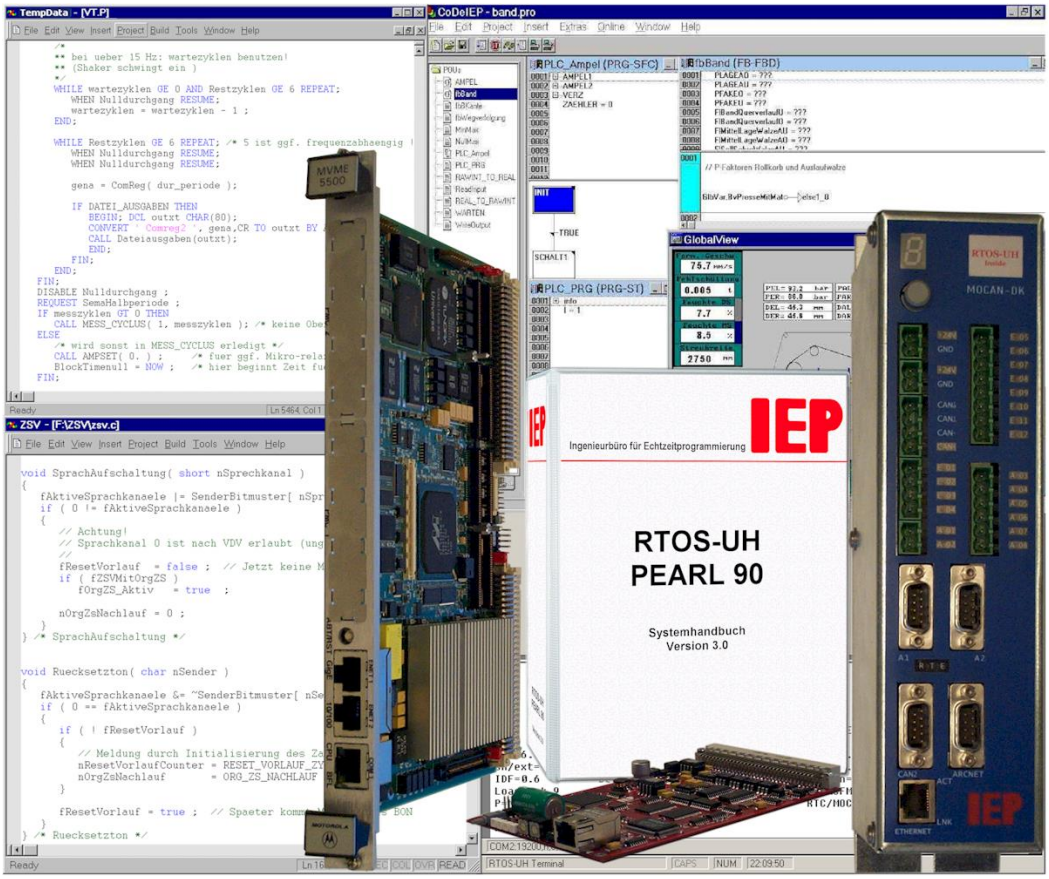
IEP

At a glance



Hard- und Software for industrial control

Ingenieurbüro für Echtzeitprogrammierung GmbH

Am Pferdemarkt 9C
D-30853 Langenhagen

Tel.: +49 (0)511 70 832 0
Fax: +49 (0)511 70 832 99

E-Mail: info@iep.de
Web: http://www.iep.de

2018

Company profile

Our house

For more than fifteen years we are acting as a competent partner to our customers in all interests of automatic control engineering. Seven high motivated and highly qualified coworkers engage themselves for customers out of the most diverse industrial fields. We develop intelligent solutions for tasks of automatic control, measuring technique and factory data acquisition, which are easy to customize and are always open to fulfill new requirements.

The three founders of 1986 are owners and managing directors of the GmbH since 1992.

Continuity

IEP was founded in 1986 by innovative engineers with the core competence in design, development and implementation of software systems, which especially must meet hard realtime constraints. The confidence of our customers and our clear business vision have led to constant growth of the company. The design of local networks and the integration of systems into heterogeneous nets are belonging to our knowledge profile for a long time. We realize modern graphical user environments, based on PC systems, using object-oriented programming languages. The realisation of many projects as complete turn-key-solutions has led to constant growth of our department for hardware development and thus continuously increases our portfolio of efficient hardware products.

Flexibility

With numerous tasks, **IEP** proves again and again that our enterprise in every respect can match with and even outshine large companies on flexibility and customer satisfaction: our customers appreciate custom-made, economical solutions and our ability to complete sophisticated projects with great reliability even under time pressure.

Innovation

Our customers and we profit from our membership in professional associations, from the cooperation in working groups and within research projects. **IEP** constantly stays up to date: with a clear look into the future we develop and ensure support for new technologies.

Quality from the outset

Our products meet the highest quality standards.

Software development with modern design methods and constant deployment of a version control system with automatic documentation assure the traceability and reliability of our programs. Code revisions and complete coverage function tests ensure quality on highest level.

By archived test reports and standardized test procedures, which can also be provided in coordination with your desires, our hardware is comprehensible to the manufacturing lot. A complete function test for each item delivered is a matter of course.

Support

Apart from expertise, effectiveness and experience, optimal customer service is our companies primary goal. Aside from reliable and well-designed products, we offer fast and uncomplicated service. **IEP** is serious about delivering first-class service to our customers.

Close co-operation with our customers and flexible reaction to their needs are leading to a rapid realisation of their and our ideas. We put new systems into operation locally and instruct your personnel reliably in their operation. We accompany the deployment of our products with support and consultation.

We will always be competent partners, familiar with your conditions of our products use and all technical innovations at the market.

Contact

Dipl.-Ing. B. Kroll, operating system design and networks

Dipl.-Ing. A. Hadler, application design and implementation

Dipl.-Ing. K. Koerth, hardware design and field bus systems

P. Lange, manufacturing and purchase

Our Offer

Software development

We develop customized and application orientated software according to state of the art engineering.

Hardware development

For special applications we develop specialised electronics.

System configuration

For each application we design and configure optimal control systems build from standard components.

Networking and network integration

With TCP/IP, Profibus, InterBus, CAN, SINEC-H1, SINEC-L2, InterChange we integrate heterogeneous systems.

Turn-key solutions

We design complete control concepts and realize projects up to turn-key delivery.

Our Products

Matured products provide you with tools to realize own projects:

- Development environment for ANSI-C in realtime systems
- PLC Programming system CoDeSys, based on IEC 61131-3
- Programming systems for realtime applications, based on PEARL and ANSI-C
- BDE systems for plant modernization, in particular under consideration of the requirements according to ISO 9000
- Modular control and automation systems based on VMEbus or PBus for custom-made, economical solutions

Our Tools

RTOS-UH

The fast and comfortable realtime multitasking operating system. We also integrate special hardware into the system.

PEARL

Easy to learn and application orientated programming language with realtime elements. We can port the PEARL system also to your hardware!

ANSI-C

The efficient programming language for systems programming and for demanding realtime applications.

IEC 61131-3 with CoDeSys

The conventional programming of controls in conjunction with the flexibility of a realtime system.

References

Our customers are working in all areas of automation. This overview represents a cutout from the broad spectrum of realized projects; on request we gladly will name a contact person.

Area of application	Project and company
Plant construction	**Plant automation with multiprocessor systems** Siempelkamp GmbH & CO, Krefeld
Industrialized concrete building	**Large scale plotter, Profibus BDE/MDE network** control Systemtechnik GmbH, Koblenz
Factory data capture	**Access control systems** Getronik GmbH, Hannover
Energy supply	**Energy distribution** Stadtwerke Hannover AG
	Wind power stations West control AB, Denmark
Driverless Transportation systems	**Vehicle control, traffic regulation, stock management** NOELL GmbH, Herzogenrath
Television	**Transmission control** CLT/UFA, Luxembourg; RTL, Cologne
Product engineering	**Laserwelding** FhG Institute for laser technology. Aachen
Research and teaching	**numerous universities**
Industrial electronics	**various large-scale installation plant controls** ATR industrial electronics, Viersen
Materials science	**Mechanical-dynamic spectrometer** GABO Qualimeter GmbH, Ahlden
Outer space technology	**Experiment data acquisition** ZARM GmbH, Bremen
Environmental technology	**Laser light scattering photometer** Lorenz Meßgerätebau, Katlenburg

Contents

RTOS-UH

Real-time operating system

Why RTOS-UH

Ever more and more complex technical systems demand the use of computers for open and closed loop control. **RTOS-UH** supports the use of control computers by a straight concept and high realtime reactivity for fast and flexible software reaction to changes and events in the processes.

Realtime

RTOS-UH enables software to react immediately upon external events (interrupts). All administrative operations of the operating system can be interrupted and the processor can be freed for the handling of external events. Even problems with very high reactivity requirements can be solved reliably by a control computer.

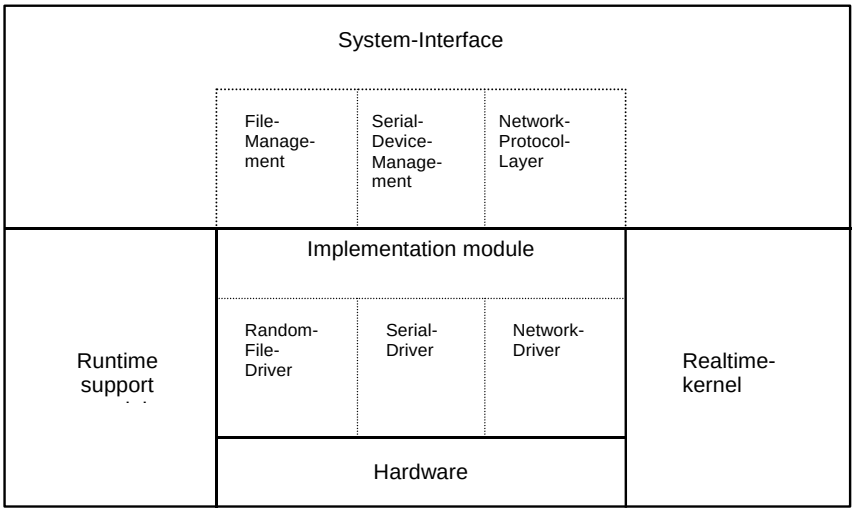
Multitasking

In practice, most often it is of great use to subdivide the complex task of steering a plant or controlling a process into smaller and simpler subtasks. **RTOS-UH** supports the breakdown of complex problem definitions by providing a framework to solve these subtasks individually and let these subtasks interact freely. The number of tasks in the system is only limited by the size of the memory. **RTOS-UH** allocates computing time to the individual task strictly based on their priority and allows for time sharing operation on each priority level. A very high number of priority levels allows for fine grained control of program interaction. The tasking is under full control by the user.

Programmingsystem

RTOS-UH is a complete realtime operating system. With its strictly modular structure, it is scalable from a small ROM-based runtime kernel up to complete workstation level with file system, network, and user interaction. Beginning by the realtime kernel the system is broadened by adding individual components. At start up, the kernel scans for additional modules and extends ist functionality according to the components found.

There is no need for special configuration tools to tailor the systems scope of services to the needs. A complete development system can be build using the same run-time kernel and delivering the same runtime behavior as a dedicated target system. User interfaces – commonly only available at workstations – can be added to targets to increase their observability and servicability.



The modular structure of the system and the automatic self-configuration of independent individual building blocks allows the system configuration even by an application designer.

Operating system

As operating system, **RTOS-UH** manages the resources of a computer. An I/O concept based on messages and message queues, an automatic memory management and an integrated error handling serve for comfortable application programming without a large system overhead. A realtime-oriented command interface supports program development and program inspection also under realtime conditions.

Realtime kernel

RTOS-UH is runnable on all processors of the PowerPC and the MC68xxx families. The assembler-coded realtime kernel provides all fundamental system services like:

- Dispatcher
- Process manager (scheduler)
- I/O management based on messages and message queues
- Memory management
- Error handling

The realtime kernel is identical in all **RTOS UH** implementations. All specific system modules as well as the resident user programs are integrated during start-up of the operating system.

Implementation

The implementation module consists of hardware-specific initialisation code and device drivers. By its modular structure, **RTOS-UH** offers

- Driver for serial and parallel interfaces
- Mass storage driver for Floppys and non removable disks
- Network driver for the deployment in LANs and WANs

All components can also be used individually and can be configured in accordance to specific requirements.

Runtime support

RTOS-UH provides all application programming interfaces in a run time module. These functions are reentrant, so the operating system and applications can use them at the same time. The commonly used integration of the respective drivers and arithmetic routines into the user programs is void, resulting in reduced memory requirements. Loadable and executable programs require only little space in RAM or ROM.

Command interface

Systems not headed for specific applications call for an efficient command interface. With commands to show and modify the state of individual tasks, **RTOS-UH** delivers insight and control over the multitasking. At any time a detailled inspection of the systems current working state is possible.

The realtime oriented commands are borrowed from the conforming PEARL-statements:

RTOS-UH realtime commands

WHEN *event* [ACTIVATE] *taskname* [PRIO *number*];
WHEN *event* C[ONTINUE] *task*;

AT *time* [ACTIVATE] *taskname* [PRIO *number*];
AT *time* C[ONTINUE] *taskname*;

AFTER *duration* [ACTIVATE] *taskname* [PRIO *number*];
AFTER *duration* C[ONTINUE] *taskname*;

AT *time* ALL *duration* [ACTIVATE] *taskname* [PRIO *number*];
AT *time* ALL *duration* UNTIL *time* [ACTIVATE] *task* [PRIO *number*];
AT *time* ALL *duration* DURING *duration* [ACTIVATE] *task* [PRIO *number*];

AFTER *duration* ALL *duration* UNTIL *time* [ACTIVATE] *task* [PRIO *number*];

The testing of applications, in particular difficult when running under realtime constraints on a multitasking system, is substantially simplified by an integrated trace facility and lots of low-level debug tools.

System interface

The system programming interface provides a uniform software interface to applications und service tools. Build upon fundamental services delivered by the implementation modules, the system interface integrates higher level administrative tasks and provides a hardware independent layer of abstraction. The system interface is identical for all versions of **RTOS-UH**, so applications can be easily ported between different processors using **RTOS-UH**.

Speed

Speed and dependability of timing are the most relevant aspects of a realtime operating system for control. Benchmark data for **RTOS-UH** show these values:

Processor Clock frequency Board	PowerPC 7455 1 GHz MVME 5500	MC68060 50 MHz MVME 177
Context switch (a to b, both tasks runnable)	**0.8 µs**	**6.5 µs**
Task wake up (by interrupt)	**2.3 µs**	**11.5 µs**
Longest interrupt lock-out (by design)	**109 instructions**	**89 instructions**

Storage requirement

RTOS-UH is famous for its low memory requirements. The following table gives an impression of memory requirements:

	PowerPC		MC68xxx	
	RAM	ROM	RAM	ROM
runtime kernel	24 kByte	52 kByte	5 kByte	23 kByte
with user interface, file systems and network	450 kByte	800 kByte	430 kByte	320 kByte
each task	300 Byte		220 Byte	

Availability

IEP supports the deployment of **RTOS-UH**

- for the MC68xxx-family of MOTOROLA processors:
  - Microcontroller 683xx
  - stand alone systems based on 68000, 68010, 68020, 68030, 68040, 68060
- for the PowerPC-family
  - Microcontroller MPC5xx, MPC8xx, MPC8xxx
  - Systems based on MPC603, MPC604, MPC750, ...

For all kinds of processors, ready-to-run boards based on standard bus systems are available (VMEbus, CompactPCI).

RTOS-UH is also available for multiprocessor systems.

Development-environment

Apart from its first class realtime behavior, **RTOS-UH** offers complete support for software development.

All programming tools as well as compilers for PEARL and CREST-C are available either as generic tools or for cross development using the Microsoft Windows operating system.

Process and
Experiment
Automation
Realtime
Language

PEARL

Programmieren von
Echtzeit-
Anwendungen
Relativ
Leicht

Why PEARL

The extraordinary increasing of software costs especially in realtime applications demands the transition from outdated assembler programming to a higher level structured programming language. **PEARL** is the only application orientated higher realtime programming language world-wide. Independent problems can be programmed as independent processes (tasks) and executed in parallel – a substantial improvement within the field of control engineering.

A modern Concept

PEARL was born in the 1970's. A goal of the development promoted by the Geman Ministry of Research and Technology was the agreement on a language, which combines the most important elements of the common high-level languages with a concise concept of realtime and tasking. **UH-PEARL** is an implementation of this language for microprocessor systems, the development started in the beginning of the 1980's at the University of Hanover under the leadership of Prof. Dr. Ing. W. Gerth.

Easy to learn

PEARL is an easy-to-learn programming language suited in particular to solve realtime-oriented problems. It is, to differentiate from e.g. process FORTRAN, a monolithic language that directly integrates process I/O and time-oriented task scheduling. Thus, a high measure of portability is given.

Universally applicable

PEARL is a block oriented, structured language. It is universally suitable also to solve complex, algorithmic problems. Apart from all common programming language elements, **PEARL** integrates interrupt handling and synchronization objects.

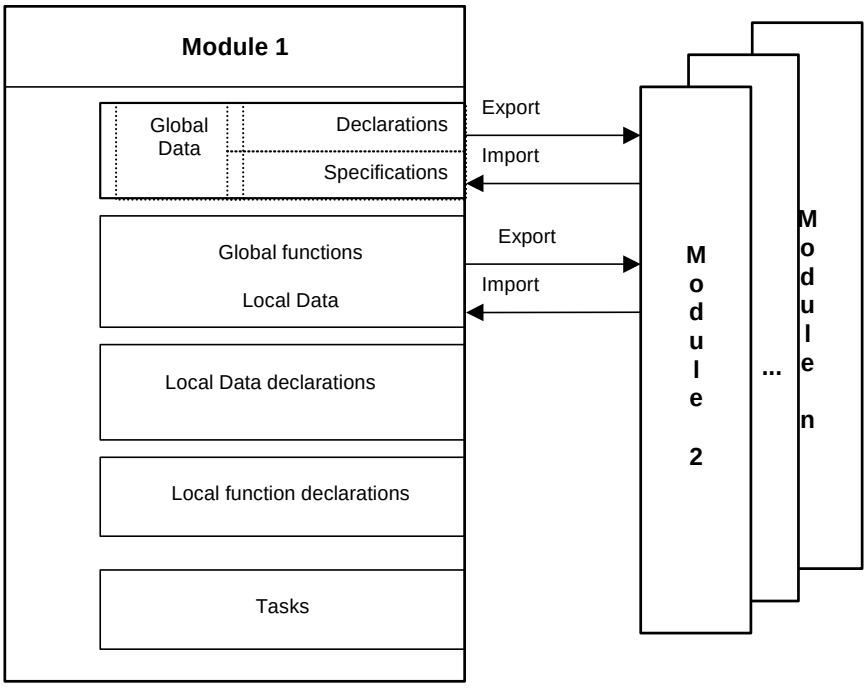
Language characteristics

A well formulated concept of multitasking, the support of all usual algorithmic control structures as well as concise realtime control statements are the special features of **PEARL**. During the language design, special attention was paid to support the writing of programs, that are easy to read and, therefore, easy to maintain, without restricting the developer or detracting him from his problem domain. Differently than e.g. Ada, **PEARL** is orientated towards the applications engineer and gives an easy start without a steep learning curve.

Modularity

The modular structure of **PEARL** programs lays ground for safe and efficient program development and easy maintenance within larger projects.

A Module is a self contained compilation unit, but not necessarily a self contained execution unit. The connection between modules is made by global declarations and the corresponding specification of global symbols. Inter-module relations can be satisfied either by an additional linker or by load-time linkage.



Portability

One **PEARL**-Module is separated in two parts, each of which may be omitted. A module starts with a SYSTEM-part, defining the system resources used in a system dependant manner. This part contains no code.

The following PROBLEM-part is system independant and may use only system services defined in either its own or another module's SYSTEM-part.

```
MODULE name;
    SYSTEM;
        hardware dependancies

    PROBLEM;
       Declarations
       Procedures
       Tasks

MODEND;
```

Block structure

The breakdown of a module into hierarchical blocks with local data allows for a program structure that mirrors the structure of the problem. Quasi parallel processing of tasks serve the ease of separating a problem in independant and simpler pieces of code. Procedures, which are reentrant and allow for recursion, provide for the hassle-free realisation of problem specific code libraries.

Control structures

PEARL supports all flow control structures common to modern programming languages:

```
IF condition THEN
...
[ELSE]
...
FIN;
```

```
CASE Variable
   ALT /* 1 */
   ...
   ALT /* 2 */
   ...
[OUT] others
   ...
FIN;
```

```
[FOR]
counter
  [FROM] start
  [BY] step
  [TO] end
[WHILE] condition
REPEAT;
  loop body
END;
```

Data types

Apart from the regular simple data types FIXED, FLOAT and CHAR, also common to other languages, **PEARL** provides the aditional data types CLOCK (time), DURATION (length of time), SEMA (synchronisation variable) and BIT (bit string) to allow for strong typing in the problem domain. New data types can be defined by problem-specific combination of elements of different basic data types into groups (STRUCT) and by own type declarations (TYPE).

Standardization

PEARL is standardized since 1981 in DIN 66,253, part 1, Basic PEARL, and since 1982 in DIN 66253, part 2, Full PEARL. PEARL is already used in over 200 large and many hundred small projects.

In 1998, with the standardization of PEARL-90 in DIN 66253-2, with caution the concept of the language was adapted to current requirements. The current advancement of **PEARL** tries to agree upon object-oriented programming procedures in combination with the safety and efficiency requirements of realtime programming.

Realtime statements

The simple time scheduling of program flow

```
AFTER 10 SEC
ALL 4 SEC
UNTIL 17:00:00
ACTIVATE control PRIO 6;
```

Scheduling the cyclic execution of the task control in a given time-frame

as well as the integrated interrupt scheduling

```
WHEN fire ACTIVATE douse;
```

to schedule task douse to execute when the interrupt fire is triggered allow for comprehensive self-documenting instructions.

I/O statements

In **PEARL**, input/output statements use so-called DATIONS. System-specific properties of dations are declared in a module's SYSTEM part; in the PROBLEM part, these dations are used in a portable manner. So, I/O of process values, e.g.

```
SEND Off  TO engine;
TAKE is_engaged FROM clutch_switch;
```

and file-oriented, alphanumeric I/O

```
PUT temperature TO  log_file;
GET target_height FROM console;
```

can be ported easily to different device configurations.

ROM code

The **PEARL** compiler can generate ROM-able code and supports targets without mass storage. At the start of the operating system, during the phase of self-configuration, programs in ROM are recognized, their RAM-Areas get initialised and their code ist executed directly out of the ROM.

Availability

IEP supports the deployment of **UH-PEARL** on all computers under the operating system RTOS-UH, based on e.g.

- the MC68xxx-family (MC68000 –MC68060, MC683xxx)
- the PowerPC family (MPC60x, MPC750, MPC5xx, MPC8xx,...)

The capabilities of these systems cover small embedded controls as well as high-powered multiprocessor-systems based on e.g. commercial off-the-shelf VMEbus-boards.

The **UH-PEARL** compiler is available either generic or as cross-compiler, runnable under all versions of the Microsoft Windows operating system since Windows '95.

Crest-C

ANSI-C for RTOS-UH

Why C

ANSI-C is one of the most flexible and most common programming languages at all. C does not target specific areas of application, but instead provides the programmer with all tools necessary to solve the problems at hand. With the standardization of ANSI-C, featuring strict type testing and prototyping, programming in C attains the security which is mandatory for the deployment under a multitasking realtime operating system.

CREST-C

CREST-C is especially designed to support the realtime operating system RTOS-UH. An expressed goal of this development was to make a reliable programming system available, which also provides for cross-development under numerous guest operating systems like e.g. Microsoft Windows, UNIX etc. Special attention was paid on compactness and efficiency of the generated code.

CREST-C provides a hosted implementation of the ANSI-C standard X3.159-1989 (ISO/IEC 9899:1990) and is also usable as free-standing implementation.

Versatile

CREST-C allows to port lots of already available sources to RTOS-UH and to reuse existing code also under the realtime-oriented environment of RTOS-UH. The successful ports of various free sources show the reliability of the **CREST** compiler**.**

System programming

Using C, even extremely time-critical driver programming can be done using a high-level language instead of assembler. Coding of interrupt handlers and their integration within the self-configuration of RTOS-UH are completely supported.

Tasking

As hosted implementation of the C89-standard, **CREST-C** generates code following the RTOS-UH model of a shell-module on default. These modules can be called by the command interpreter and provide for parameter transfer by the command-line. Each call generates a new instance of the module, so multiple instances can act in parallel. The multi-user model of RTOS-UH is supported.

The generation of single-task programs is possible also. Tasks and subtasks can be generated at runtime, the coding of system tasks and interrupt handlers is supported.

Realtime behavior

CREST-C was developed particularly according to the specifications of the RTOS-UH operating system and provides all realtime and multitasking possibilities of the system by the means of a runtime-library. By the excellent code quality of the **CREST-C** compiler, there is no reason to code in assembler. Nevertheless, an inline-Assembler is included.

ROM ability

Aside from loadable code, **CREST-C** can generate ROM-able code. A linker can be used to bind the objects to a given base address and generate a binary image, which is directly rommable. Depending of the target processor, even position independant code can be generated.

To conserve ROM space, frequently used functions can be combined to a shared library. The linker can be instructed to bind modules to the shared library, so the library can be used simultaneously by multiple modules.

PEARL interface

To the PEARL-programmer, **CREST-C** offers the possibility to migrate to a more flexible language concept. S-Records, generated by the UH PEARL compiler, can be linked with S-Records generated by **CREST-C**. Both sides can benefit: PEARL-programs can use proven C-software, as well as C-programmers can resort to established PEARL-librarys.

Target systems

CREST-68K supports processors of the M68K-Familie; a floating point unit, if available, is supported:

- MC68000, MC68010, MC68302...
- MC68020, MC68020/MC68881, MC68030/MC68882...
- MC68040, MC68060
- CPU32, CPU32+

CREST-PPC supports processors of the power PC family:

- MPC603, MPC604, MPC750 , ...
- MPC5xx, MPC8xx, MPC82xx. ...

Libraries

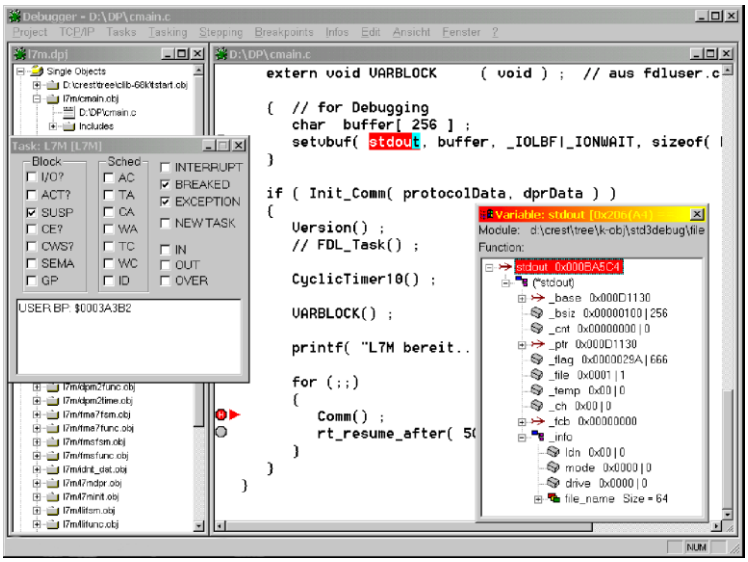
ANSI-C standard libraries for the respective processor family are in the standard scope of supply. They are delivered in different translation variants, so for each application, the user can select the appropriate library.

Cross development

CREST-C is available either as generic compiler, running under RTOS-UH, or as cross-compiler for all 32-bit Microsoft Windows operating systems since Windows 95. All tools for a complete development cycle are included: the sources can be translated to either loadable or rommable objects. Testing and debugging takes place on the RTOS-UH-target, a debugger is available separately.

RT-Debug

Source level debugging for ANSI-C and PEARL



Remote Debugging

Remote debugging gives the full comfort of a graphical environment during the program development targeting small systems. The separation between an efficient user interface and a small debugger kernel assures the almost undisturbed program behavior on the target system.

With a connection to the target system through standard networks, even remote debugging over the internet is possible.

Source level

The analysis of program behavior is constantly done on source level. Whether a program is written using Crest-C or UH-PEARL, **RT-Debug** has access to all program objects and considers the characteristics of the individual languages.

The program flow is shown in the source code. Access to variables is strongly typed, also for user-defined data types. **RT-Debug** is aware of the multitasking environment and hands full control of the programs execution under the realtime operating system RTOS-UH to the user.

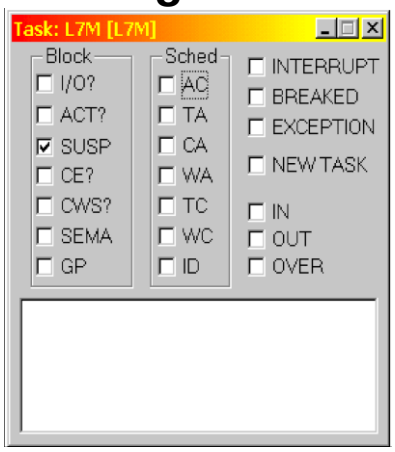
Break- and watchpoints provide for detailled examination of a programs state and flow with minimal disruptions.

Crash analysis

Even in the case of catastrophic program aborts such as bus- or address errors **RT-Debug** offers support.

Callstack and backtrace allow to inspect the program behavior before a crash. At each point in the callstack, the program state is displayed in the correct context. Seeing the valid values of variables eases the detection of either algorithmic or tasking-based programming errors as far as possible. For special cases, access to assembler code, register contents and administrative task data is provided.

Tasking control



RT-Debug shows the current state of the task under control in a task state window.

The continuous display of the task state provides a precise insight into the runtime behavior. Special events are logged in a message pane.

A task can be interrupted at any time. The current program position is graphically shown in the source code window.

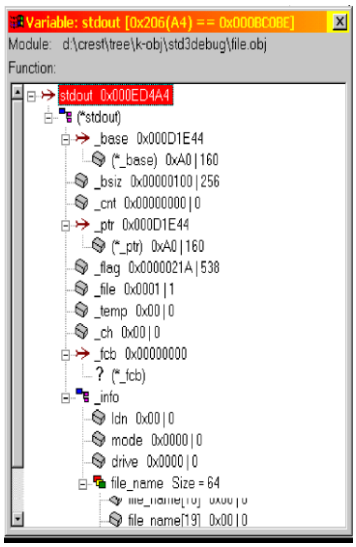
For exact control of the program flow, controlled program execution is provided by the instructions:

- Step in – executes the program in a single step mode
- Step Out – stops at the return from the current procedure
- Step Over – stops after returning from a procedure call

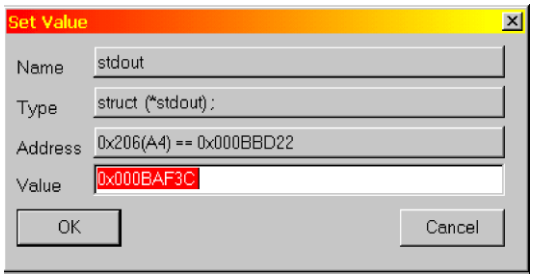
Breakpoints for systematic interruption of program execution are set directly in the source code window.

Watchpoints allow to take a snapshot of all variables currently in scope with minimal disruption of program flow.

Quick Watch



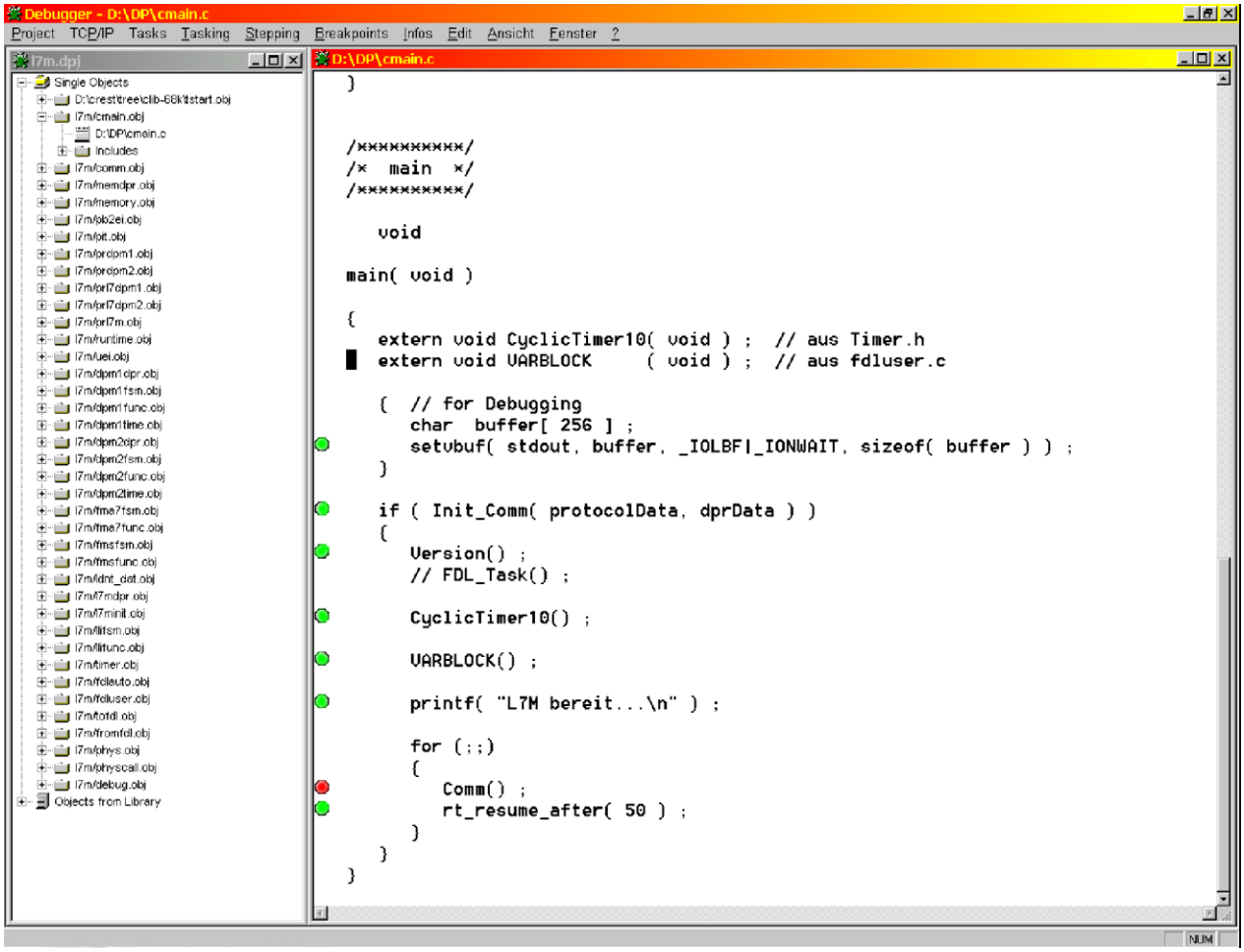
The values of all local and global variables can be inspected using the quickwatch window. Variables of complex data types are shown in a tree view, selective opening of sub-elements gives a quick overview and simple access to member variables. Different symbols for different data types give a concise view, even when working on larger projects.



A Quick Watch dialog shows all available information about individual variables or members of structured data and allows to change their values.

Project Organization

RT-Debug presents all available project information in clear, hierarchical form. Dependencies between the individual translation units as well as the linked libraries are visualized in order to ease the navigation in the source code.



The individual source files are accessable simply and quickly by navigation in the project tree. The simultaneous display of several source code panes gives an optimal view to the program flow.

Break- and Watchpoints are shown in the source code, different colors allow to differentiate between possible, active and hitten points.

When a breakpoint is hit, the program execution is suspended, all quickwatch windows are refreshed and the current point of execution is shown graphically in the source code.

On the hit of a watchpoint, program execution is interrupted only to refresh the quickwatch windows and is resumed immediately thereafter.

Target systems

RT-Debug is available for all systems based on RTOS-UH using either PowerPC or the 68xxx-family. A small debugger kernel in the target system communicates with the comfortable user interface using the TCP/IP protocol. The target can be connected either serial or by network.

The debugger kernel provides basic debugging functions. Apart from the manipulation of storage areas he sets or resets breakpoints, observes task condition changes and recognizes special events during task execution. All actions are initiated by the development computer.

The separation of the debugger into a kernel with elementary basic functions and a comfortable user interface on a commonly used workstation leads to a very small load of the target system. The target system does not have to fulfill special requirements regarding available memory or computational power. Even programs targeting small systems with little or no disks can be debugged comfortably without special hardware support. No external debugging tools are needed.

Development system

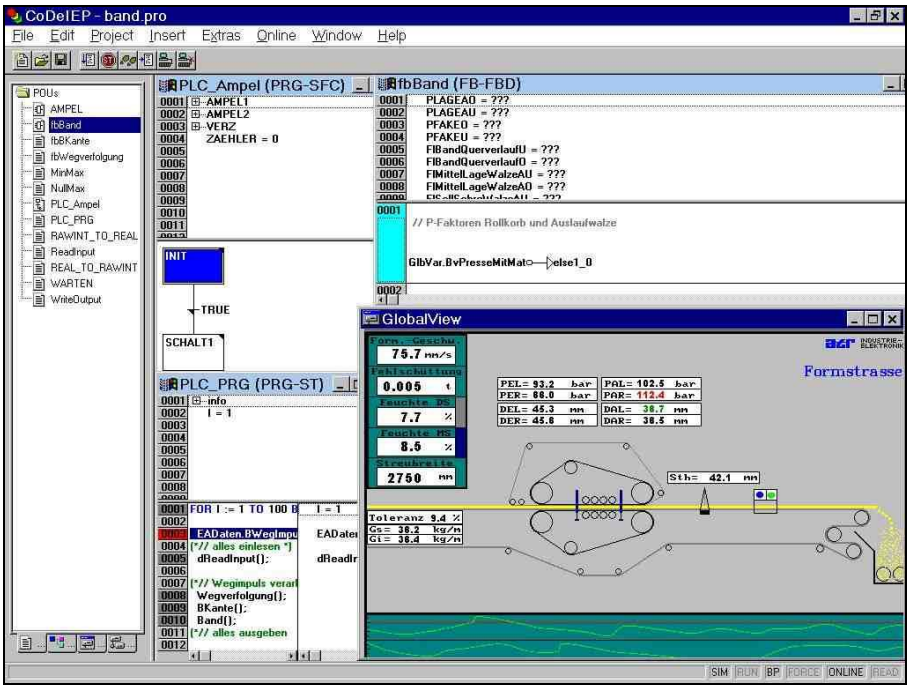
The development computer presents the main functionality of the debugger. It translates the source files, analyzes and interprets the debug informations of the compilers and gives a concise view of the program flow. The user interface follows the Look and Feel of the operating system.

All versions of the Microsoft Windows desktop or server operating system since Windows 95 are supported.

Coupling the target systems to the development system by network allows to separate the location of target from the workstation. Debug sessions can be made even over the Internet.

CoDeSys

Soft PLC and hard realtime
For all tasks of automation



Capabilities

CoDeSys consists of two parts: a complete graphical PLC software development environment, runnable under Microsoft Windows opearating systems, and a PLC runtime kernel for the **RTOS-UH** realtime operating system. **RTOS-UH** guarantees for a stable and proven runtime environment for the **CoDeSys** kernel, featuring:

- PLC programming according to the world standard IEC 61131-3, with all 5 languages: SFC, ST, IL, LD and FBD,
- IEC tasks with preemptive multitasking
- Integration of ANSI-C and PEARL

CoDeSys combines a PC's comfort and ease-of-use with the flexibility of a PLC and the reliability of the realtime system **RTOS-UH**.

Program development

With **CoDeSys,** a broad spectrum of efficient tools for program development is at hand. Programming is possible on-line as like as off-line. An integrated PLC-simulator allows to test critical program sections offline without interrupting production systems.

Integrated Editors

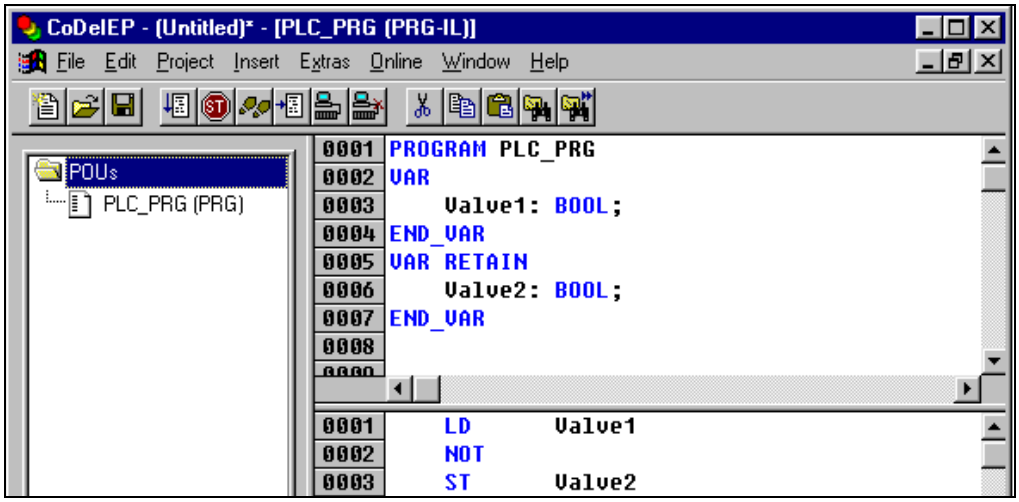
The integrated editors are providing easy programming by

- automatic formatting of the program source code
- syntactic colouring of language elements
- smooth integration into the GUI-concept of the development operating system

All 5 programming languages prescribed in IEC 61131-3 are supported.

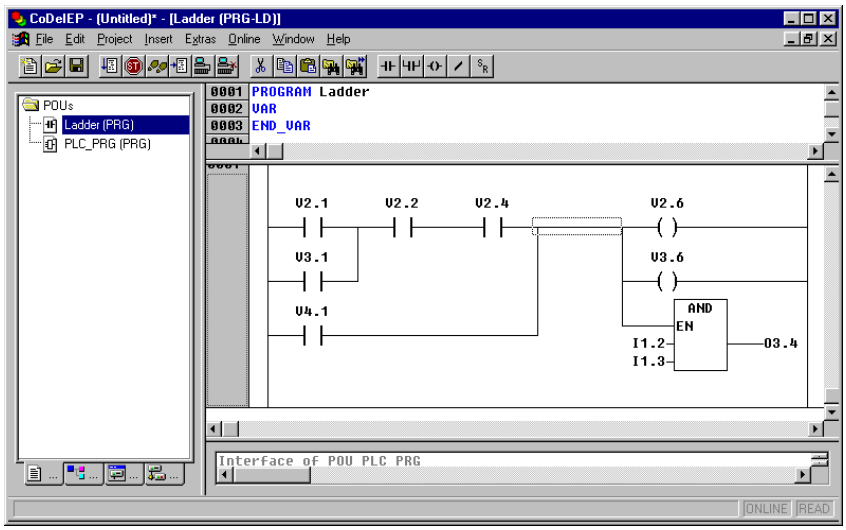
IL Instruction List

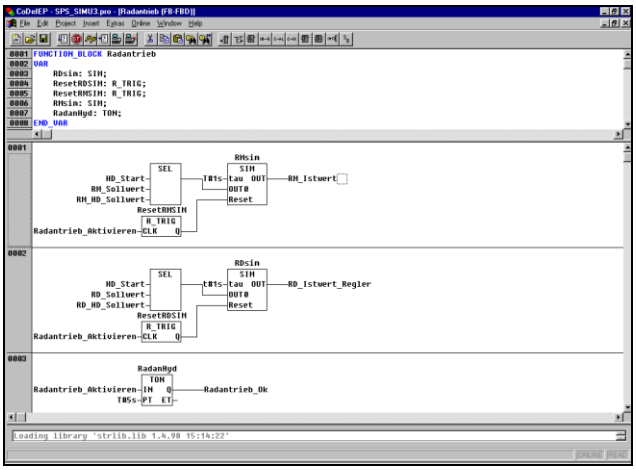
Basic language
of all controls



LD Ladder Diagramm

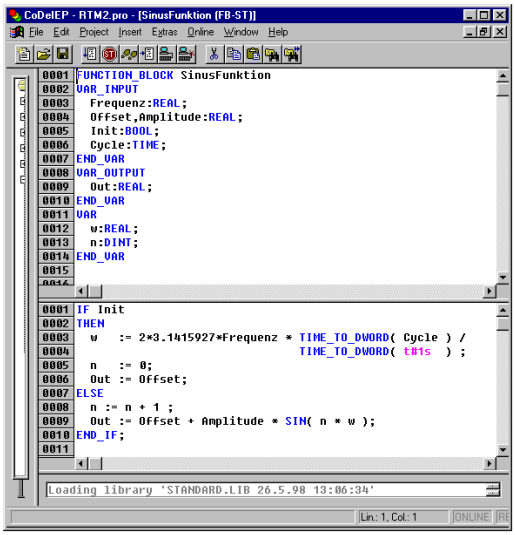
Descriptive graphic
representation
of relay logic





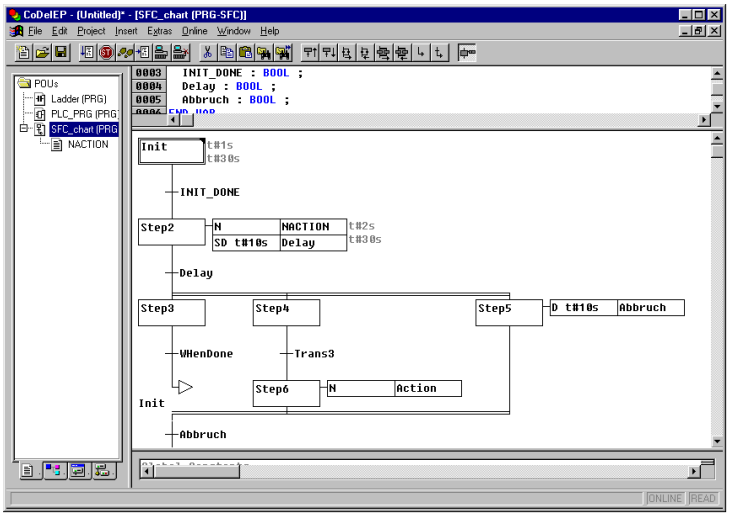
FBD Function Block Diagram

Visual representation of procedural programs



ST Structured Text

The new High-level language of the PLC



SFC Sequential Flow Chart

Graphical oriented programming showing states and state-transitions

Test Debugging

All modern programming tools are at hand:

- monitoring of input/outputs as well as of internal variables, even with the control being online
- detailled supervision of the PLC by single-cycle or continous forcing of variables
- online-changes in order to change the PLC program without interrupting a running process
- single-cycle of the control
- Inspection of the PLC's state at discret program steps by breakpoints
- full flow-control by single stepping the control from statement to statement
- state visualisation with continous display of line states and program flow
- watching of variables (with tracing of previous cycles) to catch sporadic error conditions

Operating and Visualization

Operating and graphical display is provided by the **CoDeSys** user interface:

- setting of operating conditions by batch processing and recipe administration
- visualization of the state of program and plant
- charting and archiving of plant data by variable trace

The control is operated independantly from the user interface. Headless and manual operating are supported.

Availability

CoDeSys is available for all **RTOS-UH** systems from embedded controls up to multi-processor COTS-systems with an identical behavior. For each application, an optimal trade-off between cost and capabilities of the control can be found without the need to change the operating environment. PLC software runs under the realtime operating system **RTOS-UH:** high reactivity and dependable cycle times are guaranteed, full priority control and preemptive tasking is provided.

The PLC program can use all system resources supported by the operating system: (non) removable disk, network, field busses like Profibus, CAN or InterBus are supported.

Combining PLC-programs with already existing ANSI-C or PEARL programs is integrated completely into the run time kernel.

CoDeSys for RTOS-UH is an adaption of the IEC 61131-3 development environment CoDeSys to the special possibilities of the realtime system **RTOS-UH.** CoDeSys is a product of the 3S Smart Software Solutions GmbH.

RT-LAN

Networking software for RTOS-UH

Network

RT-LAN is a modular program package for RTOS-UH to provide for communications over Ethernet. The commonly used TCP/IP protocol gives interoperability with different computers using different operating systems.

A well designed application programming interface eases the addition of networked functionality to control systems. **RT-LAN**s modular design allows the use of multiple network interfaces simultaneously.

SLIP

With the ***S**erial **L**ine **IP*** a point-to-point network communication beteen two computers can be established. Several services can run at the same time using only one serial connection.

UDP

The ***U**ser **D**atagram **P**rotocol* allows for immediate communication with low overhead, but does not provide a reliable communication path. The sender is not informed about the receiption of data packages. The loss of data packages is possible, and it is not verified, that packages are received in the order they were sent. Nevertheless, the contents of a received package is guaranteed to be correct.

TCP

The ***T**ransmission **C**ontrol **P**rotocol* is a connection oriented protocol, that guarantees either to fail or to deliver correct contents in correct sequence. As drawback, the timing of communication is not dependable. Most internet services are based on the TCP protocol.

FTP

The ***F**ile **T**ransfer **P**rotocol* is based on TCP/IP and provides the transmission of files. The **FTP** uses the client/server-model:

- The **FTP** server provides remote access to the local file system. Sending as well as receiving of files is supported.
- The **FTP** client uses the service provided by an **FTP** server. A command interpreter initiates file operation according to user input.

FTP is a presentation level application using TCP/IP. **FTP** is commonly used as least denominator for file exchanges between different operating systems.

SMB

The **SMB** protocol is specifically designed to communicate with computers using a Microsoft Windows operating system. It provides file and printing services. The file system of a computer running RTOS-UH is accessible by the graphical user interface of a PC.

Telnet

The **Telnet** protocol provides remote access to the command interface. A computer running RTOS-UH can be operated remotely.

- the **Telnet** server provides access to a local command shell. In the standard configuration, RTOS-UH provides upto 5 logins simultaneously.
- the **Telnet** client provides access to the command interface of a remote system.

Telnet is a presentation level application using TCP/IP.

Web server

A web server is commonly used to deliver either static or dynamic contents to clients using the http-protocol in conjunction with contents formulated in the HTML-language. In contrast to e.g. **Telnet** or **FTP**, the web server uses only short living connections and, therefore, uses less resources on the server.

The RTOS-UH web server allows delivering of dynamic contents with a programming interface accessible to application programmers using ANSI-C or PEARL90.

Profibus

The ***Process Field bus,*** FMS, is a deterministic field bus, standardized in EN50170, for communications on cell level. It is based on the dependable timing of a token bus and provides both multimaster and Master/Slave communications. RTOS-UH supports Profibus-FMS with drivers for the MC68302 or MC68360-processors at data rates upto 1,5 MBaud with economical 3-wire RS-485 cabling.

Profibus-DP is used as sensor/actuator bus with data rates upto 12 MBaud in direct competition to the InterBus-S. RTOS-UH supports Profibus-DP with a driver for the Siemens ASIC.

InterBus-S

The InterBus-S is a highly efficient sensor/actuator bus with strict deterministic timing, developed by the Phoenix contact company. A broad range of I/O-modules is available from various manufacturers. RTOS-UH supports the InterBus-S by drivers for different interface boards, e.g. for the VMEbus.

Our modular controller system MOCS-1100 (on basis of the MC68332) provides an InterBus-S master with remote and local bus connection.

OSI SINEC-H1

On the same physical Ethernet interface, RTOS-UH supports an OSI protocol stack as well as TCP/IP. With the use of the OSI protocol stack, RTOS-UH can be integrated easily into e.g. a SINEC-H1-network as commonly used by the Siemens S5 PLC. RTOS-UH supports OSI-layer-4-communication.

Systems
OEM-Boards

TC3004

Compact controller with touch screen

The **TC3004** is a compact controller in a standard plug-in housing, designed for controlling refrigeration or heating systems. Visualisation and operation are carried out via a 5 "colour display with touch-screen, Ethernet and USB are used for system integration.

TC3004

Signal connections for process signals:

- 4 temperature inputs PT1000 / KTY, 2- or 3-wire
- 1 current output 0... 20 mA,
- 1 frequency output 0... 150 Hz for controlling an inverter
- 3 switch inputs, 1 door connection via EasyLock

Signals

Power outputs for low voltage devices:

- 6x TRIAC, output power 230 VAC, 16 A
- 4x TRIAC, output power 230 VAC, 0.9 A
- 1 potential-free alarm contact 0.5 A

Power

Temperature inputs

The **TC3004** measures up to 4 temperatures via PT1000 sensors in two- or three-wire technology, i. e. with compensation of the conductor states. The alternative use of KTY sensors is possible.

Digital inputs

3 nc/no contacts can be connected by 2 wires each. The TC3004 detects door contacts or pressure switches via these inputs.

Power outputs

The **TC3004** offers a total of 10 power outputs 230 VAC for controlling heating and cooling units:

6 outputs, loadable with 16 A, for power consumers

4 outputs, loadable at 0.9 A, for auxiliary units

The outputs use zero crossing switches to minimize interferences.

A digital frequency output 0... 150 Hz enables the power control of frequency inverters, for example.

Notlauf, Alarme

The **TC3004** checks the temperatures for compliance with predefined limit values. Values above or below these limits trigger an internal signal transmitter and a potential-free alarm contact.

If the **TC3004** is equipped with a backup battery, it records the temperatures even in the event of a power failure and the alarm system remains active.

Operating

A colour graphic display 5" with a resolution of 800x480 pixels is supported by a graphic library and enables visualisation of the process. It is operated via a touch screen.

Network Archive

The **TC3004** records the temperature profile to archive files. The archive files can be exported to an USB stick on site.

Ethernet integrates the **TC3004** into the network and allows remote access to the archive.

A serial interface RS-485 is available for local networking of several controllers or for connecting external systems.

Construction

The **TC3004** is supplied in a slide-in housing according to the standard grid for panel mounting. For supply, a mains voltage of 100 - 240 VAC is required. The internal power consumption of the TC3004 is 10 W.

All signal connections are routed at the rear via plug-in/screw terminals.

Programming

The **TC3004** is based on a powerful PowerPC processor. The RTOS-UH real-time operating system is part of the standard scope of delivery. Crest-C, PEARL and IEC 61131-3 programming environments are available.

UCT

Mini control unit with grafic display

UCT

The **UCT** is a small controller for on-site mounting in low voltage equipment, e.g. for the control of heating and climate systems. For operating and visualisation, an OLED-display with a touch panel is integrated. With CAN as field bus, the UCT connects to a network and optionally controls additional I/O-modules.

Signals

Connections for process signals:

- 16 analog inputs, 10 Bit, PT1000 / 0..10 V / 0..20 mA
- 2 analog outputs, 0..10 V or 0..20 mA
- 8 digital inputs 24 V, optoisolated, for reading switches

Power

Power outputs for low voltage devices:

- 4x 230 V_{AC}, 2 A, with Solid State Relays
- 4x 230 V_{AC}, 2 A, with relays
- 2x relay, 2 A, floating change-over contacts
- 2x relay, 2 A, floating normally open contacts

Analog in- and outputs

In the standard configuration, the analog inputs of the **UCT** are configured for temperature sensing by PT1000-sensors. Each input can also be configured as current or voltage input (0..20 mA or 0..10 V).

The analog outputs are capable of delivering currents 0..20 mA, with an appropiate shunt, they can also be used as 0..10 V voltage outputs.

Digital inputs

The digital inputs of the **UCT** are optically isolated and share a common ground. They are designed to read ground switching sensors, e.g. switches.

CAN-Bus

The CAN-Bus is galvanically isolated. The **UCT** can be used a control unit (master) as well as intelligent I/O-Module. Baudrates 50 kB upto 1 MB are supported, CANopen ist optionally available.

Power outputs

The **UCT** provides 8 switched line outputs 230 V_{AC}, 2 A, with fast fuses. 4 of them are switched by SSRs, 4 are optionally fitted with plug-in relays or SSRs.

4 floating outputs by plug-in relays (2x normally open, 2x change over) are supporting loads upto 230 V_{AC}, 2 A.

Operation

An OLED grafic display 2,8“ with a resolution of 320x240 pixels is supported by a software library and allows process visualisation. The display comes with a touch panel for user input, e.g. by a stylus or a pen.

Size and supply

The **UCT** comes as ready-to-use module for screw mounting in an ABS case with an IP30 protection class. It measures 260x230x120 mm (LxWxH) and is supplied by 230 V_{AC}.

For wire connections, cage clamps are used and are supporting solid as well as stranded cables.

Programming

The **UCT** is based on the Motorola MC68332 microcontroller. The Realtime Operating System RTOS-UH is in the standard scope of supply. For programming, a Crest-C- and a PEARL-compiler as well as an IEC 61131-3 wokbench are available.

Options

An internal extension port is available for the mounting of different communication modules, e.g. for:

- Ethernet
- WLAN
- Modem (V92, ISDN, GSM)

All in- and outputs are configurable according to customer specifications, even in small lots. Modules in special configurations are available at low cost, please contact factory.

MOCAN DK

Networkable CAN control unit

The **MOCAN-DK** is designed as decentralized mini-controller. A galvanically isolated CAN-Bus (2.0B) provides for I/O expansion, 2 serial ports are in the standard scope of supply. Optional Interfaces, a 10/100 Mbit Ethernet, Arcnet and an additional CAN Interface, facilitate easy integration of the **MOCAN-DK** in larger control systems.

CAN Control

Two digital highside outputs 24 V / 0.5 A as well as 4 digital inputs 24V are galvanically isolated from the controller core. One 7-segment LED and a pushbutton allow for local operating

Basic I/O

The **MOCAN-DK** is based on the powerful MC68332 microcontroller. With the realtime operating system RTOS-UH and the programming languages ANSI-C, PEARL or IEC 61131-3 even complex control problems are solved quickly and reliably.

With Realtime

The **MOCAN-DK** is supplied ready to run in a sturdy steel sheet metal housing (227x57x102mm) for wall attachment (mounting plate) and requires a supply of 12..30 V_{DC}.

Serial Interfaces

Both serial ports of the **MOCAN-DK** can be used for programming or data exchange:

- 1 interface RS-232, 3-wire, upto 115 kBaud
- 1 interface RS-232, 5-wire, galvanically isolated

Memory

The memory configuration of the **MOCAN-DK** can be adjusted to fit the requirements of the application:

- 1 (optionally upto 4) MByte FLASH, on-board programmable
- 1 (optionally 2) MByte SRAM, buffered by a back-up capacitor (approx. 1 month) or by a lithium battery.

Programming

The realtime operating system RTOS-UH with drivers for all system components is in the standard scope of supply. You can begin immediately with the actual application development – compilers and run time support for ANSI-C, PEARL-90, IEC 61131-3 or 68K-Assembler are available.

Options

I/O

Additional inputs and outputs for internal mounting in the **MOCAN-DK** provide:

- 8 opto-isolated inputs 24 V
- 6 opto-isolated High Side outputs 24 V / 0.5 A, short circuit protected and capable of driving inductive loads.

All connections are made with plug/screw-terminals.

Ethernet

A RJ-45 connector for the 10/100 MBit Ethernet interface of the **MOCAN-DK** assures problem-free integration both into standard networks as well as into Industrial Ethernet.

A TCP/IP stack is in the scope of supply; FTP, Telnet, http are available. An optional OSI stack serves for the integration into control concepts based on e.g. SINEC-H1.

2 LED's (link, Activity) show the status of the Ethernet interface.

Arcnet

A 2,5 MBit Arcnet-interface with HIT-transceiver is provided by a 9p-SubD connector. A packet driver is in the standard scope of supply.

CAN

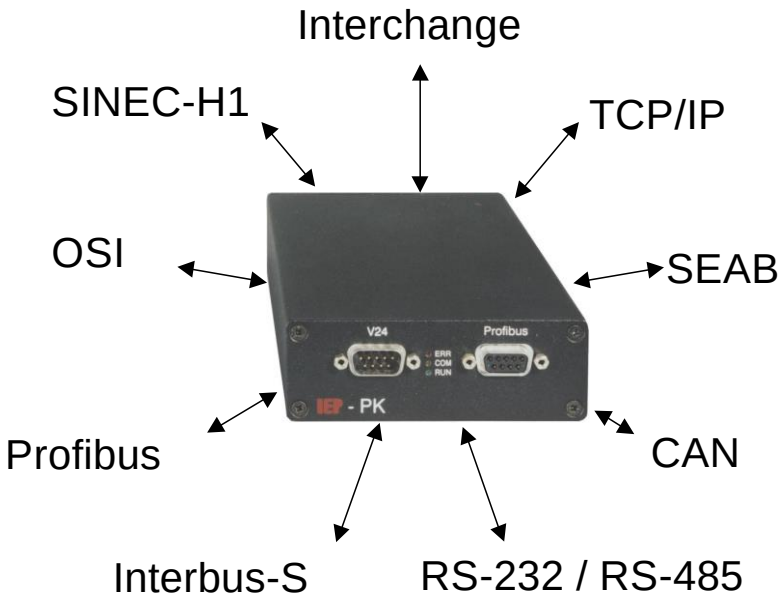
A second, isolated CAN interface (2.0B) allows simultaneous integration of the **MOCAN-DK** into different CAN-segments.

Realtime clock

An optional real-time clock is buffered by the on-board buffer capacitor and supplies time and date.

IEP-PK

The family of protocol couplers



Consistent operating is one of the main goals of modern SCADA systems. Vendor independence and an unvarying operating philosophy are trying to assure investments on a long-term basis and to ensure short training periods for the service personnel.

Long lasting manufacturing plants are characterised by large variety of control components used. The deployment of different controls, optimised for the respective purpose, assures low cost and high product quality.

A qualified technical management integrates different systems without renouncement of their specific efficiency.

IEPs process couplers provide standard interfaces both to individual controls and to different SCADA systems. Combining commodity software and free programmability, they allow to preserve proven components despite of changes in the operational and data processing environment.

Theory and

Practice

Coupled

Database

The database is the quintessential point of a process coupler. A unique, internal data base, supporting simple and complex data types, allows to present the same data differently to different SCADA or control systems.

Different data descriptions for different automation systems are not a problem: commodity software for conversion is available, special cases are taken into account by the free programmability.

Server

Connected to different SCADA systems, the process couplers present themselves as servers, just like every other control. They provide access to process data and receive input data and operating commands from the SCADA system.

All commonly used communication protocols of the respective manufacturers are supported.

Client

Connected to different controls, the process couplers present themselves as client, like every other SCADA system. They request process data and transfer operating parameters and commands to the control.

All commonly used communication protocols of the respective manufacturers are supported.

Flexible

The free programmability of the process couplers allows the integration of non-standard components.

Thus also the integration of older systems with proprietary protocols into modern production plants is feasible. Process data, fomerly only acquired and processed locally, can be distributed throughout the network to be used by e.g. production steering, quality assurance and archiving.

Scalable

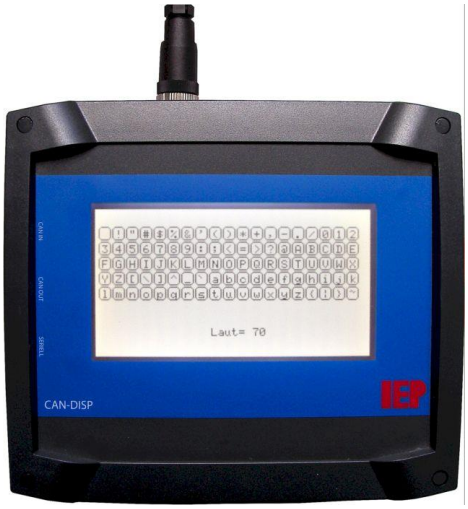
Process couplers are available in a broad range of capabilities and computational power. Starting by simple and economical systems, as shown on the front side, upto multi-processor VME systems: they cover all requirements.

Already in the simple versions, all commonly used physical interfaces (RS-232, RS-422, RS-485, Ethernet) are available and supported by the respective protocols.

More capable systems differ only by the number of available interfaces, the size of the database and the computational power.

CAN-DISP

Grafical operation panel for the CAN-Bus



CAN-DISP

The **CAN-DISP** is a rugged operating module for in-the-field deployment in control systems. A high-contrast 5,5"-LCD with a resolution of 240x128 pixel is readable even under adverse conditions, a solid front foil protects the display when used in an harsh environment.

Inputs

For operator inputs, the **CAN-DISP** provides a touch panel as well as an connector for matrix keyboards. An integrated speaker gives acoustic feedback and is controled by software in volume and frequency.

The touch panel can be used as a replacement for a keyboard as well as for grafical input.

OEM

As OEM-module, the **CAN-DISP** is also available in customer specific housings. Different configurations are supporting:

- displays based on the T6963-controler
- matrix keyboards upto 8x8 keys
- power pupplies 5-30 V_{DC}
- resistive 4-wire Touchpanel with a resolution of upto 12 Bit
- application specific firmware

Safety and reliability

The CAN-bus interface is galvanically isolated from the power supply. The supply can be fed via the CAN connector or using a separate input.

An integrated EMC protection circuitry assures trouble free operation even in industrial environments.

CAN-Bus

The **CAN-DISP** supports baudrates from 50 kB upto 1 MB. On the CAN-Bus, 4 succesive identifiers are used, baudrate as well as base identifier are configured by an internal DIP-switch.

The bus interface uses industrial M12 connectors. Both connectors are directly joined, allowing feed-through connections as well as stubs from the bus system.

RS-232

The serial port of the **CAN-DISP** supports baudrates upto 115 kBaud. The interface is a 5-wire port with RTS and CTS for handshake.

Firmware

The standard command set of the **CAN-DISP** supports displays based on the T6963-controler operating in text- or grafic mode. The base command set supports text output, including visual attributes. In grafic mode, additional drawing operations (line, circle,...) are available.

A resistive touchpanel is supported for grafical input. By definition of sensitive regions, it can also be used as a replacement for keyboard input.

Size and supply

The **CAN-DISP** comes in a sturdy aluminium cast case for direct srew mounting, either onsite or e.g. on a mounting plate, and measures 217x119x56 mm (LxWxH). The power supply of 8-30 V_{DC} can be fed via the CAN connector.

A board version of the **CAN-DISP** is available as OEM-modul for mounting in customer housings. The module is based on a euro card 100x160 mm, mounting in a 19"-rack is possible.

The visible area of the display measures 123x70 mm (5,5“).

The different configuration options apply to:

- Size of display
- Backlight (dimmable, if LED)
- Power supply
- Communication interface
- Keyboard configuration upto 8x8 keys
- Firmware (command set and communication)

Special configurations are available at low cost, even in small quantities. Please contact factory for details.

CAN on site

Power modules – CAN in the field

Our power modules are versatile process interfaces for on-site control systems in low voltage applications. **CAN-MIO**s and **CAN-HMx**s widen the versatile functionality of our IO-modules by providing power outputs, resulting in complete single-box control modules for on-site deployment.

Kind and count of process signals, switching capacity, casing, operating temperature range and environmental protection class are selected according to the applications demands.

CAN-MIO
CAN-HMx

Casing and connection technology satisfy harsh industrial requirements. M12-plug-in connectors are standard for process signals, modular configurable HAN-Connector allow for flexible interfacing to different device and plant configurations.

To satisfy manifold environmental requirements, we offer the modules in solid aluminium die casting or steel sheet metal casings, and also in different protection classes.

Tough

Signals

Process signals via M12 industrial connectors:

	MIO	M2IO	HM	HMR	KKD01
• digital inputs 24 V_{DC}	2	4	6	6	4
• high-side outputs 24 V_{DC}, 0,5 A	2	3	-	-	3
• analog inputs 0..20 mA	2	3	-	-	3
• analog inputs Pt100,	3	6	6	6	6
• analog outputs 0..20 mA	2	2	-	-	2

Power

Galvanically isolated power outputs 230 V_{AC}, HAN-Connectors

Current rating (via zero voltage turn-on SSRs)	2 A	10 A	16 A	16 A	10 A
• 1-phase AC outputs	3	6	6	3	6
• 3-phase AC outputs				1	
• AC line input, # of phases	1	3	3	3	3

CAN-Bus

The CAN-bus interface is galvanically isolated. On the CAN-bus, 5 succesive identifiers are used. Baudrate as well as base identifiers are configured by an internal DIP-switch. The bus interface uses two industrial M12 connectors. Both connectors are directly joined, allowing feed-through connections as well as stubs from the bus system.

Digital in- and outputs

The digital signals are galvanically isolated from the power supply and the bus system. The outputs can be used for inductive loads and are protected against short circuits and excessive temperatur.

Analog in- and outputs

In the standard configuration, the analog in- and outputs are implemented as current interfaces for the industrial 0..20 mA range A flexible circuit design allows different customer specific configurations even in small lot sizes.

The signals are routed via industrial M12 connectors.

Mounting and supply

The power modules are designed for wall mounting. Line power supply is fed via a HAN input connector and, to provide feed through, internally connected to a HAN output connector.

The modules require a control voltage supply of 18..36 V_{DC}.

Options

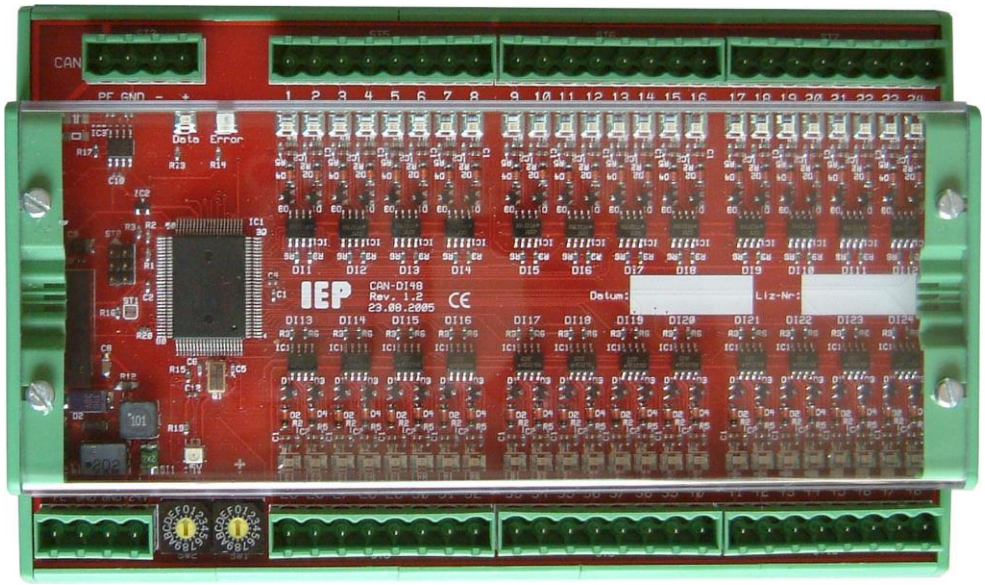
Application specific configuration is possible, especially:

- flexible analog in- and output configuration
- CAN-telegramm structure, optional CANopen®
- Signal preprocessing

User specific modules are available even in small lots, please contact factory for details.

CAN-DI48

Digital Inputs for the CAN bus



CAN-DI48

The **CAN-DI48** is a low-cost module with digital inputs for control systems based on the CAN bus. With 48 inputs, it is especially well suited for systems acquiring an exceptionally high number of digital input signals.

Features

Features of the **CAN-DI48** are:

- 1 CAN interface
- 48 optically isolated digital inputs with counter function, for 24 Volt signals, common ground
- Supply, 24 Volt, galvanically isolated
- Configuration via hexadecimal coding rotary switches
- Snap-on casing for DIN-rails

Digital Inputs

All 48 digital inputs are galvanically isolated by optocouplers. They are designed to acquire high-active signals. The switching threshold lies at 14 V, the input current at 24 V_{DC} is about 3 mA. Differing input configurations are possible, please contact our sales department for details.

Aside from the direct input state capturing, the **CAN-DI48** samples the inputs all 20 ms and counts the level changes. When transmitting the input states, the **CAN-DI48** can also report the number of state transitions since the last data request.

CAN bus

The **CAN-DI48** supports baudrates from 50 kB upto 1 MB and uses 14 consecutive identifier on the CAN bus. Baudrate as well as the basic identifier are configured by 2 hexadecimal coding rotary switches. The connection to the CAN bus is available on a 4p pluggable srew-clamp terminal.

Size and Supply

The **CAN-DI48** comes in a casing for DIN-rail mounting, measuring 182x110 mm with a height of 45 mm. All signal connections are made by 8p pluggable screw-clamp terminals.

The **CAN-DI48** uses a supply of 18 - 36 V_{DC}. The supply is connecting by a 4p pluggable srew-clamp terminal. The modules are protected from power supply polarity reversal; an EMV protection circuit assures troublefree operation in an industrial environment.

Versions

The **CAN-DI48** is customisable even in small quantities. Customer specified changes are in particular available concerning

- configuration of the input circuitry
- specialised firmware
- supply voltage ranges

Please contact our application support department fur further technical advice and support on specialised solutions.

CAN- modules

Analog und digital modules for the CAN-Bus

The **CAN-xx** modules are a line of durable and inexpensive I/O building groups for industrial control. A clear and sensible CAN communication structure simplifies their deployment. Optionally, the modules are available with CANopen® support.

CAN-xx

Module

DIO8 8 highside outputs 24 V / 0.5 A with PWM 0%...100%, galvanically isolated, can drive even inductive loads.

8 isolated digital inputs 24V with edge counter mode.

AI8 8 differential analog inputs, resolution 10 bit, input ranges 0-10V, ±10V, 0/4-20mA, PT100 etc. The versatile input circuitry allows for easy adaption to different signal level requirements.

By an automatic sample mode, the actual samples as well as their floating averages can be read.

AI8-16 8 differential analog inputs, resolution 16-24 bit, input ranges 0-10V, ±10V, 0-20mA.

Accuracy and sample frequency are user configurable. The simultaneous conversion of two different input channels is possible.

AO4 4 analog ouputs; 12 Bit; 0-10V; ±10V; 0/4-20mA

Safety Reliability

All inputs and outputs of the **CAN-xx-modules** are galvanically isolated from the CAN-bus. The power for the bus interface comes either from the bus or from a separated module supply.

Digital I/Os are isolated, using a common supply, analog signals are referenced to a common ground.

A programmable Watchdog controls CAN-Bus activity. The watchdog triggers when a module is not addressed in time. If the watchdog triggers, the outputs are switched inactive and a monitoring LED is lit.

CAN-Bus

The **CAN-xx-modules** are supporting baud rates from 50 kB upto 1 MB. They use 2...16 successive identifier on the CAN-bus. Baudrate and base identifier are set by rotary hex coding switches.

The modules base addresses can be placed in different CANOpen®-PDO-ranges. **CAN-xx-modules** can collaborate with CANOpen® modules on the same bus, modules with CANOpen® firmware are available also.

The CAN-Bus is connected either by pluggable srew-clamp terminals or by the „bus-in-the-rail“ – a 5 p bus in the DIN-rail.

Size and Supply

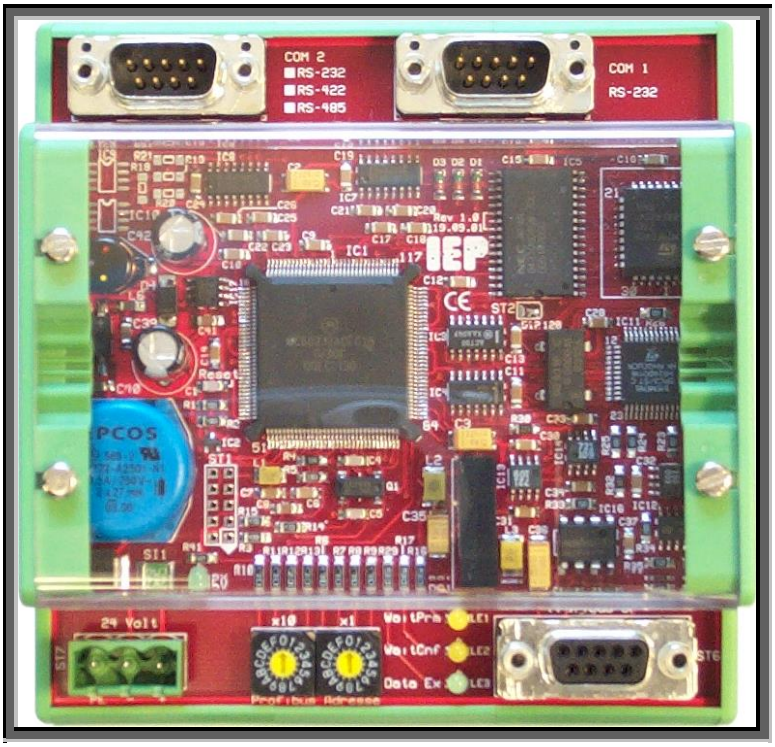
The **CAN-xx-modules** are delivered in a 85x90x22,5 mm housing for DIN-rail snap-on mounting. The CAN-bus and the power supply can be routed in the rail, so external wiring is minimised.

The **CAN-xx-modules** use a supply of 18...30 V_{DC}. The modules are protected from power supply polarity reversal; an EMV protection circuit assures troublefree operation in an industrial environment.

Special editions, even one offs, are possible at low cost, please contact us.

PK-DP

Protocol coupler for Profibus DP



The **PK-DP** protocol converter provides simple integration of devices with serial ports in Profibus DP networks:

PK-DP

Capabilities

- Profibus DP Slave up to 12 MBaud
- 1 serial port RS-232 5-wire
- 1 serial port either RS-232 5-wire or RS-485/422 (order option)
- Supply 24 V_{DC}
- Housing for the DIN rail assembly
- Variable length of I/O range (2...80 data words)
- The default ASCII driver is configured by a GSE file, customizing of drivers is possible
- Device is configured by the DP master, no special configuration tools necessary

DP Slave

The **PK-DP** protocol coupler works as Profibus-DP-slave. Using an appropriate ASIC, all baud rates required by the standard EN 50170 are supported. The employment in a Profibus-network is hassle-free. The variable length of the I/O-area helps supporting different Profibus-DP-masters.

The Profibus-address of the **PK-DP** can be set by 2 rotary hex switches.

A PLC can communicate with conventional serial devices through the **PK-DP**. Data are send to and received from the serial device, data integrity and correct timing are assured by the Profibus protocol.

Serial ports

2 asynchronous serial ports are available, thus 2 devices can be connected to the **PK-DP** simultaneously. When customizing of the serial driver software is needed, one of the ports can be used as programming console.

The first serial port has a fixed 5-wire RS-232 interface. This port is usable with Baud rates up to 76800 Baud.

The physical interface of the second port is determined by assembly as one out of RS-232 / RS-422 / RS-485. This port allows Baud rates up to 76800 Baud.

Power supply

The **PK-DP** uses a supply voltage of nominal 24 V_{DC}, voltages of 10-36 V are acceptable. The device is protected against power supply polarity reversal; an EMV protection circuit assures troublefree operation in an industrial environment.

Installation and Connections

The **PK-DP** protocol coupler comes in a 108 x 112 x 50 mm housing for snap-on mounting on DIN-rails.

The serial ports are connected with two 9p-SubD plugs. The Profibus connector is a standard 9p-SubD socket. Power supply is made by a 3 pin plug/screw terminal.

Programming

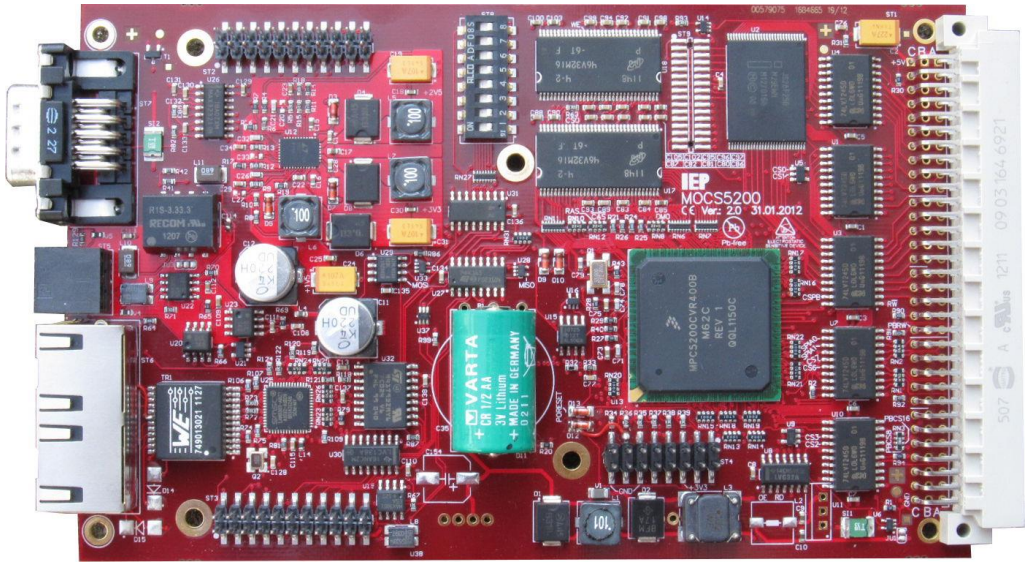
In usual applications, no programming is necessary on the **PK-DP**. An ASCII-driver, configurable by the Profibus master, and a GSE-file for automatic configuration are in the standard scope of supply.

For special requirements, customizing of the driver software is possible. A development kit for driver writing, containing an driver example in ANSI-C source and programming tools, is available. The serial communcation protocol of the **PK-DP** can be changed freely, though the Profibus-communication has to follow the Profibus standard.

The **PK-DP** is using the realtime operating system RTOS-UH. Driver can be programmed in PEARL, ANSI-C or IEC 61131-3.

MOCS5200

Versatile single board computer



As a fast computing system, the **MOCS5200** is bound for demanding applications, requiring high computational power altogether with leading edge realtime reactivity. A 400 MHz PowerPC MPC5200 with integrated FPU, originally designed by the processor manufacturer for use in automotive multimedia systems, is the heart of the **MOCS5200** and delivers upto 700 MIPS.

MOCS5200

Main features of the board are:

Versatile

- 128 MB DDR-RAM as well as 32 MB Flash
- high computing power (700 MIPS, FPU)
- 2xEthernet 10/100 Mbit, RS-232 and galv. isolated CAN
- low voltage, wide range supply 4,5 ... 30 V_{DC}, PoE possible
- realtime clock with battery backup
- EEPROM for storage of configuration data
- peripheral bus width 8/16 bit, synchronous and asynchronous operating modes
- ADD-On-interface (ATA, SPI, USB Host, 5x async. serial)

Base board

The base board of the **MOCS5200** is usable as a stand-alone controller. A RS-232 interface supports local operating, two 10/100 Mbit RJ45-Port are provided for networking and a galvanically isolated CAN-Bus connects process-IO.

The **MOCS5200** is available as plug-in board for 19“-cases and in different housings for use as stand-alone device.

Extension bus

As extension interface, the **MOCS5200** uses a 64p VG-connector with a parallel bus system. Data bus width (8/16 Bit) and access mode (synchronous/asynchronous) are user selectable.

For direct connection of serial peripherals, a SPI with select lines for upto 8 participants is available on the extension bus.

Interrupt inputs as well as programmable port pins complete the bus and provide for an easy and versatile adaption of the **MOCS5200** to special requirements.

Serial interfaces

By an Add-On interface, the **MOCS5200** provides upto 5 more serial 5-wire interfaces. 2 of these interfaces can be configured to provide an USB hostinterface instead. Line physics is user selectable, driver for RS-232 as well as RS-485 are fitted on the Add-Ons.

CAN-Bus

The **MOCS5200** supports baudrates from 50 kB upto 1 MB. The base board provides a galvanically isolated CAN-Bus, a not isolated CAN-Bus is routed via the Add-On interface. The CAN-interfaces are using Mini-Combicon-connector.

(Mass-) memory

The base board is fitted with 128 MB DDR-Ram and 32 MB Flash. Mass memory can optionally be added by an ATA interface (also usable for CF cards) or an USB hostinterface with support for memory sticks

Size and supply

The base board of the **MOCS5200** comes as 100x160mm euro-board with a 3 HE, 4 TE front panel. Depending on the Add-On used, the width can rise upto 16 TE.

The board requires a supply of 4,5...30 V_{DC}

Add-Ons

Add-Ons are available in these configurations::

-U 1x USB-Host, 1x CAN, 2x RS-232/RS-485, 1xCF/ATA

-S 4x RS-232/RS-485, 1x CAN, 1x CF/ATA

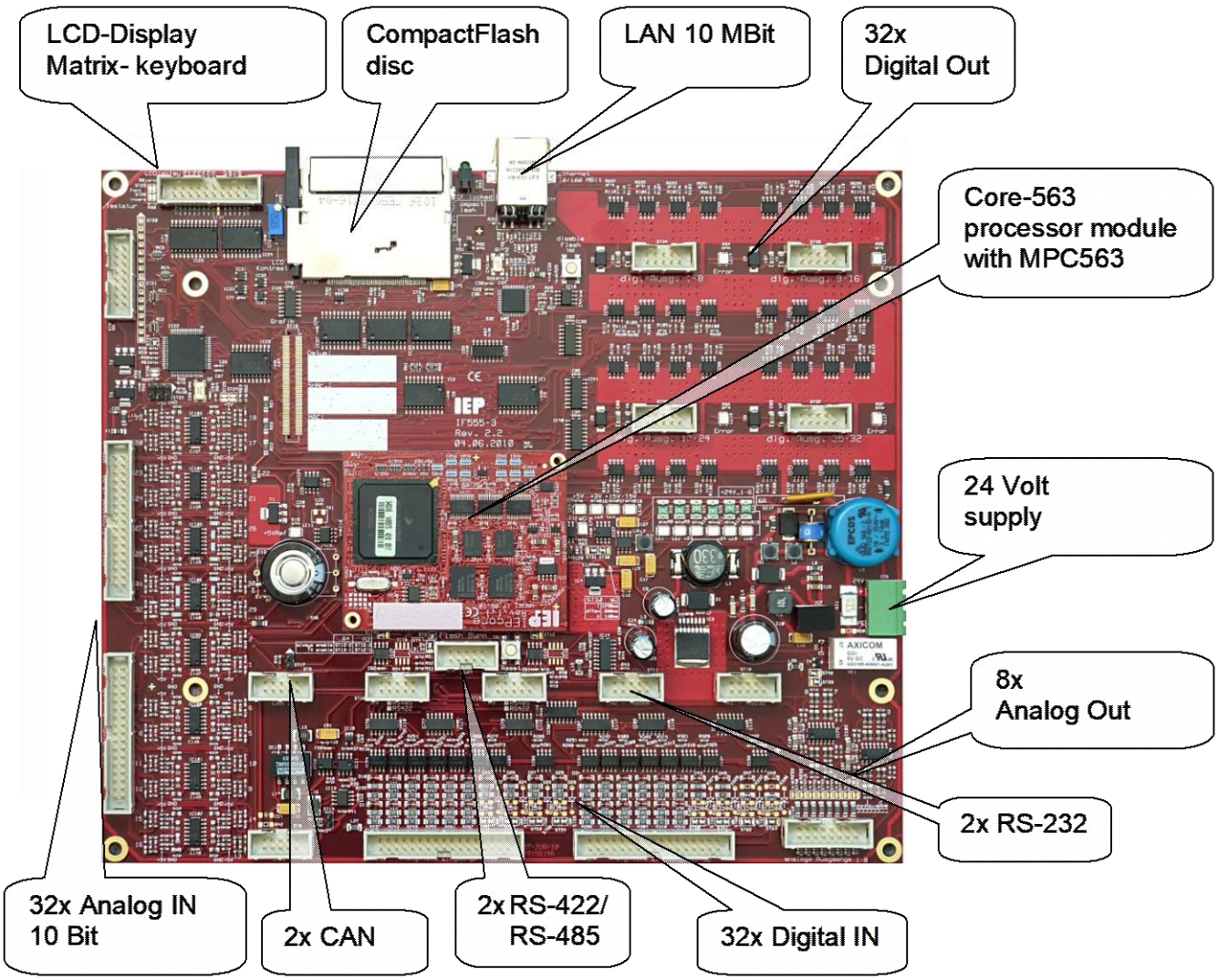
-D 6x dig. In, 2x dig. Out, 4x RS-232/RS-485, 1x CAN, 1xCF/ATA

Programming

The **MOCS5200** is based on the MPC5200 microcontroller from Freescale. The realtime operating system RTOS-UH is in the standard scope of supply. Crest-C, PEARL and IEC 61131-3 are available as programming languages.

IF555-3

Interface board for the MPC563/MPC555



Industrial I/O and high computational power – the **IF555-3** with an automotive PowerPC MPC5xx-Controller, universal I/O and standard interfaces for operating and network is a ready-to-go package for demanding applications in measuring and control engineering.

IF555-3

Capabilities

The **IF555-3** gives access to all of the controllers core modules:

- MPC563 / 56 MHz or MPC555, 40 MHz with up to 8 MB RAM and also up to 8 MB FLASH
- 2 CAN channels, one of them galvanically isolated
- 32 differential analog inputs, 10 bit resolution, input range adaptable for measurements of temperature, current and voltage
- 32 digital inputs, 24 V, opto-isolated, 24 of them usable for external lowside switches
- 8 analog outputs, using PWM, 0-20 mA / 0-10 V, adaptable
- 32 digital outputs, 24 V / 0.4 A (max. 6 A altogether)
- Supply of 24 V_{DC}

Standard interfaces allow the operating of the system as well as the integration into complete control concepts:

- 4 serial interfaces (2 x RS-232, 2 x RS-422/485)
- 10/100 MBit Ethernet 100BaseT
- CompactFlash provides external, changeable memory
- Real-time clock
- Connectors for LCD and keyboard

PowerPC 5xx

The MPC5xx-controllers are developed by Freescale for automotive applications. By their high computational power – especially at floatingpoint calculations –, their small supply current and their extensive onchip periphery they are ideally suited for general controls.

On the **IF555-3,** either an Core-563 or a phyCORE-MPC555 (made by the Phytec company) processor module with up to 8 MB SRAM and FLASH is used. The realtime operating system RTOS-UH is stored in the internal FLASH of the MPC5xx – the external FLASH is usable for applications.

Serial Interfaces

To connect further devices with conventional interfaces, the additional 4 serial ports of the **IF555-3** can be used. One of the two RS-232 5-wire interfaces serves for programming and data exchange, the second is intended e.g. for a modem or other serial devices.

Two more serial interfaces are providing differential signal transmission according to RS-422 or RS-485.

Ethernet

The Ethernet interface of the **IF555-3** uses a 100BaseT connection to provide a problem-free integration in industrial networks.

The TCP/IP stack is in the standard scope of supply. FTP, Telnet, http are available. An optional OSI stack serves for the integration into control concepts based on e.g. SINEC-H1.

The status of the Ethernet interface is indicated by 3 LED's (link, transmit, collision).

CAN interfaces

One of the two CAN interfaces is galvanically isolated from the processor core and intended for long distance connections. The second interface is set aside for local I/O-expansion in the electrical cabinet. Both CAN interfaces support baudrates upto 1 MBaud according to the CAN specification Rev. 2.0B. 16 send / receive buffers allow for high data throughput with minimal processor load.

CompactFlash

For the storage of e.g. logging data or recipes a socket for a removable CompactFlash card is provided. Type I and II cards can be used, thus also removable Microdrive CF+ discs are supported. A driver for the file system is in the standard scope of supply, data can be exchanged between the **IF555-3** and a standard PC with card reader.

LCD

LCD displays based on the Toshiba T6936 controler can be connected via a 20 header post connector. Text based displays up to 16x40 characters and graphical displays up to 240x128 pixel are supported. A terminal driver as well as an extensive graphical library are in the standard scope of supply of the operating system RTOS-UH.

Keyboard

Matrix keyboards upto 7x8 keys are supported. The keyboard is integrated into the terminal emulation – even with matrix keyboards, no additional programming is necessary. Customizing of key codes is available on request.

Analog Inputs

The 32 differential, analog inputs of the MPC5xx have a resolution of 10 bits at a typical conversion time of 10 µs. Each input of the **IF555-3** is amplified and optionally filtered by an external operational amplifier and then fed to the MPC5xx. Therefore, the measuring range of each channel can be configured separately.

The analog inputs are most simply usable by an automatic signal acquisition mode. A programmable sequencer, once initialised, samples all inputs and stores the input values in an own control store. The application can read the most recent sample values at any time.

Analog Outputs

8 analog outputs 0...20 mA / 0...10 V are provided to control external actuators. The **IF555-3** generates the desired output values using the PWM channels of the MPC5xx and on-board operational amplifiers to filter and normalize the resulting pulse train. The output frequency can be selected between 10 kHz and 20 MHz, PWM duty cycles of 0% ... 100% are possible. Accordingly, analog resolutions of 16 bit down to 1 bit can be achieved.

The analog outputs are addressed directly. No processor action is needed for the generation of the PWM pulse trains.

Digital Inputs

All 32 digital inputs are isolated by opto-couplers. 24 inputs can read external low-side switches, 8 inputs can be configured to read either low-side switches or active 24 V signals.

16 of the inputs are fed to the TPU of the MPC5xx. With its special timer functions, the TPU offers e.g. these operating modes without presenting an additional computational burden to the processor:

- Input – reading the actual signal state
- Counter –16-Bit-counter for leading and/or trailing edges
- Pulse width measurement – the time between signal slopes can be measured with a resolution of 100 ns.
- Position/angle measurement – two channels in combination can be configured to resolve quadrature signals as presented e.g. by incremental encoders. The resolved position is readable through a 16-bit-register.

The maximum incoming frequency is 75 kHz per channel.

Digital Outputs

The 32 digital high-side outputs are splitted in 4 groups. Each single output delivers up to 24 V / 0.4 A with a maximum current of 6 A altogther. Each output is protected from overcurrent and overtemperature; in case of overload, an error signal is raised. This signal is shown by a LED and can be read back by sofware. The outputs are capable of driving inductive loads, e.g. relays.

8 of the outputs are fed by the TPU of the MPC5xx offering e.g. the following operational modes:

- Output –direct setting of the output state
- PWM –automatic generation of a pulse-width modulated output signal
- Pulse control – time-controlled generation of pulse trails, also in connection with other in- or outputs of the TPU.

Power supply

The **IF555-3** requires a supply of 24 V_{DC} ±10%. The board is protected from power supply polarity reversal; an EMV protection circuit assures troublefree operation in an industrial environment.

Installation and Connections

The **IF555-3** is supplied as printed board assembly in the dimensions 222x257 mm. Voltage supply is made by a 3 pin pluggable screw terminal.

All signal connections are presented on header post connectors. The pinning is selected to allow flat-cable connections to standard connectors.

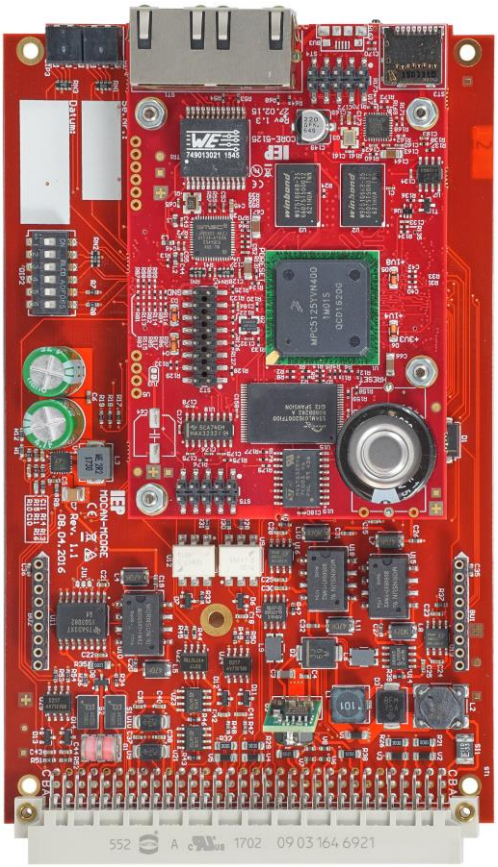
Programming

The realtime operating system RTOS-UH is in the standard scope of supply. Driver for all components of the board including software to program the internal FLASH of the processor are provided.

Crest-C, PEARL and IEC 61131-3 are available as programming languages.

MOCAN-M2

OEM PLC with CAN and network



The **MOCAN-M2** is an OEM building block for custom made PLCs and can be used as intelligent subsystem in Ethernet- and CAN-based networks. With up to two independent CAN channels (2.0b), up to 4 serial interfaces as well as digital inputs and outputs, **MOCAN-M2** offers all substantial basic functions of a PLC.

The **MOCAN-M2** is based on our powerful Core-5125 module. With the realtime operating system RTOS-UH and the programming languages ANSI-C, PEARL or IEC 61131-3, even complex control problems are solved quickly and reliably.

Different board variants offer a scope of service adapted to the problems with optimal cost-efficiency.

Network
CAN

Realtime

Interfaces

Two Ethernet-Interface integrate the **MOCAN-M2** into the factory network. A TCP/IP protocol stack is in the standard scope of supply; FTP, Telnet, http are supported. An optional OSI stack serves for the integration e.g. into control concepts based on the SINEC-H1. The interface status is signaled by LEDs.

CAN interfaces provide for integration on field bus level as well as for I/O expansion. All Baud rates up to 1 MBaud are supported.

Up to 4 asynchronous serial interfaces can be used for data exchange with external devices.

Inputs Outputs

The isolated 24 V inputs are fed to port pins of the processor and support high-level functions like interrupts, counter, etc.

The isolated high-side outputs 24 V / 0.5 A are short circuit protected and capable of driving inductive loads.

Dip-switches and hex rotary switches can be read by software; an add-on-Interface provides for additional optional in- and outputs.

Variants

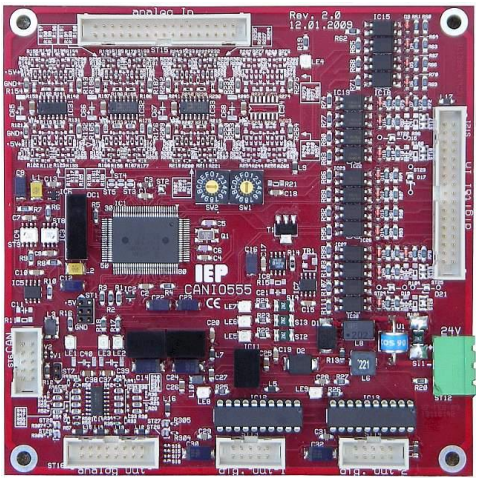
	Mocan-M2
Ethernet, 10/100BaseT	2
RS-232 / RS485	1x 5-wire / 1x isolated
TTY 20 mA, passiv	2, isolated
CAN, upto 1MBaud	2, isolated
Inputs 24V, isolated	4, with interrupt generation
Outputs 24 V, 0,5 A, high-side	2
DIP-Switch / Hex rotary switches	5/2
RTC with battery backup	Yes
RAM	128 MiB
NAND-Flash as system memory	128 MiB
Remanent storage	512 kiB FRAM
Runtime after Vcc lost	7 s, with power lost signalling
Configuration memory	EEPROM 2 KiB
µSDHC-Interface	optional
Extensions	Via Add-Ons
Firmware	Realtime operating system RTOS-UH

Size and supply

The **MOCAN-M2** boards are delivered as 100x160 mm euro-boards, require a supply of 9..38 V_{DC} and provide a 64-pin VG connector. Plug-in adapters allow for wall mounting as well as for mounting on DIN rails.

CAN-I/O

Input/Output for the CAN bus



CAN-I/O

The **CAN-I/O** is designed as inexpensive, universal I/O expansion board for systems with a CAN interface. By flexible configuration of the signal connections it integrates itself into even fastidious environments without hassle.

Features

The **CAN-I/O** provides:

- 1 CAN interface, optionally galvanically isolated
- 16 differential analog inputs, 10 bit resolution, input ranges are adaptable for measurements of temperature, current and voltage
- 16 opto-isolated digital outputs with counter operation, 24 V, 12 of them usable for external lowside switches
- 4 analog outputs, 10 bits resolution, 0-20mA (adaptable)
- 16 digital outputs with PWM, high side, 24 V / 0.3 A
- Power supply 24 V
- Configured by hex rotary switches

Digital inputs

All 16 digital inputs are isolated by opto-couplers. 12 inputs can read external low-side switches, 4 inputs can be configured to read either low-side switches or active 24 V signals.

The **CAN-I/O** scans all inputs with 20 ms cycle time and counts the number of level changes. When reading the input values, the number of level changes since last readout is given in addition to the actual input state.

Analog inputs

The 16 analog inputs of the **CAN-I/O** provide a resolution of 10 bits. Each differential input is amplified and optionally filtered by an operational amplifier. Therefore, the measuring range of each channel can be configured separately.

Digital outputs

16 digital high-side outputs are split in 2 groups. Each single output delivers upto 24 V / 0.3 A with a maximum current of 0.8 A per group. Each group is protected from overcurrent and overtemperature. In case of overload, an error signal is raised. This signal is shown by a LED and can be read back by sofware. The outputs are capable of driving inductive loads, e.g. relays.

Each ouput of the **CAN-DIO** can give a 0%... 100% PWM-signal for quasi-analog control of e.g. solenoid valves. The minimum PWM switching time is 20 ms.

Analog Outputs

4 analog current outputs 0...20 mA with a resolution of 10 bits are provided to control external actuators. The **CAN-I/O** generates the desired output values using PWM channels and on-board operational amplifiers to filter and normalize the resulting pulse train.

CAN interface

The **CAN-I/O** supports Baud rates of 50 kB upto 1 MB and uses 16 consecutive identifier on the CAN Bus. Baud rate as well as basis identifier are configured by 2 hex rotary switches.

Size and supply

The **CAN-I/O** comes as ready assembled PCB in the dimensions 232x140 mm. All connections are made by header posts.

The **CAN-I/O** uses a supply of 24 V_{DC}. Power is supplied by a 3 pin pluggable screw terminal. The module is protected from power supply polarity reversal; an EMV protection circuit assures troublefree operation in an industrial environment.

Variants

Customization of the **CAN-I/O** is feasible even for small order lots. This affects in particular:

- Range adaptions for inputs and outputs
- application specific firmware

Our application department gives advise and support during development of specialised solutions.