Home Automation Interface to IS-105

The following sections will present the HA Interface device as a Decoder device under IS-105, for the purpose of exploring the circuitry blocks which are required.

Before examining that approach we first present the logical alternative, as described in the Summary, above. Refer to Figure 2, below. In this approach the TV contains the HA interface circuitry and logic. For reasons stated previously we believe that this is the wrong approach. However, it should be clear that there are no inherent economic advantages for CEBus over other protocols in this approach, since no HA messaging is delivered on the IS-105 bus. (Whether IS-105 should use a CEBus subset at all is a different argument, as all implementations will share this cost equally.)

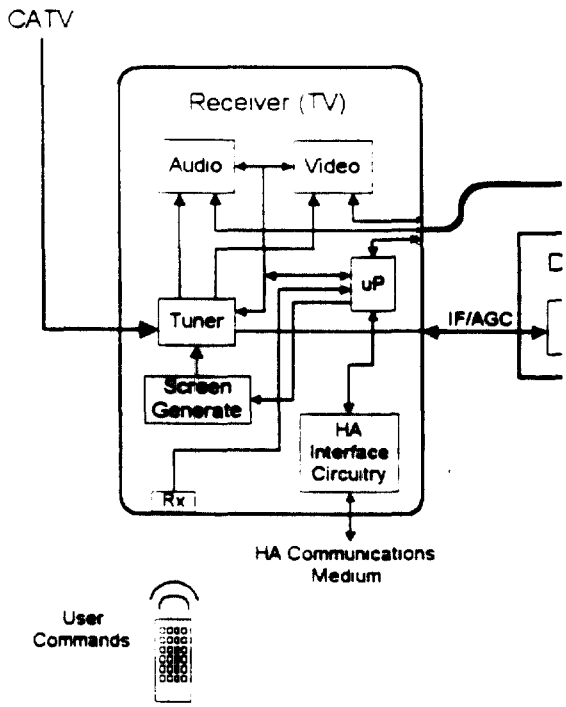

Figure 2. TV as Home Automation System Interface

We should note that Figure 2 is shown at a high level of abstraction. There are additional processors used in the communications interfaces, for example. The following Figures will move toward more detail, as we pursue the most likely architecture for HA Interface devices.

Figure 3, below, depicts the basic Receiver/Decoder pair, with a second Decoder serving as the HA Interface device. The HA Interface registers itself with the other devices on the bus, so that they can coordinate their interactions. The User must be supplied with a remote control which can generate a standard set of Decoder tokens. In addition, the remote must be able to select a particular decoder to which the tokens are to be addressed. With these features standardized, the User may direct commands to the HA Interface device when any of its features are to be accessed.

When the user directs a token to the HA Interface, it in turn takes action to cause the TV screen to come under its control.

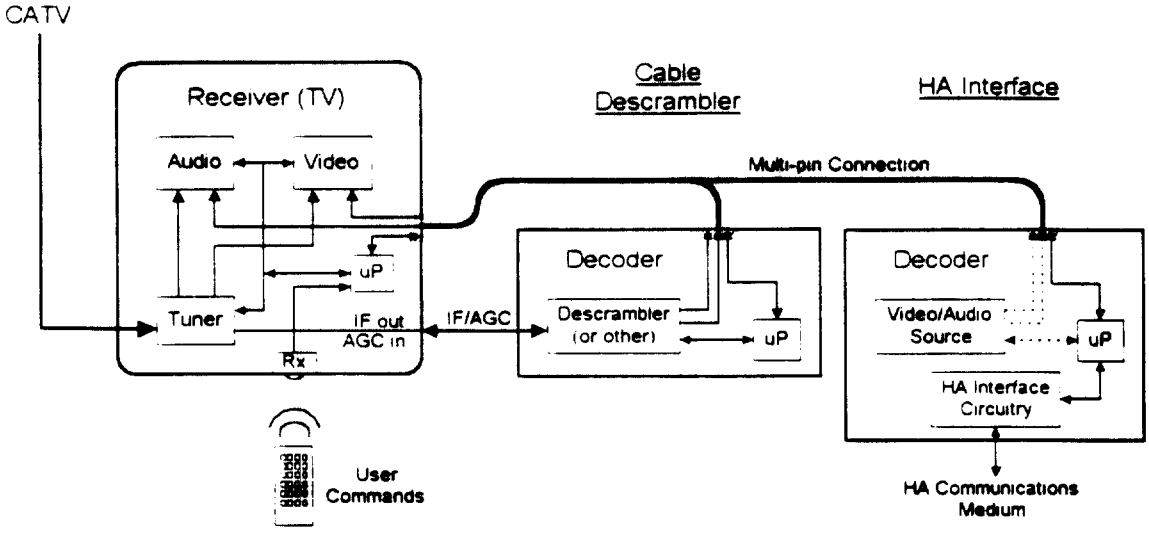

Figure 3. IS-105 Decoder as Home Automation System Interface

Once IS-105 is complete there may be a standardized way to use the character generator of the Receiver to place menus on the screen[2]. In the generalized case the Interface box will contain a video generator of its own, and potentially an audio generator which would allow voice prompts and sound effects to accompany the information display and user actions. In this case, when the user sends an initial token to the HA Interface the Interface must obtain the use of some of the lines on the bus, and must direct the TV to use these lines as the source of information for the screen/speakers.

Some of the bus lines may already be in use. For example, the VCR (another Receiver on the bus) may be recording a program which is being received by the satellite receiver (another Decoder on the bus). We would not want to interrupt this session by taking over these lines. IS-105 addresses this issue by providing a "Data Channel Locking" feature, which permits devices to guard lines which should not be interrupted. However, if the User were watching a scrambled program from the Cable Decoder, and then wanted to switch to the HA Interface for a time, the Cable box will relinquish its lines for that purpose as they need not be maintained while the User is not watching that source. Since these allocations are managed automatically they should not interfere with the User's perception of the equipment behavior.

[2] The HA Interface may be able to access the character generator of the TV, such that the Interface device sends text characters to the TV, and the TV places these characters on screen. In this case the Interface device need not contain any video generation circuitry, and need not gain access to IS-105 bus lines to carry the video to the TV. This will be a practical alternative only if all receivers are required to support this capability, so that a decoder manufacturer is guaranteed that this facility will be present.

Interface Device Block Diagrams

This section will examine the design of the IS-105 HA Interface Decoder in more detail, with emphasis on those features which will differ in two implementations as examples—an Interface to CEBus power line, versus an Interface to one of the LonWorks power line versions.

Figure 4, below, depicts one possible generic IS-105 HA Interface. The shaded blocks indicate the sections which are variable across different home automation systems. This model will be used for both CEBus and LonWorks interfaces, to evaluate obvious differences in equipment costs.

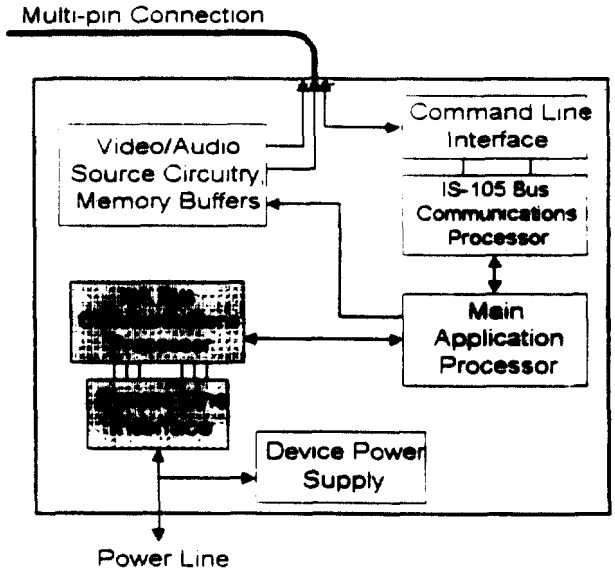

Figure 4. A Generic IS-105 Home Automation System Interface

Before moving on to HA-protocol-specific examples, let us review those circuit blocks of Figure 4 which will be common to any implementation. Figure 5 presents a more detailed diagram of the circuit elements which are unchanged by the selection of HA protocol. The following paragraphs refer to both Figures 4 and 5.

The Command Line Interface constitutes the physical layer interface to the IS-105 control bus. It uses a straightforward differential serial line to carry control messages between devices, and a differential driver and receiver circuit are needed to support this interface.

The IS-105 Communications Processor is responsible for controlling this interface, to send and receive messages on the bus. This function requires a mid-range to high-end 8-bit microcontroller, and some non-volatile RAM to store configurations in the event of a loss of power. The processor relays received messages to the Main Application Processor for disposition, and accepts messages from the main processor for transmission on the bus. Since the main processor is likely to require NVRAM as well, these requirements may be consolidated in the main processor's hardware. (There is no need to maintain separate NVRAM stores for both, since the processors will be tightly coupled.)

The Main Application Processor is probably a more capable device, maintaining larger buffers and therefore requiring more memory, and performing more complex tasks (such as supporting the User Interface at the TV), and thus perhaps needing a more powerful microcontroller. There are high-end 8-bit and 16-bit controllers which offer large RAM and ROM options, which may eliminate the need for external RAM and ROM. This processor also manages the interface to the Home Automation system, through the HA Interface Processor. The main processor is likely to require significant resources to manage the system-level issues associated with the HA interface.

The Video Generation circuitry is the hardware which the main processor uses to control the TV screen, in order to present menus and other information to the user. This circuitry will vary in implementation, depending upon the level of quality desired in the on-screen display, and single chips are available which implement much of this "character generator" functionality. The video Buffer presents the video signal to the IS-105 bus. The buffer is controlled by the main processor such that it is tristate-able and can use one of two or more video lines on the bus, in order to share the bus with another simultaneous session.

Finally, the device must contain a power supply, and since the box is not completely insulated (because of the connections to the other devices on the bus), the power supply is required to be line-isolated for safety reasons.

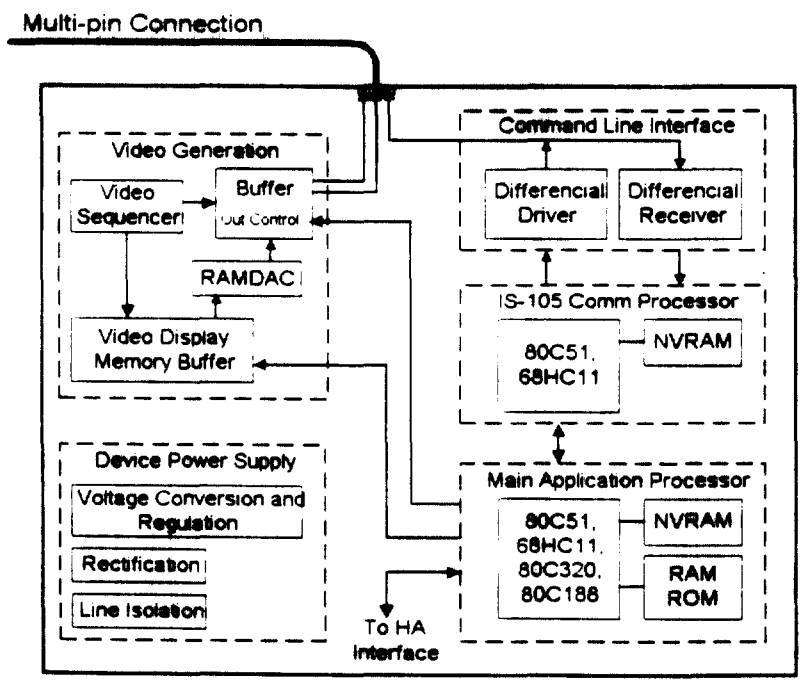

Figure 5. Common Elements in Home Automation System Interface (no audio)

The architecture described above is a straightforward approach which separates the real-time demands of the IS-105 bus from the application processor, by use of the IS-105 communications processor. It is certainly possible that the functions of both of these processors could be serviced by one microcontroller with sufficient processing capacity. Figure 6 presents a diagram illustrating this approach.

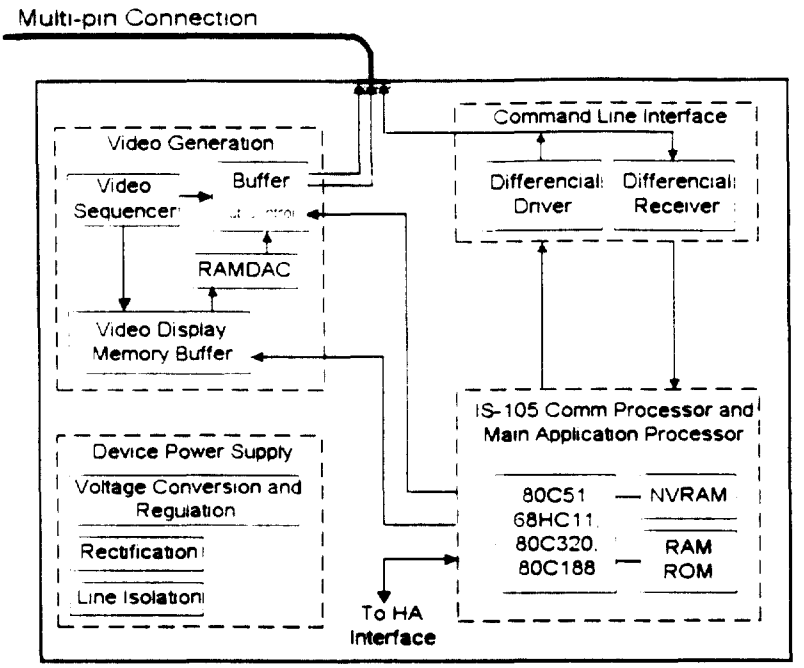

Figure 6. Common Elements, HA System Interface, Main and 105 Processors Combined

The features discussed thus far will be common to any implementation supporting the functionality outlined, independent of the HA system involved. This is an important point in the evaluation of the competitive effects of the proposed standard on various HA systems vendors—they will all share this base of hardware as a common requirement.

The selection of a particular HA system interface will determine the details of the HA protocol processor and associated PL interface circuitry, and will also affect the code in the main processor. The extent of this effect on the main processor is determined by the general requirements of that HA system. Other factors affecting this processor include the features which are to be supported by the interface device. Since the TV is such a convenient and capable display device, examples of such features include almost any feature which an installation and configuration tool might provide, as well as features associated with central controller devices in HA systems. Among these are: Automatic discovery of all devices in the home network; Assisted installation of new HA devices; Detailed reporting of device configurations; Diagnostic features to help in locating failed units; Recall and presentation of data collected by sensor devices; Direct manipulation of control values such as thermostat settings, lighting controls, sprinkler system controls, etc.; and Recall and manipulation of schedules for various automated subsystems. This list is only a sampling of the features which could be built into the HA Interface Decoder. The main processor would request/store and manipulate all of the data associated with these features, and present the formatted information on the screen.

Much of the resources required to support these User Interface features are comparable in different HA systems. It takes the same amount of memory to hold a Lighting schedule in one system as it does in the other, for example. And because the interactions with the User (mediated by IS-105 bus messages) are handled with HA language-independent tokens

which simply report remote-control button-presses, the main processor will have an equivalent task in performing translations into necessary HA system messages regardless of the type of HA system.

We turn next to protocol-specific details of two example HA system interfaces: CEBus and LonWorks.

CEBus Interface Device Block Diagrams

Figure 7 depicts the overall HA interface device with a CEBus Power Line interface.

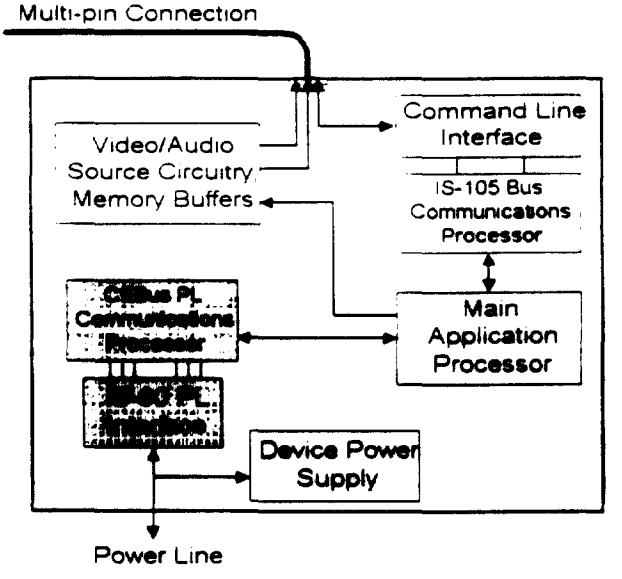

Figure 7. An IS-105 to CEBus PL Interface

The communications processor in the case of CEBus communications is likely to be an eight bit machine on the order of an 80C51 or 68HC11. Some non-volatile memory is required to store address assignments, device settings and User preferences. Depending on the implementation, the 8k to 12k of internal ROM of the mid-range versions of these chips may or may not be adequate for a complete solution. A high performance node also may require more than the 128/256 bytes of RAM available internally on mid-range 8-bit processors. High-end 8-bit microcontrollers meet these requirements.

Figure 8 presents one possible block diagram illustrating the circuitry associated with the CEBus Power Line interface. The circuitry is composed of a processor and its support circuitry, and the PL Medium Access Control (MAC) data pump controller, which deals with the transmission and reception of the spread spectrum symbols which comprise the PL messaging in IS-60.

The processor handles the higher levels of the Data Link Layer (DLL) CEBus protocol, performing error detection and correction, and duplicate packet rejection. Although it is conceptually necessary for this device to maintain non-volatile storage as shown in Figure 8, in reality the NVRAM of the main processor can provide this function.

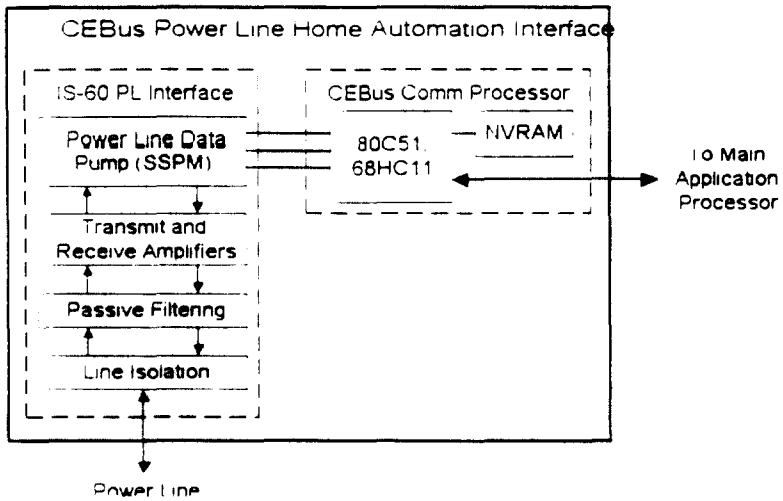

Figure 8. CEBus PL Home Automation Interface

LonWorks Interface Device Block Diagrams

Figure 9, below, depicts the equivalent interface device but with a LonWorks Power Line interface. The IS-105 Interface is unaffected by the selection of LonWorks as the main home automation bus standard.

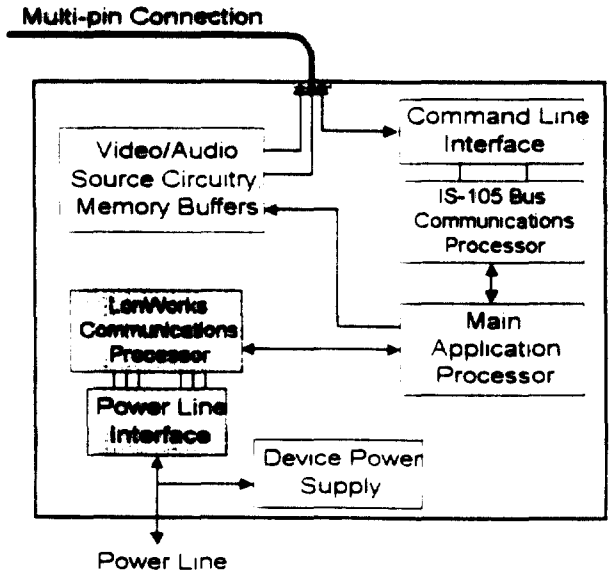

Figure 9. An IS-105 to LonWorks PL Interface

The hardware associated with the LonWorks HA system interface is considered next. Figure 10 presents one possible implementation. The communications processor in the case of LonWorks communications will be an Echelon Neuron. Since the main application processor handles all the heavy lifting the Neuron may require no external RAM or ROM, or an additional microcontroller.

It is possible that the best general design for the LonWorks version would include a Microprocessor Interface Program (MIP) enhancement to the main processor, so that the main processor would be more tightly coupled to the HA system. In this case the main processor would be unlikely to also handle the MAC and protocol layers of the IS-105 bus as shown in Figure 6, but instead would make use of a separate IS-105 processor as illustrated in Figure 5.

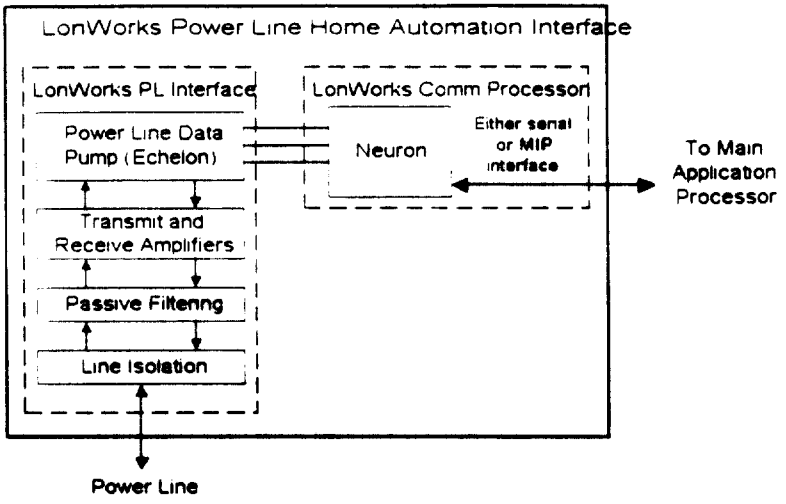

Figure 10. LonWorks PL Home Automation Interface

Conclusion

We believe that the cost of a practical implementation of a Home Automation interface to the proposed IS-105 bus will be substantially unaffected by the choice of the HA system. This conclusion is supported by the demonstration of the large proportion of circuitry which the various versions would have in common, as described in Figures 5 and 6, as well as by the substantial equality of the actual HA system interfaces required, as illustrated by specific example in Figures 8 and 10.

The IS-105 Decoder Interface is a reasonable and appropriate design. Although the use of a CEBus protocol subset on the bus may appear at first glance to favor this protocol over others, the truth is that the proposal favors no particular HA system.

References

1. EIA IS-105.1, Decoder Interface Standard, revision 5.1, with 4/18/95 mark-up
2. EIA IS-105.2, Decoder Interface Control Standard, revision 1.5, with 4/18/95 mark-up
3. Discussions with EIA staff regarding the recent addition of 4-bit-addressed, 8-bit tokens for general User Interface messaging (Markwalter; 6/2/95)