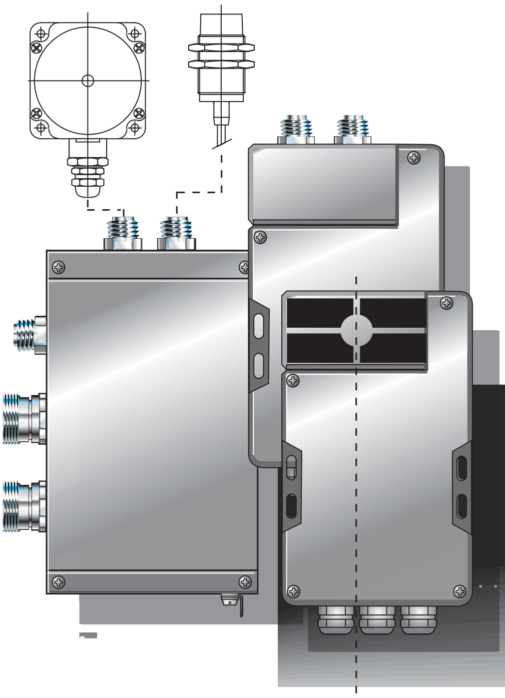


BALLUFF

sensors worldwide

Manual

Electronic Identification Systems BIS
Processor BIS C-6_0
RS 232

Deutsch – bitte wenden!

No. 836 905 D/E • Edition 0704
Subject to modification.
Replaces edition 0409.

Writing convention:
Control characters to be transmitted are in angle brackets.
Characters to be transmitted in ASCII code are enclosed in apostrophes.
Example: <STX> '1 2 3 4 5 6' BCC

This manual is valit from hardware version HV 1.40 or higher

Balluff GmbH
Schurwaldstrasse 9
73765 Neuhausen a.d.F.
Germany
Phone +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de

www.balluff.com

Content

Safety advisory

Approved operation

Series BIS C-6_0 processors along with the other BIS C system components comprise an identification system and may only be used for this purpose in an industrial environment in conformity with Class A of the EMC Law.

Installation and operation

Installation and operation should be carried out only by trained personnel. Unauthorized work and improper use will void the warranty and liability.

When installing the processor, follow the chapters containing the wiring diagrams closely. Special care is required when connecting the processor to external controllers, in particualy with respect to selection and polarity of the signals and power supply,

Only approved power supplies may be used for powering the processor. See Technical Data for details.

Use and testing

Prevailing safety regulations must be adhered to when using the identification system. In particular, steps must be taken to ensure that a failure of or defect in the identification system does not result in hazards to persons or equipment

This includes maintaining the specified ambient conditions and regular testing for functionality of the identification system including all its associated components.

Function faults

Should there ever be indications that the identification system is not working properly, it should be taken out of commission and secured from unauthorized use

Scope

This manual applies to processors in the series BIS C-600-007-...-00-KL1 and BIS C-620-007-050-00-ST2.

Introduction
BIS C Identification System

This manual is designed to assist the user in setting up the control program and installing and starting up the components of the BIS C-6_0 Identification System, and to assure rapid, trouble-free operation.

Principles

The BIS C-6_0 Identification System belongs in the category of

non-contact systems for reading and writing.

This dual function permits applications for not only transporting information in fixed-programmed data carriers, but also for gathering and passing along up-to-date information as well.

Applications

Some of the notable areas of application include

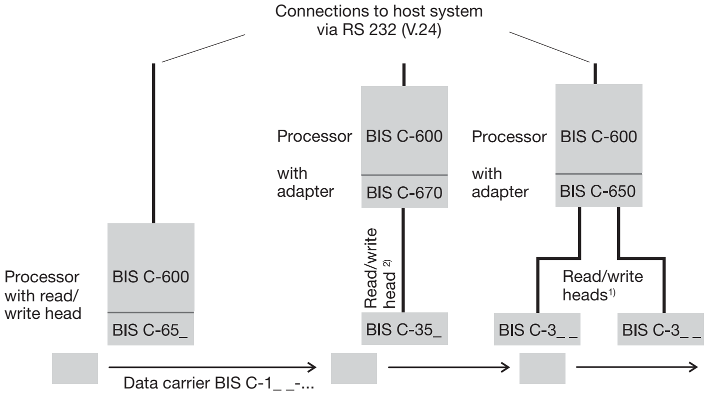
- **for controlling material flow in production processes**
  (e.g. in model-specific processes)**,**
  for workpiece conveying in transfer lines,
  in data gathering for quality assurance,
  for gathering satety-related data,
- **in tool coding and monitoring;**
- **in equipment organization;**
- **in storage systems for monitoring inventory movement;**
- **in transporting and conveying systems;**
- **in waste management for quantity-based fee assessment.**

Introduction
BIS C-600 Identification System

System components

The main components of the BIS C-600 Identification System are

- **Processor,**
- **Read/write heads, and**
- **Data carriers.**



Schematic representation of an identification system (example)

[1]) except BIS C-350 and -352

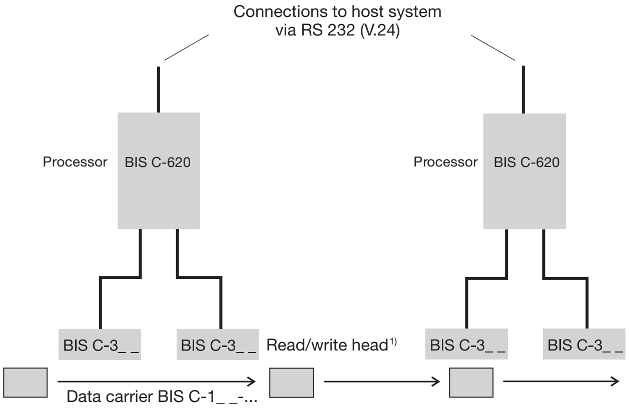
[2]) only BIS C-350 or -352

Introduction
BIS C-620 Identification System

System components

The main components of the BIS C-620 Identification System are

- **Processor,**
- **Read/write heads, and**
- **Data carriers**.



Schematic representation of an identification system (example)

[1]) except BIS C-350 and -352

Application
BIS C-6_0 Processor

Selecting system components

The **BIS C-600** processor has a plastic housing. Connections are made through a terminal strip, with the cable secured by a PG fitting. A single read/write head from series BIS C-65_ can be directly connected to the processor, which creates a compact unit. If the BIS C-650 adapter is attached instead of the BIS C-65_ read/write head, two read/write heads may be cable connected. If the BIS C-670 adapter is attached, one read/write head may be cable connected.

The **BIS C-620** processor has a metal housing. Connection is made through round connectors. Two read/write heads can be cable connected to the BIS C-620 processor.

Additional information on the read/write heads in series BIS C-65_ including all the possible data carrier combinations can be found in the manuals for the respective read/write heads.

Whether the compact version of the processor with integrated read/write head makes sense or whether the external solution is preferred depends primarily on the spatial arrangement of the components. There are no functional limitations. All read/write heads are suitable for both static and dynamic reading. Distance and relative velocity are based on which data carrier is selected. Additional information on the read/write heads in series BIS C-65_ and series BIS C-3_ _ including all the possible data carrier/read-write head combinations can be found in the manuals for the respective read/write heads.

The system components are electrically supplied by the processor. The data carrier represents an free-standing unit and needs no line-carried power. It receives its energy from the read/write head. The latter constantly sends out a carrier signal which supplies the code head as soon as the required distance between the two is reached. The read/write operation takes place during this phase. Reading and writing may be dynamic or static.

Application
BIS C-6_0 Processor

Dialog mode

The processor writes data from the host system to the data carrier and reads it from the data carrier through the read/write head, and prepares it for the host system. Host systems may be:

- **a controlling computer (e.g., industrial PC) or**
- **a programmable logic controller (PLC)**

Direct read mode

Depending on the configuration, the processor sends the read data carrier data to the host unit after recognizing a data carrier.

Protocol sequence with schematic representations

The processor controls and adminsters the data communication between data carriers and read/write heads. The serial port connects the BIS C-6_0 identification system to the external controller.

Data is exchanged between the processor and the host system using specific telegrams.

The protocol sequence is represented in the form of function blocks on pages 32/33. See the chapter on Programming Information for more details on the telegram contents.

Application
BIS C-6_0 Processor

Data checking

When sending data between the read/write head and the data carrier a procedure is required for recognizing whether the data were correctly read or written.

The processor is supplied with standard Balluff procedure of double reading and comparing. In addition to this procedure a second alternative is available: CRC_16 data checking.

Here a test code is written to the data carrier, allowing data to be checked for validity at any time or location.

Advantages of CRC_16	Advantages of double reading
Data checking even during the non-active phase (CT outside read/write head zone).	No bytes on the data carrier need to be reserved for storing a check code.
Shorter read times since each page is read only once.	Shorter write times since no CRC needs to be written.

Since both variations have their advantages depending on the application, the user is free to select which method of data checking he wishes to use (see Configuration on pages 13/16).

To use the CRC check method, the data carriers must be initialized. You use either data carriers with the data map factory configured (all data are 0), or you must use the processor to write the special initialization command 'Z' to the data carriers (see page 52).

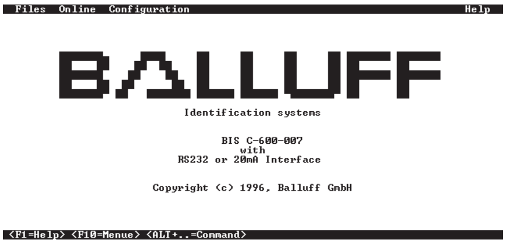
It is not permitted to operate the system using both check procedures!

BIS C-6_0
Configuration

Configuration

Before programming, the processor configuration must be carried out, in case the factory settings will not be used.

Configuration is done using a computer and the Balluff BISC600A.EXE software, and it is stored in the processor. It may be overwritten at any time. The configuration can be stored in a file, making it accessible when required.



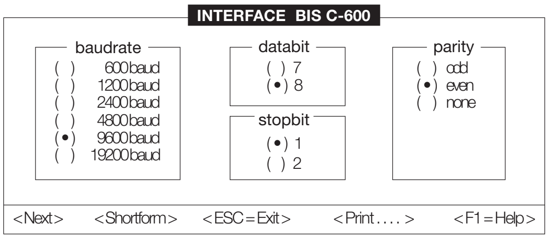
```
 Files  Online  Configuration                                    Help

                    BALLUFF
               Identification systems

                   BIS C-600-007
                       with
              RS232 or 20mA Interface

            Copyright (c) 1996, Balluff GmbH

<F1=Help> <F10=Menue> <ALT+..=Command>
```

Important. **Please note.**
Please note the selected settings on the stick-on label supplied (to be pasted on the inside of the processor cover) as well as on page 28 and 30 in the customer configuration section, so that in case of repair of the processor the settings can be saved or otherwise can also be used to set other processor units.

BIS C-6_0
Configuration, Interface

Interface Menu BIS C-6_0

Interface BIS C-600
The first screen shows the parameters baud rate, number of data and stop bits, and parity type for the serial interface selected. The graphic shows the factory settings. The other settings are carried out in the corresponding masks which are illustrated in the following pages.



INTERFACE BIS C-600

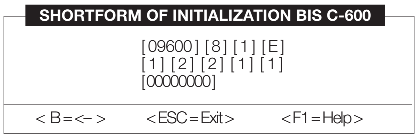
baudrate
( ) 600baud
( ) 1200baud
( ) 2400baud
( ) 4800baud
(●) 9600baud
( ) 19200baud

databit
( ) 7
(●) 8

stopbit
(●) 1
( ) 2

parity
( ) odd
(●) even
( ) none

If the initializing data are available in short form (e.g. on the processor cover after a replacement of the unit) the data can be entered directly into the "Shortform of initialization BIS C-600" mask (see also Customer cofiguration on page 28/29).

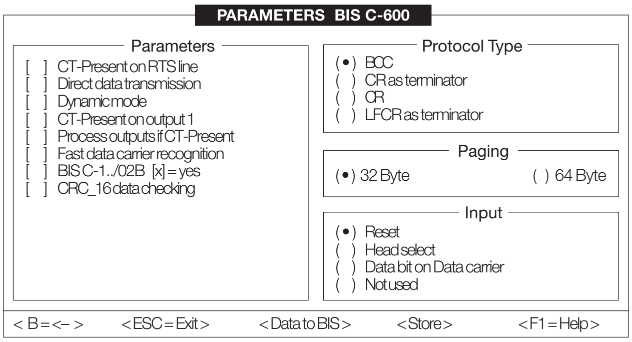


SHORTFORM OF INITIALIZATION BIS C-600

[09600] [8] [1] [E]
[1] [2] [2] [1] [1]
[00000000]

BIS C-6_0 Configuration

Menu settings BIS C-6_0



```
PARAMETERS  BIS C-600

Parameters                              Protocol Type
[ ]  CT-Present on RTS line             (•)  BCC
[ ]  Direct data transmission           ( )  CR as terminator
[ ]  Dynamic mode                       ( )  CR
[ ]  CT-Present on output 1             ( )  LFCR as terminator
[ ]  Process outputs if CT-Present
[ ]  Fast data carrier recognition      Paging
[ ]  BIS C-1../02B  [x] = yes           (•) 32 Byte        ( ) 64 Byte
[ ]  CRC_16 data checking
                                        Input
                                        (•)  Reset
                                        ( )  Head select
                                        ( )  Data bit on Data carrier
                                        ( )  Not used

< B = <– >   < ESC = Exit >   < Data to BIS >   < Store >   < F1 = Help >
```

Menu settings BIS C-6_0, Protocol Type Field

Protocol Type Field:

For host devices which require a terminator, the additional use of Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' is made available. The following page contains examples of the various possibilities.

BIS C-6_0 Configuration

Menu settings BIS C-6_0, Protocol Type Field (continued)

Examples for terminating telegrams:

Protocol Variants	Telegram with command, Address and no. of bytes	Terminator	Acknowledge	Terminator
with Blockcheck BCC	'R 0000 0001'	BCC	<ACK> '0'	
with Carriage Return	'R 0000 0001'	'CR'	<ACK> '0'	
with Terminator Carriage Return	'R 0000 0001'	'CR'	<ACK> '0'	'CR'
with Terminator Carriage Return and Line Feed	'R 0000 0001'	'LF CR'	<ACK> '0'	'LF CR'

Menu settings BIS C-6_0, Parameter Field

Parameters Field:

– **CT-Present on RTS**
The parameter CT-Present on RTS (corresponds to the LED indicator Data carrier Present on the BIS C-6_0 housing) can be used to allow the PC to check CT-Present using a hardware signal. You can connect the RTS signal, for example, to the free input RI (Ring Indicator) of the PC as this is only used with Modem operation.

– **Direct data transmission**
Each time a data carrier is recognized, the data will be read out depending on the configuration and output to the interface. With this setting the read command in the dialog mode is superfluous.

BIS C-6_0 Configuration

Menu settings BIS C-6_0, Parameter Field (continued)

– **Dynamic Mode**
This function switches off the error-message "No data carrier present", i.e.:
-> In dynamic mode, a read or write telegram is stored until a data carrier enters the working
range of the corresponding read/write head.
-> Without dynamic mode, a read or write telegram is acknowledged with an error message <NAK> '1' if there is no data carrier present in front of a read/write head; the processor goes into the ground state.

– **CT-Present on Output 1**
If CT Present is selected for Output 1, the LED message Data carrier Present is also output on Output 1. In this way the presence of a data carrier can be directly verified as a digital hardware signal.

– **Process Outputs if CT Present**
The output functions are normally processed only after a read command. But since data carrier recognition is also an automatic tag read (reads first page, either 32 or 64 bytes depending on type), the output processing can occur simultaneously with Data carrier Present. If the addresses for output processing are located on the first page, then the processor can itself carry out short control commands without a seperate command.
-> For very fast transactions, see next section.

– **Fast Data carrier Recognition**
For very fast transactions, the number of data carrier addresses used for data carrier recognition can be reduced from 32 or 64 bytes to 4 bytes. The data carrier recognition time is thereby reduced to ca. 50 ms (instead of ca. 150 ms for tags with < 2 kBytes or ca. 250 ms for data carriers with ≥ 2 kBytes of memory).
-> Please note this when using the parameter "Process outputs with CT-present".

– **BIS C-1../02B [x] = yes**
This parameter should be switched on when a data carrier of the type BIS C-1../02B is used.

BIS C-6_0 Configuration

Menu settings BIS C-6_0, Parameter Field (continued)

– **CRC_16 data checking**

If CRC_16 data checking is activated, a special error message is output to the interface whenever a CRC error is detected.

If the error message is not caused by a failed write request, it may be assumed that one or more memory cells on the data carrier is defective. That data carrier must then be replaced.

If the CRC error is however due to a failed write request, you must reinitialize the data carrier in order to continue using it.

The checksum is written to the data carrier as a 2-byte wide datum. Two bytes per page are 'lost', i.e., the page size becomes 30 bytes or 62 bytes depending on data carrier type (see next page).

Data carrier type		Usable bytes
128 bytes	=	120 bytes
256 bytes	=	240 bytes
511 bytes *)	=	450 bytes
1023 bytes *)	=	930 bytes
2047 bytes *)	=	1922 bytes
2048 bytes	=	1984 bytes
8192 bytes	=	7936 bytes

This means that the actual usable number of bytes is reduced!

*) The last data carrier page for these EEPROM-based data carriers is not fully available.

BIS C-6_0
Configuration

Menu settings BIS C-6_0, Page Size Field

Data carrier memory is organized in page sizes of 32 or 64 bytes (also referred to as block size). Factory setting: 32 bytes.

32-byte page size	BIS C-1_ _-02, -03, -04, -05
64-byte page size	BIS C-1_ _-10, -11, -30, -32

Menu settings BIS C-6_0, Input Field

The digital control input on the BIS C-600 can be specified for the desired function. Factory setting is Reset.

- **Reset**
  If Reset is selected, a High signal on this input resets the BIS C-600 processor. All pending commands are deleted.
- **Head Select**
  If Head Select is selected, this input is used to activate the desired read head.

Input Low:	Head 1 selected.
Input High:	Head 2 selected.

  -> This function always has priority. For example the function "Both read/write heads active" selected by the 'HT' command is deactivated.
- **Data bit to data carrier**
  When a new data carrier is recognized, a freely defined bit is written to the data carrier directly or inverted to a specified address. After a successful write the defined output is set until the data carrier leaved the acvtive zone of the read/write head.

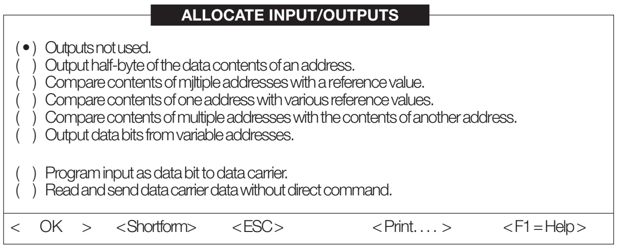
  -> The 'Dynamic Mode' parameter is automatically reset.
- **No function**
  The input is not processed.

BIS C-600
Configuration

Menu Allocate Input/Outputs (BIS C-600 only)

Allocate to input/outputs

The outputs can have various functions allocated to them. The output functions are always processed when reading. The condition for this is that the respective address was read during the preceding read request.



BIS C-600 only

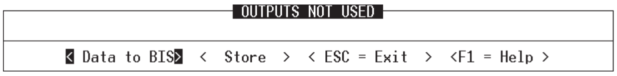
ALLOCATE INPUT/OUTPUTS

(●) Outputs not used.
( ) Output half-byte of the data contents of an address.
( ) Compare contents of mjltiple addresses with a reference value.
( ) Compare contents of one address with various reference values.
( ) Compare contents of multiple addresses with the contents of another address.
( ) Output data bits from variable addresses.

( ) Program input as data bit to data carrier.
( ) Read and send data carrier data without direct command.

< OK > <Shortform> <ESC> <Print. . . . > <F1 = Help>

"Outputs not used" deactivates processing of the outputs.



OUTPUTS NOT USED

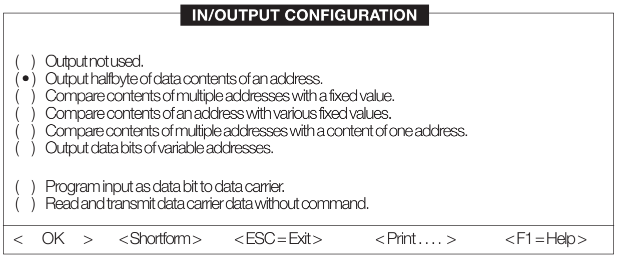
< Data to BIS> < Store > < ESC = Exit > <F1 = Help >

"Data to BIS" sends the data to the processor. "Save" stores the data in the configuration file of your computer.

BIS C-600
Configuration

Menu Allocate Input/Outputs (BIS C-600 only) (continued)

Output Halfbyte of the Data Contents of an Address:

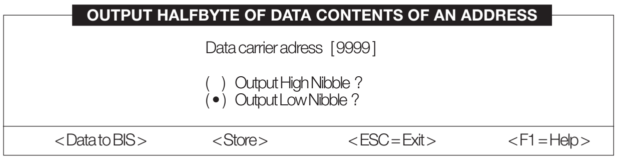


IN/OUTPUT CONFIGURATION

( ) Output not used.
(•) Output halfbyte of data contents of an address.
( ) Compare contents of multiple addresses with a fixed value.
( ) Compare contents of an address with various fixed values.
( ) Compare contents of multiple addresses with a content of one address.
( ) Output data bits of variable addresses.

( ) Program input as data bit to data carrier.
( ) Read and transmit data carrier data without command.

< OK > <Shortform> <ESC=Exit> <Print.... > <F1=Help>



OUTPUT HALFBYTE OF DATA CONTENTS OF AN ADDRESS

Data carrier adress [9999]

( ) Output High Nibble ?
(•) Output Low Nibble ?

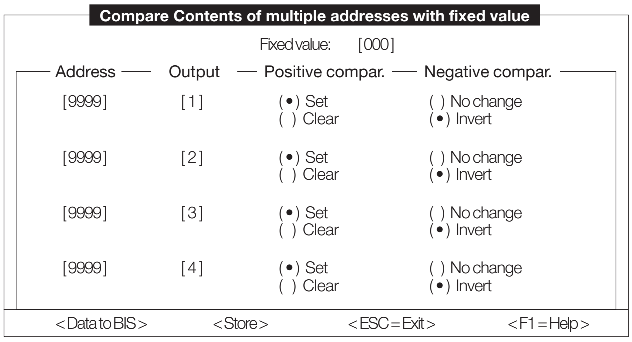
<Data to BIS> <Store> <ESC=Exit> <F1=Help>

Either the upper or lower 4 bits (Nibble) of the 8 bits data contents of an address is output (Bit 0 on Output 1, Bit 1 on Output 2, etc.). The address is given in decimal.

BIS C-600
Configuration

Menu Allocate Input/Outputs (BIS C-600 only) (continued)

Compare Contents of multiple Addresses with a fixed Value:
The data contents of up to 4 addresses are compared with a fixed decimal value. To each address can be assigned which of the Outputs 1 to 4 is set or cancelled by a positive result of the comparison an whether in case of a negative result of the comparison the output shall not be changed or shall be set in contrary to the definition with the positive result (inverted response).
All addresses found within the read command will be processed.



Compare Contents of multiple addresses with fixed value

Fixed value: [000]

Address	Output	Positive compar.	Negative compar.
[9999]	[1]	(•) Set ( ) Clear	( ) No change (•) Invert
[9999]	[2]	(•) Set ( ) Clear	( ) No change (•) Invert
[9999]	[3]	(•) Set ( ) Clear	( ) No change (•) Invert
[9999]	[4]	(•) Set ( ) Clear	( ) No change (•) Invert

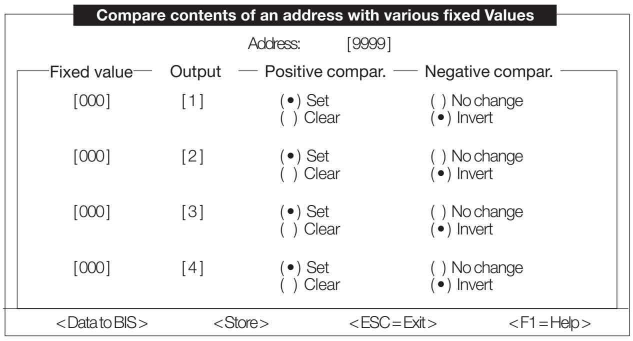
<Data to BIS> <Store> <ESC=Exit> <F1=Help>

If the parameter "Process outputs with CT-Present" is included in the initialization, then this function also will be carried out on recognition of a new data carrier (one or more of the addresses given should be on the first page of the data carrier).

BIS C-600
Configuration

Menu Allocate Input/Outputs (BIS C-600 only)
(continued)

Compare an Address with various fixed Values:
The data contents of an address is compared with 4 fixed decimal values. For each fixed value is indicated which of the Outputs 1 to 4 is set or cancelled by a positive result of the comparison and whether in case of a negative result of the comparison the output shall not be changed or shall be set in contrary to the definition with the positive result (inverted response).



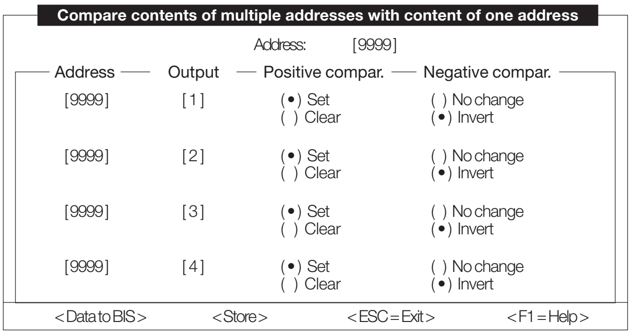
```
        Compare contents of an address with various fixed Values
                        Address:       [9999]
  Fixed value    Output    Positive compar.    Negative compar.
     [000]        [1]        (•) Set             ( ) No change
                             ( ) Clear           (•) Invert

     [000]        [2]        (•) Set             ( ) No change
                             ( ) Clear           (•) Invert

     [000]        [3]        (•) Set             ( ) No change
                             ( ) Clear           (•) Invert

     [000]        [4]        (•) Set             ( ) No change
                             ( ) Clear           (•) Invert
  <Data to BIS>      <Store>       <ESC=Exit>       <F1=Help>
```

If the parameter "Process outputs with CT-Present" is included in the initialization, then this function also will be carried out on recognition of a new data carrier (one or more of the addresses given should be on the first page of the data carrier).

BIS C-600
Configuration

Menu Allocate Input/Outputs (BIS C-600 only)
(continued)

Compare Contents of multiple Addresses with the Content of another Address:
The data contents of up to 4 addresses are compared with the data contents of another address. For each address is indicated which of the Outputs 1 to 4 is set or cancelled by a positive result and whether in case of a negative result of the comparison the output shall not be changed or shall be set in contrary to the definition with the positive result.
All addresses found within the read command will be processed provided the address to be compared is within the range.



```
      Compare contents of multiple addresses with content of one address
                        Address:       [9999]
  Address        Output    Positive compar.    Negative compar.
    [9999]        [1]        (•) Set             ( ) No change
                             ( ) Clear           (•) Invert

    [9999]        [2]        (•) Set             ( ) No change
                             ( ) Clear           (•) Invert

    [9999]        [3]        (•) Set             ( ) No change
                             ( ) Clear           (•) Invert

    [9999]        [4]        (•) Set             ( ) No change
                             ( ) Clear           (•) Invert
  <Data to BIS>      <Store>       <ESC=Exit>       <F1=Help>
```

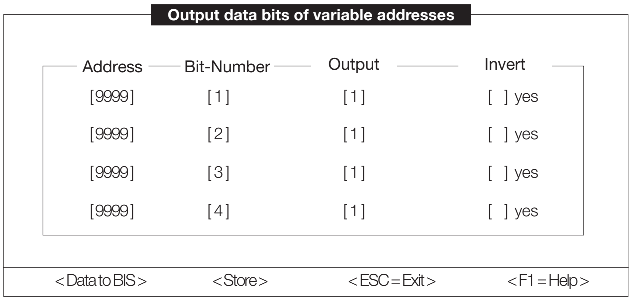
If the parameter "Process outputs with CT-Present" is included in the initialization, then this function also will be carried out on recognition of a new data carrier (one or more of the addresses given should be on the first page of the data carrier).

BIS C-600
Configuration

Menu Allocate Input/Outputs (BIS C-600 only) (continued)

Output Data Bits of variable Addresses:

1 data bit of an address or 1 bit each from up to 4 addresses can be output on one of the 4 outputs and then inverted or not inverted.



Output data bits of variable addresses

Address	Bit-Number	Output	Invert
[9999]	[1]	[1]	[ ] yes
[9999]	[2]	[1]	[ ] yes
[9999]	[3]	[1]	[ ] yes
[9999]	[4]	[1]	[ ] yes

<Data to BIS> <Store> <ESC = Exit> <F1 = Help>

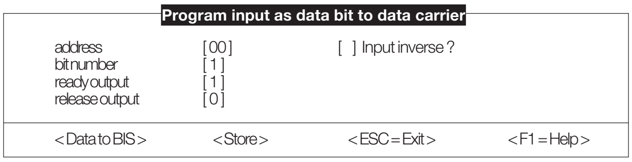
If the parameter "Process outputs with CT-Present" is included in the initialization, then this function also will be carried out on recognition of a new data carrier (one or more of the addresses given should be on the first page of the data carrier).

BIS C-6_0
Configuration

Menu Allocate Input/Outputs (continued)

Programming a Data Bit on the Data carrier depending on the Input:

On recognition of a new data carrier the state of the digital input will be written as a direct or inverted bit on the data carrier. The address range is 0...31! Bit nummer of the address is 1...8. The outputs to be used for the ready and release signals should also be given. If release output is given as "0" then the release function will not be used. The procedure is described below.



Program input as data bit to data carrier

address	[00]	[ ] Input inverse ?
bit number	[1]	
ready output	[1]	
release output	[0]	

<Data to BIS> <Store> <ESC = Exit> <F1 = Help>

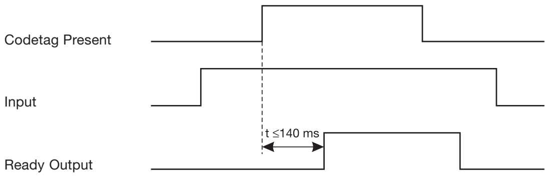
Procedure without Release Signal

On recognition of a new data carrier the defined bit of the given address will be written, direct or inverted. After a successful write operation, the given ready output is set until the data carrier leaves the active read/write range. The input must hold its state until the ready output is set.

> The input state that is to be written as information on the data carrier must be present already before the presence of the new data carrier is recognized.

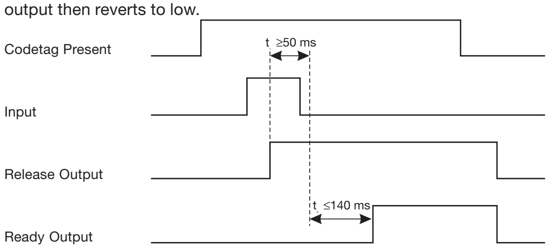
BIS C-6_0 Configuration

Menu Allocate Input/Outputs (continued)



In this example the bit is set to 1.

Procedure with Release Signal
After the recognition of a new data carrier, the input state is continuously sampled until it is set (release given). The processor sets the release output and waits 50 ms. The input is then sampled and its state is taken over as bit-value. This value is written on the data carrier directly or inverted depending on the selected configuration. After the write operation, the given ready output is set until the data carrier leaves the active read/write range. The release output then reverts to low.

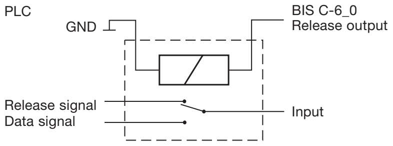


In this example the bit is reset to 0.

BIS C-6_0 Configuration

Menu Allocate Input/Outputs (continued)

The release output can be used to operate a relay in order to switch the input between the release and data signals.



As an input signal to a PLC this output can indicate that it is time to switch the data signal to the input of the BIS C-6_0. This signal is required at all places where there is a possibility that the data carrier can come into the read/write range before the input data signal is present at the processor.

This function must be additionally released for use during the initializing phase.

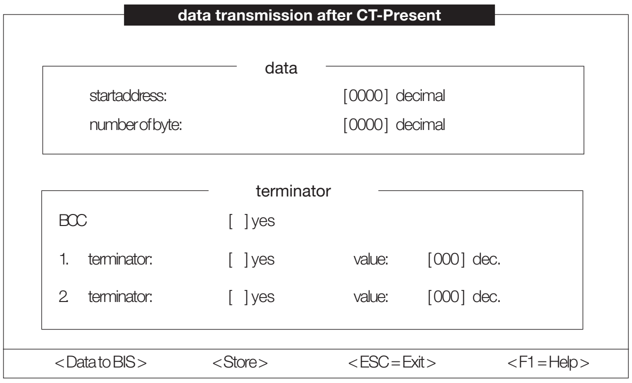
BIS C-6_0 Configuration

Menu Allocate Input/Outputs (continued)

Read and transmit tag data without command:
The given number of bytes (Number of bytes from Start address) will be read from a newly recognized data carrier.

After they are read, the data will be automatically sent to the interface.

Additionally, as termination, a BBC and/or 1 or 2 freely definable terminating characters can be sent.



data transmission after CT-Present

data

startaddress: [0000] decimal
numberofbyte: [0000] decimal

terminator

BCC [ ] yes
1. terminator: [ ] yes value: [000] dec.
2. terminator: [ ] yes value: [000] dec.

<Data to BIS> <Store> <ESC=Exit> <F1=Help>

Customer Configuration

Initialization

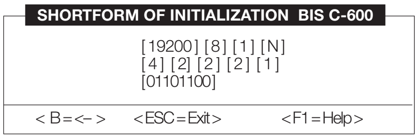
Please note the settings in the label fields on the inside of the processor cover so that in case of repair of the processor the settings can be reset in the factory. Note the settings also in the following fields so that you can set, e.g. other units, to an identical configuration.

Interface: Baudrate □□□□□ Data bit □ Stop bit □ Parity □

Protocol type: □ 2 2 Block size □ Input □

Parameters □□□□□□□□

On the following page you will find an example which shows how you can print-out after initializing. Enter the settings in the appropriate fields so that you have them handy and can reproduce the settings at any time. You can then enter the data in short form into the mask. (see also page 29).



SHORTFORM OF INITIALIZATION BIS C-600

[19200] [8] [1] [N]
[4] [2] [2] [2] [1]
[01101100]

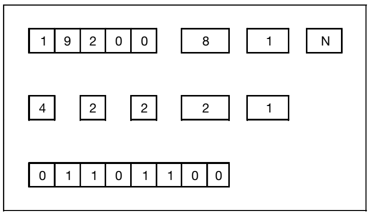
< B=<– > <ESC=Exit> <F1=Help>

Customer Configuration

Initialization
(continued)

Example of a print-out after initialization which you can print with the software BISC600A.EXE.

```
Interface settings
Baudrate          :   19200 baud
Data bit          :   8
Stop bit          :   1
Parity            :   None
Protocol
( )      BCC
( )      CR as terminator
( )      CR
( * )    LF CR as terminator
Parameter
[  ]     Output Data carrier Present on RTS line.
[ X ]    Direct data transmission
[ X ]    Dynamic mode
[  ]     Data carrier Present signal on output 1
[ X ]    Process outputs with Data carrier Present
[ X ]    Fast data carrier recognition
[  ]     BIS C-1../02B   [ X ] = yes
[  ]     CRC_16 data checking
Block size
( )      32 Byte page size
( * )    64 Byte page size
Input
( * )    Input = RESET
( )      Input = Head selection
( )      Input = Data bit on data carrier
( )      Input = Not used
```



```
[1|9|2|0|0]   [8]   [1]   [N]

[4]  [2]  [2]  [2]   [1]

[0|1|1|0|1|1|0|0]
```

The entries in the field are either in clear text (as with Interface settings) or the number of the line marked. In the case of 'Parameters' the marked line is indicated by a 1.

Customer Configuration

Input/Output Configuration

Please note the settings in the label fields on the inside of the processor cover so that in case of repair of the processor the settings can be reset in the factory. Note the settings also in the following fields so that you can set, e.g other units, to an identical configuration.

Address □□□□ □ □ □

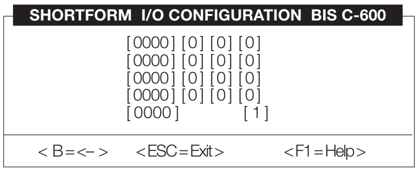
Address □□□□ □ □ □

Address □□□□ □ □ □

Address □□□□ □ □ □

Fixed value □□□□ □ Type

On the following page you will find an example which shows how you can print-out after initializing. Enter the settings in the appropriate fields so that you have them handy and can reproduce the settings at any time. You can then enter the data in short form into the mask. (see below).



```
SHORTFORM  I/O CONFIGURATION  BIS C-600
          [0000] [0] [0] [0]
          [0000] [0] [0] [0]
          [0000] [0] [0] [0]
          [0000] [0] [0] [0]
          [0000]           [1]

  < B=<– >    <ESC=Exit>       <F1=Help>
```

In case the initialization settings are available (e.g. on the cover after exchanging the processor), the settings can be directly entered into the mask "Shortform I/O configuration BIS C-600".

Customer Configuration

Input/Output Configuration (continued)

Example of a print-out after initialization which you can print with the software BISC600A.EXE.

Input/output configuration
(   ) Output not used.
(   ) Output halfbyte of data contents of an address.
(•) Compare contents of multiple addresses with a fixed value.
(   ) Compare contents of an address with various fixed values.
(   ) Compare contents of multiple addresses with content of one address.
(   ) Output data bits of variable addresses.

(   ) Program input as data bit to data carrier.
(   ) Read and transmit tag data without command.

Definition

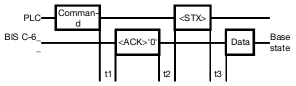
Fixed value:	123	
Address:	0010	
Output:	1	
Positive comparison:	(•) Set	( ) No change
Negative comparison:	( ) Clear	(•) Invert
Address:	0072	
Output:	2	
Positive comparison:	( ) Set	( ) No change
Negative comparison:	(•) Clear	(•) Invert
Address:	0114	
Output:	3	
Positive comparison:	(•) Set	(•) No change
Negative comparison:	( ) Clear	( ) Invert
Address:	0254	
Output:	4	
Positive comparison:	( ) Set	(•) No change
Negative comparison:	(•) Clear	( ) Invert

0	0	1	0	1	S	I
0	0	7	2	2	L	I
0	1	1	4	3	S	N
0	2	5	4	4	L	N
0	1	2	3			3

Protocol

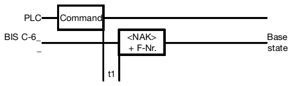
Dialog mode without Head Select

Read: a) If no error:



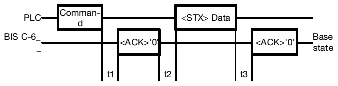
t1 depending on no. of bytes to read (see page 55/56)
t2 ≥ 0 (is not monitored by the processor)
t3 = max. 50 ms

b) With error:



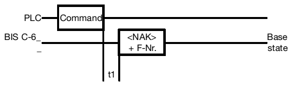
t1 depending on no. of bytes to read (see page 55/56)
and error type (recommended monitor time: 15 s)

Write: a) If no error:



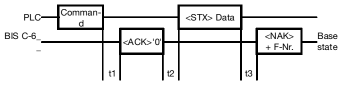
t1 = max. 50 ms
t2 ≥ 0 (is not monitored by the processor)
t3 depending on no. of bytes to write
(see page 55/56)

b) With error in command:



t1 = max. 50 ms
t2 ≥ 0 (is not monitored by the processor)
t3 depending on no. of bytes to write (see page 55/56)
and error type (recommended monitor time:
30 s for data carriers with 32 byte block size,
60 s for data carriers with 64 byte block size)

c) With error in writing:



t1 = max. 50 ms

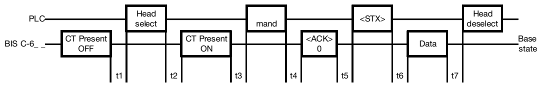
The examples are valid only if:

- The processor is in the base state.
- A data carrier is present in front of a read/write head.

Protocol

Dialog mode with Head Select

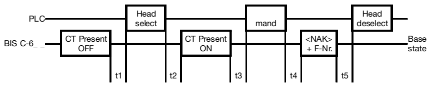
Read:
a) If no error:



t1, t3, t7 ≥ 0
t2 = max. 500 ms
t4 depending on no. of bytes to read (see page 55/56)

t5 ≥ 0 (is not monitored by the processor)
t6 = max. 50 ms

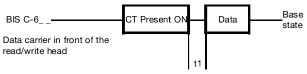
b) With error:



t1, t3, t5 ≥ 0
t2 = max. 500 ms

t4 depending on no. of bytes to read (see page 55/56) and error type (recommended monitor time: 15 s)

Direct read mode



t1 depending on no. of bytes to read (see page 55/56)

The examples are valid only if:

- The processor is in the base state.
- A data carrier is present in front of a read/write head.

Programming Information

The previous sections have described the basic telegram sequence and configuration. Following is information on how to correctly construct telegrams.

There are specific telegrams for the individual operations in the BIS C identification system. They always begin with the command which corresponds to the telegram type:

Telegram types with associated command (ASCII)

'L'	Read data carrier with head and block size selection
'P'	Write to data carrier with head and block size selection
'C'	Write a constant value to the data carrier with head and block size selection
'R'	Read data carrier
'W'	Write to data carrier
'H'	Select read/write head and block size with the variants
'?'	Find next data carrier (one-time)
'!'	or Find next data carrier (continuous)
'B'	Process outputs
'Q'	Restart processor (Quit)
'S'	Check status message
'Z'	Initialize CRC_16 data check

Please note:
– Continuous querying on the interface is not permitted!
– The minimum wait time between two commands is 300 ms!

Programming Information

Explanation of selected telegram contents

Start address and no. of bytes	The start address (A3, A2, A1, A0) and the number of bytes to send (L3, L2, L1, L0) are transmitted as ASCII characters. For the start address a range of 0000 to 8191 and for the number of bytes 0001 to 8192 can be used. A3 ... L0 stand for 1 ASCII character each. **Please note:** Start address + number of bytes may not exced the data carrier capacity.

Head number and block size	For the 'L' (read with head select and page size) and 'P' (write with head select and page size) commands, first the number of the read/write head K ('1' or '2') and then the block size B ('0', '1') of the data carrier are sent.
	B = '0' corresponds to 64 bytes,
	B = '1' corresponds to 32 bytes.

Acknowledgement within	The <ACK> '0' is sent by the identification system if the serially transmitted characters were correctly recognized and a data carrier is the active zone of a read/write head. FOr the 'R' command, <ACK> '0' is only given if the data are ready for sending. <NAK> + 'Error No.' is sent as an acknowledgement if an error is detected or if there is no data carrier within the active zone of a read/write head.
Start	<STX> starts data transmission.
Bytes sent	The data are transmitted code-transparent (not converted).

Programming

BCC Block Check

The BCC block check is formed as an EXOR of the serially transmitted binary characters of the telegram block. Example: Read 128 bytes starting at address 13.
The command line without BCC is: 'L 0013 0128 20'. The BCC is formed:

	'L	=	0100 1100	EXOR
	0	=	0011 0000	EXOR
	0	=	0011 0000	EXOR
	1	=	0011 0001	EXOR
	3	=	0011 0011	EXOR
	0	=	0011 0000	EXOR
	1	=	0011 0001	EXOR
	2	=	0011 0010	EXOR
	8	=	0011 1000	EXOR
	2	=	0011 0010	EXOR
	0'	=	0011 0000	EXOR
Block check result:	BCC	=	0100 0111	= 'G'

Variants for finish with BCC, Terminator

If necessary the finish using block check BCC can be replaced with a special ASCII character. This is:

– Carriage Return 'CR'

For hosts which always require a terminator character, this must always be included in the telegrams. Available are:

– Carriage Return 'CR' or
– Line Feed with Carriage Return 'LF CR'.

The various protocol variations are represented on the following page (see also page 13/14).

Programming

Description of Various Protocol Variants

Reference is now made to the command string 'L 0013 0128 20 G' with 'G' as BCC (see preceding page). This command string is here shown in its possible variants; also shown are the various forms of acknowledgement with and without terminator:

Command line from host system to BIS	Acknowledge from BIS for correct reception	Acknowledge from BIS for incorrect reception
with BCC, but no terminator 'L 0013 0128 20 G'	No terminator <ACK> '0'	No terminator <NAK> '1'
with 'CR' instead of BCC, no terminator 'L 0013 0128 20 CR'	No terminator <ACK> '0'	No terminator <NAK> '1'
no BCC, with terminator 'CR' 'L 0013 0128 20 CR'	with terminator 'CR' <ACK> '0 CR'	with terminator 'CR' <NAK> '1 CR'
no BCC, with terminator 'LF CR' 'L 0013 0128 20 LF CR'	with terminator 'LF CR' <ACK> '0 LF CR'	with terminator 'LF CR' <NAK> '1 LF CR'

For <NAK> with error number a '1' was used here (no data carrier present) as an error example.

The respective positions for the additional terminator are shown in the tables in italics.

Programming

Read from data carrier with head select and block size
Write to data carrier with head select and block size

Task	Data Flow	Com-mand	Start address of first byte to be sent	Number of bytes to be sent	Head No.	Block size	End 2)	Acknow-ledge 3)	*Termi-nator 4)*	Start trans-mission	*Termi-nator 4)*	Data (from start address to start address + no. of bytes)	End 2)	Acknow-ledge 3)	*Termi-nator 4)*
Read	from host system to BIS	**'L'**	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L2 L1 L0 '0 0 0 1' to '8 1 9 2'	K '1' or '2'	B '0' or '1'	BCC or see 2)			<STX>	*'CR' or 'LF CR'*				
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*			D1 D2 D3 ... Dn	BCC or see 2)		
			1)								1)				
Write	from host system to BIS	**'P'**	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '8 1 9 2'	K '1' or '2'	B '0' or '1'	BCC or see 2)			<STX>		D1 D2 D3 ... Dn	BCC or see 2)		
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*					<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*
			1)									1)			

1) The commands Status and/or Quit are not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
4) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here as a terminator.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming

Telegram example for page 38: ***Read from data carrier with head select and block size*** *with block check (BCC)*

-> Head 1 is selected. Read 10 bytes starting at address 50 of the data carrier at read/write Head 2. The data carrier at Head 4 has a block size of 64 bytes.

The host sends 'L 0 0 5 0 0 0 1 0 2 0 J' BCC ($4A_{HEX}$)

Address of first byte to read
Number of bytes to read
Read/write Head No. 2
Block size 0 = 64 bytes

The BIS processor acknowledges with <ACK> '0'
The host system gives the start command <STX>
The BIS processor provides the data from the data carrier '1 2 3 4 5 6 7 8 9 A F' BCC (70_{HEX})
After the telegram sequence, Head 2 remains selected, with 64 byte block size.

Telegram example for page 38: ***Write to data carrier with read/write head select and block size*** *with block check (BCC)*

-> Head 1 is selected. Write 5 bytes starting at address 500 of the data carrier at read/write Head 2. The data carrier at Head 2 has a block size of 64 bytes.

The host sends 'P 0 5 0 0 0 0 0 5 2 0 R' BCC (52_{HEX})

Address of first byte to write
Number of bytes to write
Read/write Head No. 2
Block size 0 = 64 Byte

The BIS processor acknowledges with <ACK> '0'
The host system gives the start command and data <STX> '1 2 3 4 5 3' BCC (33_{HEX})
The processor acknowledges with <ACK> '0'
After the telegram sequence, Head 2 remains selected, with 64 byte block size.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming

Writing a constant value in the data carrier with read/write select and block size

This command can be used to erase a data carrier data. One saves the time for the transmission of the write byte.

Task	Data Flow	Com-mand	Start address of first byte to be sent	Number of bytes to be sent	Head No.	Block size	End 2)	Acknow-ledge 3)	*Termi-nator 4)*	Start trans-mission	*Termi-nator 4)*	Data (from start address to start address + no. of bytes)	End 2)	Acknow-ledge 3)	*Termi-nator 4)*
Write	from host system to BIS	**'C'**	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '8 1 9 2'	K '1' or '2'	B '0' or '1'	BCC or see 2)			<STX>		D	BCC or see 2)		
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*					<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*
			1)									1)			

1) The commands Status and/or Quit are not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
4) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here as a terminator.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming

Telegram example for page 40: ***Write to data carrier with read/write head select and block size*** *with block check (BCC)*

-> Head 1 is selected. Write 500 bytes of ASCII data value 0 (30_{HEX}) starting at address 20 of the data carrier at read/write Head 2. The data carrier at Head 2 has a block size of 64 bytes.

```
The host sends                                      'C 0020  0500  2 0 F'  BCC (46HEX)
                      Address of first byte to write
                       Number of bytes to write
                         Read/write Head No. 2
                          Block size 0 = 64 Byte
The BIS processor acknowledges with                     <ACK> '0'
The host system gives the start command and data        <STX>  '0 2'  BCC (32HEX)
The processor acknowledges with                         <ACK> '0'
```

After the telegram sequence, Head 2 remains selected, with 64 byte block size.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming

Read from Data carrier, Write to Data carrier

Task	Data Flow	Com-mand	Start address of first byte to send	Number of bytes to send	End 2)	Acknow-ledge 3)	*Termi-nator 4)*	Start trans-mission	*Termi-nator 4)*	Data (from start address to start address + no. of bytes)	End 2)	Acknow-ledge 3)	*Termi-nator 4)*
Read	from host system to BIS	**'R'**	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '8 1 9 2'	BCC or see 2)			<STX>	*'CR' or 'LF CR'*				
	from BIS to host system					<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*			D1 D2 D3 ... Dn	BCC or see 2)		
			1)										
Write	from host system to BIS	**'W'**	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '8 1 9 2'	BCC or see 2)			<STX>		D1 D2 D3 ... Dn	BCC or see 2)		
	from BIS to host system					<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*					<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*
			1)							1)			

1) The commands Status and/or Quit are not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) <ACK> '0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
4) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here as a terminator.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming

Telegram example from page 42: ***Read from Data carrier*** *with block check (BCC)*

Read from Data carrier: -> Read 10 bytes starting at address 50.

The host sends	'R 0 0 5 0 0 0 1 0 V' BCC (56HEX)
Address of first byte to read	
Number of bytes to read	
The BIS processor acknowledges with	<ACK> '0'
The host gives the start command	<STX>
The BIS processor provides the data from the data carrier	'1 2 3 4 5 6 7 8 9 0 SOH' BCC (01HEX)

Telegram example from page 42: ***Write to Data carrier*** *with block check (BCC)*

Write to Data carrier: -> Write 5 bytes starting at address 500.

The host system sends	'W 0 5 0 0 0 0 0 5 W' BCC (57HEX)
The BIS processor acknowledges with	<ACK> '0'
The host sends the data	<STX> '1 2 3 4 5 3' BCC (33HEX)
The BIS processor acknowledges with	<ACK> '0'

The 'R' and 'W' commands represent a subtype of the 'L' and 'P' commands.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming

Selecting a Read/Write Head

The 'H1' command selects Read/Write Head 1, 'H2' Read/Write Head 2, and 'HT' (Head Twin) both Read/Write Heads.

If both heads are selected, please note:

1. Only one data carrier is allowed to be in the active zone of a read/write head at a time.
2. The read or write time increases by ca. 40 ms - regardless of the data amount to be read or written. (This does not apply to the data carrier recognition).
3. The positive acknowledgement for a read or write action is no longer <ACK> '0' but rather <ACK> '1' or <ACK> '2', depending on at which read/write head there is a data carrier to be read from or written to.

Task	Data Flow	Com-mand	Head number	End 2)	Acknowledge 3)	*Terminator 4)*
Select Read/Write Head	from host system to BIS	**'H'**	'1', '2' or 'T'	BCC or see 2)		
	from BIS to host system				<ACK>'0' resp. <NAK> + Error-No.	*'CR' or 'LF CR'*
			1)			

1) The commands Status and/or Quit are not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) <ACK>'0' is returned as acknowledgement if there is no error, or <NAK> + 'Error No.' if an error occurs.
4) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here as a terminator.

Telegram example: ***Selecting a Read/ Write Head*** *with block check (BCC)*

-> Switch to Head 1.

The host sends	'H 1 y' BCC (79HEX)
The BIS processor acknowledges with	<ACK> '0'

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Find Next Data carrier (one time)

The following telegram is used to find the next data carrier. The next following read/write head is selected and checked to see if a data carrier is in front of this read/write head. If yes, the first 4 bytes of the data carrier are read. The telegram reply then contains the corresponding number of the read/write head and the four bytes read. If no tag is found, the original read/write head is reselected and checked. If no data carrier is found here, then the telegram reply is: 'H ? 0000 w'.

'H ?' recognizes any data carrier, regardless of the preset block size, assuming that read/write head and data carrier are compatible.

Task	Data Flow	Com-mand	Des.	End 2)	Acknow-ledge	*Termi-nator 3)*	Reply	Head number	Data from data carrier	End 2)
Find next data carrier (one time)	from host system to BIS	**'H'**	'?'	BCC or see 2)						
	from BIS to host system				<ACK>'0'	*'CR' or 'LF CR'*	'H'	'1', '2' or '?'	D1 D2 D3 D4	BCC or see 2)
			1)							

1) The commands Status and/or Quit are not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here as a terminator.

Telegram example:
Find Next Data carrier (one time)
with block check (BCC)

-> Head 1 is selected. Only read/write head 2 has a data carrier in front of it, whose first four bytes are 9876.

```
The host sends                          'H ?              w'     BCC (77HEX)
The BIS processor acknowledges with     <ACK> '0'
                 and sends the data      'H 2  9 8 7 6    z'     BCC (7AHEX)
```

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Find Next Data carrier (continuous)

The following telegram is used to find the next data carrier. The next following read/write head is selected and checked to see if a data carrier is in front of this read/write head. If yes, the first four bytes of the data carrier are read. The telegram reply then contains the corresponding number of the read/write head and the four bytes read. If no tag is found, the original read/write head is reselected and checked. This procedure is repeated until a data carrier is found.

H !' recognizes any data carrier, regardless of the preset block size, assuming that read/write head and data carrier are compatible.

Task	Data Flow	Com-mand	Desi-gnator	End 2)	Acknow-ledge	*Termi-nator 3)*	Reply	Head number	Data from data carrier	End 2)
Find next data carrier (contin.)	from host system to BIS	**'H'**	'!'	BCC or see 2)						
	from BIS to host system				<ACK>'0'	*'CR' or 'LF CR'*	'H'	'1' or '2'	D1 D2 D3 D4	BCC or see 2)
			1)							

1) The commands Status and/or Quit are not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here as a terminator.

Telegram example:
Find Next Data carrier (continuous)
with block check (BCC)

-> Read/write head 2 has a data carrier in front of it whose first four bytes are 9876.

```
The host sends                          'H !              i'     BCC (69HEX)

The BIS processor acknowledges with   <ACK> '0'
                  and sends the data    'H 2  9 8 7 6     z'     BCC (7AHEX)
```

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming

Processing the Outputs

A telegram can be sent to set or cancel the four outputs.

Task	Data Flow	Com-mand	Designator	Termi-nator 2)	Acknow-ledge	Termi-nator 3)
Process outputs (set or cancel)	from host system to BIS	**'B'**	'00' bis 'A1' (see below)	BCC or see 2)		
	from BIS to host system				<ACK>'0'	'CR' or 'LF CR'
			1)			

Designator meaning:	Output No.	0	1	2	3	all outputs
	Cancel output	00	10	20	30	A0
	Set output	01	11	21	31	A1

1) The commands Status and/or Quit are not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here as a terminator.

Telegram example: ***Processing the Outputs*** *with block check (BCC)*

The host sends 'B 21 A' BCC (41_{HEX})
The BIS processor acknowledges with <ACK> '0'

After the telegram is completed, output 2 is set.

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming

Show Output Condition

This telegram is used to check the condition (status) of all four outputs.

Task	Data Flow	Com-mand	Desig-nator	End 2)	Acknow-ledge	Condition of the 4 outputs	Termi-nator 3)	End 2)
Show output condition	from host system to BIS	**'B'**	'AO'	BCC or see 2)				
	from BIS to host system				<ACK>'0'	'XXXX' '0' =cancelled, '1' = set	'CR' or 'LF CR'	BCC or see 2)
			1)					

Output status is shown in sequential order 0 1 2 3

1) The commands Status and/or Quit are not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.
3) For protocol variants which always require a terminator, either 'CR' or 'LF CR' must be inserted here as a terminator.

Telegram example: ***Show Output Condition*** *with block check (BCC)*

-> Outputs 0 and 1 are set, outputs 2 and 3 are cancelled.

The host sends 'B AO L' BCC ($4C_{HEX}$)

The BIS processor acknowledges with <ACK> '0'
and sends the data '1100 NUL' BCC (00_{HEX})

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming

Restart the Processor (Quit)

Sending the Restart command causes a telegram in process to be aborted and puts the processor in the ground state. After this telegram is acknowledged, a minimum of 1600 ms pause should be allowed before starting a new telegram.

Important! The Quit command is not permitted while the processor is waiting for a terminator (BCC, 'CR' or 'LF CR'). In this situation, the Quit would be incorrectly interpreted as a terminator or datum.

Task	Data Flow	Command	Terminator 2)	Acknowledge	Terminator 2)
Restart (Quit)	from host system to BIS	**'Q'**	BCC or see 2)		
	from BIS to host system			'Q'	BCC or see 2)
			1)	1)	1)

1) The commands Status and/or Quit are not permitted at this point.
2) Instead of block check BCC, depending on protocol variant either Carriage Return 'CR' or LIne Feed with Carriage Return may be used.

Telegram example with block check (BCC):

Put the BIS system into the ground state.

The host sends 'Q Q' BCC (51_{HEX})

The BIS processor acknowledges with 'Q Q' BCC (51_{HEX})

Data within angle brackets are control characters.
Values inside apostrophes represent the respective character(s) in ASCII code.

Programming Information

Querying the status message

The status telegram checks to see what kind of telegram is in process.

Important! The Status command is not permitted while the processor is waiting for a termination character (BCC, 'CR' or 'LF CR'). In this situation Status would be misinterpreted as a termination or data character.

☞ **Important:** A status check during a red or write operation on a data carrier (Data carrier Operating LED on) increases the read or write time. Especially in dynamic mode this can result in insufficient time for a full read or write while the tag is in the active zone of the read/write head. Continuous status checking disturbs the processing of the data carrier, and the tag may not be recognized!

The characters between the apostrophes represent the respective character(s) in ASCII code.
'_' = Space = ASCII 20_{HEX}.

Task	Data Flow	Command	Terminator 2)	Status message	Terminator 2)
Check Status Message	From host system to BIS	**'S'**	BCC or see 2)		
	From BIS to host system			'S' '_', 'R', 'W', 'L', 'P' or 'H'	BCC or see 2)
			1)		

1) The Status and/or Quit commands are not permitted at this point.
2) Instead of BCC block check, depending on the protocol variant either Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' can be used.

Programming Information

Status messages and their meaning:

'S L'	=	Read data carrier with head select and block size
'S P'	=	Write to data carrier with head select and block size
'S R'	=	Read from data carrier
'S W'	=	Write to data carrier
'S H'	=	Select head and block size
'S _'	=	No telegram in process

Telegram examples for page 50:
Query status message
with (BCC) blockcheck

-> To check the BIS status just after a **read telegram** has been sent.

Host sends	'S	S'	BCC (53HEX)
BIS acknowledges with	'S L	US'	BCC (1FHEX)

-> To check the BIS status just after a **write telegram** has been sent.

Host sends	'S	S'	BCC (53HEX)
BIS processor acknowledges with	'S P	ETX'	BCC (03HEX)

-> To check the BIS status just after a **Select read/write head telegram** has been sent.

Host sends	'S	S'	BCC (53HEX)
BIS processor acknowledges with	'S H	ESC'	BCC (1BHEX)

-> To check the BIS status when **no telegram** has just been sent.

Host sends	'S	S'	BCC (53HEX)
BIS processor acknowledges with	'S	_'	BCC (20HEX)

Programming Information

Initialize CRC_16 data check

This telegram initializes a data carrier located at the active read/write head for use of CRC_16 data checking. This telegram must also be send again if a CRC error results from a failed write operation, i.e., the data carrier must be reinitialized in order to use it again.

Please note the table on page 16! The indicated number of usable bytes may not be exceeded, i.e., the sum of start address plus number of bytes must not exceed the data carrier memory capacity!

Task	Data Flow	Com-mand	Start address of first byte to be sent	Number of bytes to be sent	Head No.	Block size	End 2)	Acknow-ledge 3)	*Termi-nator 4)*	Start trans-mission	Data (from start address to start address + no. of bytes)	End 2)	Acknow-ledge 3)	*Termi-nator 4)*
Initialize CRC_16 range	from host system to BIS	**'Z'**	A3 A2 A1 A0 '0 0 0 0' to '8 1 9 1'	L3 L3 L1 L0 '0 0 0 1' to '8 1 9 2'	K '1', '2', '3', or '4'	B '0' or '1'	BCC or see 2)			<STX>	D1 D2 D3 .... Dn	BCC or see 2)		
	from BIS to host system							<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*				<ACK>'0' or <NAK> + Error-No.	*'CR' or 'LF CR'*
			1)								1)			

1) The Status and/or Quit commands are not permitted at this point.
2) Instead of BCC block check, depending on the protocol variant either Carriage Return 'CR' or Line Feed with Carriage Return 'LF CR' may be used.
3) <ACK> '0" is sent as an acknowledgement if there was no error, or <NAK> + 'Errorno.' if there was an error.
4) For protocol variants which always need a terminator, either 'CR' or 'LF CR' must be inserted here.

The characters between the apostrophes represent the respective ASCII character(s). '_' = Space = ASCII 20HEX.

Error Numbers

Error Numbers

The BIS C-600 always outputs an error number. The meaning of these error numbers is indicated in the following table.

No.	Error Description	Effect	
1	No data carrier present	Telegram aborted, processor goes into ground state.	
2	Read error	Read telegram aborted, processor goes into ground state.	
3	Read aborted, since the data carrier was removed.	Processor goes into ground state.	
4	Write error	Write telegram aborted, processor goes into ground state.	**CAUTION:** An aborted write could cause new data to be written to the data carrier which may be incomplete! *)
5	Writing aborted,since the data carrier was removed.	Processor goes into ground state.	
6	Interface error	Processor goes into ground state. (parity or stop bit error)	
7	Telegram format error	Processor goes into ground state. Possible format errors: - Command is not 'L'/'P'/'C'/'R'/'W'/'H'/'B'/'Q'/'S'/'Z'. - Start address or number of bytes exceed permissible range	

*) **Note:** If a CRC data check is used, error message "E" could result if error 4 or 5 was not cleared.

Error Numbers

Error Numbers (continued)

No.	Error Description	Effect
8	BCC error, the transmitted BCC is wrong.	Telegram is aborted, processor goes into ground state.
9	Cable break from read/write head or cable not connected, Codetag Present LED flashes.	Telegram is aborted, processor goes into ground state. If both read/write heads were selected using 'HT', one head may not be connected. If both read/write heads are selected, the cable break message only comes if there is no data carrier in front of the connected, functional head.
A	New command not possible, since a read command is already in process.	After error message the read command is stopped, internally, but not acknowledged. Processor goes into ground state.
B	New command not possible, since a write command is already in process.	After error message the write command is stopped, internally, but not acknowledged. Processor goes into ground state.
C	New command not possible, since a head select is already in process.	After the error message, no positive acknowledge is given, even thought the head select was successful. Processor goes into ground state.
E	CRC error: the CRC on the data carrier is wrong. *)	Telegram aborted, processor goes into ground state.
I	EEPROM error	Telegram aborted, processor goes into ground state.

*) **Note:** If a CRC data check is used, error message "E" could result if in the preceding command error 4, 5 or B was reported.

Read/Write Times

Read Times in Static Mode
(Configuration: without dynamic mode, without CRC_16 check)

For double read and compare:

Data carrier with 32 byte blocks	
No. of bytes	Read time [ms]
from 0 to 31	110
for each additional 32 bytes add	120
from 0 to 255	= 950

Data carrier with 64 byte blocks	
No. of bytes	Read time [ms]
from 0 to 63	220
for each additional 64 bytes add	230
from 0 to 2047	= 7350

Write Times in Static Mode
(Configuration: without dynamic mode, without CRC_16 check)

Including readback and compare:

Data carrier with 32 byte blocks	
No. of bytes	Write time [ms]
from 0 to 31	110 + n * 10
for 32 bytes or more	y * 120 + n * 10

Data carrier with 64 byte blocks	
No. of bytes	Write time [ms]
from 0 to 63	220 + n * 10
for 64 bytes or more	y * 230 + n * 10

n = number of contiguous bytes to write
y = number of blocks to be written

Example:
Read 17 bytes starting at address 187. Data carrier with 32 byte blocks.
Blocks 5 and 6 have to be accessed, because start address 187 is in block 5 and end address 203 is in block 6.

t = 2 * 120 + 17 * 10 = 410

The indicated times apply after the data carrier has been recognized. Otherwise an additional 45 ms must be added to allow for the energy field to be established until the data carrier is recognized.

Read/Write Times

Read Times in Dynamic Mode
(Configuration: with dynamic mode, without CRC_16 check)

Read Times for 1 Block with double read and compare:

Data carrier with 32 byte blocks	
No. of bytes	Read time [ms]
from 0 to 3	14
for each additional byte add	3.5
from 0 to 31	112

Data carrier with 64 byte blocks	
No. of bytes	Read time [ms]
from 0 to 3	14
for each additional byte add	3.5
from 0 to 63	224

m = highest address to be read

Formula: t = (m + 1) * 3.5 ms

Example: Read 11 bytes starting at address 9. Hence the highest address to be read is 19.
This results in 70 ms.

Write Times in Dynamic Mode
(Configuration: with dynamic mode, without CRC_16 check)

Including readback and compare:

Data carrier with 32 byte blocks	
No. of bytes	Write time [ms]
from 0 to 3	14 + n * 10
for each additional byte add	3.5

Data carrier with 64 byte blocks	
No. of bytes	Write time [ms]
from 0 to 3	14 + n * 10
for each additional byte add	3.5

n = Number of contiguous bytes to be written

The indicated times apply after the data carrier has been recognized. Otherwise an additional 45 ms must be added to allow for the energy field to be established until the data carrier is recognized.

LED Display

LED Display:
System Ready
Data carrier Present
Data carrier
Operating

The BIS C-600 Processor uses three LED's on the front panel to indicate the most important operating conditions.

Condition	LED	Meaning
System Ready	on (green)	Supply voltage OK; no hardware error
	off	Supply voltage or hardware not OK or read/write head cable break or not connected.
Data carrier during Present	on (yellow)	Data carrier ready to read or write. (If a read/write error occurs a read/write operation, System Ready LED goes out, if the protocol variant "without error number" is used!)
	flashes	Read/write head cable break or not connected. If both read/write heads were selected using 'HT', one head may not be connected. If both read/write heads are selected, the cable break message only comes if there is no data carrier in front of the connected, functional head.
	off	No data carrier in active zone of read/write head
Data carrier Operating	on (yellow)	Command being processed
	off	No command in process

If all three LED's flash on and off in synch, the processor needs to be returned to the factory for repair.

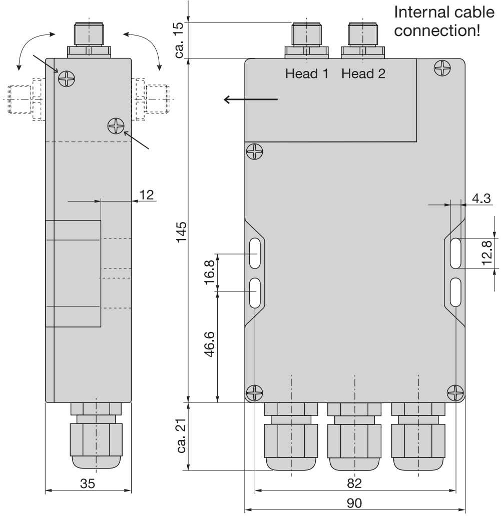
BIS C-600
Processor / Head Assembly

Assembling the BIS C-600 processor and configuring the read/write head or BIS C-650 adapter

The processor is attached at the 4 side through-holes.

Depending on the processor version either a read/write head or the adapter for remote read/write head is fitted. Both the read/write head and the adapter can be rotated by the user by + or - 90° to the desired orientation (see illustration). Be sure that the device is turned off. Remove both screws (indicated by arrows in the illustration). Carefully pull the head or adapter out towards the side (direction of arrow, right illustration). **Caution: Internal cable connection!** Attach it in the desired position and tighten down with the screws.

Caution:
Internal cable connection!

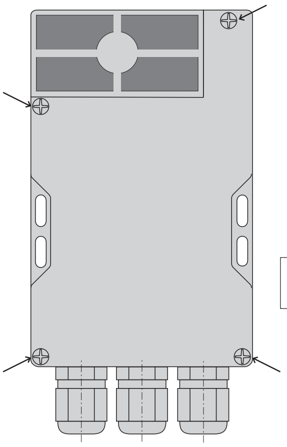


BIS C-600
Processor / Head Assembly

Opening the processor BIS C-600

The BIS C-600 processor must be opened in order to make the connections.

Ensure that the device is turned off. Remove the 4 screws on the BIS C-600 and lift off the cover. Guide the connection cable through the cable fittings. See following pages for additional details.



Attaching the cover (4 screws),
max. permissible torque 0.15 Nm

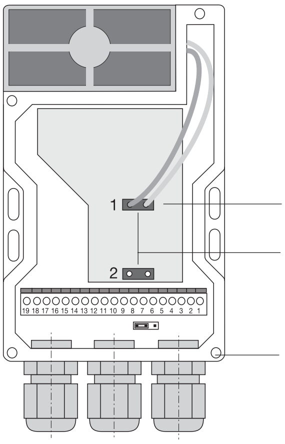
Write your configuration data on the sticker supplied and apply it to the interior of the housing cover.

Opening the processor

BIS C-600
Processor / Head Assembly

Attaching a read/ write head or BIS C-650/-670 adapter

To change out a read/write head: Turn processor off and open. Remove mounting screws on the read/write head (see page 58) and remove processor cover (see page 59). Unplug the read/write head from the circuit board and draw the connection cable out through the cable opening. To attach the new head, proceed in reverse.



To attach the adapter, proceed as described above. For the BIS C-650 both connection cables must be plugged into the circuit board.

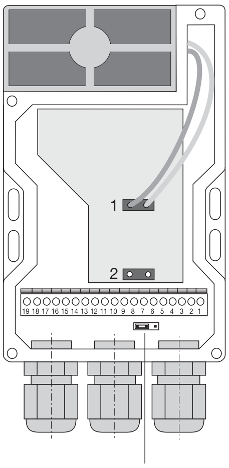
Connecting the integrated read/write head or
BIS C-670 adapter

Connections for the BIS C-650 adapter
1 = Head 1
2 = Head 2

Attaching the cover (4 screws),
max. permissible torque 0.15 Nm

BIS C-600 Interface Information

Wiring diagram for BIS C-600 processors with integrated read/write head



Terminal strip

Shunt for handshake setting when using RS 232 (see following page)

19	18	17	16	15	14
+VS	–VS	⏚	TxD	RxD	COM
POWER			Service		

13	12	11	10	9	8	7	6
+VS	–VS	1	2	3	4	+IN	–IN
OUTPUT						INPUT	

5	4	3	2	1
COM	RxD	CTS	TxD	RTS
RS 232				

BIS C-600...00

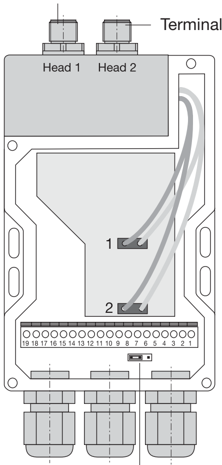
Connection names and locations

BIS C-600 Interface Information

Wiring diagram for BIS C-600 processors with adapter BIS C-650

Terminal for read/write head 1

Terminal for read/write head 2



Terminal strip

19	18	17	16	15	14
+VS	–VS	⏚	TxD	RxD	COM
POWER			Service		

13	12	11	10	9	8	7	6
+VS	–VS	1	2	3	4	+IN	–IN
OUTPUT						INPUT	

5	4	3	2	1
COM	RxD	CTS	TxD	RTS
RS 232				

BIS C-600...00

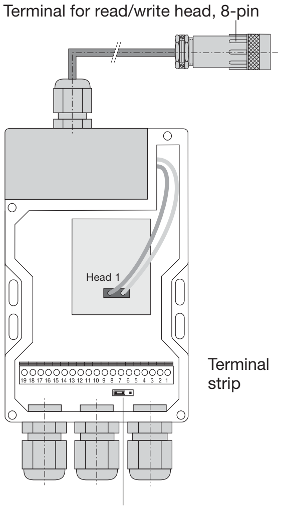
Connection names and locations

Shunt for handshake setting when using RS 232 (see following page)

BIS C-600
Interface Information

Wiring diagram for BIS C-600 processors with adapter BIS C-670

Terminal for read/write head, 8-pin



19	18	17	16	15	14
+VS	–VS	⏚	TxD	RxD	COM
POWER			Service		

13	12	11	10	9	8	7	6
+VS	–VS	1	2	3	4	+IN	–IN
OUTPUT						INPUT	

5	4	3	2	1
COM	RxD	CTS	TxD	RTS
RS 232				

BIS C-600...00

Connection names and locations

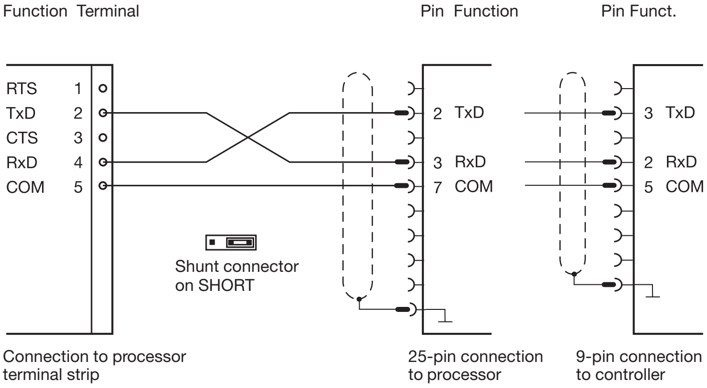
Shunt for handshake setting when using RS 232 (see following page)

BIS C-600...00
Wiring diagrams

Interface functions

A serial port RS 232 interface (V.24) is provided for connecting the BIS C-600 processor to a host system (e.g., PC or PLC).

Interface RS 232 (V.24) without hardware handshake

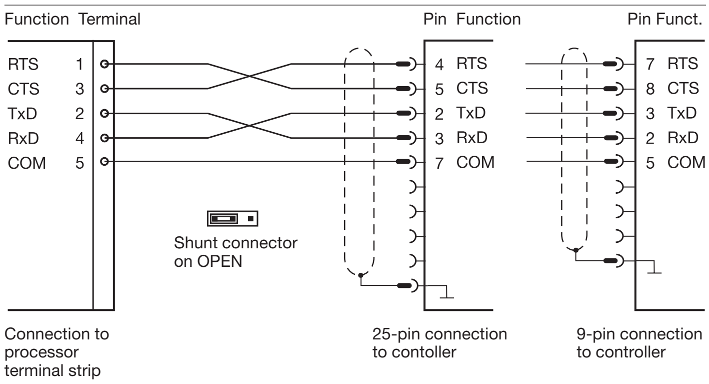


OPEN / SHORT

CTS Factory setting = SHORT.
Since the CTS signal is not used, the shunt connector remains in the SHORT position.

BIS C-600...00
Wiring diagrams

Interface RS 232 (V.24) with hardware handshake



OPEN / SHORT
CTS ▣ Factory setting = SHORT.
Since the CTS signal is used, set the shunt connector in the OPEN position.

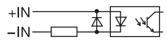
BIS C-600
Technical Data

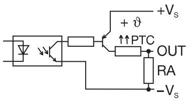
Dimensions, Weight	**Housing**	Plastic PS
	Dimensions with BIS C-652 read/write head	approx. 166 x 90 x 35 mm
	Dimensions with BIS C-650 adapter	approx. 181 x 90 x 35 mm
	Weight	approx. 400 g
Temperature range	Ambient temperature	0 °C to +60 °C
Connections	Terminal strip	19-point
	Cable fitting	3 x clamping PG 9
	Cable diameter	4 to 8 mm
	Wire gauges	AWG 26 to AWG 17
	with crimp contacts	AWG 24 to AWG 22
Protection	Protection	IP 65 (when connected)
Electrical connections	**Supply voltage V_S, input**	DC 24 V ± 20 %
	Ripple	≤ 10 %
	Current draw	≤ 400 mA
	Read/write head	integrated, BIS C-65_ and following*);
	alternate for BIS C-650 adapter *)	2 x integrated connectors 4-pin (male) for all BIS C-3_ _ read/write heads with 4-pin connector (female), **not** BIS C-350 and BIS C-352
	alternate for BIS C-670 adapter *)	1 x 8-pin male connector for one of the read/write heads BIS C-350 and BIS C-352
	Serial interface	RS 232 (V.24)

*) can be rotated by ± 90°

BIS C-600
Technical Data

Electrical connections (cont.)	**Digital input (+IN, –IN)**	optocoupler isolated
	Control voltage active	4 V to 40 V
	Control voltage inactive	1.5 V to –40 V
	Input current at 24 V	11 mA
	Typical delay time	5 ms
	Control outputs (01 to 04)	optocoupler isolated
	Output current	max. 20 mA
	Voltage drop at 20 mA	approx. 2.5 V
	Output resistance R_A	10 kΩ to V_S
	Output circuit	PNP (current sourcing)
	Supply voltage, output V_S	DC 24 V ± 20 %
	Ripple	≤ 10 %
Function displays	System Ready	LED green
	Data carrier Present	LED yellow
	Data carrier Operating	LED yellow





CE *With the CE Mark we affirm that our products are in accordance with the requirements of the EU (European Union) Guideline*
89/336/EEC (EMC Guideline)

and the EMC Law. It has been verified in our EMC Laboratory, which is accredited by the DATech for Testing of Electromagnetic Compatibility, that Balluff products meet the EMC requirements of the Harmonized Standard

EN 61000-6-4 (Emission), EN 61000-6-2 (Noise Immunity)

BIS C-600
Ordering Information

Ordering Code

BIS C-600-007-_ _ _-00-KL1

Balluff Identification System

Series C

Hardware type
600 = Compact

Software type
007 = Balluff protocol

Read/Write Head
000 = no read/write head
651 = with read/write head BIS C-651 (circular antenna on top)
652 = with read/write head BIS C-652 (circular antenna on front)
653 = with read/write head BIS C-653 (bar antenna)
650 = adapter with two connections for external read/write heads BIS C-3_ _ (except BIS C-350 and -352)
670 = adapter with cable connection for an external read/write head BIS C-350 or BIS C-352

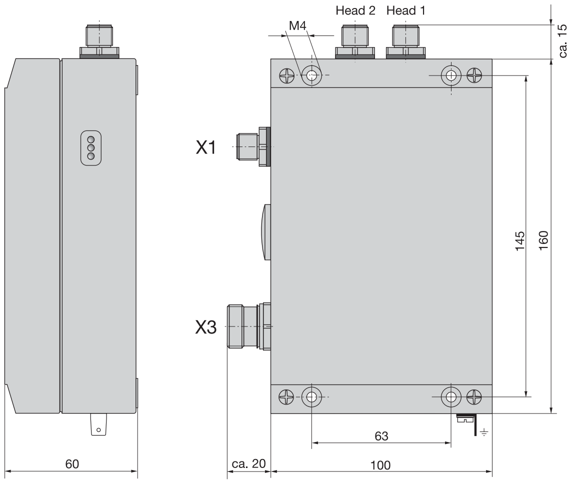
Interface
00 = RS 232 interface (V.24)

User Connections
KL1 = 3x PG 9 cord seal fittings

BIS C-620
Processor Installation

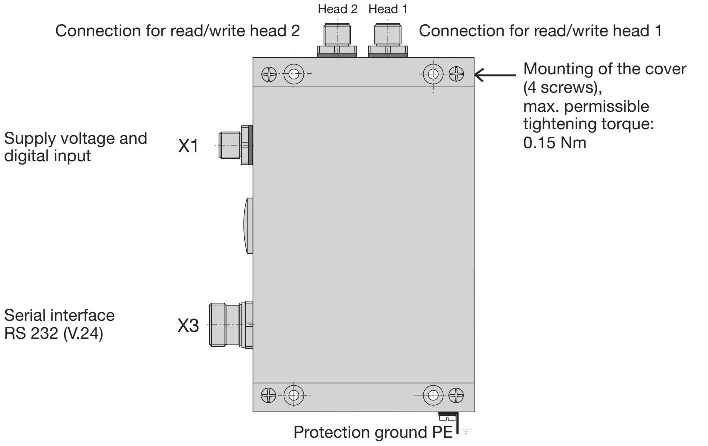
Installing the BIS C-620 processor

The processor is mounted using 4 size M4 screws.



BIS C-620
Interface Information

Wiring for the BIS C-620 processor



Head 2
Head 1
Connection for read/write head 2
Connection for read/write head 1
Mounting of the cover (4 screws), max. permissible tightening torque: 0.15 Nm
Supply voltage and digital input
X1
Serial interface RS 232 (V.24)
X3
Protection ground PE

Connection locations and names

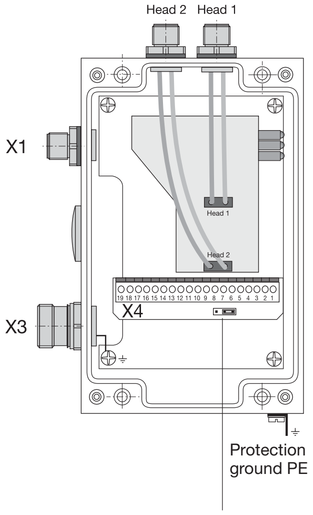
Opening the BIS C-620 processor

To insert the shunt connector for the handshake setting when using RS 232, the BIS C-620 processor must be opened.

Ensure that the device is turned off. Remove the 4 screws on the BIS C-620 and lift off the cover. See following pages for additional details.

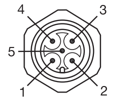
BIS C-620
Interface Information

Wiring for the BIS C-620 processor



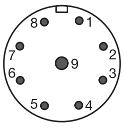
Shunt connector for the handshake setting when using RS 232, see following pages

X1, Supply voltage and digital input



Pin	Function
1	+Vs
2	–IN
3	–Vs
4	+IN
5	n.c.

X3, RS 232
BIS C-620...00



Pin	Function
1	n.c.
2	RxD
3	TxD
4	n.c.
5	COM
6	n.c.
7	RTS
8	CTS
9	n.c.

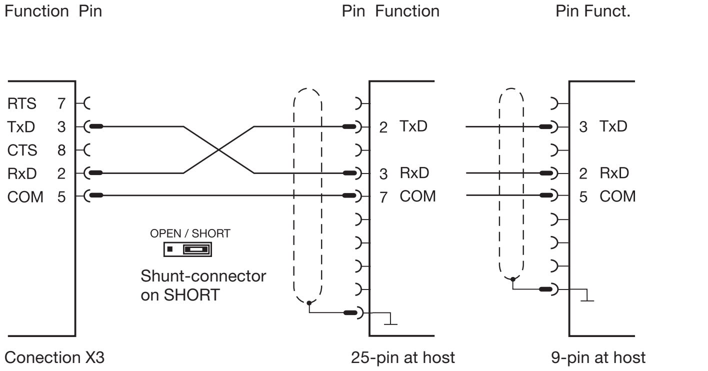
n.c. = leave empty

BIS C-620...00
Wiring Diagram

Function of the Interfaces

The BIS C-620 processor uses a serial interface RS 232 (V.24) for communicating with a host system (e.g. PC or PLC).

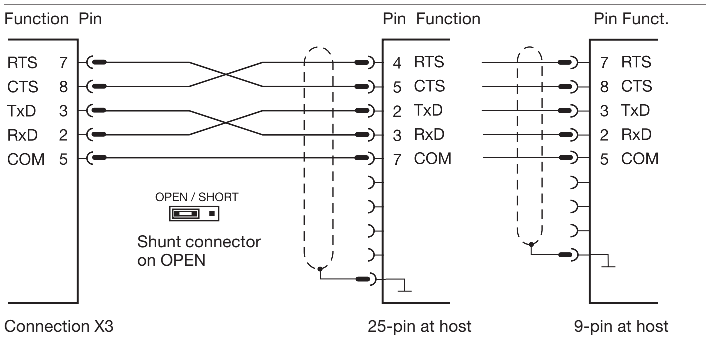
Interface RS 232 (V.24) without hardware handshake



OPEN / SHORT
CTS ▭ Factory setting = SHORT.
Since the CTS signal is not used, the shunt connector remains in the SHORT position.

BIS C-620...00
Wiring Diagram

**RS 232 (V.24)
Interface with
hardware handshake**



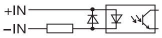
OPEN / SHORT
CTS ▣ Factory setting = SHORT.
Since the CTS signal is used, set the shunt connector in the OPEN position.

BIS C-620
Technical Data

Dimensions, weight	**Housing** Dimensions Weight	Metal 175 x 120 x 60 mm 820 g
Temperature range	Ambient temperature	0 °C to +60 °C
Connections	Integrated connector X1 Einbaustecker Head 1, Head 2 Rundsteckverbinder X3	5-pole (male) 4-pole (male) 9-pole (male)
Protection	Protection	IP 65 (when connected)
Electrical connections	**Input X1, supply voltage V_s** Ripple Current draw	DC 24 V ± 20 % ≤ 10 % ≤ 400 mA
	Terminal X3, serial interface	RS 232 (V.24)
	Connections to read/write head Head 1, Head 2	2 x integral connectors 4-pole (male) for all BIS C-3_ _ read/write heads with 4-pole connector (female), **not** BIS C-350 and BIS C-352

BIS C-620
Technical Data

Electrical Connections	**Digital Input (+IN, –IN)**	Optocoupler isolated	
	Control voltage active	4 V to 40 V	
	Control voltage inactive	1.5 V to –40 V	
	Input current at 24 V	11 mA	
	Typical delay time	5 ms	
Function Displays	System Ready	LED green	
	Data carrier Present	LED yellow	
	Data carrier Operating	LED yellow	



With the CE Mark we affirm that our products are in accordance with the requirements of the EU (European Union) Guideline
89/336/EEC (EMC Guideline)

and the EMC Law. It has been verified in our EMC Laboratory, which is accredited by the DATech for Testing of Electromagnetic Compatibility, that Balluff products meet the EMC requirements of the Harmonized Standard

EN 61000-6-4 (Emission), EN 61000-6-2 (Noise Immunity)

BIS C-620
Ordering Information

Ordering code

BIS C-620-007-050-00-ST2

Balluff Identification System

Series C

Hardware Type
620 = Metal housing

Software Type
007 = Balluff protocol

Read/write head
050 = with two connectors for external read/write heads
BIS C-3_ _ (except BIS C-350 and -352)

Interface
00 = RS 232 (V.24)

User Connections
ST2 = Connector variant
1 round connector for supply voltage,
1 round connector for RS 232 (V.24)

Accessories (optional, not included)

Article		Ordering code
Mating connector	for X1	BKS-S 79-00
	for X3	BKS-S 84-00
Protection cap	for Head 1, Head 2	BES 12-SM-2

Appendix, ASCII Table

Deci- mal	Hex	Control Code	ASCII
0	00	Ctrl @	NUL
1	01	Ctrl A	SOH
2	02	Ctrl B	STX
3	03	Ctrl C	ETX
4	04	Ctrl D	EOT
5	05	Ctrl E	ENQ
6	06	Ctrl F	ACK
7	07	Ctrl G	BEL
8	08	Ctrl H	BS
9	09	Ctrl I	HT
10	0A	Ctrl J	LF
11	0B	Ctrl K	VT
12	0C	Ctrl L	FF
13	0D	Ctrl M	CR
14	0E	Ctrl N	SO
15	0F	Ctrl O	SI
16	10	Ctrl P	DLE
17	11	Ctrl Q	DC1
18	12	Ctrl R	DC2
19	13	Ctrl S	DC3
20	14	Ctrl T	DC4
21	15	Ctrl U	NAK

Deci- mal	Hex	Control Code	ASCII
22	16	Ctrl V	SYN
23	17	Ctrl W	ETB
24	18	Ctrl X	CAN
25	19	Ctrl Y	EM
26	1A	Ctrl Z	SUB
27	1B	Ctrl [	ESC
28	1C	Ctrl \	FS
29	1D	Ctrl]	GS
30	1E	Ctrl ^	RS
31	1F	Ctrl _	US
32	20		SP
33	21		!
34	22		"
35	23		#
36	24		$
37	25		%
38	26		&
39	27		'
40	28		(
41	29		)
42	2A		*
43	2B		+

Deci- mal	Hex	ASCII
44	2C	,
45	2D	-
46	2E	.
47	2F	/
48	30	0
49	31	1
50	32	2
51	33	3
52	34	4
53	35	5
54	36	6
55	37	7
56	38	8
57	39	9
58	3A	:
59	3B	;
60	3C	<
61	3D	=
62	3E	>
63	3F	?
64	40	@

Deci- mal	Hex	ASCII
65	41	A
66	42	B
67	43	C
68	44	D
69	45	E
70	46	F
71	47	G
72	48	H
73	49	I
74	4A	J
75	4B	K
76	4C	L
77	4D	M
78	4E	N
79	4F	O
80	50	P
81	51	Q
82	52	R
83	53	S
84	54	T
85	55	U

Deci- mal	Hex	ASCII
86	56	V
87	57	W
88	58	X
89	59	Y
90	5A	Z
91	5B	[
92	5C	\
93	5D	]
94	5E	^
95	5F	_
96	60	`
97	61	a
98	62	b
99	63	c
100	64	d
101	65	e
102	66	f
103	67	g
104	68	h
105	69	i
106	6A	j

Deci- mal	Hex	ASCII
107	6B	k
108	6C	l
109	6D	m
110	6E	n
111	6F	o
112	70	p
113	71	q
114	72	r
115	73	s
116	74	t
117	75	u
118	76	v
119	77	w
120	78	x
121	79	y
122	7A	z
123	7B	{
124	7C	|
125	7D	}
126	7E	~
127	7F	DEL