Inductive
Read/Write Identsystem
CIS3

System - Manual
for
Read-Only Head CIT3PL..
Read/Write Head CIT3SX..

Ident. Nr. 071652

EUCHNER

EUCHNER GmbH + Co.
Industrie-Elektrik und -Elektronik

Kohlhammerstraße 16
D-70771 Leinfelden-Echterdingen

Telefon: 07 11 / 75 97-0
Telefax: 07 11 / 75 33 16

www.euchner.de
info@euchner.de

Contents

1. Overview

This Manual describes on the whole the communication between PLC resp. CNC control and the CIT3SX... read/write head.
The Protocol R transfer procedure serves to connect the CIT3SX1 read/write head to PLC or NC controls.

The transfer telegrams for the commands

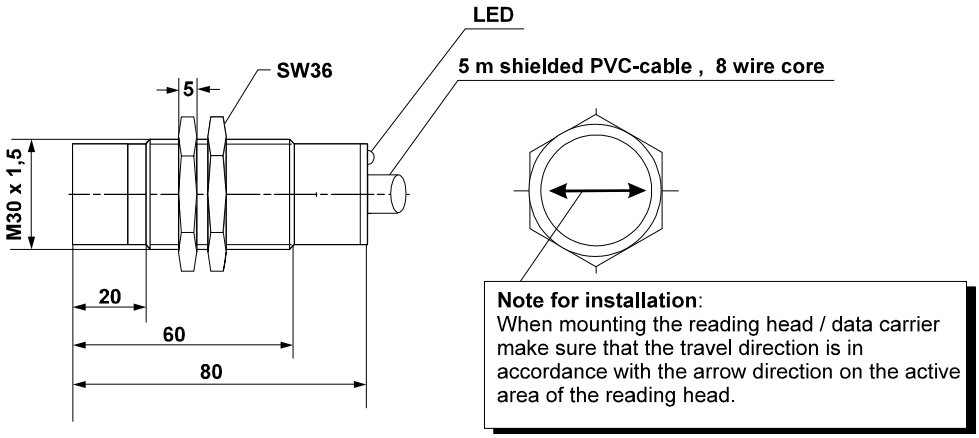
- Program (write) data carrier

- Read data carrier

are based on the 3964-R transfer procedure /1/

1.1. Cable- and terminal assignment

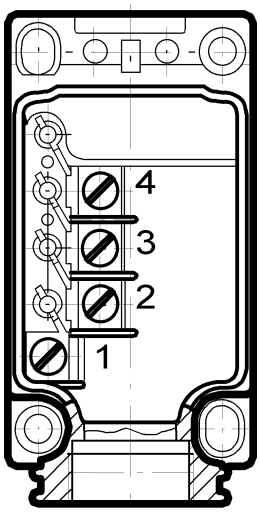
1.1.1 Read-Only Head CIT3PL...



Core wire colour	white	brown	green	yellow	grey	pink	blue	red	transp.
Signal	0V/GND	24V/UB	A	B	C	D	SKIP	STROBE	shield

Figure 1: Wiring diagram of the read-only head CIT3PL...

1.1.2 Read/Write Head CIT3SX...



Klemme
4 TxD
3 GND
2 RxD
1 24V

Figure 2: Terminal assignment read/write head CIT3SX1R1G05...
(basic unit with terminal cover open)

1.2 Connection example for read/write head CIT3SX...

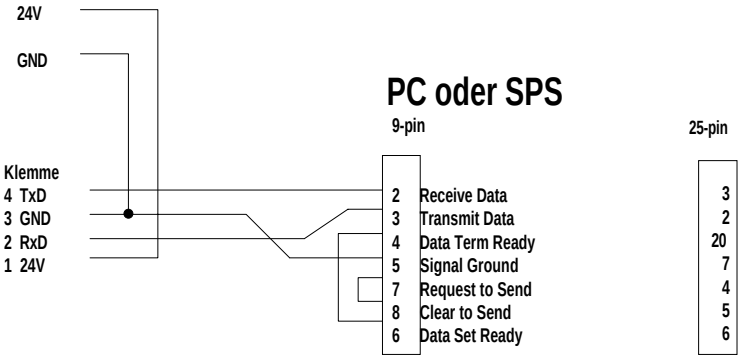
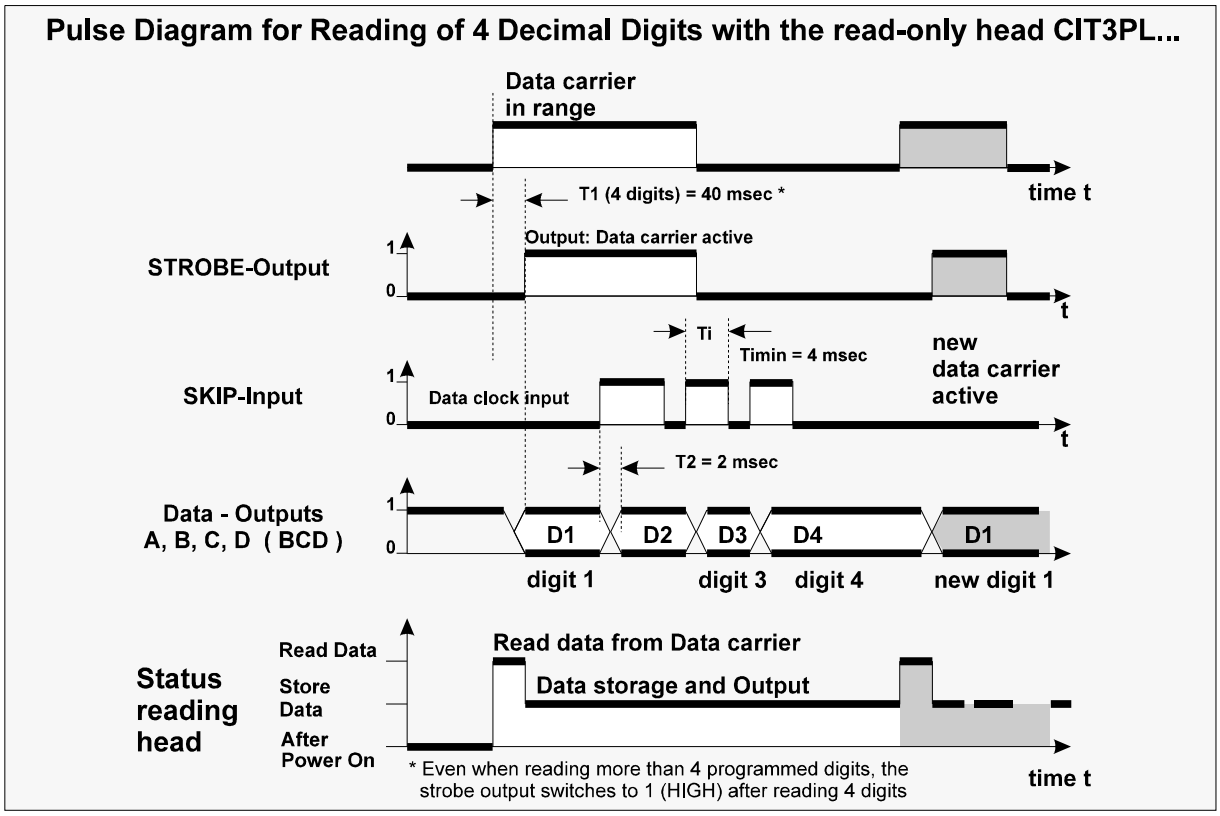


Figure 3

2. Notes on installation

- Do not connect the read/write head CIT3SX... as well as the read-only head CIT3PL... if it is still connected to the electrical power supply.
- The Read-Only head CIT3PL... should be mounted isolated, to avoid any interference's
- When mounting the facilities make sure that the travel direction is in accordance with the arrow direction on the active area of the reading head.

3. Pulse Timing Diagram for the Read-Only Head CIT3PL...



The above pulse timing diagram shows the time response of the input and output signals of the read-only head with respect to the PLC.

As soon as a data carrier is read in, the first digit is disposed at the data outputs of the read-only head. To switch the following digits to the data outputs, you have to give a HIGH-pulse to the Skip-input.
With every HIGH-pulse to the Skip-input, the digits are switched on.

The read-only head stores the coded information read from one data carrier until another data carrier has been read.

The storage function offers the following advantage:
At high relative speeds between Read head and data carrier, the dwell time of a data carrier in the read-only head's response range may, under certain circumstances, be shorter than the scan time of the PLC control. Owing to the storage characteristics of the Read head, the code can nevertheless be read by the PLC in the next scan time cycle.

The additional STROBE output, when at HIGH level, indicates that a data carrier is located in the read head's response range.
This output is set to HIGH as soon as the Read head has been able to read 4 decimal digits. If more than 4 digits are required, it is **not** possible to check with the STROBE output whether all required digits have been read in.
If any error should occur during reading, an error message F_{hex} is given at the point in the data string. All preceding data is valid.

4. Basic telegram structure without response telegram

Each command and any related data blocks are transferred within the telegram frame in accordance with Procedure 3964-R from and to the read/write head (Figure 4).

With Protocol 3964-R, the relevant receiver acknowledges the received telegram by returning an acknowledgement character (DLE). In the case of negative acknowledgement (NAK), the entire protocol is repeated. If it is not possible to transfer the protocol error-free after a total of six attempts, the operation is aborted.

4.1 Basic command structure

	Description	Byte No.	C O N T E N T S ASCII	Acknowledgement of the receiver + -
Connection set-up	3964-R procedure start		STX	
				DLE NAK
Telegram data max. 128 Bytes (telegram core)	Number of telegram Bytes	0		
	Command identification	1 2	T or R command	
	*Header address *)*	3	01h	
	User data description	4 5 6	Start address Start address Number of data items	
	User data	7 to n		
Connection cleardown	3964-R Procedure end		DLE ETX BCC	
				DLE NAK

Figure 4 : Basic command structure

\) *For downward compatibility with the CIS2 identification system. The content of the header address must always be 1 in the case of CIS3.*

4.2 Special features of the 3964-R data transfer protocol /1/

The 3964-R data transfer protocol is a comparatively reliable program for electronic data interchange between a control and a connected peripheral since data transfer is handled with a standardised protocol.

On controls with integrated 3964-R driver (see /1/ for instance), it is **not** necessary for the user to bother with the details of connection set-up and cleardown resp. data integrity.
It suffices to transfer the telegram core to the 3964-R driver via the program.

On controls without 3964-R driver or if a read/write head is connected to PC's, the user must, however, also program the connection set-up and cleardown and the retry attempts.

4.2.1 Basic information on data transfer procedures with protocol /1/

Numerous conventions must be agreed for a data transfer procedure: codes, operating modes, transfer speeds and the algorithmic transfer sequence. The stipulation of the algorithmic sequence is referred to as **transfer protocol** (*protocol for short*). A transfer protocol generally defines the following phases of data transfer:

- Request from A to B for data interchange
- Data interchange
- End of data interchange

The transfer protocol is essentially a matter of the control. This means that it handles data transfer independently on the basis of this protocol.

4.2.2 The 3964-R transfer protocol /1/

Unlike non-protocol-based data transfer procedures, 3964-R is a data transfer procedure with protocol. This means that the actual data to be transferred is enclosed in specific control characters. The 3964-R driver allows comparatively reliable data transfer by virtue of the fact that the receiver must first signal to the transmitter that it is ready to receive (connection set-up) and, after data interchange, must acknowledge correct reception. Data integrity is enhanced by an additional block check character with the 3964-R transfer protocol.

The 3964-R driver interprets the following control characters:

- DLE (10_{hex}) **D**ata **L**ink **E**scape
- STX (02_{hex}) **S**tart of **Tex**t
- NAK (15_{hex}) **N**egative **Ac**k**k**nowledgement
- ETX (03_{hex}) **E**nd of **Tex**t

With the 3964-R transfer protocol, a **block check character *(BCC for short)*** is transmitted for data integrity at the end of each data block. The block check character BCC is the **even longitudinal parity (EXORing of all data Bytes)** of a transmitted or received block. Generation **starts** with the **first user data Byte (first Byte of the telegram)** after connection set-up and **ends after** characters **DLE** and **ETX** on connection cleardown.

4.2.2.1 Control sends /1/

The control sends the control character STX in order to set up the connection. If the peripheral responds before expiry of the acknowledgement delay time (QVZ, typically: 2 seconds) with control character DLE, the transfer protocol reverts to Transmit mode. If the peripheral responds with control character NAK or any other character (apart from DLE) or if the acknowledgement delay time elapses with no response, this means that connection set-up has failed. The procedure is aborted after a total of 6 unsuccessful attempts (specification of the 3964-R protocol).
If connection set-up is successful, the user information characters contained in the control's transmit buffer are transmitted to the peripheral at the selected transfer speed. The peripheral monitors the interval between the incoming characters. The interval between two characters may not exceed the character delay time (ZVZ, typically: 100 ms).

Each control character DLE (10_{hex}) contained in the user information **must** be transmitted **twice** so that the communication partner recognises that the data is user data and not the control character DLE. **(DLE doubling)**.

After transmission of the user data, the control appends the following characters as **end identifier**: DLE, ETX, BCC

The control then waits for an acknowledgement character from the peripheral. If the peripheral sends control character DLE within the acknowledgement delay time (QVZ, typically: 2 seconds), the data block has been accepted error-free.
By contrast, if the peripheral responds with control character NAK or any other character or if the acknowledgement delay time elapses with no response, the control starts transmission again with connection set-up STX. The procedure is aborted and the control sends the control character NAK to the peripheral after a total of 6 unsuccessful attempts (specification of the 3964-R protocol).

If the peripheral sends control character NAK during a running transmission, the control aborts the block and repeats it in the manner described above. In the case of any other character, the control initially waits for the character delay time (ZVZ) to elapse and then sends control character NAK in order to set the peripheral to idle state. The control then starts transmission again with connection set-up STX.

4.2.2.2 Control receives /1/

If the control receives control character STX from the peripheral in idle state, it responds with DLE. If the control receives another character (apart from STX) in idle state, it waits for the character delay type (ZVZ, typically: 100 ms) to elapse and then sends the control character NAK. After each character, the next character is awaited during the character delay time (ZVZ). If the character delay time elapses without reception, control character NAK is sent to the peripheral.

If the control detects character string DLE ETX BCC, it terminates reception. It compares the received block check character BCC with the internally generated longitudinal parity. If the block check character is correct and no other reception error has occurred, the control sends control character DLE. If the BCC is errored, control character NAK is sent to the peripheral. A retry is then awaited. If it is not possible to receive the block error-free even after a total of 6 attempts (specification of the 3964-R protocol), or if the retry is not started by the peripheral within the block waiting time of 4 seconds, the control aborts reception.

If transmission errors (lost character, frame error, parity error) occur during reception, reception continues through to connection cleardown and control character NAK is then sent to the peripheral. A retry in the manner described above is then awaited.

4.2.3 Summary of the most important points

- **DLE doubling:**
  In order for the control to be able to distinguish between control character DLE and any randomly occurring DLE as user information character, **a further DLE must be sent in the case of a DLE as user information character**. This means that if a byte with ASCII value DLE (10_{HEX}) occurs within the telegram core, this character must be transmitted again so that it is not interpreted by the distant station as a control character for connection cleardown.

- **Block check character (BCC):**
  A block check character is sent at the end of each data block for data integrity. The block check character BCC is the **even longitudinal parity (EXORing of all data Bytes)** of a transmitted or received block. Generation **starts** with the **first user data Byte (first Byte of the telegram)** after connection set-up and **ends after** characters **DLE** and **ETX** on connection cleardown.

- **Retry attempts in the case of errors:**
  If an error occurs for any reason during data transfer, **a total of 6 attempts** are made to transfer the data correctly.

Please refer to the program example in the Annex of this Manual for further details.

5. Commands for writing and reading the data carrier

Read and write operations are always initiated by the higher-level control (NC, PLC) with a "command telegram".

The read/write head then sends a response telegram to the control.

Control	Read/write head CIT3SX...
Command telegram ⟶	
	Response telegram ⟵

5.1 Write operation

The data carrier must be in front of the read/write head in the case of this command and may be removed from the active area only after reception of the response telegram.

Command telegram (telegram core, PLC → CIT3SX , see also Figure 5):

TP (read/write head address) (start address) (number of Bytes user data) (user data)

Response telegram (telegram core, CIT3SX → PLC, see also Figure 6):

RF (read/write head address) (0.0) (Error No.)

Content or possible value range

Byte No.	Description	CONTENTS ASCII	HEX	Decimal
0	Number of telegram Bytes			8 ... 23
1	Command	T	54	84
2	identification	P	50	80
3	Read/write head address ***)**		01	01
4	Start address of the		00	00
5	user data		0 ... 5Fh	0 ... 95
6	Number of Bytes of the user data		1 ... 10h	1 ... 16
7 ... 22	User data	ASCII or HEX resp. BCD (code-transparent) ****)**		

Figure 5: Command telegram "Write data carrier" (telegram core)

)** ***For downward compatibility with the CIS2 identification system. The content of the header address must always be 1 in the case of CIS3.

****)** ***If read/write head CIT3SX.. is used in conjunction with read head CIT3PL, please note that a Byte comprises 2 BCD digits (0 ... 9) when programming the user data.***

Byte No.	Description	CONTENTS ASCII	HEX	Decimal
0	Number of telegram Bytes			7
1	Command	R	52	82
2	identification	F	46	70
3	Read/write head address *		01	01
4	Padding data		00	00
5			00	00
6	Error number		Error No. **	

Figure 6: Response telegram "Write data carrier" (telegram core)

*) *For downward compatibility with the CIS2 identification system. The content of the header address must always be 1 in the case of CIS3.*

**) Error number f = 00: No error
02: Data carrier not in active area
03: Read operation aborted
05: Write operation aborted, move data carrier out of active area

5.2 Read operation

Command telegram (telegram core, PLC → CIT3SX , see also Figure 7):
TL (read/write head address) (start address) (number of Bytes user data)

Response telegram (telegram core, CIT3SX → PLC, see also Figure 8 or Figure 9):
There are two different possible responses for this command

1. RL (read/write head address) (start address) (number of Bytes user data) (user data)
or
2. RF (read/write head address) (0.0)(error No.)

The response telegram RL (see also Figure 8) means error-free reception of the data.

If it is not possible to read a data carrier, an RF response telegram is received (see also Figure 9). The error number then indicates the cause of the error.

Byte No.	Description	Content or possible value range CONTENTS ASCII	HEX	Decimal
0	Number of telegram Bytes			7
1	Command	T	54	84
2	identification	L	4C	76
3	Read/write head address *)		01	01
4	Start address of		00	00
5	user data		0 ... Fh	0 ... 15
6	Number of Bytes of user data		1 ... 10h	1 ... 16

Figure 7: Command telegram "Read data carrier" (telegram core)

Byte No.	Description	CONTENTS ASCII	HEX	Decimal
0	Number of telegram Bytes			8 ... 23
1	Command	R	52	82
2	identification	L	4C	76
3	Read/write head address *)		01	01
4	Start address of		00	00
5	user data		0 ... Fh	0 ... 15
6	Number of Bytes of user data		1 ... 10h	1 ... 16
7 ... 22	User data	ASCII or HEX resp. BCD (code-transparent)		

Figure 8: Response telegram "Read data carrier" (telegram core)

Byte No.	Description	CONTENTS ASCII	HEX	Decimal
0	Number of telegram Bytes			7
1	Command	R	52	82
2	identification	F	46	70
3	Read/write head address *)		01	01
4	Padding data		00	00
5			00	00
6	Error number		Error No.	

Error number f = 02: Data carrier not in active area

Figure 9: Response telegram "Read data carrier" (telegram core)

*) *For downward compatibility with the CIS2 identification system.*
The content of the header address must always by 1 in the case of CIS3.

5.3. Command overview:

Description	Command telegram	Response telegram
Program data	(Num. Tele-Bytes) **TP** (Read/write head address *) (Start address) (Number of Bytes user data) (User data.....)	(Num. Tele-Bytes) **RF** (read/write head address *) (Error No.)
Read data carrier in active area	(Num. Tele-Bytes) **TL** (Read/write head address *) (Start address) (Number of Bytes user data)	(Num. Tele-Bytes) **RL** (read/write head address *) (Start address) (Number of Bytes user data (User data) *or* (Num. Tele-Bytes) **RF** (read/write head address *) (Error No.)

**)* *For downward compatibility with the CIS2 identification system.*
The content of the header address must always by 1 in the case of CIS3.

6. Error messages

02h Data carrier not in active area
03h Read operation aborted
04h Error while programming or while check-reading the data carrier
05h Write operation aborted, remove data carrier from active area
16h Data length greater than 16 Bytes

7. Example of a PC-Program in Q-BASIC under MS-DOS

```
' ******* Demoprogramm für EUCHNER S/L System ******************************'
DECLARE FUNCTION Lesebefehl$ (Start!, Anzahl%, Kopfnr%)
DECLARE FUNCTION Schreibbefehl! (Start!, Anzahl%, Kopfnr%, Daten$)
DECLARE FUNCTION Telegrammsenden! (Telegramm$, AnVersu!)
DECLARE FUNCTION Antworttelegramm$ (AnVersu!)
DECLARE FUNCTION BCCErmittlung$ (Telegramm$)
DECLARE FUNCTION Datenauswerten$ (A$)
DECLARE SUB Fehleranzeigen (A$)

REM ***** Einstellung fuer COM1 *****
REM OPEN "COM1:9600,N,8,1,CD,CS,DS,OP0,RS,TB0,RB0" FOR RANDOM AS #1
REM OUT &H3FB, &H1B  '8 Datenbit 1 Stopbit Even Parity für COM1

REM ***** Einstellung fuer COM2 *****
OPEN "COM2:9600,N,8,1,CD,CS,DS,OP0,RS,TB0,RB0" FOR RANDOM AS #1
OUT &H2FB, &H1B  '8 Datenbit 1 Stopbit Even Parity für COM2

REM ***** Verbindungsabbau **********
END$ = CHR$(&H10) + CHR$(&H3)

DO
  CLS
  PRINT "Datenträger lesen              :1"
  PRINT "Datenträger schreiben          :2"
  PRINT "Daten ausdrucken               :3"
  PRINT "Beenden                        :4"
  PRINT "  "
  PRINT "  "
  INPUT "Auswahl :?", A

  SELECT CASE A

  REM ********* Datentraeger lesen *************
  CASE 1
    INPUT "Datenträger Startadresse           :", A!
    INPUT "Anzahl der Daten 0 bis 16          :", B%
    INPUT "Kopfadresse                        :", C%
    INPUT "Schleife J/N                       :", D$
SCHLEIFE:
    Returnwert$ = Lesebefehl(A!, B%, C%)
    IF Returnwert$ = "0" THEN
      PRINT "Keine gültigen Daten eingelesen"
    ELSE
      PRINT Returnwert$
    END IF


    FOR i = 0 TO 10
    NEXT i

    IF D$ = "J" THEN GOTO SCHLEIFE
    INPUT H
```

```
CASE 2
  REM ********* Datentraeger schreiben *************
    INPUT "Datenträger Startadresse            :", A!
    INPUT "Daten max. 16 Bytes                 :", D$
    INPUT "Kopfadresse                       :", C%
    B% = 16
    REM  B% = LEN(D$)
    Returnwert = Schreibbefehl(A!, B%, C%, D$)

  REM ** Datentraegerinhalt nach lesen ausdrucken **
  CASE 3
    IF Returnwert$ = "0" THEN
      PRINT "Keine gültigen Daten eingelesen"
    ELSE
      OPEN "LPT1:" FOR OUTPUT AS #2
      PRINT #2, Returnwert$
      INPUT H
      CLOSE #2
    END IF
  CASE 4
    REM ******* Programmende ************************
    END
  CASE ELSE
    INPUT "Fehlerhafte Eingabe", H
  END SELECT

LOOP UNTIL 0

CLOSE #1

END

'*
'******************** 3964R Antworttelegramm einlesen **********************
'*
FUNCTION Antworttelegramm$ (AnVersu)
END$ = CHR$(&H10) + CHR$(&H3)
STX$ = CHR$(2)
DLE$ = CHR$(&H10)
NAK$ = CHR$(&H15)
Wiederholzaehler = 0
Startantworttelegr:
A$ = INPUT$(1, #1)                    'STX von Bussystem einlesen

PRINT #1, DLE$;                       'DLE senden

Wiederholzaehler = Wiederholzaehler + 1

A$ = INPUT$(1, #1)                    'Telegrammlänge einlesen
B = ASC(A$) - 1
BCCWE = ASC(A$)

IF A$ = DLE$ THEN                       'DLE Vedopplung nach 3964R
  D$ = INPUT$(1, #1)
  BCCWE = BCCWE XOR ASC(D$)
END IF

FOR i = 1 TO B                        'Telegramm einlesen
  C$ = INPUT$(1, #1)
```

```
BCCWE = BCCWE XOR ASC(C$)

  IF C$ = DLE$ THEN                    'DLE Vedopplung nach 3964R
    D$ = INPUT$(1, #1)
    BCCWE = BCCWE XOR ASC(D$)
  END IF
  A$ = A$ + C$
NEXT i
C$ = INPUT$(1, #1)                     'DLE einlesen
BCCWE = BCCWE XOR ASC(C$)
C$ = INPUT$(1, #1)                     'ETX einlesen
BCCWE = BCCWE XOR ASC(C$)
C$ = INPUT$(1, #1)                     'BCC einlesen
IF Wiederholzaehler = AnVersu THEN         'nach mehrmaligen ungültigen Versuch, Verbindung abbrechen
  PRINT "Kein gültiges Antworttelegramm"
  PRINT #1, NAK$
  GOTO Abbruch
END IF
IF BCCWE <> ASC(C$) THEN                'Vergleich ob Telegramm-BCC richtig ist
  PRINT "BCC-Fehler", BCCWE, ASC(C$)
  PRINT #1, NAK$
  GOTO Startantworttelegr               'Erwartet Protokollwiederholung
END IF
PRINT #1, DLE$;                        'positiv quittieren

Antworttelegramm$ = A$
Abbruch:
END FUNCTION

'*
'************************ Ermittlung des BCC ******************************
'*
FUNCTION BCCErmittlung$ (Telegramm$) STATIC
BCC = 0
FOR i = 1 TO LEN(Telegramm$)
  BCC = BCC XOR ASC(MID$(Telegramm$, i, 1))
NEXT i
BCCErmittlung$ = CHR$(BCC)
END FUNCTION

'*
'***************** Daten aus dem Antworttelegramm ausfiltern ***************
'*
FUNCTION Datenauswerten$ (A$)
IF MID$(A$, 3, 1) = "K" THEN PRINT "Daten korrigiert"
Z = ASC(MID$(A$, 7, 1))
B$ = MID$(A$, 8, Z)
Datenauswerten$ = B$
END FUNCTION

'*
'********************* Fehler oder Statusermittlung ************************
'*
SUB Fehleranzeigen (A$)
PRINT "Fehler Nr. : ";
B$ = MID$(A$, 7, 1)
PRINT ASC(B$)
INPUT X
END SUB
```

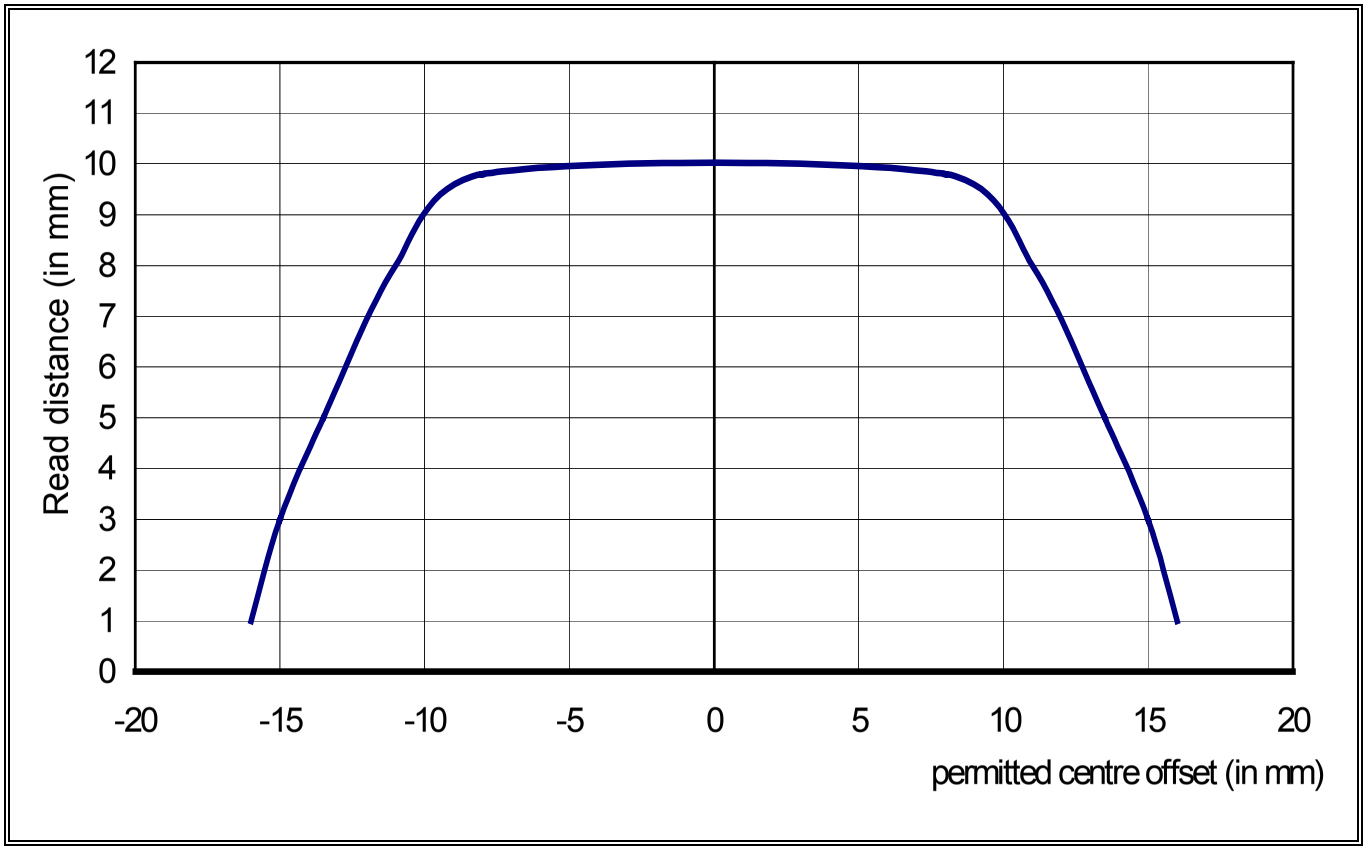
```
'
*
'************************** Datenträger lesen ****************************
'*
FUNCTION Lesebefehl$ (Start!, Anzahl%, Kopfnr%)

Starthi% = Start \ 256              'Startadresse in high und low Teil aufteilen
Startlo% = Start MOD 256

Telegramm$ = CHR$(7) + "TL" + CHR$(Kopfnr%) + CHR$(Starthi%) + CHR$(Startlo%) + CHR$(Anzahl%)

IF Telegrammsenden(Telegramm$, 6) <> 0 THEN GOTO ende1

A$ = Antworttelegramm(6)            'Antworttelegramm einlesen

PRINT "Antworttelegramm auswerten"
                          'Antworttelegramm auswerten
IF MID$(A$, 3, 1) = "F" THEN Fehleranzeigen (A$): Lesebefehl$ = "0"
IF MID$(A$, 3, 1) = "L" THEN Lesebefehl$ = Datenauswerten(A$)
IF MID$(A$, 3, 1) = "K" THEN Lesebefehl$ = Datenauswerten(A$)

ende1:
END FUNCTION

'*
'****************** Daten auf Datenträger schreiben ************************
'*
FUNCTION Schreibbefehl (Start!, Anzahl%, Kopfnr%, Daten$)

Starthi% = Start \ 256              'Startadresse in high und low Teil aufteilen
Startlo% = Start MOD 256

Telegramm$ = CHR$(LEN(Daten$) + 7) + "TP" + CHR$(Kopfnr%) + CHR$(Starthi%) + CHR$(Startlo%) +
CHR$(Anzahl%) + Daten$

A = Telegrammsenden(Telegramm$, 6)
IF A <> 0 THEN GOTO ende

PRINT "Programmiertelegramm gesendet"

A$ = Antworttelegramm(6)            'Antworttelegramm einlesen

PRINT "Antworttelegramm auswerten"
                          'Antworttelegramm auswerten
IF MID$(A$, 3, 1) = "F" THEN Fehleranzeigen (A$)

Schreibbefehl = 0
ende:

END FUNCTION

'*
'********************* 3964R Telegramm senden ******************************
'*
FUNCTION Telegrammsenden (Telegramm$, AnVersu!)
END$ = CHR$(&H10) + CHR$(&H3)
STX$ = CHR$(2)
DLE$ = CHR$(&H10)
```

```
Wiederholzaehler = 0

Protokolstart:

PRINT #1, STX$;                 'STX ausgeben
A$ = INPUT$(1, #1)               'DLE einlesen

IF Wiederholzaehler = AnVersu! THEN
  PRINT "Kein Verbindungsaufbau": INPUT " ", Q
  Telegrammsenden = -1
  GOTO ENDE2
END IF

IF A$ <> DLE$ THEN
  Wiederholzaehler = Wiederholzaehler + 1
  GOTO Protokolstart
END IF

telegraus$ = ""
FOR i = 1 TO LEN(Telegramm$)
  C$ = MID$(Telegramm$, i, 1)

  IF C$ = DLE$ THEN telegraus$ = telegraus$ + C$ 'DLE Verdopplung nach 3964R
  telegraus$ = telegraus$ + C$
NEXT i

'       Kommandotelegramm senden

PRINT #1, telegraus$ + END$ + BCCErmittlung$(telegraus$ + END$);

A$ = INPUT$(1, #1)                      'DLE von Bussystem einlesen
IF Wiederholzaehler = AnVersu! THEN PRINT "negative Quittierung": INPUT " ", Q: Telegrammsenden = -1:
GOTO ENDE2
IF A$ <> DLE$ THEN Wiederholzaehler = Wiederholzaehler + 1: GOTO Protokolstart
PRINT "Anforderung gesendet"
Telesenden = 0:
ENDE2:

END FUNCTION
```

8. Maximum permitted centre offset of the CIT3PL... read only head

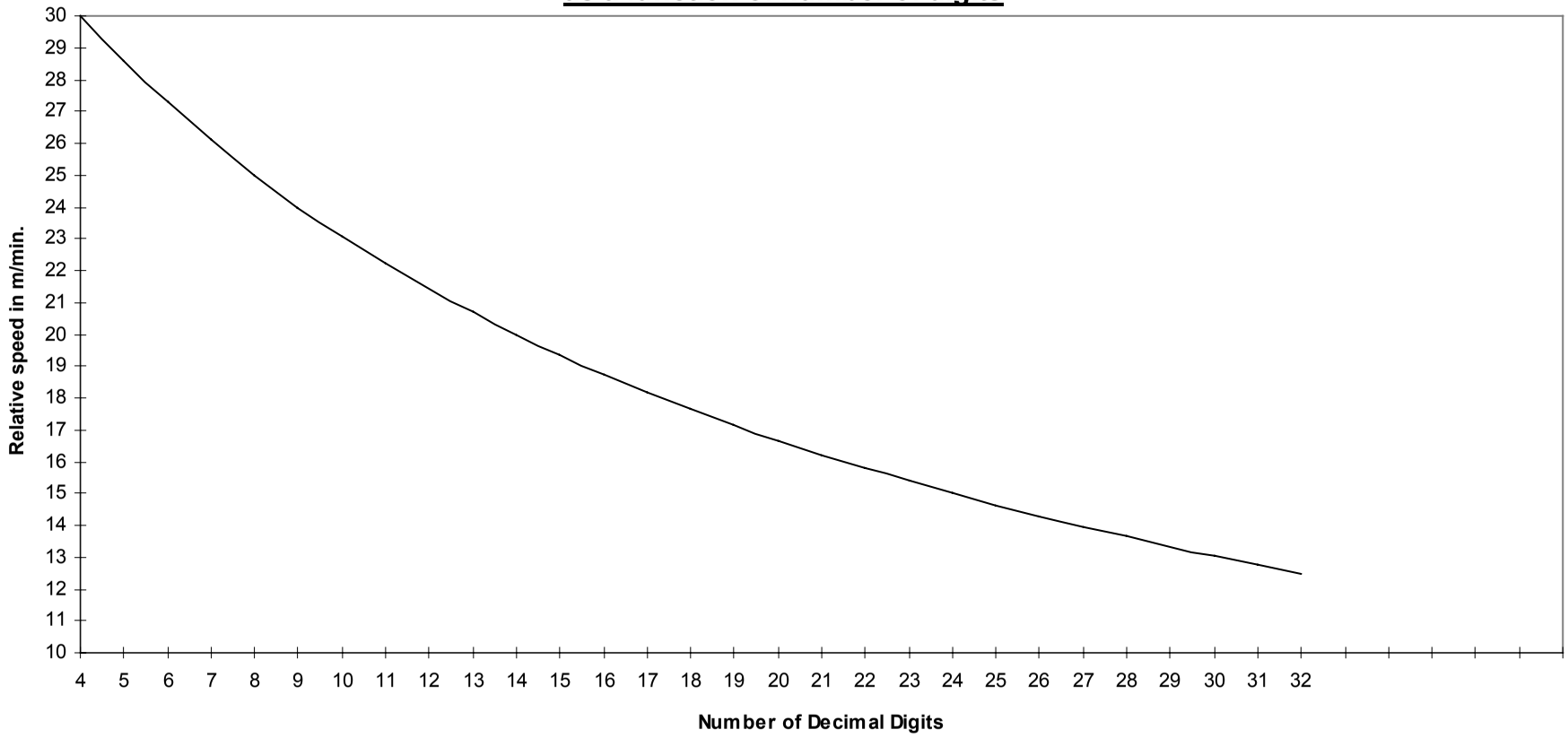
Test conditions: Ambient temperature: 25 °C

Units used: Read-only head CIT3PL1N30-5000 (Order No. 040085)
Data carrier CIS3P35X16SH01KH (Order No. 040045)



9. Maximum relative speed with the CIT3PL... read only head

max. relative read speed
as a function of number of digits



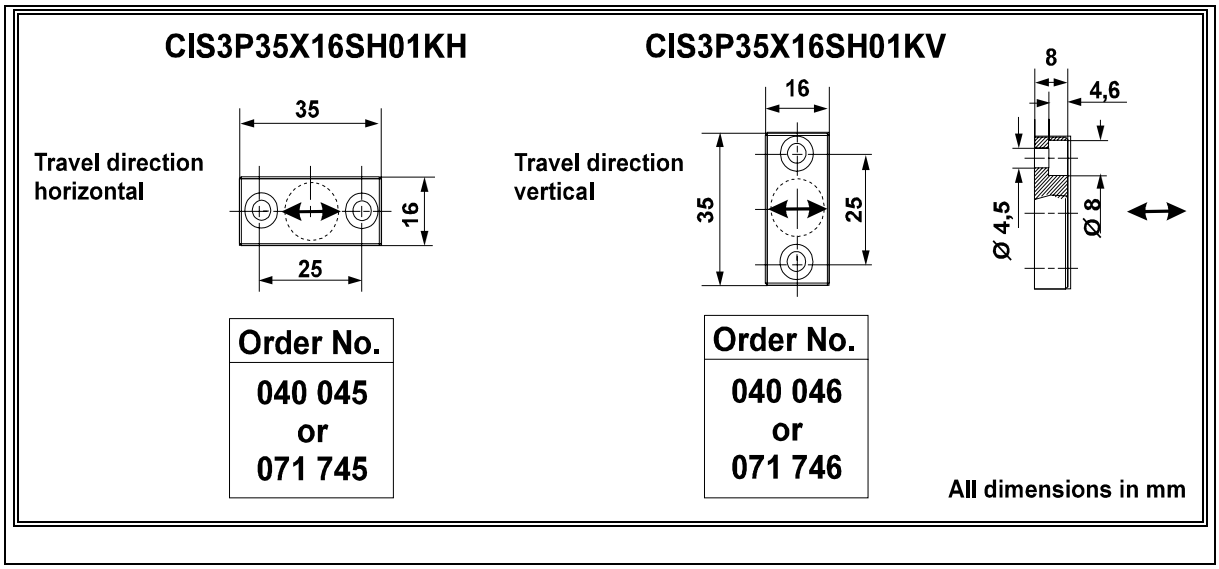
Note: Since *at least* 4 decimal digits must be read, the diagram also only starts at the 4th decimal digit

10. Installation instructions for CIS3 data carriers

Please comply with the following points when installing the data carrier in order to comply with the technical specifications contained in the data sheets for the CIS3 ident system even in the case of **flush installation in metal**:

- On data carrier, Type CIS3P35X16SH01KH (Order No. 040045 or 071745), there **must** be a **minimum clearance of 5 mm per side** with respect to the surrounding metal on the long side (35 mm side). The short side (16 mm side) may directly adjoin the surrounding metal.

- Data carrier, Type CIS3P35X16SH01KV (Order No. 040046 or 071746) can be installed flush in metal with no problems.
  No minimum clearance with respect to the surrounding metal needs to be complied with on any side.



11. Technical data

CIT3PL1N30-5000

Operating voltage:	15 - 28 V
Current consumption:	100 mA (max.)
Load current per output:	30 mA (max.)
Storage temperature	0 - 80°C
Ambient temperature	0 - 50°C
Read distance:	0 - 12 mm (see also page 18)
Centre offset:	nominal ±10 mm (see also page 18)
Response time (for 4 decimal digits):	40 msec. (max.)
Relative speed:	max. 30m/min. (see also page 19)

CIT3SX1R1G05KS:

Operating voltage:	15 - 25 V
Current consumption:	140 mA (max.)
Storage temperature	0 - 80°C
Ambient temperature	0 - 50°C
Type of connection:	Screw terminals
Line length (RS232):	5 m (max.)
Read distance:	0 - 12 mm
Write distance (Static writing):	0 - 6 mm
Write time	
4 decimal digits:	230 ms (max.)
32 decimal digits:	420 ms (max.)

The following values are set for the serial interface (V24, RS 232):

1 Start Bit
8 Data Bits
1 Parity Bit (even parity)
1 Stop Bit

Baud rate: 9600 baud

Bibliography:

/1/ SIEMENS

Manuals :

Connection components for S5 controls:

Communication processor CP 521 SI

Designation :	*SIEMENS Order No.*
Device Manual CP 521 SI	GES5 998 - 1 UD 11

Communication processor CP 523

Designation :	*SIEMENS Order No.*
Device Manual CP 523*(d/e/f/s/i)*	GES5 998 - 0 DD *d* 1

Communication processor CP 544

Designation :	*SIEMENS Order No.*
Device Manual CP 544*(d/e/f)*	GES5 998 - 2 DB *d* 1

/2/ Link, W.

Coding and code integrity on programmable data carrier IDENT´88 , Sindelfingen