Multi Standard Gateway
for Home Control Networks

document version: 1.0

document date: 02/08/2005

subject: Konnex Scientific Conference 2005 – Paper submission

authors: G. Grifoni
L. Lesa
M. Lutman

company : SIPRO Sistemi Integrati per l'Innovazione di Processo srl.
Autoporto Capannone A c/o BIC
34170 Gorizia
Italy
Tel. +39.0481.527778
fax. +39.0481.527731
Email: m.lutman@sipro.biz

1 ABSTRACT

Unlike other areas in Telecommunications and Computer Networking, Home-automation and, more generally, distributed-control networks suffer from the lack of a universally recognized standard. International organizations for standardization just gave generic “indications” (the O.S.I. model in particular) and, in many areas, different systems established themselves as standard de-facto and, in some cases, provided with dedicated hardware and proprietary protocols. This all relates to strong restrictions in the product choice for final users and, for the companies, in a substantially rigid market.

To ward off this situation the proposed solution concerns a Multi-standard Gateway which should allow communication between Home network devices, adopting different protocols. A Gateway for EHS and LONWorks systems is currently under accomplishment, whereas its extension to the Konnex system, is intended to be the next step. The device includes hardware and relative functional blocks typical of each single system involved, which communicate, by means of an internal protocol, with the core gateway application. The latest handles, besides the necessary translation coding routines between different protocols and the message dispatching between the networks, the dynamic creation of an internal database of connected devices. In this way devices located in different networks, using also different media, are allowed to communicate and to operate each other in a totally transparent way in the respect of the actual protocol adopted. Moreover, it is permitted to keep typical features of the different systems, as, for example, plug&play procedures, where they are usually allowed, or the use of normal configuration tools.

This paper, after summarizing the state-of-the-art in the area and the features of the involved systems, illustrates the general requirements and the architecture the gateway device must have. Then the solutions adopted for a specific double-standard Gateway are described. Finally the actual results and the current limits are

summarised together with the possible improvements and developments to obtain the multi-standard extension.

2 MULTI-STANDARD BENEFITS

Many reasons lead to a combined use of different standards in the context of the same Home Automation system. In particular, it should be emphasized first the chance to pick out from a wider range of products, enhancing the ways of easily satisfying the requirements asked both by end-users and companies, in terms of functional behaviours and commercial aspects. Second, but not less important, the possibility of taking advantage from the strength points of the various systems.

Leaving out details, it should be useful at this point to underline that these strength points are allowed not only by different protocols, but also by different concepts and background philosophies of the most widely-used standards. In particular:

- EHS bases itself on a command-oriented control form, identified by Object/Service couples, and on a strictly asymmetric and hierarchical structure of the devices, with controlling units (Feature Controller - FC) and controlled ones (Complex Device - CoD). It is intended to cover a complete system for domestic environment, covering both network communication and home application. The main strength points regard the insertion of new devices in a Plug&Play manner, the accurate control on individual applications and the easiness in building complete end-users interfaces. This is achieved at the expense of a relative complexity, both in development phases and in protocol implementations.
- LonWorks is intended as a completely distributed, generic purpose control system; it is data oriented, by means of application-level objects as Network Variables and predefined I/O types. This allows a great freedom and simplicity for developers in projecting applications, while pure protocol aspects are handled by dedicated processors. This implies a wide

availability of off-the-shelves devices, variety of transmission media, good overall performances. A Lon network, on the contrary, requires dedicated tool to be configured and is not well suitable for centralized control.

- KNX technology is a specialised form of automated process control, dedicated to home and building applications by means of distributed control with standardised data types and "functional block" objects. It provides different operation methods, from S-("System")Mode which uses centralised free binding and parameterisation, to E_("Easy")Mode that can be configured according to a structured binding principle through simple manipulation, to A_("Automatic")Mode which achieve plug&play configuration aimed mainly at consumer products; like LonWorks, it supports twisted pair and powerline as media, as well as radio frequency. It is therefore a widely flexible system for many applications.

It is therefore evident that an interconnection device for these standards should be worth of interest not only for commercial purpose, but also in order to increase performances and potential of the joined system in itself, on condition that profitable features of single systems are maintained.

3 GATEWAY ARCHITECTURE

Subject of the present paper is the implementation of a Multi-standard Gateway device to permit the communication between the above-mentioned systems, maintaining as much as possible features and functional characters. A device for EHS and LonWorks systems, depicted in figure 3.1, has been developed, and will be presented in the next sections. Its extension to other standards as Konnex is in progress; some guidelines for this purpose will be expounded in section 6. Here the general features of a Multi-standard Gateway are highlighted.

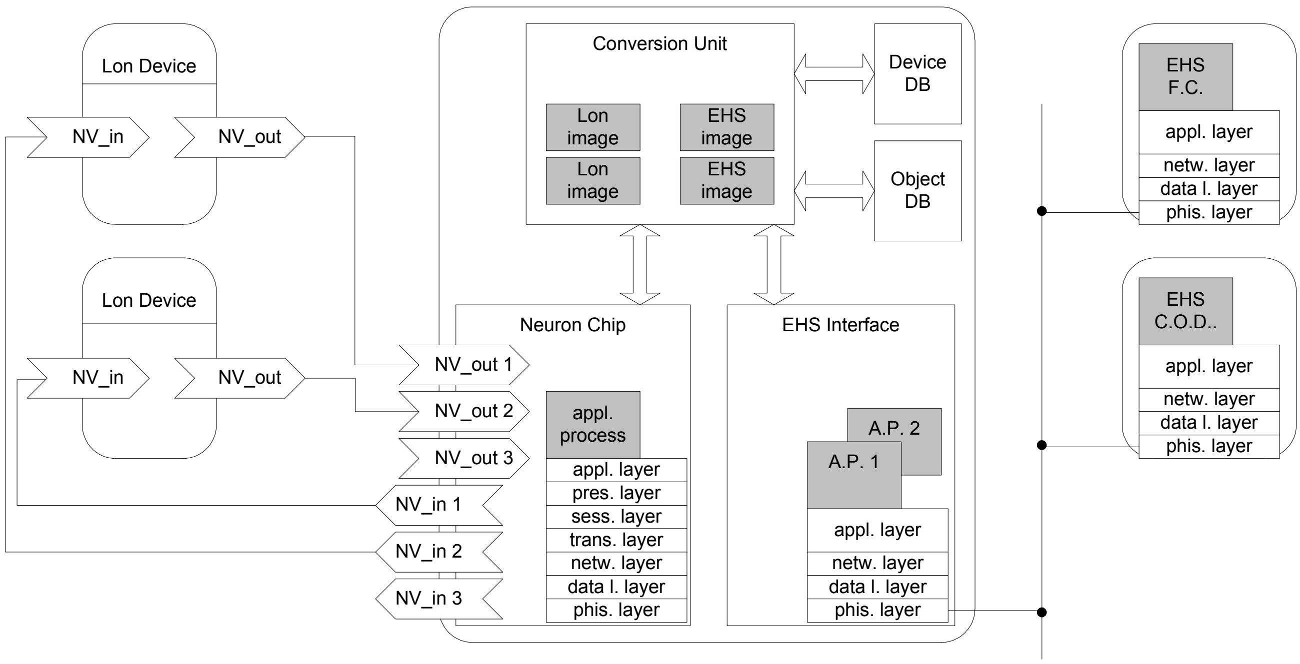

Figure 3.1 - architecture of the Lon-EHS Gateway

As reported in the above, many specifications that such a device must respect should be already manifest. First of all, transcoding operations should be made at the application level, at least because LonTalk protocol, which covers all seven levels of O.S.I. model, is a proprietary protocol. Operations most strictly connected to single different systems should however be carried out more appropriately by suitable, specific functional blocks (in particular, by a Neuron chip for the Lon side). These blocks not only menage low-level aspects of the protocols, but also provide queuing systems for incoming messages.

Since the maximum allowable flexibility is wished in the choice of supported devices, and Plug&Play methods should be implemented for EHS and KNX systems, the Gateway must in some way be updated about devices actually requiring services. So it includes a dynamic database of connected devices on both sides (DeviceDB), containing fields about type of device, address and current state, as well as fields relative to available objects and services. To allow communication between devices at a semantic level, a further database (ObjectDB) is also useful: this last one is static and updated from outside, depending from the particular application area of interest (security, safety, white goods and so on). It includes all useful data about potentially connected device types and parameters to convert compatible items from a system to the other.

Another essential feature is related to the requirement to obtain the maximum "transparency": to observe the set purposes, application processes in connected nodes should not be modified, and should communicate with those belonging to other systems as if they were using the same protocol. To achieve this goal, the Gateway must provide the different subnets with "image devices" with which to communicate using usual routines. An image device can be thought to as a virtual node, equivalent with the real device at functional level, but containing information (address, data, parameters, objects) proper to another protocol. The same databases mentioned above manage the data for this purpose.

Finally, it is necessary to provide a common platform, basing on which the Gateway should be able to convert messages keeping usefulness and meaning; this is not a trivial task, because different standards - as said before - make use not only of different protocols but also different philosophies. In the implementation, the choice was to refer to the EHS concept of Objects and Services, which is suitable to correspondence with KNX Datapoints and LonWorks Network Variables. In particular, provided services include Read, Write and Event Notification.

It is worth to underline that this choice only establishes a common level for semantic understanding of messages, and it does not interfere with other differences between systems that are instead entirely maintained. This means that, for example, an EHS FC can request data and send commands to Lon devices in the usual hierarchic and command-oriented way, while a Lon node can send and receive updates about Network Variables from a EHS device according to normal distributed and data-oriented criterion. This and other aspects should be clearer after briefly explaining the behaviour of the Gateway device.

4 INSTALLATION AND CONFIGURATION

As mentioned above, one of the main differences between systems taken into consideration is that Lon standard defines links between devices during installation phase, through provided configuration tools: this requires in any case some knowledge of the overall network structure, studied at development time or defined on-site; on the contrary, the EHS protocol provide for dynamic insertion of new devices in a Plug&Play manner, while KNX, depending on configuration mode, provides for both Plug&Play and tool-supported binding.

As the essential features of involved systems are to be maintained, the Multi-standard Gateway provides for a first phase of Network Variables binding on the Lon side, made with the usual tools. The same phase can be used for connecting KNX Datapoints in S-Mode, but this point will be expounded in more details in a subsequent section. Afterwards, if the bindings have been correctly defined, the Gateway automatically identifies connected devices and constructs the related part of the database. The Device Descriptor objects are then sent to the EHS side of the network, which can finally operate enrolments in the usual way.

Configuration process yet includes another sub-phase. In fact, in some situations it could be useful and simple to define a fixed, data oriented link, in a Lon fashion, between two devices belonging to different subnets. For this purpose ObjectDB contains also possible compatibilities and conversion parameters between objects of different device types, for example the output of a sensor and the input of an actuator or alarm. However, the Gateway cannot arbitrarily choose the devices between which these type of connection is appropriate, so this operation is carried out by the user in a specially provided binding sub-phase, using respectively Service Pins within LonWorks environment and Key_Pressed objects on the other side.

5 BEHAVIOUR OF THE DEVICE

Some examples should probably clarify how the device works.

Referring in particular to the Lon_EHS Gateway, suppose first that there are two devices, a sensor and an actuator, connected to the Lon side, and a Feature Controller on the EHS side.

- During initiation phase, the Gateway updates the DeviceDB with Lon devices, and creates their equivalent EHS images.
- At the beginning of working phase, the Gateway sends the Device Descriptors of the images to the EHS side.
- The FC can than try to enrol the devices in the usual way; the Gateway updates the database with the FC .
- The FC can ask an Event Notification to the sensor; this is stored as a EVE (EVent Enable) link in the DeviceDB
- The sensor sends updates of its output N.V. to the Gateway, which stores the new values; if there is an EVE link to the FC, the Gateway converts the N.V. data to an Object EVN by means of the ObjectDB and sends it to the FC.
- The FC can also ask an explicit Read to the sensor: if the device has an appropriate input N.V. to press the output, this N.V. is found in the ObjectBD and updated, else the converted stored value is sent.
- Last, the FC can Write an object of the actuator; the corresponding input N.V. is found via ObjectDB and updated.

As a second case, suppose that no FC is connected, and a direct communication between, for example, a Lon sensor and a EHS alarm device is required.

- In the link phase, the user must select the Lon device via Service Pin.
- The gateway finds compatibility for a EHS sensor output in the ObjectDB and tries to enrol such a device on the EHS side.
- The user activates the object Key_pressed on the EHS device to select it; the Gateway than stores this connection in the form of an EVE link in the DeviceDB.

- Whenever the sensor updates its output N.V., the link is found; the Gateway converts the value in the appropriate object with service Write and sends it to the alarm.

Multi-standard Gateway allows both this communication schemes to work in a network at the same time.

6 EXTENDING TO KNX STANDARD

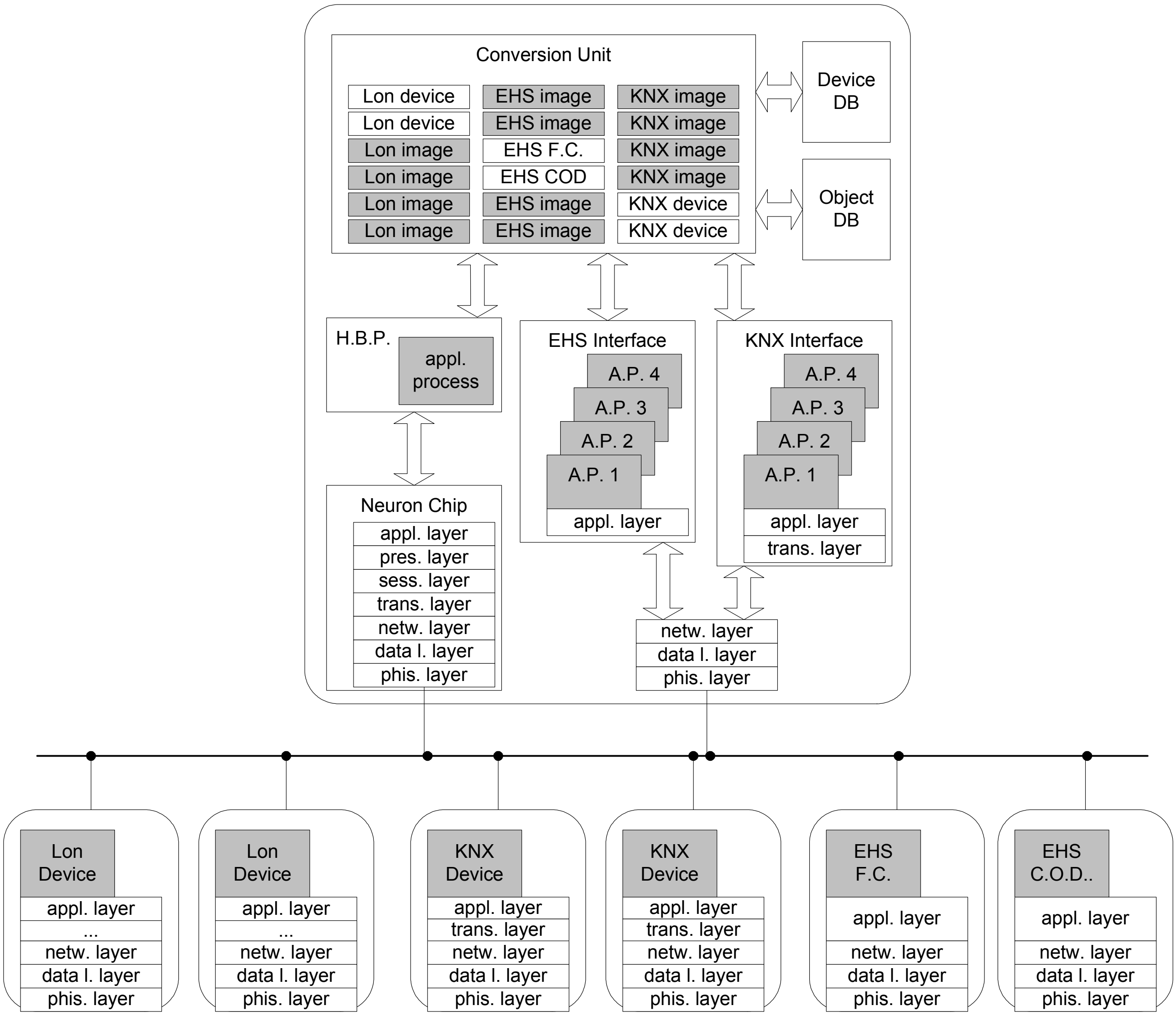

Figure 6.1 – complete tri-standard architecture

An obvious improvement, currently under analysis, of the device actually developed consists in extending the Gateway functions to other standards, first of all the KNX one. It is worth to underline anyway that both ideal architecture and functioning mode of the device, and the philosophy based on transparency and equivalent image devices are automatically suitable for this purpose. An architecture for this tri-standard device, as well as some improvements expounded in the next section (host based processor), is depicted in figure 6.1.

Entering in deeper details, it must be noted first of all that even if databases in the Gateway are organized in terms of devices, these are in all effects only collections of objects and relative services, in no way tied with physical identification of devices; so the same structure suites to represent collections as Interface Objects and Functional Blocks.

An important point to be treated is about which mode type (Simple, Easy, Automatic) is more appropriate to use within a multi-standard network. An intuitive approach shows that S-Mode is in some way similar to Lon concepts, requiring the binding of independent data-objects at configuration time by means of provided tools; A-Mode instead has some affinity with EHS system, allowing plug&play and focusing on point-to-point, server/client communications. It has been shown that the multi-standard Gateway can support both these styles in the mean time, provided a way to store permanent links between compatible objects and a model to compare and translate messages at a semantic level. In this sense Lon Network Variables and EHS objects can be compared, for the first mode, with single Datapoints, linked by tools or user intervention, and for the second with Property / Interface Object couples, assigned in a tagged binding way according to device applications; moreover, both of them adopt essentially the same set of services (Read, Write and Information) considered up to now.

The E-Mode and relative Group Objects are less straightforward to implement. This is essentially for two reasons: first, the Gateway should implement structured binding, that is assigning Datapoints in base of potentially unknown Functional

Blocks, second, it should in some way detect and store the membership to a group also for non-KNX objects. These requirements would increase substantially the Gateway complexity, so it probably would be worthwhile only for particular networks.

7 LIMITATIONS AND POSSIBLE UPGRADES

Up to now the efficiency aspects of the Gateway and the network using it haven't been mentioned: that's because this factor is strongly dependant from the actual implementation of the architecture.

The device currently implemented is not optimized under this point of view, mainly because it makes use of the same functional block and internal protocol for communications between central conversion unit and different subnets. However, since the various subnets operate with considerably different speeds, it would be worthwhile to manage communication by means of specific blocks, one per standard, operating independently and in accordance with different criterions and speeds; this would be consistent with the idealized Gateway architecture expounded above. One aspect of this question is handling internal protocol and queues: the device in fact includes, as one can imagine, a reserved memory space for incoming and waiting messages, and a simple internal protocol for the communication between different blocks. Overall performances, as well as memory allocation, would be substantially improved if both queue dimensions and timeout and polling timing should be scaled in function of response time and statistical frequency of messages on different subnets.

Another strongly limiting aspect is the need of having at disposal a fixed (and limited) set of Network Variables, as well as a tool and probably an installer for binding them correctly. The first problem could be solved implementing the Lon-side block with an host-based processor, instead of a simple Neuron chip: this would allow not only to increase considerably the number of usable N.V, but also to define them according to requirements of the particular network by means of Dynamic

Network Variables (supported by LonMark 3.0). The second aspect is strictly correlated with the first, because both would be easily solved if Lon devices could in some way identify themselves towards the Gateway. Useful in this sense should be the identification and configuration files of the devices (XIF), present in LonMark 3.0 specifics, even if not completely and universally standardized. This involves, however, the implementation of more complex, lower level communication methods between Gateway and Lon devices, which currently make use only of high level objects as Network Variables and predefined I/O.