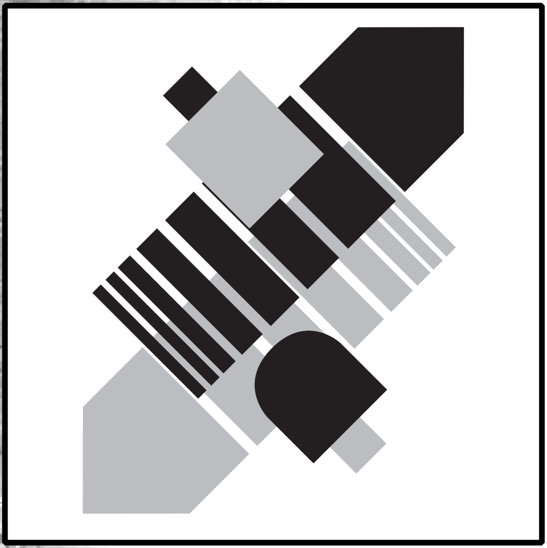
FACTORY AUTOMATION

# Manual

## IVT-P and IWP-12K-P

Read/Write Unit and
Read/Write Wand



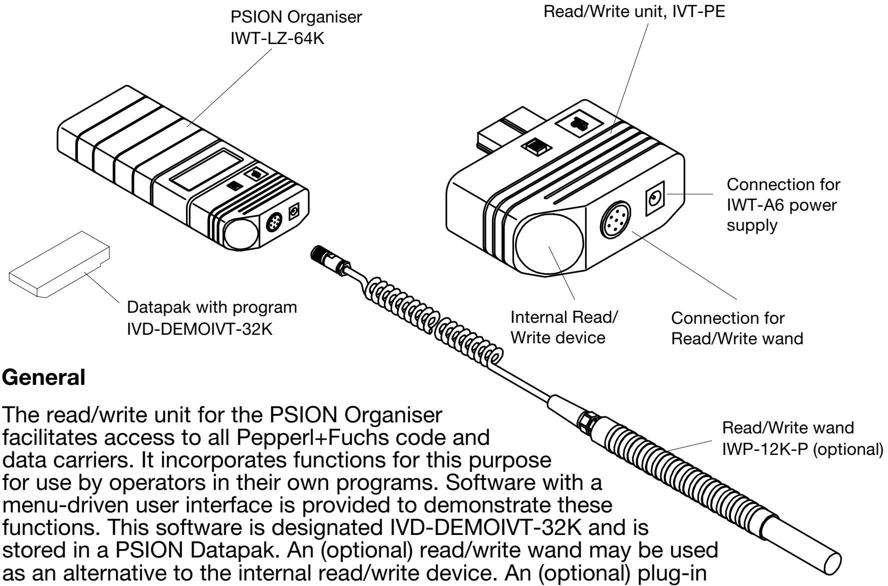
PEPPERL+FUCHS

We at Pepperl+Fuchs recognise a duty to make a contribution to the future.
For this reason, this printed matter is produced on paper bleached without the use of chlorine.

# Operating instructions

Page

Subject to reasonable modifications due to technical advances.



## 1 General

The read/write unit for the PSION Organiser facilitates access to all Pepperl+Fuchs code and data carriers. It incorporates functions for this purpose for use by operators in their own programs. Software with a menu-driven user interface is provided to demonstrate these functions. This software is designated IVD-DEMOIVT-32K and is stored in a PSION Datapak. An (optional) read/write wand may be used as an alternative to the internal read/write device. An (optional) plug-in RS232 Comms Link module can be fitted to provide a serial RS232 interface.

Order code IVT-P covers the following items:

| | |
|---|---|
| - PSION Organiser LZ64 | IWT-LZ-64 |
| - Read/write unit | IVT-PE |
| - Datapak with program | IVD-DEMOIVT-32K |
| - Power supply | IWT-A6 |
| - Manual | IVT-P/IWP-12K-P |

The following are available as options:

| | |
|---|---|
| - Read/write wand | IWP-12K-P |
| - RS232 Comms Link | WT-1R1D |

## 1.1 Power supply

The PSION Organiser is powered by a battery, with or without an additional power supply unit. The 9V battery must, therefore, first be inserted in the Organiser in all cases. The use of alkaline batteries is recommended (e.g. VARTA - Alkaline Type 4022), as weaker types may lead to data losses. Using the Organiser without a battery will result in the loss of all data and settings when the power is disconnected.

**Note:** Refer to the section on the battery in the operating manual for the PSION Organiser.

## 2 Peripheral devices and Datapaks

In addition to the standard peripheral devices, other accessories are available from Pepperl+Fuchs. These are described on the following pages.

Subject to reasonable modifications due to technical advances.

## 2.1 Connecting the read/write unit to the PSION Organiser

Switch off the Organiser before connecting the read/write unit. Open the Organiser by operating the slide at the top, and insert the read/write unit until it clicks into place. Press the “ON/CLEAR” key on the Organiser twice to switch the unit on and to load the software. The main menu should now include the option “IVTP”. If it does not appear, press the “ON/CLEAR” key again.

## 2.2 Removing the read/write unit

Switch off the Organiser before removing the read/write unit. Now depress the catch and remove the read/write unit. The software will remain in memory. To clear it from memory, again press the “ON/CLEAR” key twice. The option “IVTP” will no longer appear in the main menu.

## 2.3 External read/write wand (optional)

To read or write to code or data carriers in inaccessible locations, a type IWP-P external read/write wand may be used. This is simply inserted into the flange type socket on the top of the read/write unit, and screwed into place. When a read/write wand is connected, the read/write circuit of the read/write unit is automatically disabled.

## 2.4 Datapak

There are two slots on the base of the Organiser for external memory modules. Remove the cover and insert the Datapak supplied. This contains the IVD-DEMOIVT-32K software.

**Note:** Avoid pressing any keys when inserting or removing a Datapak. For further information, refer to the section on storage media in the PSION Organiser operating manual.

## 2.5 RS232 Comms Link (optional)

This plug-in module enables the Organiser to be connected to another computer, or a printer, for example, via an RS232 interface. The Comms Link module is plugged into the socket on the top of the PSION Organiser in the same way as the read/write unit. Note that the Organiser must be switched off first. Open the Organiser by operating the slide at the top, and insert the Comms Link connector until it clicks into place. The connector fits into the socket one way only, so that the name PSION is visible on the top of the connector. Depress the “ON/CLEAR” key twice to switch the device on and load the auxiliary software. A new menu option, “COMMS”, will appear in the main menu.

**Note:** Detailed instructions will be found in the sections on connecting the Comms Link cable and linking to an IBM PC, in the manual for the Comms Link module.

If the Organiser is connected to an IBM PC/XT/AT or compatible PC via the Comms Link cable, the DOS communications software supplied must be installed and started before any files or programs can be transferred. All communication is then controlled by the PSION Organiser.

## 3 Software

The IVT-P read/write unit includes demo programs and additional functions which enable user-specific programs to be devised.

### 3.1 To call-up the software

To use the IVT-P software functions, first plug the IVT-PE read/write unit into the PSION Organiser. Switch on the Organiser and load the software by pressing the “ON” key twice. The option “IVTP” will now appear in the main menu. This menu option is used to call-up a program named “IVTP”, which may be either the user’s own application program, or the IVD-DEMOIVT-32K demo software. The latter is stored on the Datapak supplied.
If no such program is available (e.g. the Datapak containing IVD-DEMOIVT-32K has not been inserted), a message will be displayed to this effect, and a simple demo program will be started from the IWT-P.

### 3.2 Description of the IVD-DEMOIVT-32K software

#### General

The IVD-DEMOIVT-32K software is used to read and write to data carriers, and to read code carriers. The program is contained on a PSION Datapak, on delivery. The software is used in conjunction with PSION Organiser LZ64 and the IVT-PE read/write unit. The software can also be used to extend the capability of the PSION Organiser LZ64 and IWT-PE, so that all the functions of the IVT-PE are made available.

#### Description

The IVT-PE or IWT-PE read/write attachment must be plugged into the LZ64 PSION Organiser. The IVD-DEMOIVT-32K Datapak is inserted in Slot B: (see also the operating manual for the PSION Organiser).
The PSION Organiser is then switched on and the software loaded by pressing the “ON/ CLEAR” key twice. The option “IVTP” will now appear in the main menu.

#### To call-up the program

To activate the software, select the menu option “IVTP” with the arrow keys and press “EXE”. A Menu item can also be selected by pressing the key representing the first letter of the name of the item.
Press “ON” to leave a menu or an input without making any changes.
In the main menu, “ON” can be used to exit from the program (IVD-DEMOIVT-32K). In this case, however, an additional prompt must be accepted to confirm the command.

Subject to reasonable modifications due to technical advances.

## Menu structure

| Main menu | Level 2 | Level 3 | Level 4 |
|---|---|---|---|
| Settings | | | |
| | Range | | |
| | | Start address<br>Length<br>Read<br>Write | |
| | Data carriers | | |
| | | 256-bit<br>1K<br>64K<br>256K | |
| | Format | | |
| | | Address | |
| | | | Decimal<br>Hex |
| | | Data | |
| | | | ASCII<br>Hex |
| Read | | | |
| Write | | | |
| | Data<br>New_data<br>Reset<br>Restore (for 256-bit data carriers only) | | |
| MakeICC | | | |
| | Input<br>Copy<br>Read only | | |
| Off | | | |
| Version | | | |

## Editor:

When data is read, and before data is written, it is displayed in a so-called “full-screen editor”, where it can be edited (‘write’ data) or displayed only (‘read’ data).
In order to accurately analyse data, the arrow keys may be used to move the cursor around in the data display. Where more data is available than can be displayed within the screen, the contents scroll up or down automatically when the cursor reaches the top or bottom of the display.
The address of the character currently marked by the cursor is always shown in the first line of the display. The “Settings/Format/Address” menu can be used to define whether the address concerned is to be shown in decimal or hexadecimal notation. If HEX is selected, the address will be preceded by the character “$”.
The currently selected character is displayed again in the top right corner of the display, but in a format other than in the Editor itself. If the option HEX was selected in the “Settings/ Format/Address” menu, the data will be shown in hexadecimal notation in the Editor, and in ASCII format in the top right-hand corner of the display. If, however, the option ASCII was selected from the “Settings/Format/Address” menu, then the data will be shown in ASCII format in the Editor, and in hexadecimal, preceded by the character “$”, in the top right-hand corner of the display.

### Limits:

In IVD-DEMOIVT-32K, communication with all data carriers is such that the user receives the data one character at a time (one character = 8 bits). This results in the following address ranges for the various data carriers:

| Name | Size | Possible addresses from ... to ... | |
|---|---|---|---|
| IDC-... | 256 bits | 0...31 | |
| IDC-...-1K | 1K | 0...127 | (from Version 2.0) |
| IMC-40-64K | 64K | 0...8191 | |
| IMC-40-256K | 256K | 0...32767 | |

In this context, the term ‘address’ refers to the start and end address of the required data block. The start address and block length must be selected so that both addresses are within the address range shown above for the data carrier concerned. The end address is calculated as follows:

End address = Start address + block length – 1

The maximum block length possible for a single read or write command is 256.

### Entering data:

The “Write/Data” menu item allows data to be entered or modified via the Editor. If no Read command has yet been carried out, all positions will contain the byte “zero”. After a Read command has been carried out, however, the most recently read data is displayed.
If the data format is set to HEX, each byte consists of two hexadecimal characters. To enter a byte, therefore, two keys from “0” to “9” or from “A” to “F” must always be pressed. When entering ASCII data however, each byte consists of one character, and any letters, numbers or other characters may be entered.

### “EXE” key:

When all the data has been entered, the “EXE” key is pressed to execute the Write command. For this purpose, a data carrier must be within the detection range of the IVT-PE. The data and addresses to be written in are displayed briefly while the Write operation is in progress.
On completion of the Write operation, the Edit screen will again appear, with the same data. The data can now be modified, or the same Write command can be repeated by again pressing the “EXE” key.
In the same manner, after a Read command, the command can be repeated by pressing “EXE”. Similarly, in the event of a Read or Write error, the command can be repeated directly by pressing the “EXE” key.
The “ON” key is used to exit from the Editor.

### Entering a start address and block length:

The range to be accessed in the data carrier is selected in the “Settings/Range“ menu. To do this, the first address in the range is entered (start address) and the number of data items (length). The lowest possible address is “0”, and the minimum length is “1”. Selecting “Start address“ or “Length“ will result in the display of the currently valid setting followed by two “>>” characters. The displayed value can be retained by simply pressing the “EXE” key. To enter a new value, type in the new number in decimal notation or press the “$” character and enter a hexadecimal value. These two options are always available, irrespective of the option selected in the “Settings/Format/Address” menu.

### Saving the configuration automatically

When an exit is made from the software or the PSION Organiser is switched off while in the "IVTP" main menu, the current settings are automatically saved in a configuration file so that they are available when the software is next restarted. The configuration data is as follows:

- The selected data carrier type
- The start address and block length of the data record
- The display format for data and addresses (ASCII/hex or decimal/hex)

The configuration is stored in the internal memory A: under the name IVTPCFG.

## 3.3 Description of the menu options

| Menu option | Description |
|---|---|
| **Settings** | Sub-menu used to set the display format, select a data carrier and set the start address and block length of the required data record. |
| **Range** | Used to enter the start address and length of the data block to be read. |
| **Start address** | Used to enter the start address of the required data block. |
| **Length** | To input the length of the required data block. |
| **Read** | This option is the same as in the main menu, and is provided here for ease of use. |
| **Write** | This option is the same as in the main menu, and is provided here for ease of use. |
| **Data carrier** | Used to select the data carrier in use. |
| **256-bit** | Sets the data carrier mode for type IDC-.. data carriers. |
| **1K** | Sets the data carrier mode for type IDC-...-1K data carriers (from Version 2.0). |
| **64K** | Sets the data carrier mode for type IMC-40-64K data carriers. |
| **256K** | Sets the data carrier mode for type IMC-40-256K data carriers. |
| **Format** | Sets the display format for data and addresses. |
| **Address** | Used to define whether the addresses are to be displayed in hexadecimal or decimal format. |
| **Data** | Used to define whether the data is to be entered and displayed in ASCII or hexadecimal form. |
| **Read** | Used to read a data carrier using the selected settings. (For code carriers see "MAKEICC/Read only".) The data will then appear in the display in the display format selected. |

**Menu option** **Description**

**Write**
Sub-menu used to write to or erase a data carrier.

**Data**
Starts the Editor, for entry and modification of Write data. The data displayed is the data last read.

**New data**
Starts the Editor, for entry of write-data. All data bytes are preset to “00”.

**Reset**
Erases the data carrier after a prompt to confirm the command. All data is reset to zero.

**Restore**
This menu is available only when the data carrier mode is set to “256-bit” and is used to restore the parity information, e.g. in cases where a data carrier is no longer readable. Often this can be traced to a parity error, caused, for example, by the interruption of a Write operation. Faulty data in the data carrier cannot be corrected with this option, however.

**MakeICC**
Sub-menu used to read and write to data carriers which are to be read like code carriers **(see separate section).**

**Input**
Used to enter a fixed code which is then programmed into a data carrier, enabling it to be read like a code carrier.

**Copy**
Used to read a code carrier and subsequently write its contents into a data carrier.

**Read only**
Reads a code carrier and displays the code.

**Off**
Switches the Organiser off.

**Version**
Displays the current software version.

## 3.4 Main menu option MAKEICC

### Purpose of this program option

Code carriers used with an inductive fixed code system are allocated a unique code during manufacture. The manufacturer guarantees that each code number will be allocated once only. This is necessary for security-related applications, where protection against fraud is a consideration.
For other applications, this security measure is irrelevant. Code carriers with the same code may indeed be required, to replace faulty code carriers without the need to modify the control software. In some applications, control is simplified by the possibility of working with several identical codes.
In applications of this type the facility to combine code carriers and data carriers is a requirement. This means that one 'Read' command can be used to read both code and data carriers. Note that this is not possible with the read/write system, which has separate commands for reading data carriers and code carriers. A 'Read' command for data carriers cannot be used to read code carriers, and one for code carriers cannot read data carriers.
The program option "MAKEICC" can be used to programme data carriers with a code which can be read with the fixed code system in the same way as for a code carrier.

### Operating MAKEICC

After starting the "IVTP" program, the cursor keys (arrow keys) are used to move the cursor over the menu item "MAKEICC" and the program is started by pressing the "EXE" key. The "MAKEICC" menu will appear, with the following options:

"Input -- Copy -- Read only"

Using "Input", the code to be written in the data carrier can be entered from the keyboard. "Copy" enables an existing code carrier to be read and its code written to a data carrier. The "Read only" option reads a code carrier (or a data carrier programmed as a code carrier) and displays the code.

### Manual code entry

After selecting "Input" with the "EXE" key, the following prompt will appear:
"Enter code -- (7 digits) :"

The code to be entered should now be input, using data format 10. For details of the data formats, refer to the manuals for the relevant interface units. For example, enter the code "A123456" and press "EXE".
A total of 7 digits must be entered. An error message will be displayed if too few characters or invalid characters are entered. When the code has been entered correctly, the following message appears:

"Target: -- Data carrier in position? -- Press any key."

The data carrier is now held facing the Read/Write face of the IVT-PE and any arbitrary key is pressed to start the Write operation.
After successful programming, the code entered by the user will be displayed together with the programmed code. The two codes should, of course, be identical, for example:

"Verification: -- Input: A123456 -- Target: A123456"

Press any key to return to the "Input' prompt.

If the code Write operation was unsuccessful, the following message will appear:

“Write error -- Press any key”

Press any key to return to the “Input’ prompt.
To return to the previous menu press “EXE” or “ON”.

Codes startes with 00 or FF are ignored by all interface devices. This prevents these codes from being read as a result of interference in cases where no code carrier is present.

### Copy code carrier

The “Copy” menu item is selected from the “MAKEICC” sub-menu by use of the cursor keys and activated by pressing “EXE”. A prompt will then appear, inviting the user to move the code carrier to be copied into the Read/Write range. The message is as follows:

“Source: -- Code carrier in position? -- Press any key”

Holding the code carrier in front of the Read/Write face, press any key to read the code. If the code cannot be read, the following error message will be displayed:

“Read error ! -- Press any key -- Retry: "EXE"

Press any key to return to the ‘Read’ prompt. If the “EXE" key is pressed, an attempt will be made to read the code again.
When the code has been read successfully, the user will be prompted to hold the data carrier in front of the Read/Write face. The prompt is as follows:

“Target: -- Data carrier in position ? -- Press any key”

After a key has been pressed, the code will be programmed. If the Write operation was successful, the following will appear, for example:

“Verification: -- Input : A123456 -- Target: A123456”

In the event of a Write error, an error message will be displayed, and the Copy operation should be repeated. Press “S” to return to the sub-menu.

### Code carrier - Read-only

Before activation of the “Read only” function, a code carrier (or a data carrier programmed as a code carrier) must be positioned in front of the Read/Write face. After activation with the “EXE” key, the code will be read and displayed as in the following example:

“Code : A123456”

Press any key to return to the “MAKEICC” menu. If no code could be read (e.g. the code carrier was not placed in front of the read/write unit) the following error message will appear after an audible alarm signal:

“Read error ! -- Press any key. -- Retry: "EXE" ”

Press “EXE” to initiate a further attempt to read the code. Press any other key to return to the “MAKEICC” menu.

### Exit from program

From the “MAKEICC“ sub-menu, the “ON” key is pressed to exit from the program and return to the main menu.

Subject to reasonable modifications due to technical advances.

## 3.5 Description of the IWT-P demo software

If the program “IVTP” is not available in the PSION Organiser (e.g. the Datapak containing IVD-DEMOVIT -32K has not been inserted), selecting the menu option “IVTP“ will start the program “IWTP“.
This provides a simple demonstration of how to read ICC-... code carriers and how to write and read IDC-... 256-bit data carriers. The program has a menu with four options:

| Menu item | Description |
|---|---|
| DemoFix | A continuous attempt is made to read a code carrier. The code is displayed, or, if no code carrier can be identified, the error message “No Code“ will be displayed. Pressing the “S“ key terminates the Read operation. |
| DemoRW | Reads and displays the contents of a data carrier. The “S“ key terminates the Read operation. Pressing “W“ allows the entry of up to 32 characters, which will be written to the data carrier when the “EXE“ key is pressed. The Read operation will then continue. |
| Reset | Restores the data carrier parity information (see RD$). |
| Ver | Displays the software version of the IVT-PE. |

## 3.6 Description of the IVT-PE software

The programming language of the PSION Organiser (OPL) includes the following new commands for use with the IVT-PE.

| | |
|---|---|
| **CR$** | Read fixed code |
| **UHEX$** | Convert the read code for display in hexadecimal notation |
| **UDEC$** | Convert the read code for display in decimal notation |
| **RW$** | * Read word (IDC-...) |
| **WW$** | * Write word (IDC-...) |
| **RD$** | Restore parity data in IDC-... data carrier |
| **REIK$** | * Read word (IDC-...-1K, from Version 2.0) |
| **WEIK$** | * Write word (IDC-...-1K, from Version 2.0) |
| **HEXIN%** | Enter an integer in decimal or hexadecimal form |
| **WIM$** | Read byte (IMC-64K/256K) |
| **RIM$** | Write byte (IMC-64K/256K) |
| **BATS$** | Check battery status in the IMC-64K/256K |

The commands marked * communicate one word at a time with the associated data carrier, depending on its internal structure. One data word consists of 16 bits. By contrast, the

IVD-DEMOIVT-32K software, for example, operates byte by byte with each data carrier, one byte consisting of 8 bits:

| | | |
|---|---|---|
| Word 0 = | Byte 1 | Byte 0 |
| Word 1 = | Byte 3 | Byte 2 |
| Word 2 = | Byte5 | Byte 4 |
| etc. ... | ... | ... |

One word consists of two consecutive bytes. For this reason, the word addresses (e.g. Word 0, Word 1 etc.) are not identical to the byte addresses (Byte 0, Byte1, Byte 2 etc.). The sequence of bytes within a word is as shown above.

## 3.7 Detailed description of functions

### CR$:

This function is called-up without parameters. It reads the 64 bits of a code carrier and returns the 28 bits of user-relevant data as a binary-packed string with a length of four characters. The same format should be used when storing the code in a databank. If it was not possible to read the code carrier, an empty string will be returned.

Example:

```
CR:                             REM Program name
AT 1,2                          REM Positions cursor on screen
PRINT “CR$=>”,CR$:              REM Displays result
GET                             REM Wait for keystroke
```

The result still needs to be converted into readable form (see UHEX$ and UDEX$).

### UHEX$:(Code$)

This function converts a binary-packed character string into a readable format suitable for output. The parameter it expects is a string with a length of four characters, containing the code in binary-packed format. The first 12 bits of the code are converted into hexadecimal format and the next 16 bits into decimal. The code is then returned as a string with a length of seven characters.

Example:

```
UHEX:                           REM Program name
LOCAL CODE$(4)                  REM Declaration of variables
AT 1,2                          REM Position cursor on screen
CODE$ = CR$:                    REM Read code and assign variables
IF LEN(CODE$)=4                 REM Interrogate string length
PRINT “UHEX$=>”, UHEX$:(CODE$)  REM Convert code and display
GET
ELSE                            REM Branch
PRINT “UNABLE TO READ CODE”
GET
ENDIF                           REM End of loop
```

### UDEC$:(CODE$)

This function converts a binary-packed character string with a length of four characters into a readable format suitable for output. The code is converted into decimal notation for this purpose. The result is eight characters in length.
The parameter expected by the function is a string with a length of four characters, containing the code in binary-packed format. The code is converted into decimal format and returned as a string with a length of eight characters.

Example:

```
UDEC:                               REM Program name
LOCAL CODE$(4)                      REM Declaration of variables
AT 1,2                              REM Positions cursor on screen
CODE$ = CR$:                        REM Read code and assign variables
IF LEN(CODE$)=4                     REM Interrogate string length
PRINT "UDEC$=>", UDEC$:(CODE$)         REM Convert code and display
GET
ELSE                                REM Branch
PRINT "UNABLE TO READ CODE"
GET
ENDIF                               REM End of loop
```

### RW$:(ADRS%)

This function reads a word (16 bits) from a 256-bit data carrier at address ADRS%. ADRS% must be between 0 and 15.
The address of the data word to be read is transferred to the function, which reads out the data word and returns it as a string with a length of two characters. The ASCII value of the first character corresponds to the high byte of the data word, and the ASCII value of the second character to the low byte. If it was not possible to read the data word, an empty string will be returned.

Example:

```
RW:                                 REM Program name
LOCAL ADRS%                         REM Declaration of variables
PRINT "ADRS% (0-15) ="              REM Display input
AT 16,1                             REM Position cursor after expression
INPUT ADRS%                         REM Read-in command for the address
AT 1,3
PRINT "RW$:(ADRS%)=",RW$(ADRS%)        REM Read word and display
GET                                    REM Wait for keystroke
```

### WW$:(ADRS%,DATA%)

This function writes the word DATA% to the address ADRS% of a 256-bit data carrier. ADRS% must be between 0 and 15. DATA% is a 16-bit integer. The result of function WW$ is a character string with a length of two characters representing DATA% in ASCII HEX notation.
The parameters transferred to the function are the address and the value of the data word to be written. The data word is written and re-read for verification purposes. If the Write operation was successful, the data word is returned in the same format as for the function RW$.

Example:

```
WW:
Local ADRS%, INPUT$(2), Z1$(1), Z2$(1), DATA%
PRINT "ADRS% (0-15) ="
AT 16,1
INPUT ADRS%                     REM Read command for this address
PRINT "INPUT ="
AT 16,2
INPUT INPUT$                    REM Read command for 2 characters
Z1$ = MID$(INPUT$, 1, 1)        REM Isolate first character
Z2$ = MID$(INPUT$, 2, 1)        REM Isolate second character
DATA% = ASC(Z1$) + ASC(Z2$) * 256     REM Conversion into ASCII value
AT 9,3                                     and 16-bit integer
PRINT WW$:(ADRS%, DATA%)        REM Write word to address
GET
```

If the read/write unit outputs a signal tone during the Write operation, it must be brought closer to the data carrier to avoid an error while the parity information is being written. If an error occurred while writing, the function will return an empty string.

**RD$:**

This function restores the parity information of a 256-bit data carrier. It can be used if a data carrier can no longer be read (e.g. after a Write error). In many cases the problem is caused by a parity error. This function cannot be used, however, to correct faulty data in the data carrier. The function returns a string with a length of two characters, containing the last word written.
The function reads all memory locations three times, compares the data items read, and writes them, if identical, back to the same memory location. The new parity information is then written.
If the data carrier could be written to, the most recently written data word will be returned as a string with a length of two characters. Otherwise an empty string will be returned.

Example:

```
RD:
LOCAL A$(2)                     REM Declaration of variables
AT 3,1
PRINT "PARITY CHECK"
PRINT "-- PLEASE WAIT--"
A$ = RD$:                       REM Store result of parity check in A$
IF LEN (A$) = 2                 REM If length of A$ is 2 characters then ...
AT 4,4
PRINT "DATA CARRIER OK"
GET
ELSE                            REM ... else...
AT 4,4
PRINT "UNSUCCESSFUL"
GET
ENDIF                           REM End of loop
```

**Note:** The data recovered by this method is valid only if the parity error was caused by an error when writing the parity information.

### REIK$:(ADRS%), WEIK$:(ADRS%,DATA%), (from Version 2.0)

These two functions operate in the same way as RW$ and WW$ except that in this case, access is made to IDC-...-1K data carriers. For this reason, ADRS% must be in the range 0 to 63.

Example: (REIK$: only)

```
REIK:
LOCAL ADRES%
PRINT “ADRES% (0-63) =”            REM Address between 0 and 63
AT 16,1
INPUT ADRES%
AT 1,3
PRINT “REIK$: (ADRS%) =”, REIK$:(ADRS%)
GET
```

### HEXIN%:

This function is used to enter a number via the keyboard, which will then be converted to an integer value. The numerical range is therefore restricted to values between -32768 and +32767. Hexadecimal values can also be entered by preceding the number with the character $. In this case the value range will be between $0000 and $FFFF.

Example:

```
HEXIN:
LOCAL A
PRINT “ENTER VALUES BETWEEN  REM Note the input range
         -32768 ... 32767 OR
    $0000 ... $FFFF”
PRINT “INPUT =”
AT 9,4
A = HEXIN%:                        REM Read, convert and assign to A
PRINT A                            REM Display variable A
GET
```

### RIM$:(ADRS%)

Reads a byte at ADRS% in the IMC-... data carrier. The result is a character string containing one character. ADRS% is between 0 and 8191 for the IMC-40-64K and between 0 and 32767 for the IMC-40-256K.

Example:

```
RIM:
LOCAL ADRS%, INPUT%
DO
CLS
PRINT “ADRS% (0 - 8191) =”         REM Address range 0-32767 for
AT 16,1                                  IMC-40-256K
INPUT ADRS%
AT 1,3
PRINT “RIM$:(ADRS%) =”, RIM$:(ADRS%)       REM Read one byte to address
PAUSE -100
INPUT% = KEY
```

```
UNTIL INPUT% = %S OR INPUT% = %s
```

### WIM$:(ADRS%,DATA%)

Writes the byte DATA% at address ADRS% in the IMC data carrier. DATA% is an integer and must be between 0 and 255. For ADRS% the same applies as for RIM$.

Example:

```
WIM:
LOCAL ADRS%; DATA$(1)
PRINT “ADRS%(0-32767) =”          REM Address range 0-8191 for IMC-40-64K
AT 16,1
INPUT ADRS%                       REM Read-in address
PRINT “DATA$ =”
AT 16,2
INPUT DATA$                       REM Read-in data
CLS
AT 1,1
PRINT “ADRS%,DATA$ =”
AT 9,3
PRINT WIM$:(ADRS%;ASC(DATA$))  REM Convert data, transfer to
                                  address and display
GET
```

### BATS$:

Checks the battery status in an IMC-... data carrier. The result is a string containing one character.

‘*’ indicates that the battery is still in order.
‘U’ means that the battery charge is very low. In this condition the data carrier can be used as normal for a certain period, but needs to be replaced.

Example:

```
BATS:
LOCAL VOLT$(1)
CLS
PRINT “CHECK PRG. BATTERY”
VOLT$ = BATS$:                          REM Check battery
AT 1,3
IF VOLT$ = “*”                          REM Interrogate “OK”
PRINT “BATTERY OK”
ELSEIF VOLT$ = “U”                      REM Interrogate “Replace soon”
PRINT “REPLACE DATA CARRIER SOON”
ELSE                                    REM  .... else....
PRINT “NO DATA CARRIER”
ENDIF                                   REM End of loop
GET
```

It is recommended that the command BATS$ should be used at regular intervals, before or after each access to an IMC-... data carrier, for example.

## 4 Technical data

| | |
|---|---|
| Dimensions: | 40 x 75 x 30 mm |
| Current consumption: | Approx. 30 mA including PSION Organiser, for read operations |
| Service life: | With an alkaline manganese battery (e.g. VARTA - Alkaline - TYPE 4022) the Read/Write unit gives approx. 20 h read time . |
| Temperature range: | 273 Kelvin to 343 Kelvin (0° C to +70° C) |

### 4.1 Read/write distance table

| Key data: | Code or data carrier | Code- or data carrier mounting | Read/write distance |
|---|---|---|---|
| **Read:**<br>With 9 V battery voltage<br>(maximum voltage)<br>and 298 Kelvin (25 °C) | ICC-12 | not in metal | 13 mm |
| | | steel, flush-mounted | 10 mm |
| | IDC-12 | not in metal | 12 mm |
| | | steel, flush-mounted | 6 mm |
| | IDC-15 | not in metal | 16 mm |
| | | steel, flush-mounted | 10 mm |
| **Read:**<br>With 6.5 V battery voltage<br>(minimum voltage)<br>and 298 Kelvin (25 °C) | ICC-12 | not in metal | 9 mm |
| | | steel, flush-mounted | 7 mm |
| | IDC-12 | not in metal | 9 mm |
| | | steel, flush-mounted | not possible |
| | IDC-15 | not in metal | 12 mm |
| | | steel, flush-mounted | 7 mm |
| **Write:**<br>With 9 V battery voltage<br>(maximum voltage)<br>and 298 Kelvin (25 °C) | IDC-12 | not in metal | 7 mm |
| | | steel, flush-mounted | not possible |
| | IDC-15 | not in metal | 10 mm |
| | | steel, flush-mounted | 5 mm |
| **Write:**<br>With 6.5 V battery voltage<br>(minimum voltage) | IDC-12 | not in metal | not possible |
| | | steel, flush-mounted | not possible |
| | IDC-15 | not in metal | 7 mm |
| | | steel, flush-mounted | 2.5 mm |

Subject to reasonable modifications due to technical advances.

## 4.2 ASCII table

| hex | dec. | ASCII | hex | dec. | ASCII | hex | dec. | ASCII | hex | dec. | ASCII |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 0 | NUL | 20 | 32 | Space | 40 | 64 | @ | 60 | 96 | ‘ |
| 01 | 1 | SOH | 21 | 33 | ! | 41 | 65 | A | 61 | 97 | a |
| 02 | 2 | STX | 22 | 34 | “ | 42 | 66 | B | 62 | 98 | b |
| 03 | 3 | ETX | 23 | 35 | # | 43 | 67 | C | 63 | 99 | c |
| 04 | 4 | EOT | 24 | 36 | $ | 44 | 68 | D | 64 | 100 | d |
| 05 | 5 | ENQ | 25 | 37 | % | 45 | 69 | E | 65 | 101 | e |
| 06 | 6 | ACK | 26 | 38 | & | 46 | 70 | F | 66 | 102 | f |
| 07 | 7 | BEL | 27 | 39 | ‘ | 47 | 71 | G | 67 | 103 | g |
| 08 | 8 | BS | 28 | 40 | ( | 48 | 72 | H | 68 | 104 | h |
| 09 | 9 | HT | 29 | 41 | ) | 49 | 73 | I | 69 | 105 | i |
| 0A | 10 | LF | 2A | 42 | * | 4A | 74 | J | 6A | 106 | j |
| 0B | 11 | VT | 2B | 43 | + | 4B | 75 | K | 6B | 107 | k |
| 0C | 12 | FF | 2C | 44 | , | 4C | 76 | L | 6C | 108 | l |
| 0D | 13 | CR | 2D | 45 | - | 4D | 77 | M | 6D | 109 | m |
| 0E | 14 | SO | 2E | 46 | . | 4E | 78 | N | 6E | 110 | n |
| 0F | 15 | SI | 2F | 47 | / | 4F | 79 | O | 6F | 111 | o |
| 10 | 16 | DLE | 30 | 48 | 0 | 50 | 80 | P | 70 | 112 | p |
| 11 | 17 | DC1 | 31 | 49 | 1 | 51 | 81 | Q | 71 | 113 | q |
| 12 | 18 | DC2 | 32 | 50 | 2 | 52 | 82 | R | 72 | 114 | r |
| 13 | 19 | DC3 | 33 | 51 | 3 | 53 | 83 | S | 73 | 115 | s |
| 14 | 20 | DC4 | 34 | 52 | 4 | 54 | 84 | T | 74 | 116 | t |
| 15 | 21 | NAK | 35 | 53 | 5 | 55 | 85 | U | 75 | 117 | u |
| 16 | 22 | SYN | 36 | 54 | 6 | 56 | 86 | V | 76 | 118 | v |
| 17 | 23 | ETB | 37 | 55 | 7 | 57 | 87 | W | 77 | 119 | w |
| 18 | 24 | CAN | 38 | 56 | 8 | 58 | 88 | X | 78 | 120 | x |
| 19 | 25 | EM | 39 | 57 | 9 | 59 | 89 | Y | 79 | 121 | y |
| 1A | 26 | SUB | 3A | 58 | : | 5A | 90 | Z | 7A | 122 | z |
| 1B | 27 | ESC | 3B | 59 | ; | 5B | 91 | [ | 7B | 123 | { |
| 1C | 28 | FS | 3C | 60 | < | 5C | 92 | \ | 7C | 124 | l |
| 1D | 29 | GS | 3D | 61 | = | 5D | 93 | ] | 7D | 125 | } |
| 1E | 30 | RS | 3E | 62 | > | 5E | 94 | ^ | 7E | 126 | ~ |
| 1F | 31 | US | 3F | 63 | ? | 5F | 95 | _ | 7F | 127 | DEL |

# Notes

Subject to reasonable modifications due to technical advances.

We at Pepperl+Fuchs recognise a duty to make a contribution to the future.
For this reason, this printed matter is produced on paper bleached without the use of chlorine.

# One Company, Two Divisions.

## Factory Automation Division

### Product Range

- Digital and analogue sensors
- in different technologies
  - Inductive and capacitive sensors
  - Magnetic sensors
  - Ultrasonic sensors
  - Photoelectric sensors
- Incremental and absolute rotary encoders
- Counters and control equipment
- Identification Systems
- AS-Interface

### Areas of Application

- Machine engineering
- Conveyor or transport
- Packaging and bottling
- Automotive industry

## Process Automation Division

### Product Range

- Signal conditioners
- Intrinsically safe interface modules
- Remote Process Interface (RPI)
- Intrinsically safe field bus solutions
- Level control sensors
- Process measuring and control systems engineering at the interface level
- Intrinsic safety training

### Areas of Application

- Chemical industry
- Industrial and community sewage
- Oil, gas and petrochemical industry
- PLC and process control systems
- Engineering companies for process systems

### Service Area

Worldwide sales, customer service and consultation via competent and reliable Pepperl+Fuchs associates ensure that you can contact us wherever or whenever you need us. We have subsidiaries worldwide for your convenience.

## The Pepperl+Fuchs Group

**USA Headquarters**
Pepperl+Fuchs Inc. • 1600 Enterprise Parkway
Twinsburg, Ohio 44087 • Cleveland-USA
Tel. (330) 4 25 35 55 • Fax (330) 4 25 46 07
**e-mail: sales@us.pepperl-fuchs.com**

**Asia Pacific Headquarters**
Pepperl+Fuchs Pte Ltd. • P+F Building
18 Ayer Rajah Crescent • Singapore 139942
Tel. (65) 7 79 90 91 • Fax (65) 8 73 16 37
**e-mail: sales@sg.pepperl-fuchs.com**

**Worldwide Headquarters**
Pepperl+Fuchs GmbH • Königsberger Allee 87
68307 Mannheim • Germany
Tel. +49 621 7 76-0 • Fax +49 621 7 76-10 00
**http://www.pepperl-fuchs.com**
**e-mail: fa-info@de.pepperl-fuchs.com**

PEPPERL+FUCHS