

PAC : BASE SOFTWARE CHARACTERISTICS

PAC: BASE SOFTWARE CHARACTERISTICS

Operating system

On Achille model the Windows XPembedded operating system with real time extension is present. This system offers a programming interface (API) similar to the typical Win32 for operating system NT/2000/XP. The Windows XPembedded operating system is compatible at binary level with the equivalent desktop type and therefore the same application can run on both environments.

On Aiace and Ulisse the Windows CE operating system is available. This system has been expressly conceived for embedded application, it offers Real Time functionality together with some typical office functionalities, and it is equipped with powerful developing tools.

The supported technologies are:

- MFC: library similar to MFC for desktop applications;
- ATL: library for developing of COM components;
- COM: Microsoft technology to create reusable components.
- .NET compact framework: collection of functionalities that allow to execute interpreted code (metacode) widely independent from the beneath hardware and software platform.
- Script with support of Automation;
- Web server: it allows to publish information with HTTP protocol
- FTP server: it allows to transfer file from and to unit in an efficient way.

PAC environment

The PAC is structured by an execution environment (framework) and by PLC logic that is made of one or more modules. The framework handles the protocols, schedules the execution of PLC module functions and interfaces with other applications, while the PLC modules implement the control law. All these operations are performed with sure execution times (real time performances).

COM interface


All framework functionalities are available by means of a COM interface that allows the other applications to acquire information and to actively control the framework. The functionalities, offered by PLC modules (synonyms, routines of events, definition of structures,), are shared in order that the external applications can directly interact with implemented logic. In this way, for example, a script is able to interact and manage the execution flow of recipes of axis movement or controlling logic, without compromising the real time execution.

The interaction modalities with other application are:

- Bi-directional sending of messages;
- Access to synonym data;
- Activation/Deactivation of event handler;
- Centralized management of alarms, attentions and information messages;

The external applications are also able to load and unload the PLC modules.

There is a library that allows the interaction with PLC by the side of applications for .NET

<p>ISAC S.r.l. CAPITALE SOCIALE 100.000,00 € C.F. e P.I. 01252870504</p>	<p>VIALE CAMPANIA, 61 – ANG. VIA ETRURIA, 36 56021 CASCINA (PI) ITALY</p>	<p>TEL 050 711131 FAX 050 711472</p>	<p>WWW.ISACSR.L.IT ISACSR.L@ISACSR.L.IT</p>	<p>Azienda con sistema di qualità certificato UNI EN ISO 9001:200 </p>
---	---	---	---	--

environment, which are able to connect itself with PAC unit running on the same CPU or other PAC unit connected by net. In this case the information are available by means of “web service”.

Native Interface

The native interface PLC Interface (*PLCI*) is an application running in .NET Framework, it is able to visualize all information relative to PLC and its modules. PLCI can run on PAC or on a PC that is connected by Ethernet with the unit to be controlled.

I-O Configuration

The configuration of I-O devices (the *physical* nodes) is described by means of dedicated interface and the information are organised in *logical* nodes at disposal of PLC modules. In this way the PLC modules are independent from the I-O device technology and from their physical organization. The assignment of values to the associated synonyms is made during the module loading in accord to the current configuration. The physical node can be re-directed, duplicated or masked.

If a synonym is associated to a unavailable resource the framework detects the fault condition, signals it and disables the synonym. The module can still use the synonym but no data are exchanged with the field.

Memory Areas

The PLC modules can use an area of memory with their own variables (volatile area) between the range from 128KB to 8MB. They can memorize data in retentive manner (32/128KB) too.

Model of PLC programming

The code of PLC logic responds to the events that the framework can rise, or it is executed in background (MAIN routine). All the event handlers relative of an event are executed in descending priority order. During the execution of MAIN routine the status of inputs and outputs reflects the real status; the PLC logic can execute some efficient waits and not-passing operations. The routines associated to events (for example the TIMER event) can not make waits; the input status is frozen immediately before the routine execution and the output status is updated immediately after the execution.

Inside of the code it is possible to use the sub-routines of the same module, or to call functions that are exported from other modules (library modules). Similarly the use of “function block” allows to implement state machines with logic implemented in external modules.

Each PLC module shows maintenance information that can be read without the need of source code: these information are very useful in case of administration or maintenance of plant. A particular event handler can be the implementation of a custom interpolator: the PLC can thus make a custom trajectory generator for special purposes and applications.

Programming for PLC logic

The PLC modules can be written in languages that are in agreement with standard IEC61131-3:

- IL (Instruction List)
- ST (Structured Text, similar to PASCAL)
- LD (Ladder Diagram)
- FBD (Functional Block Diagram)
- SFC (Sequential Functional Chart)

The develop system is Multiprog and Multiprog Express by KW-Software.

Moreover some or all modules can be written in C language using the environment Microsoft Visual Studio .NET 2003, 2005 and 2008, integrated with an extension that eases the development of PLC modules.

The PLC logic can use a selection of functions of C standard library, the framework standard functions and the others at disposal of optional modules.

Among the standard tools of framework there are:

- Programmable delay lines (Timers);
- Deriving;
- Mono-stable oscillators;
- Counters of transitions;
- Function Block.
- Current date and hour.

Optional modules

The base functionalities offered by framework can be extended by means of the adding of Optional Modules. These are software and hardware components that offer hardware or software functionalities to PLC modules and to other external applications in order to assure an homogeneous usability.

List of main optional modules:

- OMAX = Module for Axes

- management
- OMCOM = Module Serial Ports (to manage devices on serial lines with MODBUS protocol, as inverter, PLC and operator interface, or devices with free protocol, as RFID and BARCODE)
- OMETH = Module communication Ethernet Real-Time (for example to synchronize PLC in Real Time via Ethernet (TCP/IP))
- OMKB = Module keyboard management (management of keyboard by means of PLC)
- OMFILE = Module accessing file (management of file and database)
- OMSMS = Communication through SMS, e-mail.

Axis Handler Module (OMAX)

The axis handler module makes a direct management of the drives by means the communication protocols and it offers an homogeneous interface that is independent from the technology and from the peculiarity of controlling of each axis. During the application set-up it is necessary to assign the axes to their channel, specify their technology and all the needed parameters. The axes belonging to one channel can accept commands in an independent manner or as a group (for a certain type of axes). These commands can be set up by the PLC modules or by external applications. Each channel can handle a motion command recipe, a linear interpolator and a command dispatcher. There can be up to 8 channels and it is possible to associate up to 16 axes to each channel.

The working modalities of axes can be:

- Free servo: the axis does not accepts commands and it only shows the actual height;
- Positioning: the axis receives the positioning commands in and independent way from the others;
- Channel interpolation: the axis is associated to the linear interpolator of the channel and it moves itself in a synchronized way with respect to the

axes in the same modality;

- Types of interpolation: Linear, circular, extern through CAMMA or through PLC (free interpolation). In this last case the PLC module can house a customized interpolator that is designed in function of application needs.

The available commands for the axes are the same at disposal of the command list and of the collective commands for channel:

- Zero searching for linear axis with parameter micro, mark, entering feed, exit, height to be set up, ...
- Zero searching for round axis;
- Movements with ending on input commutation;
- Movements with programmable velocity profile;
- Interpolating movements;
- Change of working mode;
- Change of value of height;
- Change of value of height in zero position;
- Synchronized axes (GANTRY);
- Synchronized axes with more *slaves* (GEAR);
- Movements of positioning in space and in velocity, with incremental and absolute positions.

The velocity of movements can be controlled by a percentage value (OVERRIDE).

Utility libraries and Applicative models

There are some libraries available that make easier the axis management and other functionalities useful in order to develop the applications. These libraries export functions and functions block that can be called in other PLC modules.

Among the available functionalities:

- Management of zero searching for axes;
- Movement in JOG;
- Override management;
- Management of functioning modes: Home, Manual Jog, Manual Positioning, Automatic for the execution of cycle of movement, Block of moving axes along

trajectory (Hold) and following unblock, Reset of cycle and/or movement command .

Development tools

The framework offers a series of extremely useful tools for developing and verifying in order to reduce the application time-to-market.

Oscilloscope

In PLCI it is possible to set up a series of synonyms (not necessary physical signals) and to show the trend during the time. The acquisition can be continuous or triggered by a trigger logic. The data are acquired contemporaneously and in a synchronous mode with respect to the execution of PLC events. The data are showed on video as on last generation digital oscilloscope, they can be saved on a file in order to be analyzed by external application and they can be reloaded at every time.

All oscilloscope functionalities are accessible also by other applications through the COM interface of the framework.

Execution trace

It is possible to insert a series of strings and binary data into the PLC logic in order to be used as execution markers, recorders of values of variables, executing flow tracks etc... These strings are recorded by the 'PLC trace data Logger' (FWLOG). The use of the trace does not significantly modify the execution times of PLC logic.

PLC Remote Debugger (PRD)

It allows to verify of PLC code directly in Visual Studio debugger. PRD is a powerful tool

particularly suitable during the beginning steps of verifying. The logic execution times are not guaranteed. With PRD it is possible to set up breakpoints on code or on synonym values, to view the instant values of the synonyms and to execute all possible verifying operations as with on any desktop applications. During the execution of routine inside of PRD, the others are executed inside of the framework with real-time performances, so it is possible to make a test keeping a part or all logic of safety management fully responsible.

Crash analysis (under development)

When the PLC logic raises up a unmanaged exception the framework collects all the necessary information in order to execute the logic in the same conditions. The information are saved in a file that can be read by a dedicated application. In this way it is possible to execute the frozen cycle inside of Visual Studio debugger and analyse the causes of the exception. This tool is very powerful in order to understand the faulty causes, that are hard to be identified otherwise.

Use statistics

By means of PLCI or dedicated scripts it is possible to acquire information about the execution count of each event routine, main function, and the statistics of execution time.

SIMU Implementation

ISAC offers a software package that makes a simulation of the PLC framework on PC. By using this package it is possible to make and test the PLC logic, to simulate the I/O devices and axes, and to perform external application of auxiliary task or human interface.