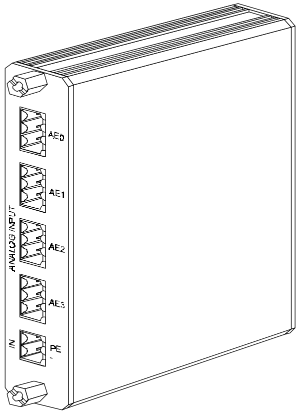
The analog input modules E.IPC - IO40 / IO41



- Module IO40: 4 analog voltage inputs
  resolution 12 bit / 10 ms
  ranges: -10 V ... +10 V / 0 V ... +10 V / -5 V ... +5 V / 0 V ... +5 V
- Module IO41 4 analog voltage inputs
  resolution 16 bit / 100 ms
  ranges: -10 V ... +10 V / 0 V ... +10 V / -5 V ... +5 V
- Signal preprocessing with microcontroller PIC16C64
- Programmable average generation
- Connection of input lines via plug-in screw contacts
- Easy assembly and secure attachment via two fixing screws
- High noise immunity thanks to screened electronics in aluminium housing and electronics design conforming to EMC requirements

The Festo IPC concept

Thanks to the modular design of the Festo IPC concept individual system components can be suited to very different areas of application. The system's modularity combines minimal space requirements with high PC capacity and low purchasing and maintenance costs.
Existing systems are freely extendible should requirements increase, but costs are minimised as only modules which are to be used are necessary.
The modularity of the system guarantees minimal logistic requirements as all systems use the same basic building blocks (busboard, CPU, I/O, and communication modules).
Special requirements such as integration into existing field bus systems are not a problem for the Festo IPC concept. The current range of more than 80 modules fulfils almost every need.
Thanks to the simple and robust plug-in system, users can install or exchange individual components in seconds without the need for dealing with special complex wiring or the removal of the housing. The ease of handling means that the costs and time required for servicing and training are minimised.
The many uses of the modules and the software are not limited to one particular system. They can be rearranged in any combination to develop modified or completely new possibilities for use.
The design of the modules is strictly geared to meet the demands of industry (demonstrated by the durable aluminium housing) and guarantees full compatibility with today's mechanical and electrical environment.
All the main plugs in the Festo IPC are in accordance with the standard for industrial PCs. The use of expensive special cables with unusual pin allocations has been avoided.

Contents

Update status 5

Hardware updates 5

Notes regarding installation and use 6

Scope of delivery 6
Usage and implementation 6
The design of the E.IPC module 7
The design of the E.IPC busboard 7
Connecting the module 8
The configuration of the module 9
The assignment of output channels 10
The configuration of the input voltage of the E.IPC-IO40 module 11
The configuration of the input voltage of the E.IPC-IO41 module 11
The internal structure 12

Maintenance 13

General notes regarding maintenance 13

Solutions to problems 14

Instructions regarding fault-finding 14

Programming 15

Reading and writing to the internal registers 15
The autoincrement function 15
Reading an analog value 15
The internal registers 16
The control port (writing) 17
The control port (reading) 17
The mode register (register 12) 18
The command register (register 16) 19
Programming example in Basic for the module IO40 19
Programming example in Basic for the module IO41 19
The safeguard functions ("PLC safeguard") 21
The common register (address E0h, write) 22
The module check register (E0h, read) 24
Allocation of the EEPROMs for modules with a PLC safeguard function 26
Structure of the parameter EEPROM (example IO41) 27

Data overview 29

Performance characteristics 29

Main details in brief 30
Technical data 31

We make every effort to ensure that our manuals are always up-to-date. If you should find over-complicated descriptions or even errors, please inform us at the following EMail address (Internet):**ps1docu@bcl.de**, indicating the title of the manual and the version number.
Any suggestions for improvement are also most welcome.
Unfortunately technical questions cannot be answered here. Please contact your dealer if you require technical support.

Update status

Hardware updates

Serial number	Board revision	Hardware revision	PAL revision	Alteration
as PM950401-300110	3	H00	P14	Actual version

Notes regarding installation and use

Scope of delivery

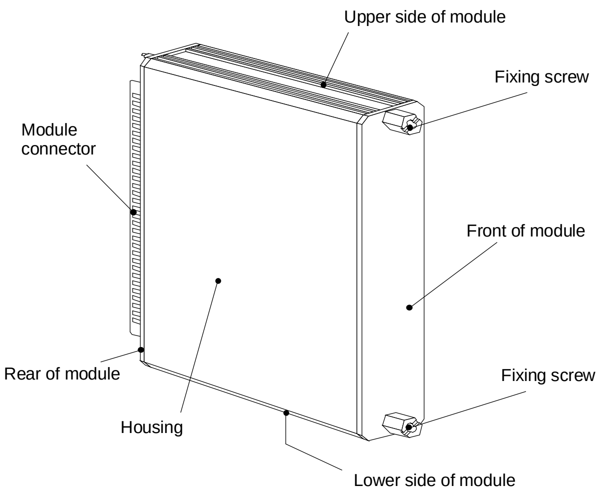
The following are included in the scope of delivery

- E.IPC-IO40 or E.IPC-IO41 module
- Manual
- Diskette with sample programs

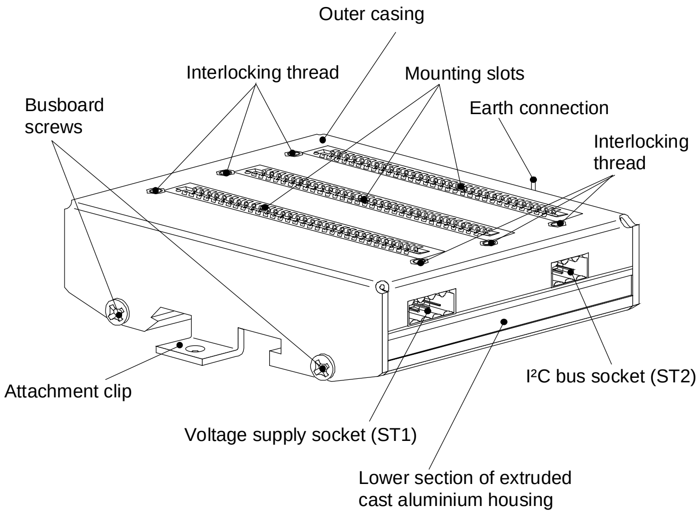
Usage and implementation

The modules E.IPC-IO40 and E.IPC-IO41 are Festo IPC components, designed for further processing of analog signals.
These modules are intended for use in a Festo IPC installation only. Whilst in operation the modules must be screwed tightly to the Festo IPC busboard and the voltage supplied via the E.IPC bus.
The modules can also be operated via the E.IPC-PCA1 PC adapter on a standard PC with ISA bus in the laboratory and development field.

The design of the E.IPC module



The design of the E.IPC busboard



Connecting the module

Configure the module using the rotary switch on the back of the module (please refer to the chapter "The configuration of the module"). Then insert the module into a free location on the Festo IPC busboard.

Warning

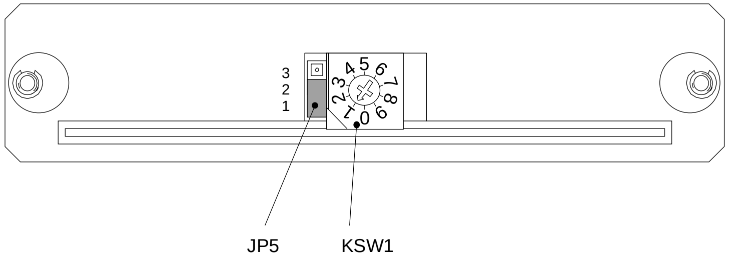
Make sure that all power to your system is switched off before installing your Festo IPC!

In order to guarantee a secure attachment, screw the module to the front of the busboard with the two fixing screws provided. While tightening the screws, use one hand to gently press the module into place.

Now connect the module to your installation.

The configuration of the module

The module is configured via a rotary switch underneath the module. The jumper JP5 has no function at present.



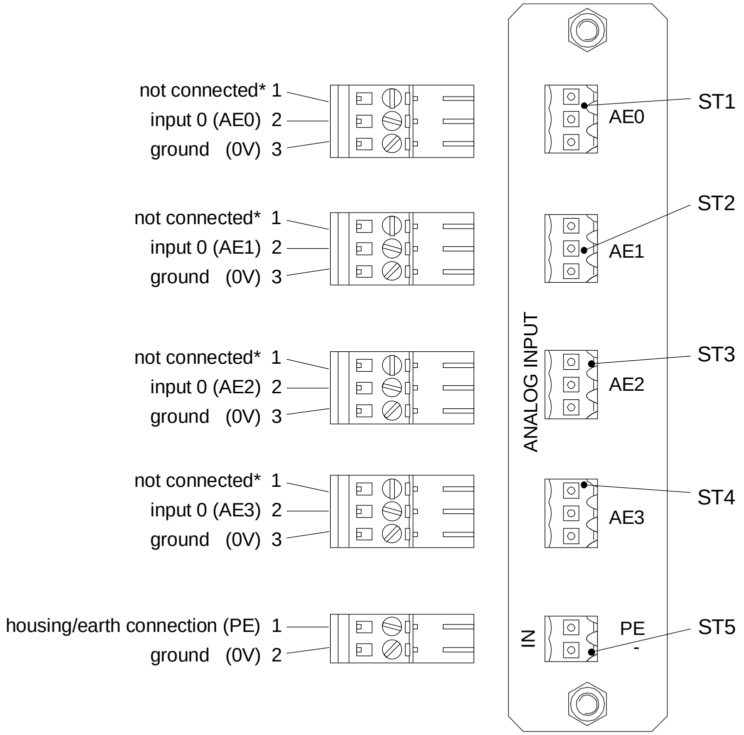
IO40/41	PAL version *	
KSW1	Control port	Data port
1	**1A4h (420)**	**1A5h (421)**
2	2A4h (676)	2A5h (677)
3	3A4h (932)	3A5h (933)
4	1A6h (422)	1A7h (423)
5	2A6h (678)	2A7h (679)
6	3A6h (934)	3A7h (935)
7	-	-
8	-	-
9	-	-
0	Reserved for customer specified design	Reserved for customer specified design

Address data in hexadecimal notation. Decimal notation in brackets.

Status at delivery indicates in bold type.

*) PAL version: P14

The assignment of output channels



*) Optionally, connecting the housing/earth connection of pin 1 on the ST5 plug to pin 1 of the plugs ST1 to ST4 provides an earth connection (PE) which acts as a protective earth conductor.

The configuration of the input voltage of the E.IPC-IO40 module

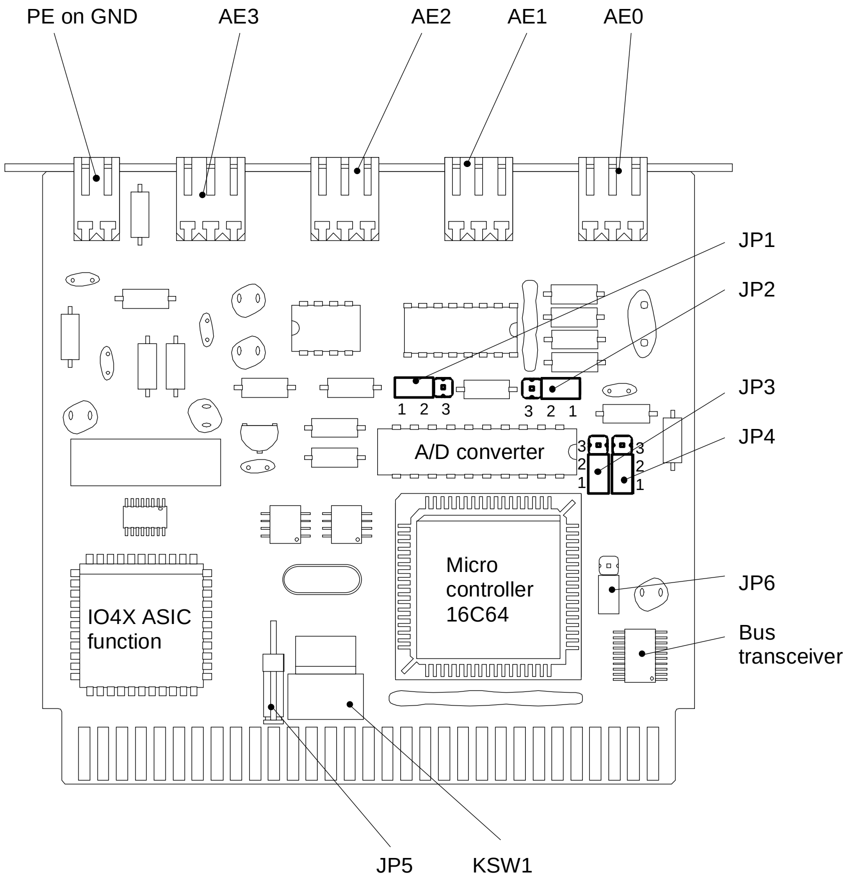
The input voltage range is set using software by the internal mode register.
After a system start or reset the module is set to -10 V .... +10 V.

The configuration of the input voltage of the E.IPC-IO41 module

Input voltage	Jumper switching			
	JP1	JP2	JP3	JP4
-10 V ... +10 V *	**1-2**	**1-2**	**1-2**	**1-2**
0 V ... 10 V	2-3	2-3	2-3	2-3
-5 V ... +5 V	2-3	2-3	2-3	1-2

* as delivered

The internal structure



Module IO40: The jumpers JP1 ... JP6 are factory settingS. They don't change the position.

Module IO41: Both the jumpers JP5 and JP6 have no function at present.

Maintenance

General notes regarding maintenance

Thanks to its robust design and the avoidance of batteries or similar components which may have to be exchanged, the E.IPC-IO40 and E.IPC-IO41 modules are absolutely maintenance-free.

Solutions to problems

Instructions regarding fault-finding

Please ensure that the installation instructions in the manual have been carried out correctly. Check that the E.IPC system has been connected using undamaged, standard cables. Moreover, it is absolutely necessary that the E.IPC system is supplied with the correct voltage.

Programming

Reading and writing to the internal registers

The analog input/output modules E.IPC-IO40 and E.IPC-IO41 has 18 internal registers, which can be read or written by the application program.
The registers are adressed by writing the register index (0..17) to the control-register and than reading or writing the data-register.
When the low-byte of a analog value ist read, all analog values are froozen until a high-byte of any analog value has been read. So the high-byte can be read with out changing.

The Busy bit is set automatically in the control register whenever the control or data registers are accessed. The Busy bit is reset once the E.IPC-IO41 module has processed the data. Therefore before each access to the control or data register, it is necessary to check that or wait until the busy bit is equal to 0.

The autoincrement function

When a register value is read from the data register, the register index in the control-register is incremented automaticaly, so that all 6 analoge values can be read without the control register having to be addressed each time.
The low and high byte values of an input channel always have to be read using the autoincrement function.

Reading an analog value

As the analog value of an A/D channel is read by accessing the register twice, it is necessary to follow the following sequence in order that the measured value is not overwritten by another value during the reading process.

1. The A/D channel to be read is selected by writing the internal register address of the low byte to the control register.
2. The data register is read (contains low byte)
3. The data register is read (contains high byte)

It is necessary to check whether the module is ready before each access. Please refer to the chapter "Reading and writing to the internal registers".

The internal registers

Index (control)	Register name (data)	
0	R	ADC channel 1, LSB
1	R	ADC channel 1, MSB
2	R	ADC channel 2, LSB
3	R	ADC channel 2, MSB
4	R	ADC channel 3, LSB
5	R	ADC channel 3, MSB
6	R	ADC channel 4, LSB
7	R	ADC channel 4, MSB
8	-	reserved
9	-	reserved
10	-	reserved
11	-	reserved
12	R/W	Mode
13	R/W	Channel active mask
14	R	Firmware revision, LSB
15	R	Firmware revision, MSB
16	R/W	Command

The control port (writing)

Bit number	R/W*	Function
0 ... 4	W	Register address
5 ... 7	-	not used

The control port (reading)

Bit number	R/W*	Function
0	R	0 = Module ready 1 = Module busy
1 ... 7	-	not used

*) R = read
W = write

The mode register (register 12)

	IO40	IO41	
Bit number	R/W*	R*	Function
0 ... 1	R	R	Measuring range
2 ... 3	R/W	R/W	Averaging
4 ... 7	-	-	Reserved for extensions

Mode register		Measuring range
Bit 1	Bit 0	
0	0	-10 V ... +10 V (IO40 and IO41)
0	1	0 V ... +10 V (IO40 and IO41)
1	0	-5 V ... +5 V (IO40 and IO41)
1	1	0 V ... +5 V (only IO40)

Mode register		Function
Bit 3	Bit 2	
0	0	No averaging
0	1	not used
1	0	not used
1	1	max. averaging

*) R = read
W = write

The command register (register 16)

Bit number	R/W*	Function
0	W	1 = Command active
1	W	1 = Read calib. data
2	W	1 = Write calib. data
3 ... 6	-	Reserved for extensions
7	-	Reserved for diagnostics

Programming example in Basic for the module IO40

```
100 ‘SAMPLE1.BAS
110 ‘reading IO40 analog values
120 CTL% = &H1A4        ‘KSW1 = 1
130 DAT% = &H1A5
140 VMIN = 0! : VMAX = 10!
150 VRNG = VMAX - VMIN
160 VLSB = VRNG / 4096!
170 N = -VMIN / VLSB
180 FOR CH%=0 TO 3      ‘read channel 1 to 4
190   GOSUB 320         ‘wait for ready
200   OUT CTL%,CH%*2    ‘set internal register address
210   I1 = INP(DAT%)    ‘read internal register (low-byte)
220   GOSUB 320         ‘wait for ready
230   I2 = INP(DAT%)    ‘read internal register (high-byte)
240   I = I1 + I2 * 256!
250   V = (I - N) * VLSB
260   PRINT CH%+1,HEX$(I),V
270 NEXT CH%
280 GOTO 180
290 ‘
310 ‘wait for busy-bit = 0
320 IF (INP(CTL%) AND &H01)<>0 THEN GOTO 320
330 RETURN
```

Programming example in Basic for the module IO41

```
100 ‘SAMPLE1.BAS
110 ‘reading IO41 analog values
120 CTL% = &H1A4        ‘KSW1 = 1
130 DAT% = &H1A5
140 VMIN = 0! : VMAX = 10!
```

```
150 VRNG = VMAX - VMIN
160 VLSB = VRNG / 65536!
170 N = -VMIN / VLSB
180 FOR CH%=0 TO 3      ‘read channel 1 to 4
190   GOSUB 320         ‘wait for ready
200   OUT CTL%,CH%*2    ‘set internal register address
210   I1 = INP(DAT%)    ‘read internal register (low-byte)
220   GOSUB 320         ‘wait for ready
230   I2 = INP(DAT%)    ‘read internal register (high-byte)
240   I = I1 + I2 * 256!
250   V = (I - N) * VLSB
260   PRINT CH%+1,HEX$(I),V
270 NEXT CH%
280 GOTO 180
290 ‘
310 ‘wait for busy-bit = 0
320 IF (INP(CTL%) AND &H01)<>0 THEN GOTO 320
330 RETURN
```

The safeguard functions ("PLC safeguard")

Several of the new E.IPC I/O modules are equipped with additional safeguard functions which allow the user program to check the correct functioning of the module both during the starting up process and runtime.

By reading a parameter EEPROM on the module when the system starts, it is possible to determine whether the correct module is at the correct address.
Further information such as version number and production date can be ascertained for servicing and maintenance purposes. This parameter EEPROM is addressed via the I²C bus.

Checks during runtime are made and the PLC safeguard functions controlled via additional registers (the safeguard registers). These registers are to be found in one common memory range (processor I/O address E0h-EFh).

The safeguard registers are divided into so-called “paged” and “not paged” registers

Register address	R/W	Register name	paged / not paged
E0h (224)	R	Modul check register	paged
E0h (224)	W	Common register	not paged

The „paged“ registers are only active when a write access is made to one of the normal function registers immediately beforehand (e.g. for the module E.IPC-OM21 with switch position at address 110h). The „paged“ registers only influence the function of the module addressed immediately beforehand.

The “not paged” registers are always active and influence the operating mode of all the modules with additional safeguard functions simultaneously. The “not paged” registers can only be written to.
The processor interrupt must be disabled whilst the standard function register and the “paged” safeguard register are accessed, in order to avoid a possible processor I/O access at another address between the two.

Example for IO41 in Turbo-C (excerpt)

```
disable();                  /* Disable interrupts */
outportb(0X1A4,value);      /* Access to standard function register for addressing
                            module*/
chk = inportb(0XE0);        /* Access to safeguard function register */
enable();                   /* Enable Interrupt */
```

As the SuperState interrupts of the CPU modules HC1X cannot be disabled, please refer to the section “The module check register” .

The common register (address E0h, write)

The common register is always active and controls the operating mode of all the modules with safeguard functions on the busboard. The register can only be written to, but not read.

Bit no.	R/W*	Name	Functions	Value**
0	W	PAGE MODE	0 = PLC safeguard functions inactive 1 = PLC safeguard functions active	0
1	W	EE MODE	0 = EEPROM deactivated 1 = EEPROM active, when module active***	0
2	W	IO RST	0 = Module reset not triggered 1 = Trigger module reset	0
3-7	W	-	Reserved, must always be set to "0"	0

*) R = Read access, W = Write access
**) Must always be reset to “0” after the EEPROM has been accessed.

Bit 0: PAGE MODE
The safeguard functions of all the modules can be activated and deactivated via this bit. If the bit is set to 0 then the safeguard functions are inactive and the modules behave in the same way as modules without safeguard functions. This corresponds to the standard setting. If the bit is set to 1, then the safeguard functions are active.

Please note that when the safeguard function is deactivated (bit 0 = 0), then the modules behave differently than when the safeguard function is active.
For further information, please refer to the section “Treatment of errors when safeguard function is deactivated”.

Bit 1: EE MODE
The parameter EEPROM on the module is usually addressed via the I²C address ACh. When this bit is set to 1, the module-internal parameter EEPROM is switched to the I²C bus address AEh every time the standard function registers of the module are accessed. The EEPROM can then be read via the service interrupt functions of the CPU.
If the bit is set to 0 then the parameter EEPROM is not switched (remains at address ACh). Once the EEPROM has been read, the bit must be reset to 0 by the user program.

Bit 2: IO RST (IO RESTART)
If this bit is set to 1, all of the modules with a safeguard function are reset to switch-on state (reset) and the outputs reset to their initial state. At the same time the bit IO RST is set automatically to 0.
It is only necessary to reset a module, when an error occurs within the module. This function is not required during normal operation.

The module check register (E0h, read)

Each time one of the standard function registers is accessed, the inverted value can be read bit by bit in this register.

Example for IO41 in Turbo C (excerpt):

```
  outportb(0XE0,0X01);         /* Activate page mode */
  disable();                   /* Disable interrupts */
  outportb(0X1A4,0X01);        /* Output value, relay output 1 active */
  chk = inportb(0XE0)          /* Read module check register */
  enable();                    /* Enable interrupts */
  if (chk!=0XFE)               /* Check whether inverted value */
    goto error                 /* can be read */
```

One particular feature should be observed when reading the module check register

Once the standard function register has been accessed, the module check register can only be read until another processor IO address is accessed. On the odd occasion a hardware or software interrupt may occur after the standard function register has been accessed and before the module check register has been read which will cause any number of processor IO addresses to be accessed and thus trigger an error.
As the user cannot deactivate all of the interrupts of the E.IPC CPU, the whole procedure (writing to the standard function register, reading the module check register) should be repeated at least once before the user program reports an error, if ever an error occurs whilst the module check register is being read.

Example for IO41 in Turbo C (Excerpt):

```
outportb(0XE0,0X01);         /* Activate page mode */
disable();                   /* Disable interrupts, except SuperState interrupts */
outportb(0X1A4,0X01);        /* Output value */
chk = inportb(0XE0);         /* Read module check register */
enable();
if (chk!=0XFE)
  {                          /* Error has occurred -> Repeat */
  disable();                 /* access */
  outportb(0X1A4,0X01);
  chk = inportb(0XE0);
  enable();
  if (chk!=0XFE)
    goto error;              /* Genuine error */
    }
```

Example for IO41 in Turbo Pascal (excerpt):

```
Port[$E0] := $01;            (* Activate page mode *)
disable;                     (* Disable interrupts, except SuperState interrupts
                             (HC1X)*)
Port[$1A4] := $01;           (* Output value *)
chk := Port[$E0];            (* Read module check register *)
enable;                      (* Enable interrupts *)
if chk<>$FE then
  begin                      (* Error has occurred -> Repeat access *)
    disable;
    Port[$1A4] := $01;
    chk := Port[$E0];
    enable;
    if chk<>$FE then
      goto error;            (* Genuine error *)
    end;
```

Enable and disable are not available in standard Pascal. Here is an example of these procedures. They are dependent on the pascal version beeing used.

```
procedure disable;
begin
  asm
    cli;
  end;
end;

procedure enable;
begin
  asm
    sti;
  end;
end;
```

Allocation of the EEPROMs for modules with a PLC safeguard function

All those modules with a PLC safeguard function have a 256 byte EEPROM which contains all of the important data concerning the module (see table).
The EEPROM is addressed via the I²C bus using the service interrupt function of the E.IPC CPU modules.
In order that any amount of E.IPC modules with a PLC safeguard function can be operated simultaneously in the system without causing a conflict of the EEPROMs I²C addresses, all of the EEPROMs are addressed via the same I²C bus address (ACh).
The EEPROM of a module only switches to the I²C bus address AEh when the module is addressed via the parallel E.IPC bus. The EEPROM can then be read via the I²C bus.
In contrast to the module check register which is automatically invalidated, the EEPROM must be reset explicitly by setting the EE MODE bit (bit 1 in the common register) to zero. If the EEPROM is not reset, it will conflict with the EEPROM of the next module to be activated. As a result the EEPROM of the next module cannot be read.
Furthermore, the EE MODE bit must be set to 1 before the standard function register is accessed, otherwise the EEPROM will not be activated.

For reasons of speed and to avoid burdening the I²C bus unnecessarily, the EEPROM should be read once only when starting up the installation or by activating a diagnostic routine in the user program. The module check register is available to check the module during run-time.

Using the data in the EEPROM, the user program can determine automatically whether the actual hardware configuration corresponds to the specified configuration.

In order that the module can be identified as quickly as possible, the name of the module is to be found at the beginning of the EEPROM so that only 11 characters must be read in order to establish the module type.

If two modules of the same type (e.g. 2 x IO41) are set to identical addresses, this will not cause a check sum error when the header is read out since these are identical. A conflict of addresses can only be determined by reading the whole EEPROM and establishing the check sum thereof.

Structure of the parameter EEPROM (example IO41)

Pos.*	Length	Comment	Value (hex) (Example)
00	1	Type	02
01	1	Check sum via header 256 - (< cross-check sum via address 00 ... 0Ah > module 256)	02
02	6	Module designation in ASCII (IO41)	49 4F 34 31 00 00
08	1	Downward compatibility (not yet used)	00
09	1	Upward compatibility (not yet used)	00
0A	1	Reserved	00
0B	1	End of the complete allocated range	38
0C	1	Check sum via complete allocated area 256 - (< cross check sum via alloc. range (00 ...) > module 256)	16
0D	3	Serial number of the module, 6 digits BCD	30 00 50
10	1	Hardware version of the module (0-255)	00
11	2	PAL version of the module (e. g. "A1" or "29")	31 34
13	2	Software version of the module (e. g. 01 02 = 1.02)	02 01
15	2	Special variant (low, high: 100-9999)	00 00
17	2	Production date (packed in 2 bytes) Bit 0 ... 5: calender week Bit 6 ... 9: year from 1994 onwards (1994 = 0) Bit 10: 0 = Serial module, 1 = "red module" (pilot series) Bit 11 ... 15: check information for the production	C2 00 00 00
19	2	Servicing info (last service, date packed in 2 bytes) Bit 0 ... 5: calender week Bit 6 ... 9: year from 1994 onwards (1994 = 0) Bit 10 ... 15: Servicing counter	00 00
1B	1	Card type (IO41 = 04h) 01 = digital in 10 = communication 02 = digital out 20 = memory module 04 = analog in 40 = not yet defined 08 = analog out 80 = miscellaneous	04
1C	1	Number of outputs	00

*) Statements in hexadecimal notation

Pos.*	Length	Comment	Value (hex) (example)
1D	1	Bit 0 ... 2: output bits per byte - 1 / bit 5 ... 7: output bytes	00
1E	1	Number of inputs	04
1F	1	Bit 0 ... 2: input bits per byte - 1 / Bit 5 ... 7: input bytes	00
20	1	Number of PLC function status registers (as of I/O addresse E1h) Bit 0 ... 3: number Bit 7 = 0: Mask in the following 4 bytes Bit 7 = 1: Bit 0 ... 6 = pointer to the mask	00
21	4	Bit masks for status register (max. 4 status registers)	00 00 00 00
25	2	Offset in EAM memory (low/high)	00 00
27	9	Reserved	00
30	1	Pointer to customer range (16 bytes)	00
31	1	Pointer to EAM init routine	00
32	1	Type of EAM init routine (Assembler, Macro language, ...) 0/FF = Routine not available 1 = 8088 Assembler 2 = Macro language 3 = ...	00
33	1	Pointer of EAM de-init routine	00
34	1	Type of EAM de-init routine (as for init routine)	00
35	1	Pointer to EAM read routine	00
36	1	Type of EAM read routine (as for init routine)	00
37	1	Pointer to EAM write routine	00
38	1	Type of EAM write routine (as for init routine)	00
39	1	Pointer to range for modifiable, module specific configuration data e.g. calibration data for analog modules, additional data for communication modules (network node address for network modules, telephone numbers for modem/Fax module etc.)	00
3A	6	Reserved	00
40 ...		Beginning of free range	00

*) Statements in hexadecimal notation

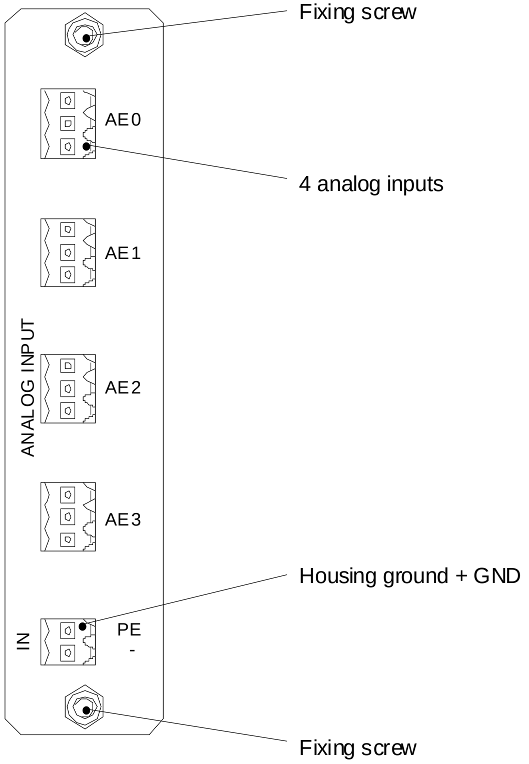
Data overview

Performance characteristics

- Module IO40: 4 analog voltage inputs
  resolution 12 bit / 10 ms
  ranges: -10 V ... +10 V / 0 V ... +10 V / -5 V ... +5 V / 0 V ... +5 V
- Module IO41 4 analog voltage inputs
  resolution 16 bit / 100 ms
  ranges: -10 V ... +10 V / 0 V ... +10 V / -5 V ... +5 V
- Signal preprocessing with microcontroller PIC16C64
- Programmable average generation
- Connection of input lines via plug-in screw contacts
- Easy assembly and secure attachment via two fixing screws
- High noise immunity thanks to screened electronics in aluminium housing and electronics design conforming to EMC requirements

Main details in brief

The modules E.IPC-IO40 and E.IPC-IO41 are Festo IPC components, designed for further processing of analog signals.
These modules are intended for use in a Festo IPC installation only. Whilst in operation the modules must be screwed tightly to the Festo IPC busboard and the voltage supplied via the E.IPC bus.
The modules can also be operated via the E.IPC-PCA1 PC adapter on a standard PC with ISA bus in the laboratory and development field.



Technical data

Climatic features:

Temperature

- Transport/storage (up to 3300 mNN) . . . : -25 °C to +70 °C
- Normal operation (up to 2000 mNN) . . . : 0 °C to +55 °C
- Change in temperature: Transport/storage . . : max. 20 °C/h
  Normal operation . . : max. 10 °C/h

Humidity requirement DIN 40040

- Humidity class E . . . . . : 75 % on 30 days
- Relative humidity (at 25 °C non condensing) . . : 10 to 95 %

Mechanical features:

- Cast aluminium housing
- Weight . . . . . . : approx. 170 g
- Dimensions . . . . . : 75x21x96 mm
- MTBF (at 35 °C) . . . . . : better than 100 000 h

Mechanical environmental operating conditions:

Vibratory load (according to IEC 68-2-6)

- Vibration (when screwed on) . 10 to 57 Hz . : const. ampl. 0,075 mm
  57 to 150 Hz : : const. accel. 1 G

Shock (according to IEC 68-2-27)

- Type of shock . . . . . : Half sine
- Strength of shock (when screwed on) . . . : up to 15 G in 11 ms

Electrical environmental operating conditions:

- Electromagnetic compatibility . . . : DIN IEC 1131-2 (EMV)
- Fast burst firing (burst) . . . . : DIN IEC-801-4-L3
- Electrostatic discharge (ESD) . . . : DIN IEC-801-2

- Electromagnetic HF field . . . . : DIN IEC-801-3
- Magnetic LF field . . . . . : DIN IEC-770-6.2.9
- Electrical safety requirements . . . : VDE 0160
- Insulation group A . . . . . : VDE 0160

Electrical features:

- Maximum current consumption without external keyboard (at 5 V) : typ. 120 mA
- Power consumption . . . . : typ. 6.5 W

Protection against ingress of solid foreign bodies and water:

- For top-hat rail components . . . : IP 40

AD converter E.IPC-IO40

- Converter: Burr-Brown ADS7809
- Integral linearity error: ± 1 LSB (max.)
- Input voltage: -10 V ... +10 V / 0 V ... +10 V / -5 V ... +5 V / 0 V ... +5 V
- Resolution: 12 bit incl operational sign
- Conversion time: 10 ms incl. average generation

AD converter E.IPC-IO41

- Converter: Burr-Brown ADS7809
- Integral linearity error: ± 3 LSB (max.)
- Input voltage: -10 V ... +10 V / 0 V ... +10 V / -5 V ... +5 V
- Resolution: 16 bit incl operational sign
- Conversion time: 100 ms incl. average generation

All of the brand names mentioned are registered trade marks of the respective owners