

## **Programming the APAD Interface**

P: 1.888.CRESTRON F: 201.767.6011

**APAD wall mount LCD controllers** provide versatile control opportunities. With its large white-on-black LCD display, scrolling nav-wheel and 4 backlit softkeys, the APAD affords full view of radio and music server information such as station, album, artist, song title, and genre, with easy navigation for browsing and control.

**Processor Support:** APAD are supported by 2-series processors running firmware versions 3.155 or later. APADs are not supported on x-gen processors.

**APAD Firmware:** Initial Release 1.00.12

**Cresnet Device:** Draws 3 Watts of Cresnet Power

**Reset Button** on the front Top left of the APAD circuit board (behind the face plate and next to the LCD). Touch the reset button while holding down a soft button to enter setup.

**Setup:** Adjust Cresnet ID, LCD Brightness, Screen Invert, LCD Contrast, Key Backlight, Sleep Timeout, Key click sounds.

### **VT-Pro-e**

The APAD LCD supports 2-color graphics only. Graphics with more than 2 colors will be reduced to 2 colors, which will result in poor image quality. Crestron therefore recommends using only 2-color graphics.

Very flexible APAD pages/sub pages can be created which employ Serial Indirect text fields that can be updated and changed through programming, or more traditional pages and sub pages can be created employing fixed text fields allowing for easier programming by eliminating the need to drive the APAD Display and menu's with serial text.

The APAD's buttons can be setup with project default join numbers, or these soft keys can be assigned different joins on a page-by-page basis.

APADS support many new features such as:

- Scrolling text with scroll rate analog joins, and scroll delay analog joins.
- Return Page Flips- Displays the previous (most recently viewed) page on the rising edge of the input.
- Cursor Position Analog Join - Moves the cursor to the corresponding character and highlights that character.

### **Programming for an APAD**

The documentation below references the following example programs and files.

- *APAD Basic Demo.zip* – Archived program and VT Pro-e project
- *APAD Advanced Demo.zip* – Archived program and VT Pro-e project
- *AADS-XM 12 Room Demo v1\_0\_6 w ARQ - 3-6-07 v2.zip* – Archived program and VT Pro-e project
- *Export Of AADS - APAD - ARQ.sba* – SystemBuilder Archive

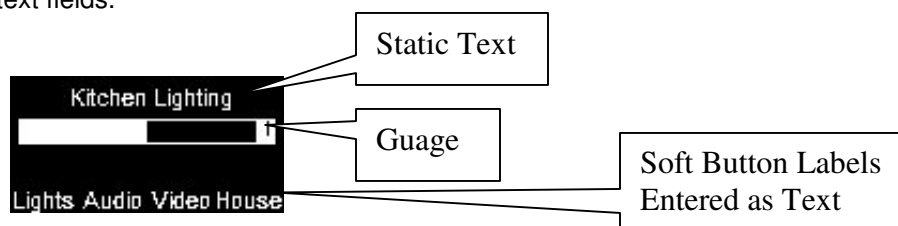
An APAD interface is more than a keypad as it contains an LCD display to which you can load projects created in VT Pro-e, however it does not provide all of the capabilities of a touch panel as the APAD's LCD display is not a touch screen. This hybrid interface provides some new

opportunities when building control interface projects and when programming the APAD in VT Pro-e and SystemBuilder. This document and supporting example programs provide guidance for building projects that employ the APAD in basic as well as more complex interface designs.

#### APAD Basic Example:

From the most basic level you can create pages and sub-pages in VT Pro-e and add text fields to the screen to display this information to a user. It is typical to label the four soft buttons below the LCD by including text labels along the bottom 1/4" of the LCD display.

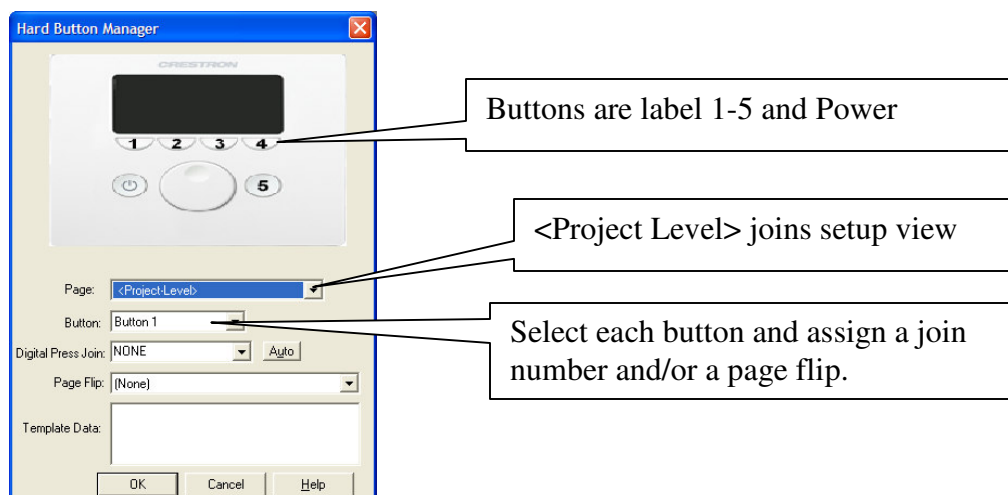
While many advanced features such as dynamic menu's are implemented using serial indirect text fields to which serial strings are sent from the control processor's program, a basic panel project can be created by adding static text fields to the pages in the VT Pro-e project. In this regard you create a page for the APAD in a similar manner to that of a touch panel, but instead of inserting buttons on the page you instead add borders, legends, and gauges. The *APAD Basic Demo* primarily utilizes static text fields.



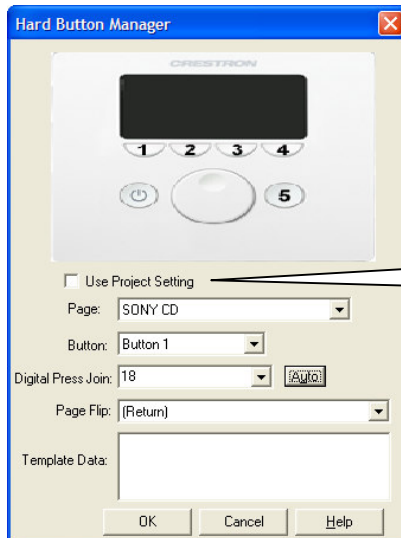
The physical buttons on an APAD are assigned join numbers, which then are tied into your program just as you would with a touch panel. There are 2 options for assigning join numbers to the APAD buttons. Both options are configured using the Hard Button Manager, which is accessed through the 'Tools' drop down menu of VT Pro-e.

Through the 'Hard Button Manager' dialogue window you have the option of selecting to use project level join numbers for the APAD, or to assign unique joins numbers to each button on a page-by-page basis.

- **Project Level Joins** – When you assign a join number or a page flip to any of the buttons while you have the 'Project Level' option selected in the Page drop down menu, you are assigning joins that will be sent for each of the buttons whenever 'Use Project Level Joins' is selected for the page. Essentially use project level joins when you will be managing the buffering of the button presses within your program.



- Assign Joins on a Page-by-Page basis - Use "Page Level" assignments when you want different join numbers (or page flips) to be assigned to each of the buttons for each page of the project. When you select a page from the page drop down menu a check box appears underneath the image of the APAD allowing you to assign project level joins to this particular page. If you leave this box unchecked you can assign joins and page flips for each button on the APAD and these joins will be specific to this page (i.e. these page specific joins will be transmitted when a button is pressed while the page is active).



Check to use project level joins  
for this particular page

**Note:** Assigning join numbers on a page-by-page basis simplifies your Simpl Windows program by allowing each button to be associated with a unique join, and thereby, a unique programmed function such as the "play" command on a CD player. Using "Project Level" assignments require that you use an interlock/buffer combination with the APAD button press signals so that you can enable/disable various buffers to pass the button press to the appropriate logic based upon the navigation of your APAD panel project. We recommend assigning joins on a page-by-page basis when this will simplify your Simpl Windows programming.

### Advanced Techniques

Many advanced features can be created by building panel projects that employ serial indirect text fields. With 'Indirect text' fields, the text that will be displayed is sent from the program running on the control processor.

#### Dynamic Soft-Labels for hard buttons

Serial indirect text can be used to label the hard buttons so as to allow the label to be changed by the control system based upon various conditions or states. For example perhaps you want soft button 2 to be connected to the "Play/Pause" signal on the CD player, which toggles between these two functions. If the CD player provides feedback as to its current state you could change the soft button label from "Play" to "Pause" by sending a new serial string triggered by the feedback from the CD player.

(Note that this simple example could also be accomplished with a "Legend" using the active and inactive states and employing static text, but as soon as you require more than 2 labels, indirect text is a better choice. The APAD does not support multi-mode buttons as the LCD screen is display only, it is not a touch panel)

Now playing Song  
Title and Artist fields



P: 1.888.CRESTRON F: 201.767.6011

138 139  
142 143  
144 145  
555 556 557 558

Soft Button Labels configured as  
Serial Indirect Text fields displayed  
on an APAD page in VT Pro-e

In the example shown above the bottom ¼" of the page includes four serial indirect text fields for soft button labels. Above the soft button labels are a number of additional serial indirect text fields. In this case these fields are arranged to provide 'Now Playing' information from a source that provides meta data such as an iDoc or media server.

#### Dynamic menu's

Dynamic Menu's such as scrolling lists can also be created using serial indirect text. In the page shown below, the top of the page is designed to display a scrolling list. The Serial Indirect text field at the top of the page (Serial Join 144) is designed to display the title of the scrolling list. For example the program could be designed to send the serial string " Cable Presets" to this field. This scroll list title field was created as a 'Border' in VT Pro-e but it could also have been defined as a text field; either way it was configured to display Indirect Text. The three fields below the title are the 3 lines of the scrolling list. They are defined as 'Legends' within VT Pro-e. These legend fields were configured to accept indirect text, which is evidenced by the serial join numbers displayed in the field when viewed in VT Pro-e. A 'legend' also allows you to define a digital join that can be used to change the state of the legend from 'Inactive' to 'Active' as you drive the digital join high/low in your program. In this example both the active and inactive state of the button are configured with the same Serial Join for indirect text, but the active state has the Font properties set to 'Underline'. When the digital join of the legend is asserted high the text displayed is underlined. With a scrolling list the underline is the cursor position showing which of the 3 fields is the currently active selection.

144 145 146 147  
555 556 557 558

Legend with indirect text join 145  
and digital feedback join 609.

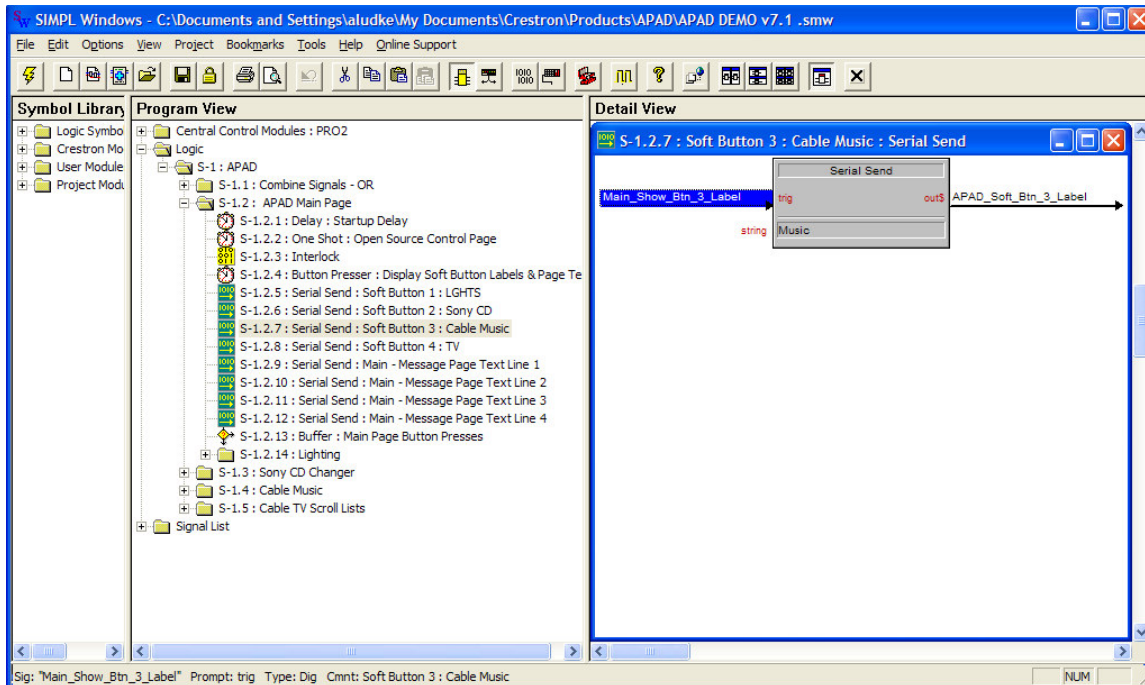
3 lines of the scrolling  
list defined as Legends

Note that the VT Pro-e project that is included with the *APAD Advanced Demo* program essentially uses only one page. All of the soft button labels and now\_playing strings are on sub-pages. The project has a number of sub-pages that can be held high through programming so that various combinations of sub pages can be displayed to create a dynamic page design that can be changed during runtime through programming. This provides a flexible means of building dynamic APAD projects that are driven through logic. Through programming you can enable various combinations of sub-pages on which you can display dynamic data. The dynamic data is also sent to the APAD from the program as serial text string. The APAD projects that are included with the AES and AADS demo programs work similarly; all text fields are on sub-pages. Within the AES and AADS demo programs you may notice that each soft button label is on a separate sub page. In fact each soft button has two sub pages associated with it. On one of these sub pages the serial indirect text field has the font defined as Arial with the active state set to underline so that driving the feedback high will display the text as underlined (this is the button feedback). The second sub page associated with each soft button is very similar with the exception that the font is set to the 'Crestron Transport' font. In this way a program could be written to display a text label using the Arial font over the button at one moment and later be changed to a transport icon. This flexibility is accomplished by displaying the appropriate sub page and then sending the appropriate characters to the indirect text field for the transport icon.

## Programming Dynamic Menu's

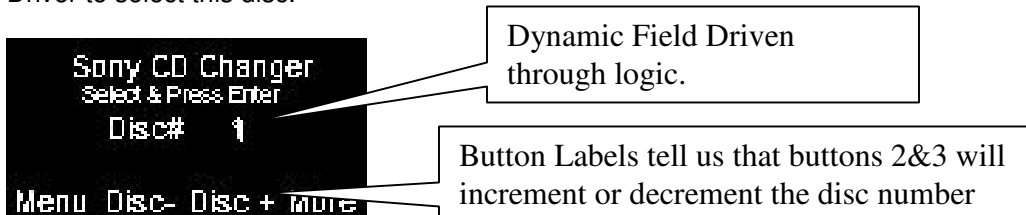
P: 1.888.CRESTRON F: 201.767.6011

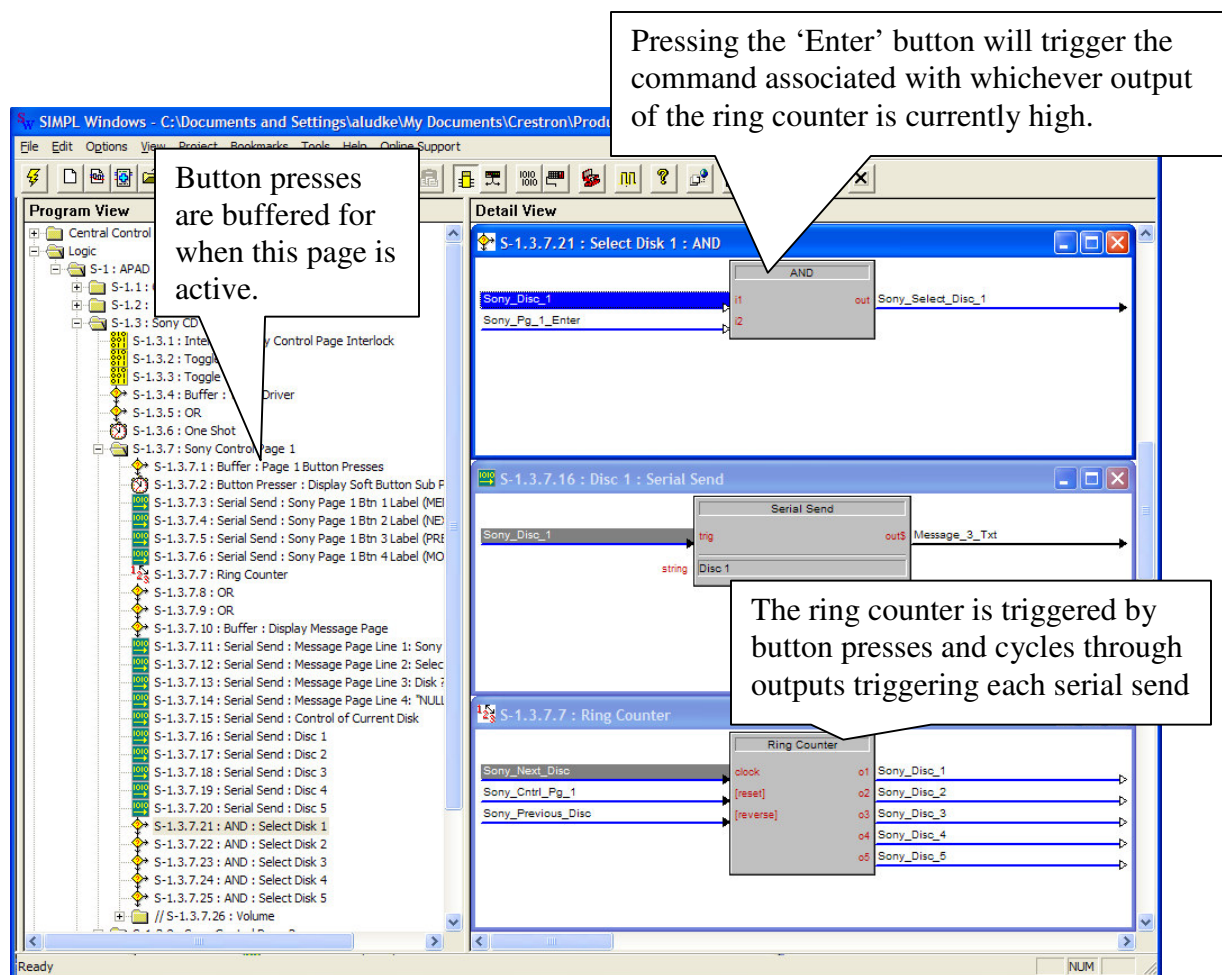
In programming there are a number of ways to drive the dynamic menu's and text on an APAD. To begin with, a serial indirect field requires that you send a serial string to the serial join associated with the field to display some text. There are a number of logic symbols that can be used within a Simpl Windows program to send a serial string to a serial indirect text field on a panel. In the screen shot below, the 'Detail View' of the Simpl Windows programming screen shows a 'Serial Send' symbol transmitting the string 'Music' to the 'APAD\_Soft\_Btn\_3 Label' field.



When creating dynamic menu's you need to create logic that will drive the menu's serial indirect fields with different values. In the *APAD Advanced Demo* program there are a number examples showing different methods for driving dynamic menus.

**Basic single line dynamic menu.** The Sony CD page in the '*APAD Advanced Demo*' program allows a user to select 1 of 5 discs. Buttons 2 and 3 allow the user to cycle through the discs and the Disc# field in the middle of the page changes to show the disc number. When the display shows the disc they want to play, pressing the 'Enter' button on the APAD would send the IR Command to the CD player to select this disc. To see how this menu is driven in logic take a look at Logic Folder S-1.3.7 in the program. The APAD button presses are buffered so that when this is the active page the hard buttons 2 and 3 are tied through an 'Or' symbols to a 'Ring Counter'. The output of the ring counter triggers the serial sends to send the changing disc number to the APAD display. When the desired disc number is displayed, pressing the 'Enter button' (button 5) send the correct command to the IR Driver to select this disc.





Note that there are other examples of simple dynamic menus in both the *APAD Basic Demo* and *APAD Advanced Demo* programs.

1. The cable box music pages provide a number of examples of how to create single line scrolling lists of channels and presets. These examples employ the APAD Knob to drive the scrolling list.
  - a. Scrolling through channels – In the *APAD Advanced Demo*, logic folder S-1.4.5.11 provides an example of employing the 'Clockwise' and 'Counterclockwise' pulses from the APAD Knob to drive a 'Numeric Keypad' symbol. The analog output of the keypad symbol is displayed on the APAD as an incrementing/decrementing channel number. The 'Enter' on the APAD sends the currently displayed channel to the cable box driver by passing it through a 'Decade', which with a 'Stepper' and some 'Buffers' pulses the appropriate numbers on the IR Driver. The 'Clockwise' and 'Counterclockwise' pulses from the APAD knob can be set to output either 8 or 16 pulses per revolution. In this instance setting this to 8 seems to be a better choice as the granularity of the pulses allows for easy tuning of a particular channel while 16 pulses per revolution seems to be too sensitive to knob movement to easily select a particular channel.
  - b. The *APAD Advanced Demo* program also provides another example of how to accomplish similar functionality using the 'APAD\_Knob(Acc)' signal. This is demonstrated in the commented out folder S-1.4.5.12. You should be able to uncomment folder S-1.4.5.12, and comment out S-1.4.5.11 to see this alternate method in action. Using the Accelerated Analog signal from the APAD Knob is a good choice when scrolling through long lists such as a list of 999 cable channels. The accelerated analog output traverses the analog values 0-65535 slowly when the knob is turned slowly and more quickly as the rate of knob rotation increases.

As such, you can quickly move through a lot of values with the same amount of granularity required to select an individual channel.

2. The second page of the Cable Music option also provides the facility to scroll through store and recall presets. There are also examples of 2 different methods of driving these menu's in Logic folders S-1.4.6.7.3 and S-1.4.6.7.3.

### Multi-Line Scrolling Lists

Crestron has created a user module to simplify the creation of Multi-line scrolling lists that contain up to 50 list items. The 'TV Page' in the 'APAD Advance Demo' demonstrates the implementation of the 'Scrolling List (cm)' module. This module will only work with a static list of items such as a list of 50 presets. The module is not intended for integrating a scrolling list from a device such as an Audio Request from which the list items are served up from another device.

#### Inputs

- The module contains serial Inputs labeled 'Item1\_Text' through 'Item50\_Text' to which you send the strings that you want displayed in the list.
- There are also 50 analog inputs allowing you to associate an analog value with each list item; channel numbers for example. 'Item1\_Data' through 'Item50\_Data'
- There are also serial inputs which are intended to be used for displaying dynamic graphics in the scrolling list such as Channel icons. The image fields would not be used when implementing a scrolling list on an APAD. Please reference the separate scrolling list example projects to see how this is implemented on a touch panel using dynamic graphics.
- Finally there are digital inputs for controlling the navigation of the scrolling list

#### Outputs

- The digital outputs labeled 'Item1\_Selected' through 'Item50\_Selected' are the signals that are used to trigger the logic associated with a scrolling list items when the item is selected from the list.
- The serial outputs 'Line1\_Text' through 'Line15\_Text' drive the scroll list serial indirect text fields on your APAD or touch panel. So you can create a scrolling list with up to 15 lines displayed at a time.
- The Serial outputs 'Line1\_Image' through 'Line15\_Image' drive the scroll list dynamic graphic field when utilized on a touch panel that supports dynamic graphics.
- The digital outputs 'Line1\_Highlighted' through 'Line15\_Highlighted' handle driving the feedback on the serial indirect text fields on your APAD or touch panel indicating the cursor position or the currently active line.

In the examples provided in the *APAD Advanced Demo*, the scrolling lists modules are driven by the 'Clockwise' and 'Counter Clockwise' digital signals from the APAD knob, and the currently highlighted scroll list item is selected by the APAD 'Enter' button.

### Controlling devices that provide dynamic data

Device such as the Audio Request or a Kaleidescape System are designed to be controlled using dynamic menu's. In some cases existing modules for these devices are designed specifically for integration with a touch panel. For example the Audio Request modules have signals to page up and down through the list of albums and artists ect.. In this case the module provides digital inputs for moving through the list by 4,5,6, or more lines at a time. When integrating with an APAD you really want to be able to move through the browse list 1 line at a time as with the APAD you'd typically control the navigation of the browse menu with the APAD Knob and the Enter Button. Many of these modules do not currently provide the ability to scroll one line at a time...they only provide page up and down functions. Often it is easy to modify existing modules to provide this 1 line scroll functionality.

### AADS/Audio Request Simple Windows Example:





This example is based on the AADS 12 Room Demo program. We have integrated the 'ARQConnect Basic v4\_5\_5' and the unmodified 'ARQConnect Advanced V4\_5\_5' module. The only change we made to the original Audio Request module was to modify the navigation to facilitate moving 1 line at a time through the browse menu. These examples are setup for a serial controlled single zone Audio Request Fusion Pro. Changing the Simpl Windows example over to Ethernet should be straightforward; integrate a TCP/IP client and change the connection signals on the ARQ module to Ethernet. You will need to load the APAD OOTBF v1.0.8 with Type 7-11 Generic Source Pages.vtp to the APAD's

**Required Files:**

AADS-XM 12 Room Demo v1\_0\_6 w ARQ - 3-6-07 v2.zip  
APAD OOTBF v1.0.8 with Type 7-11 Generic Source Pages.vtp

**AADS SystemBuilder Example:**

We have also created a SystemBuilder example, which shows how to control the Audio Request from an APAD in SystemBuilder. This example includes a user module that packages the Audio Request modules for use within SystemBuilder. The attached version does not provide the facility to control the ARQ from a touch panel. The decision was made to keep this example simple and easy to follow. A programmer can easily modify the user module to expose additional signals on the modules 'Define Arguments' symbol so as to expose additional features and functions.

After dragging in the Fusion Pro module from the UserDB and allowing SystemBuilder to create the APAD pages for the ARQ, you'll need modify the Audio Request pages created by System Builder to create/add a scrolling list sub-page with the serial indirect text fields for the Browse Menu. In this example we made a copy of the AAS sub-page and rejoined the fields and then connected the newly joined legend and border fields to the Audio Request module.

**Required Files:**

Export Of AADS - APAD - ARQ.sba  
Audio Request APAD SystemBuilder.umc