks:



```
cp474
TC#4
af101
B&R
io_lib
```

I/O Library

Before using the function blocks to handle configuration words, you must import the I/O library (starting with V 01.60) into the project database. The function, *Import Library* in the *File* menu is provided for this purpose. The I/O library must be entered in the GDM as a system module.

AF101

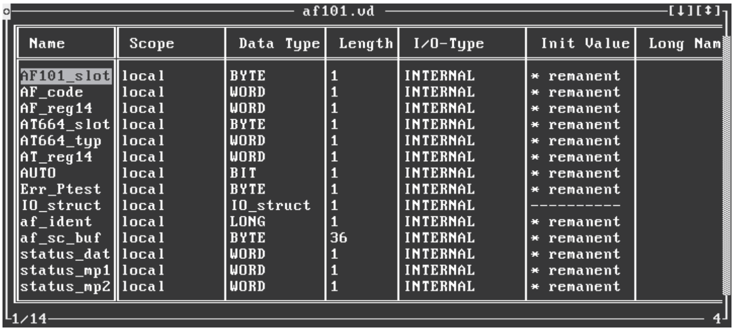
This task processes the screw-in modules on the AF101 adapter module.
The shovel instructions required for carrying out commands on the AF101 module are defined in the INIT-SP.

INIT-SP

```
Project: af_auto                                         File: InitSP : af101

Err_Ptest = 0
AF101_slot = 1
AT664_slot = 1

;#############################################################################
;## Multipler for physical shovel instructions to read                           ##
;## parameter word 14 for the AT664 on the AF101, Slot 1                 ##
;#############################################################################
IO_struct.io_type=5                      ;2003 IO
IO_struct.master_no=1                    ;Master always 1
IO_struct.slave_no=0                     ;Slave number always 0
IO_struct.module_adr=AF101_slot          ;Module slot
IO_struct.intern_off= 32*(AT664_slot -1)+ 14 *2 ;Register no. 14 on ScrM = AT664_slot
IO_struct.mode=%00100000                 ;Read, normal
IO_struct.data_len=2                     ;2*Byte
IO_struct.reserve=0                      ;Not used
IO_struct.data_adr=adr(AT_reg14)         ;Address of the data area containing the
                                         ;image for parameter word 14
AT_reg14 = $0000                         ;0 required to read the module type


;Declare shovel instruction #0
IO_mphydef(1,adr(IO_struct),adr(af_sc_buf),0,status_mp1,af_ident)

if status_mp1=0 then
  loop            ;Carry out shovel instruction cyclically...
    IO_data(1,af_ident,status_dat)
    exitif status_dat <> 5559    ;...until shovelling is no longer active
  endloop
endif

if status_dat=0 then
    AT664_typ = AT_reg14
endif


;#############################################################################
;## Multipler for physical shovel instructions to read                           ##
;## parameter word 14 (=code) on the AF101 module                             ##
;#############################################################################
IO_struct.io_type=5                      ;2003 IO
IO_struct.master_no=1                    ;Master always 1
IO_struct.slave_no=0                     ;Slave number always 0
IO_struct.module_adr=AF101_slot          ;Module slot
IO_struct.intern_off= 14 *2              ;Register no. 14 on the AF101
IO_struct.mode=%00100000                 ;Read, normal
IO_struct.data_len=2                     ;2*Byte
IO_struct.reserve=0                      ;Not used
IO_struct.data_adr=adr(AF_reg14)         ;Address of the data area containing the
                                         ;image for parameter word 14
AF_reg14 = $8000                         ;Read AF101 code

IO_mphydef(1,adr(IO_struct),adr(af_sc_buf),0,status_mp2,af_ident)

if (status_mp2=0) then
  loop            ;Carry out shovel instruction cyclically...
    IO_data(1,af_ident,status_dat)
    exitif status_dat <> 5559    ;...until shovelling is no longer active
  endloop
endif
```

```
if status_dat=0 then
    AF_code = AF_reg14
endif

;#############################################################################
;##                    AF101 Switch off automatic mode                        ##
;#############################################################################

;Shovel instruction can be used further
AF_reg14 = $8400                        ;Command switch off automatic mode
AUTO = 1    ;Automatic mode still active

if (status_mp2=0) then
  loop            ;Carry out shovel instruction cyclically...
    IO_data(1,af_ident,status_dat)
    exitif status_dat <> 5559    ;...until shovelling is no longer active
  endloop

  if (status_dat=0) and (AF_reg14=$F400) then
    ;Automatic mode switched off successfully
    AUTO = 0      ;Automatic mode inactive
  else if (status_dat=0) and (AF_reg14=$C000) then
    ;AF101 SW Rev. < 2.00  -> No automatic mode
    AUTO = 0      ;Automatic mode inactive
  endif
endif
```

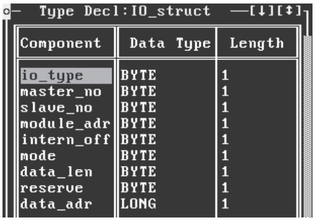
Variable Declaration



Name	Scope	Data Type	Length	I/O-Type	Init Value	Long Nam
AF101_slot	local	BYTE	1	INTERNAL	* remanent	
AF_code	local	WORD	1	INTERNAL	* remanent	
AF_reg14	local	WORD	1	INTERNAL	* remanent	
AT664_slot	local	BYTE	1	INTERNAL	* remanent	
AT664_typ	local	WORD	1	INTERNAL	* remanent	
AT_reg14	local	WORD	1	INTERNAL	* remanent	
AUTO	local	BIT	1	INTERNAL	* remanent	
Err_Ptest	local	BYTE	1	INTERNAL	* remanent	
IO_struct	local	IO_struct	1	INTERNAL	----------	
af_ident	local	LONG	1	INTERNAL	* remanent	
af_sc_buf	local	BYTE	36	INTERNAL	* remanent	
status_dat	local	WORD	1	INTERNAL	* remanent	
status_mp1	local	WORD	1	INTERNAL	* remanent	
status_mp2	local	WORD	1	INTERNAL	* remanent	

Data Type

A structure will be defined for the variable „IO_struct“ using the data type *typedef* . Once the structure name *IO_struct* has been entered, the structure elements can be entered in a dialog box.



Component	Data Type	Length
io_type	BYTE	1
master_no	BYTE	1
slave_no	BYTE	1
module_adr	BYTE	1
intern_off	BYTE	1
mode	BYTE	1
data_len	BYTE	1
reserve	BYTE	1
data_adr	LONG	1