

Distributing Composite Radar Data

How to use SPx Server to distribute packed radar data from an HPx card and separate the data at the client.

Summary

HPx radar input cards provide two channels of analogue video input and up to eight 1-bit digital video inputs.

These inputs may be combined together by the HPx in a number of different ways, to give a composite data output.

For example, the data from the two analogue channels may be combined into 16-bit data samples, comprising 8 bits from channel A and 8 bits from channel B.

This application note explains how such data may be handled by SPx Server and/or at a receiving client.

Overview

Composite radar video data samples from an HPx card may be separated within SPx Server and distributed in separate channels or they may be distributed as composite data and separated at the client. Generally the former method would be preferred because it reduces the processing within the client and also allows SPx Server to operate (for example perform tracking) on a single radar video channel. However, there may be occasions when the latter is preferred, or even necessary, e.g. if network bandwidth is limited.

The diagrams below summarise these options.

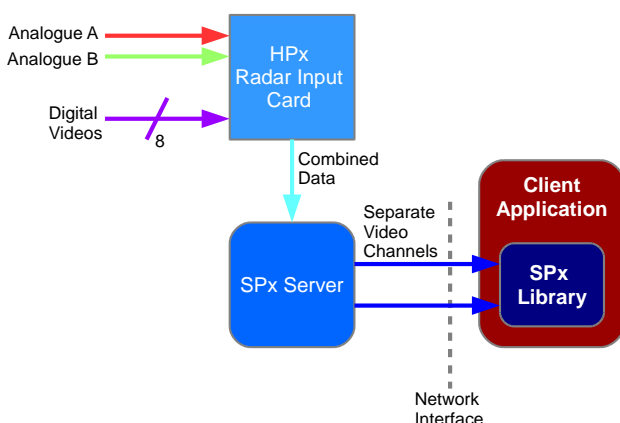


Figure 1a: Using SPx Server to separate composite data and distribute on multiple channels.

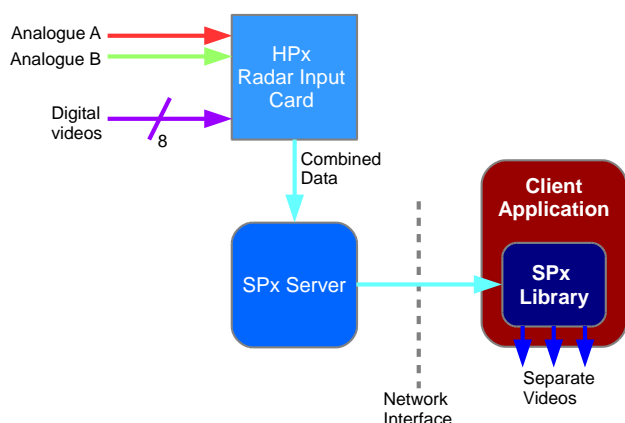


Figure 1b: Distribution of composite data and its separation at the client.

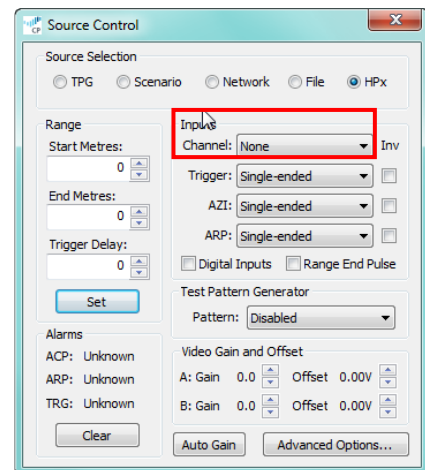
SPx Server Configuration

This section describes how to configure SPx Server to select the different input options through the HPx card and to separate and distribute the data.

Selecting HPx Inputs:

The composition of the data samples output from the HPx card is simply controlled by selecting one of the predefined modes for the "Channel" field within the HPx input source dialog. This channel mode can also be selected via the `SrcHw.Channel` parameter.

The supported channel modes for the HPx card are described in the HPx user manual and corresponding values for the `SrcHw.Channel` are given on the SPx Server Configuration Guide. For example, setting `SrcHw.Channel` to a value of 16 will configure the HPx card to output 16-bit samples comprising digitised analogue channel A data in the top 8 bits and digitised analogue channel B data in the bottom 8 bits.



Note that the `Svr.MaxBytesPerSample` parameter, which sets the expected number of bytes per sample from the HPx card, must be set to a value of 2 for any channel modes that result in sample sizes greater than 8 bits. This is a startup parameter and can only be set in the configuration file, there is no GUI control.

Splitting the Video Samples:

SPx Server provides a LUT (look-up table) process which may be used to separate composite samples into separate data streams. The LUT process provides a number of built-in modes and also allows a custom LUT to be loaded. In most practical situations there will be a built-in LUT already available to split the video as required.

By distributing the "processed" video, i.e. after it has passed through the LUT, SPx Server will be sending separated data rather than the composite data. For example, if SPx Server is receiving composite 8-bit Channel A + 8-bit Channel B data from the HPx card, an SPx Server configuration file to separate and distribute the top 8-bit data (Channel A) might contain the following settings:

```
Svr.MaxBytesPerSample = 2      # Set to 2 because expecting 16-bit samples from HPx card
SrcHw.Channel = 16             # Channel mode 16 = 8-bit Channel A + 8-bit Channel B
ProLUT.Available = 1          # Allow the LUT process
ProLUT.Enabled = 1            # Turn on the LUT process
LUT8.Type = 101               # Sets LUT to "SPX_LUT16_TOP8", takes top 8 bits of each 16-
bit sample
Dist.Available = 1             # Distribution available
Dist.Enabled = 1              # Turn on distribution
Dist.Source = 0               # Distribute processed data
```

The auxiliary channel is available in SPx Server to facilitate splitting out a second channel of data from composite samples. The auxiliary channel is a copy of the raw input to SPx Server and has its own LUT process that can be used to separate sample data, as shown in Figure 2.

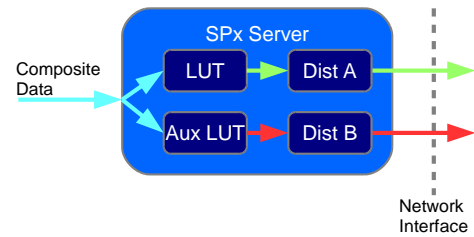


Figure 2: Primary and Auxiliary Channels of processing and distribution

For the example configuration above, the bottom 8-bit data (Channel B) could be sent via the auxiliary channel by making the following settings:

```

ProAux.Available = 1      # Allow the auxiliary channel
ProAux.Enabled = 1       # Turn on the auxiliary channel
AuxLUT.Type = 102        # Sets LUT to "SPX_LUT16_BOTTOM8", takes bottom 8 bits of each 16-bit
                           sample
Dist.Enabled = 2         # Configure SPx Server for 2 channels of distribution
DistB.Source = 3         # Set second distribution channel to be the auxiliary data
    
```

Distribution of Composite Data:

SPx Server may be configured to distribute the composite data samples from an HPx card, without splitting them into constituent data parts. This is simply achieved by distributing "raw" data, rather than "processed", onto the network:

```

Dist.Source = 1          # Distribute the raw data
    
```

It is possible to distribute the "raw" samples onto the network but still split them (via a LUT) for further processing within SPx Server. Attempting to track on composite data may not be a valid, so splitting the data prior to tracking could be desirable, even when distributing the composite data.

In cases where SPx Server has been configured to receive 12-bit or 16-bit data samples from an HPx card the distributed data will be sent as 16 bit samples. Note that the PPI window within SPx Server automatically displays the top 8 bits of any input samples as the "raw" data display.

Client-Side Processing

LUT processing, similar to that employed by SPx Server, may be used within client application code to separate composite data.

Figure 3 shows the general outline structure of SPx library object used to receive and separate composite data into two streams. This could be extended to split the data more streams by adding further "SPxProLUT" processes.

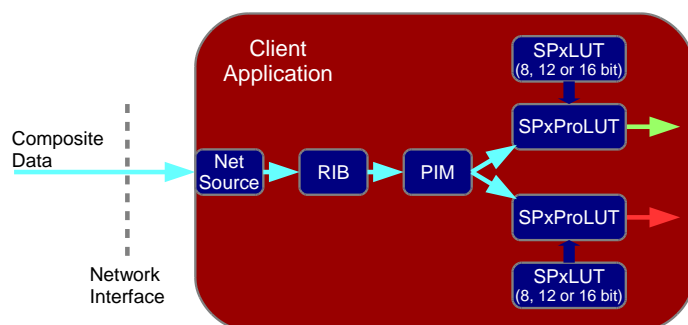


Figure 3: LUT processing to separate composite data into two separate data streams

Additional processing, for example scan conversion, may simply be added to the output of the "SPxProLUT" processes, in the usual SPx fashion.

< End of document >