Advanced Simulation Technology inc.
441-A Carlisle Drive
Herndon, Virginia 20170 U.S.A.
Tel. (703) 471-2104
Fax. (703) 471-2108

ASTi: Digital Audio Communication System

Model Builder Reference Manual

Version 3.10A

May, 1997

Introduction 9

Model Builder Menus 11
- Main Menus 12
- Models Menus 14
- Controls List Menus 20
- Signals List Menus 22
- Feeders List Menus 24
- Functions List Menus 26
- Sound Files List Menus 28
- Host Interface Menus 30
- Digital Signal Processor Menu 32
- DIS Network Menus 36
- Voice Network Menus 38
- Local Network Menus 40

Object Inspector Panels 43
- Inspector Panel 44

Signals 47
- Basic Signals 48
  - Sine Wave Signal 49
  - Triangle Wave Signal 52
  - Sawtooth Wave Signal 55
  - Square Wave Signal 58
  - Filter Signal 61
  - Play Soundfile Signal 64
  - Amplitude Modulator 67
  - Analog Input 70
  - VOX 72
  - Signal Selector 76
  - Signal Mixer 79
- Pulse Stream and Pulse Sequence (Pulse Stream Package) 83
  - Pulse Signal 84
  - Pulse Stream Signal 87
  - Pulse Sequence Signal 96
  - Pulse Step Signal 102
- Engine Signals (Engine Package) 107
  - Engine 108
  - Small Engine 111

Propeller 114
Rotor 119
Radio Signals (Radio Package) 125
Radio 126
Receiver Signals (Radio and Nav-Aids Packages) 139
Receiver 140
Receiver 141
Signal PDU Receiver 145
Receiver with BFO 147
Transmitter Signals (Radios & Nav-Aids Packages) 151
Transmitter 152
Jammer 156
Signal PDU Transmitter 159
Navigation Aids Signals (Nav-Aids Package) 163
VOR Beacon 164
NDB Beacon 167
DME Beacon 170
TACAN Beacon 173
Continuous Wave Beacon 176
Collocated VOR/DME 179
Collocated VOR/TACAN 183
Marker Tone 187
Intercom Signals (Intercom Package) 191
Intercom 192
Intercom over DIS or VoiceNet 195
Record Replay Signals (Record/Replay Package) 197
Record / Replay Signal 198
Play Sound Signal 201

Feeders 205
Basic Feeders 206
Buffer 207
Balancer 209
Stereo Pan 211
Link 213
Link Filter 215
RIU Feeders (RIU Package) 219
RIU Audio Input 221
RIU Audio Output 224
Local RIU 227
Analog Out Feeder 231
Output 232

Record Replay Feeders (Record/Replay Package) 235
Record / Replay Feeder 236

Communication Panels (Radio & Intercom Package) 241
Communications Selector Panel 242

Global Channels 247

Global Channels 248
Global In Signal 251

Controls 253

Host Inputs 254
Float Input 255
Integer Input 258
Unsigned Integer Input 261
Boolean Input 264
8 Bit Input 267
Message List Input 271

Host Morse Inputs (Nav-Aids Package) 275
Morse Keyer Input 276
Morse Identifier Input 279

Host Outputs 283
Float Output 284
Integer Output 287
Boolean Output 289

RIU Digital I/O (RIU Package) 291
RIU Digital Input 292
RIU Digital Output 295

Internal Controls 297
Math Function 298
Signal Function 301
Logic Table 304
Integer Table 307
Bit to Byte 310
Counter 313

Navigation Controls (Radios & Nav-Aids Packages) 317
Geodetic World Position 318
Geodetic World Position 321
Entity 323

Functions 327

Table Function 328

Polynomial 330
Scale and Limit 332
Log / Antilog 334
Lag Filter Function 336
Adder/ Subtracter/ Multiplier 338
Divider 340
Random Number 342
Comparator and MaxMin 344
Switch 346
Function Group 348
Swap / Duplicate 350

Sound Files 353
Sound File 354
Sound File Group 359

System Configuration 361
Configuration Files 362
What is a Configuration file? 362
Saving System Configuration Files 362
Using the Configuration file 362
Configuration file commands 363
Sample Configuration File 363
System Configuration Pages 364
Sound Model Configuration file commands 365
Model 365

Ethernet Configuration Commands 373
Overview of Ethernet Header Conventions and Message Passing 374
Introduction 374
The Ethernet Header 375
The I/P Header 376
UDP Header 376
Ethernet Configuration Commands 382
Bit3 Configuration Commands 391

Voice Net Configuration Commands 393
Voice Net Configuration Commands 396

DIS Configuration Commands 401
DIS Configuration Commands 406

Figure 1	Main Menus	13
Figure 2	Models Menus	15
Figure 3	Controls List Menus	21
Figure 4	Signals List Menus	23
Figure 5	Feeders List Menus	25
Figure 6	Functions List Menus	27
Figure 7	Sound Files List Menus	29
Figure 8	Host interface Menus	31
Figure 9	DSP Menus	33
Figure 10	DIS Network Menus	37
Figure 12	Local Network Menu	41

Introduction

This manual provides detailed information on the model builder menu structure, and the operation of each sound object.

NOTE: Not all of the features and menu items that appear in this manual may be on your system, depending on what software and hardware features your system has.

Model Builder Menus

The ModelBuilder software contains the following major areas:

- Up to six sound model development, status, and runtime areas.
- Host computer interface control and status.
- DIS or Voice Network interface, control and status.
- Waveform DSP software and hardware monitoring.
- System status and error displays.

A sound model consists of a group of sound, mixing, and control objects which control a single DSP, hence a 4 DSP system would contain 4 models.

Each Model contains five lists of objects segregated by function, namely:

- Control objects which interface the model with the incoming ethernet packet.
- Signal objects which provide the fundamental sound sources via the DSP.
- Feeder objects which determine the mixing ratios for each signal onto the signal highway, and hence to the analog outputs.
- Function objects which provide mathematical relationships, such as polynomials and table look-ups.
- Sound file objects and groups which provide a directory of prerecorded sounds.

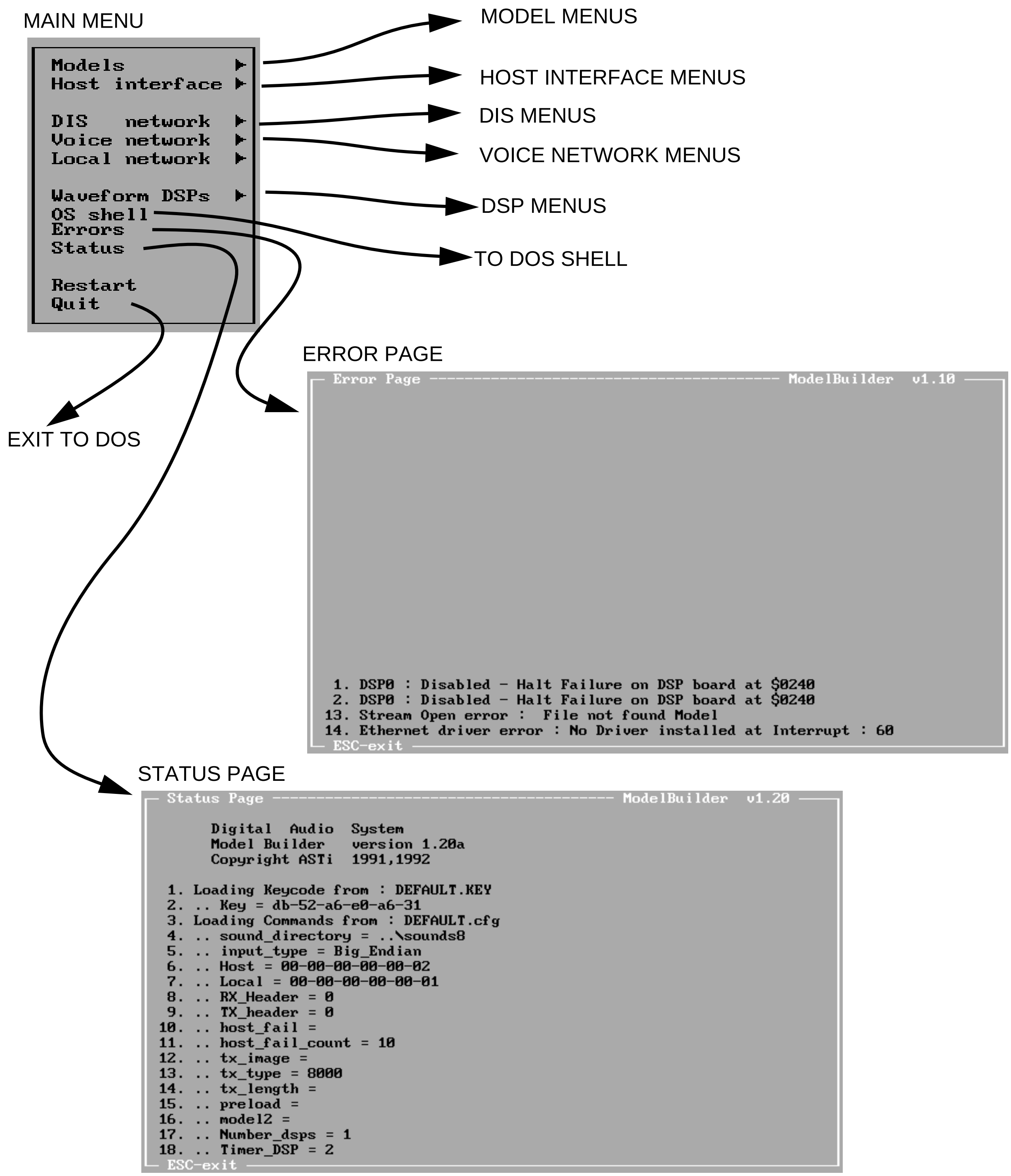
The user interface for each of the above is via a series of pull-down menus, which are graphically documented in the following pages.

Main Menus

The main menu provides access to:

- Model development, and runtime environment monitoring and control.
- The host computer interface.
- The waveform DSP status and control pages.
- The DOS shell.
- The DIS and/or Voice Network status and control areas.
- The error reporting page.
- The status message page.
- DOS by quitting ModelBuilder.

Figure 1 Main Menus



Models Menus

The model menu provides:

- Access to up to six models. (The number of models is the same as the number of waveform DSP boards in the system.)
- A timing page which lists the current duration of each model, the total time spent processing all models, the system master model rate, and the individual model rates. This menu is described in detail below.
- A memory page which lists the current filename of each model, the total memory used for all models, and the Model Builder Version under which the model was saved.
- An options display page displaying the currently installed software packages.
- A paste board model which facilitates inserting parts of other sound models into an existing model.

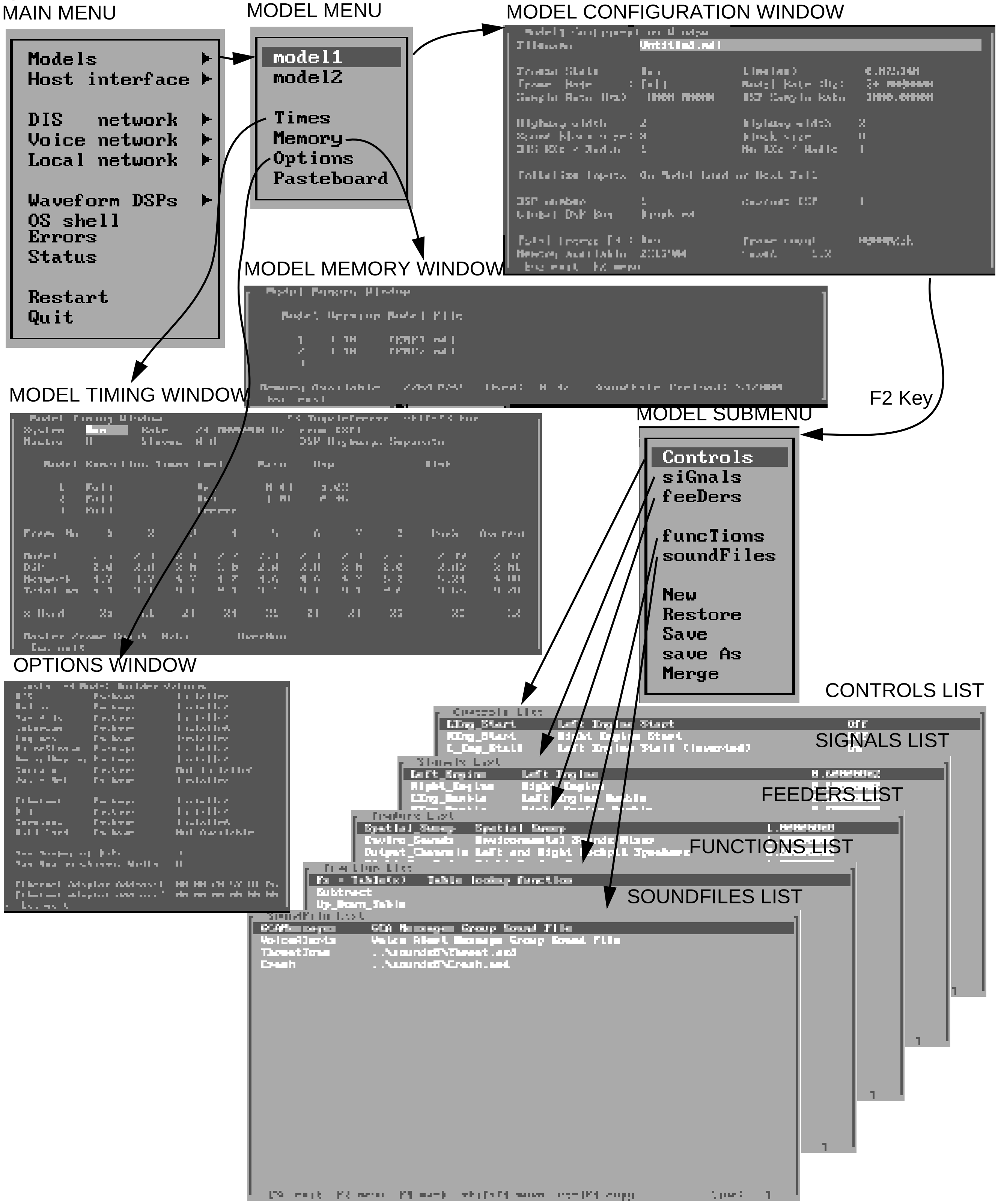
Each model has a configuration window which provides control over:

- Model filename
- Freeze state
- Model Iteration Rate
- DSP sample rate
- Highway width
- Sound file buffer size
- Control value initialization selection.
- Communication between a pair of DSPs
- Analog input preamplifier gain control

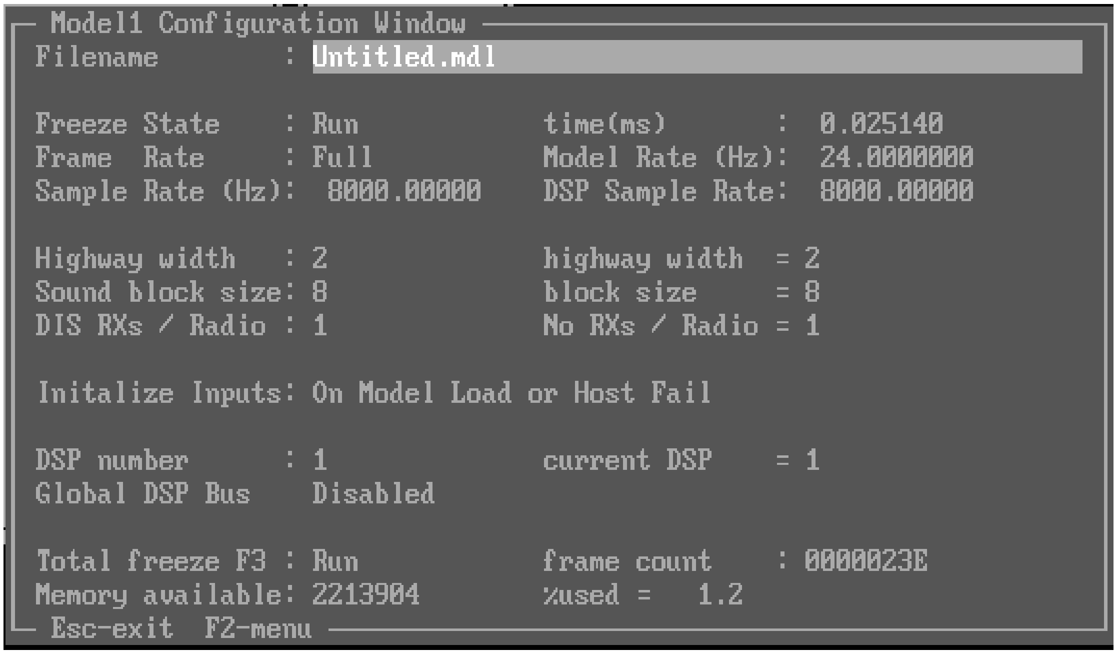
The F2 key provides access to the model sub-menu, which in addition to bringing up the five object lists that constitute the bulk of the model, also provide:

- Model reinitialization
- Model loading
- Model saving
- ASCII export capability
- Merging of objects from the pasteboard

Figure 2 Models Menus



Model Configuration Window



Sample Rate, DSP Sample Rate

The DSP sample rate is the number of digital samples of a waveform that are processed per second, as well as the rate at which external audio input is sampled. The highest frequency reproducible is one half of the DSP sample rate.

The sample rate is any number entered by the user. The DSP can only run specific sample rates - the DSP sample rate will be the closest rate possible to the user entered sample rate.

The sample rate can run from 7,000 to 48,000 Hz. Increasing the sample rate increases the processing load on the DSP board.

Model Rate

The model rate is a parameter that is set in the Model Timing Window or from the start up configuration file, and is the same for all models. It determines the maximum rate at which changes in the model parameters will be sent to the DSP cards. It should be equal to or slightly higher than the rate at which the host computer sends ethernet packets to the DACS.

The model parameters are changed, for example, when a Press to talk button is pushed or a radio frequency is changed. The model will react to these changes no faster than the model rate.

Model Configuration Window

Frame Rate

The frame rate determines at what fraction of the model rate the given model will have its parameters updated. If set to full, the model gets updated at the model rate. If set to half, it gets updated at half the model rate. This is useful because there is one model rate for all the DSP boards - if some models do not need to be updated as often the frame rate for those models can be reduced to reduce the processing load.

The frame rate can be set to full, half, quarter, or eighth. For fractional rates, the frame in which the model gets updated is also specified. For example, if the model rate is 24 Hz and one model is set to Half -1 and the other is set to Half -2, both models will be updated at 12 Hz, and will be updated in alternate frames.

Frame Count

This is a counter that indicates the number of frames processed. If the model is running normally this counter will be incrementing.

Highway Width

Number of output sound channels. A channel is one analog input and one analog output - so a channel width of two would be two inputs and two outputs.

The waveform synthesizer can only use up to eight channels. A TDM controller card (with RIUs) can support up to 24 channels.

Global DSP Bus

Indicates whether or not the DSP is connected to the Global DSP bus. The Global DSP Bus allows sound to be passed from one DSP card to another.

Enabling the global bus for the DSP card is done through either the Model Timing Page, or the configuration file. See the sections on Global Channels and System Configuration for more information.

DSP Number

The number of the current DSP board.

DIS RX / Radio

Determines the number of independent simultaneous sound streams that can be taken from the DIS network and listened to on a single radio.

Increasing this number increases the buffer memory allocated to each radio on the DSP card - so the higher the number, the fewer radios which can be used.

When simulating FM communications over the DIS network, a value of one is sufficient, as an FM radio can only pick up one audio stream at a time. When using AM radios or Intercom buses over the network, a higher value may be necessary.

Freeze State

If the freeze state is set to run, the model runs. If it’s set to freeze, the model freezes. This only affects one model.

Total Freeze F3

Hitting the F3 key at any time will freeze all of the models. Hitting F3 again will unfreeze them. This field indicates whether F3 has been pressed.

Sound Block Size

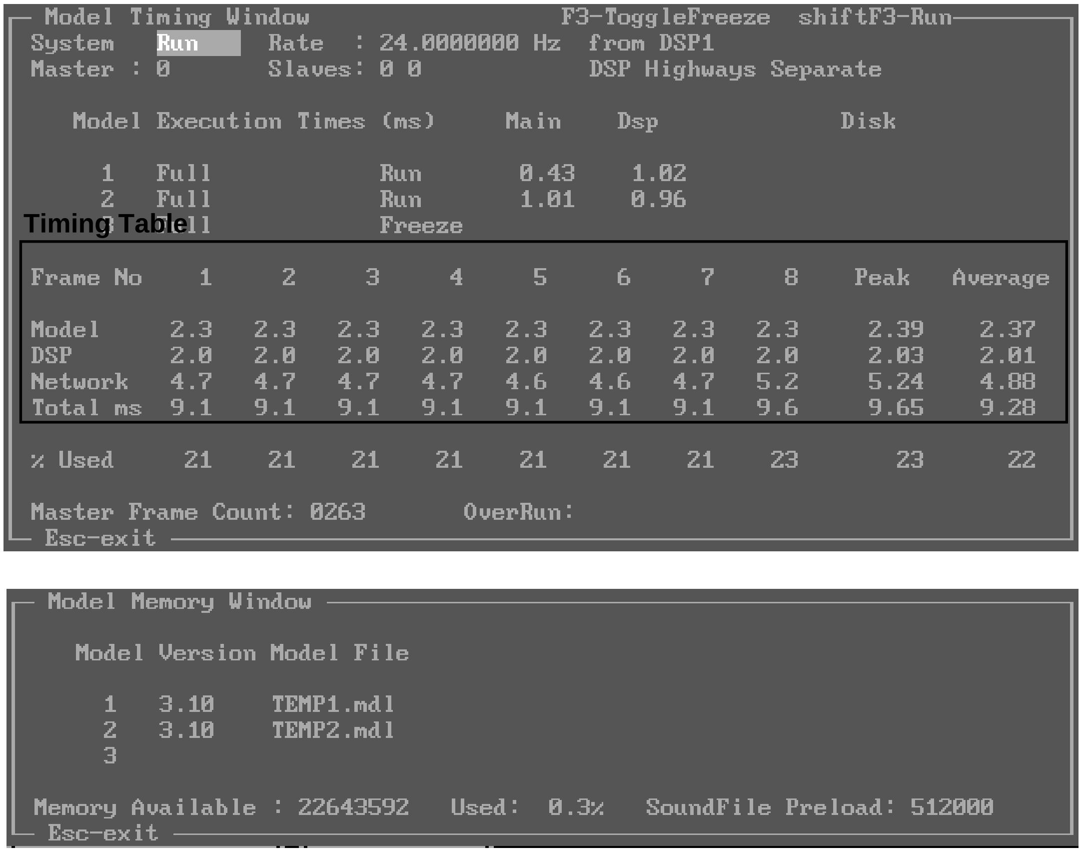
The sound block size is only used when sounds are being played from files on the disk. It indicates the size of the buffer memory on the DSP card allocated to holding the recorded sound before it’s played back.

For larger sound block size, fewer sound files can be played at the same time, and less cpu time is used. A smaller block size allows more sounds to be played from the disk but could cause gaps in the sounds if the CPU is heavily loaded.

time (ms)

This is the time per frame that the CPU spends servicing the model.

Model Timing and Memory Windows



System

A flag for freezing all of the models. Changing this field is equivalent to hitting F3.

Rate

The model rate. Each individual model will have its parameters updated at this rate, or at some even fraction (half, quarter or eighth) of this rate.

DSP Highways Separate / Linked

This field is a legacy to provide backwards compatibility to models made on earlier versions of Model Builder. New models should keep this set to separate.

Master/Slave

For systems with two communicating DSP boards, one must be set as a master and one as a slave. Leaving these fields at zero will disable communication between DSP boards. In addition to these fields, master and slave jumpers must be set on the DSP boards.

The sample rate of communicating boards must be the same. The rate is taken from the master board. Other than this, there is no distinction between master and slave boards.

Model Timing and Memory Windows

Model Execution Time

Lists execution times and parameters for all the models.

First, the frame rate as a fraction of model rate is given. This is the same as the Frame Rate field in the model configuration window.

Next is the Run/Freeze state of each model. This is the same as the Freeze State field in the Model Configuration window.

Next are listed execution times for Main, DSP, and Disk for each model, in milliseconds.

Timing Table

The timing table gives a detailed account of the use of CPU time on the system. The times needed for the CPU to complete the various necessary tasks are given in milliseconds. Model is the amount of time spent computing the radio received powers, updating counters, etc. DSP is the time the CPU spends sending updated information to the DSP boards. Network is the amount of time the CPU spends servicing the ethernet connection.

Eight frames are listed because a given model could be running at a fractional frame rate, so it wouldn’t use full resources on every frame. If all the models are running at the Full frame rate, each column of numbers should be roughly equal.

The total time CPU time available per frame is one over the frame rate (e.g. for a frame rate of 25 Hz, there are 40 ms /frame to complete the necessary tasks.) At the bottom is the percentage of the frame being used, which is just the total ms per frame divided by the time available per frame. These numbers can go up to 100% before OverRuns start occurring.

Master Frame Count

This counts the master frames being executed.

OverRun

This counts the number of frames that were not processed properly because of an excessive burden on the CPU. This number should be zero in normal operation - if it is incrementing, the CPU cannot keep up with the demands of the models. If this is occurring, possible solutions would be reducing the frame rate, reducing the model complexity, or upgrading the CPU board.

Model Memory Window

The model memory window lists the model name for each model, the version of Model Builder the model was saved from, the available memory and the memory used, and the soundfile preload. The soundfile preload indicates the number of bytes of a soundfile that are preloaded into the computer memory from the disk when the model is replaying a recorded sound.

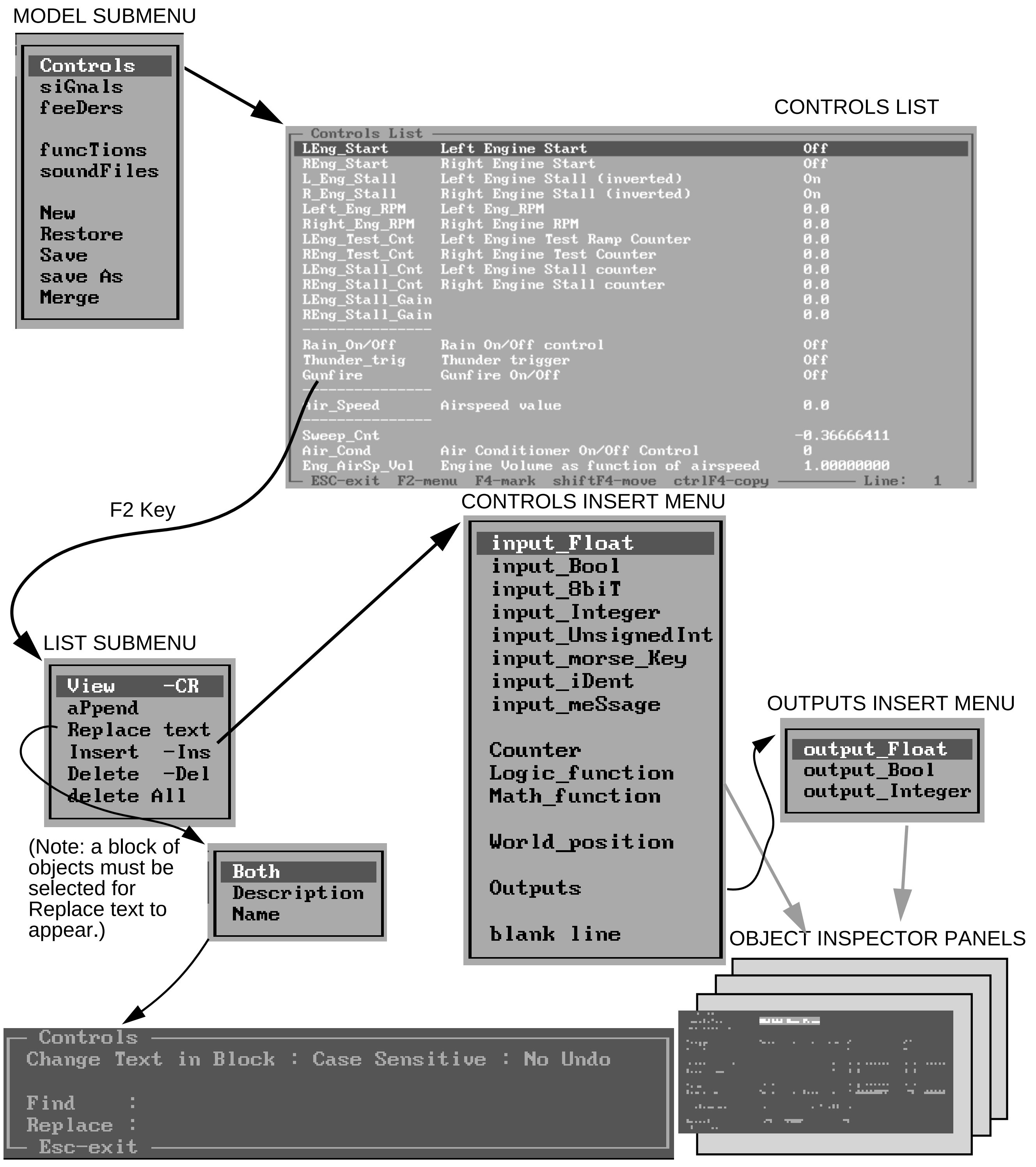
Controls List Menus

The controls option on the model sub-menu provides access to the list of control objects. These are principally the ethernet interface variables used by this particular sound model.

In common with all of the object lists the following keys are active:

- Cursor arrows, Page up, page Down, Home, and End for movement of object highlight.
- Enter (CR): brings up inspector panel for currently highlighted object.
- Ins: Inserts a new object in list.
- Del: Deletes current object.
- F1: steps through alternative list display formats
- F2: brings up list submenu.
- F3: freezes/ unfreezes all models.
- F4: marks an object as well as the beginning and end of a block.
- shift F4: moves marked object or block of objects to current position.
- control F4: creates a new copy of the marked object or block of objects.
- Replace text feature: allows the user to change text in the name and description fields of all the objects in a selected block. (A block of objects must be selected with the F4 key).

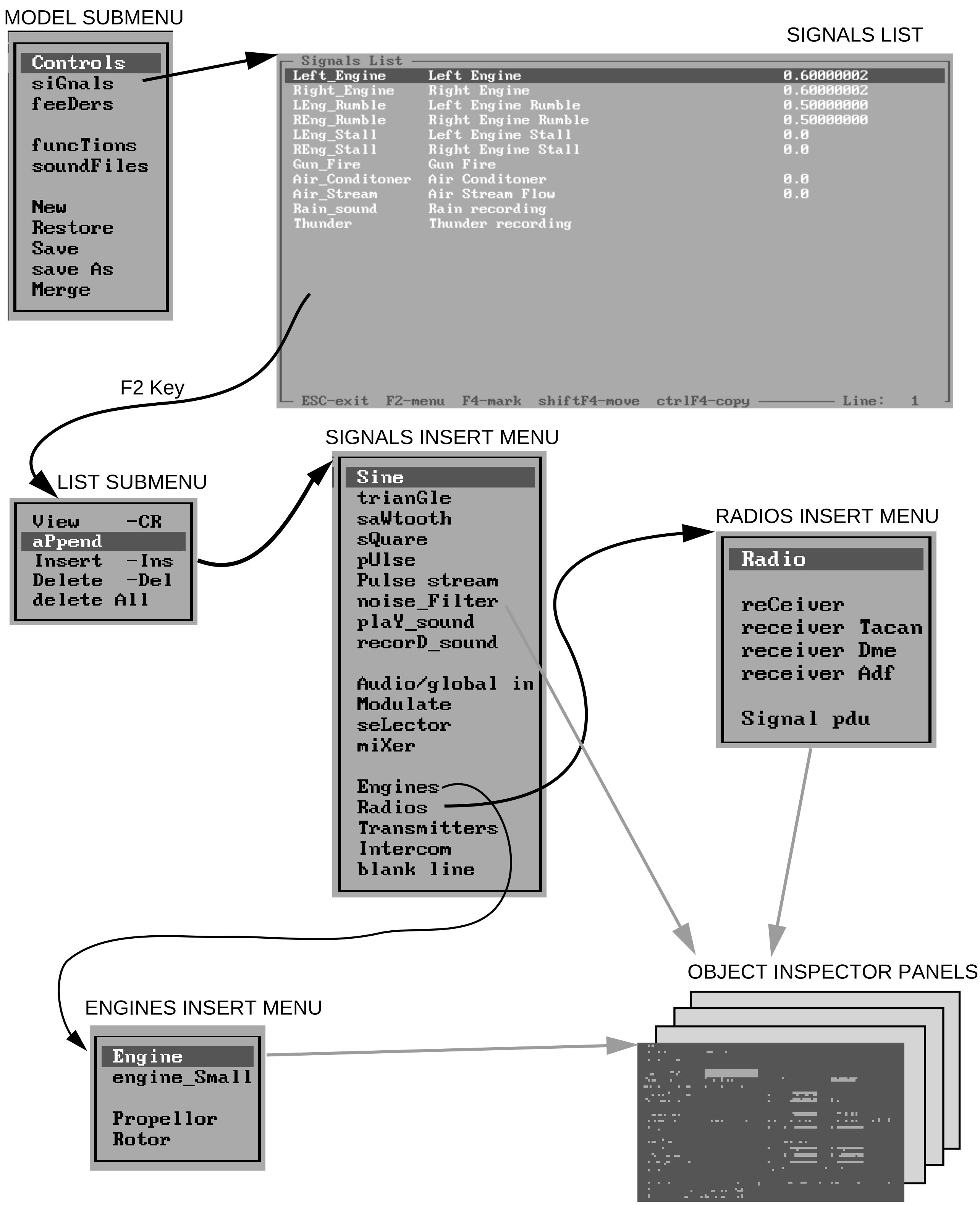
Figure 3   Controls List Menus



Signals List Menus

The signals list is the set of sound source objects in the model, and has the same keystroke access as the control list.

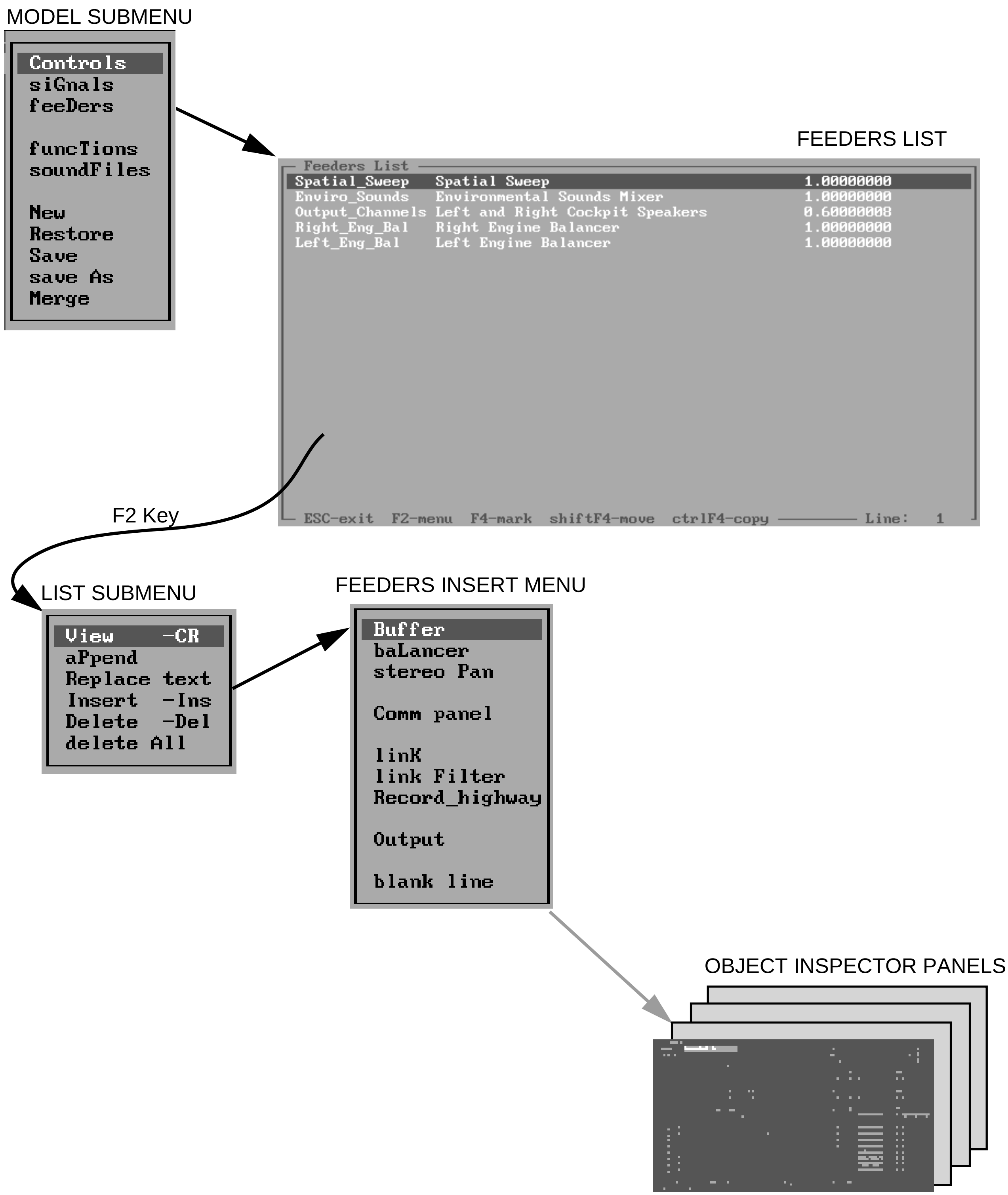
Figure 4 Signals List Menus



Feeders List Menus

The feeders list is the set of highway mixing objects in the model, and has the same keystroke access as the control list.

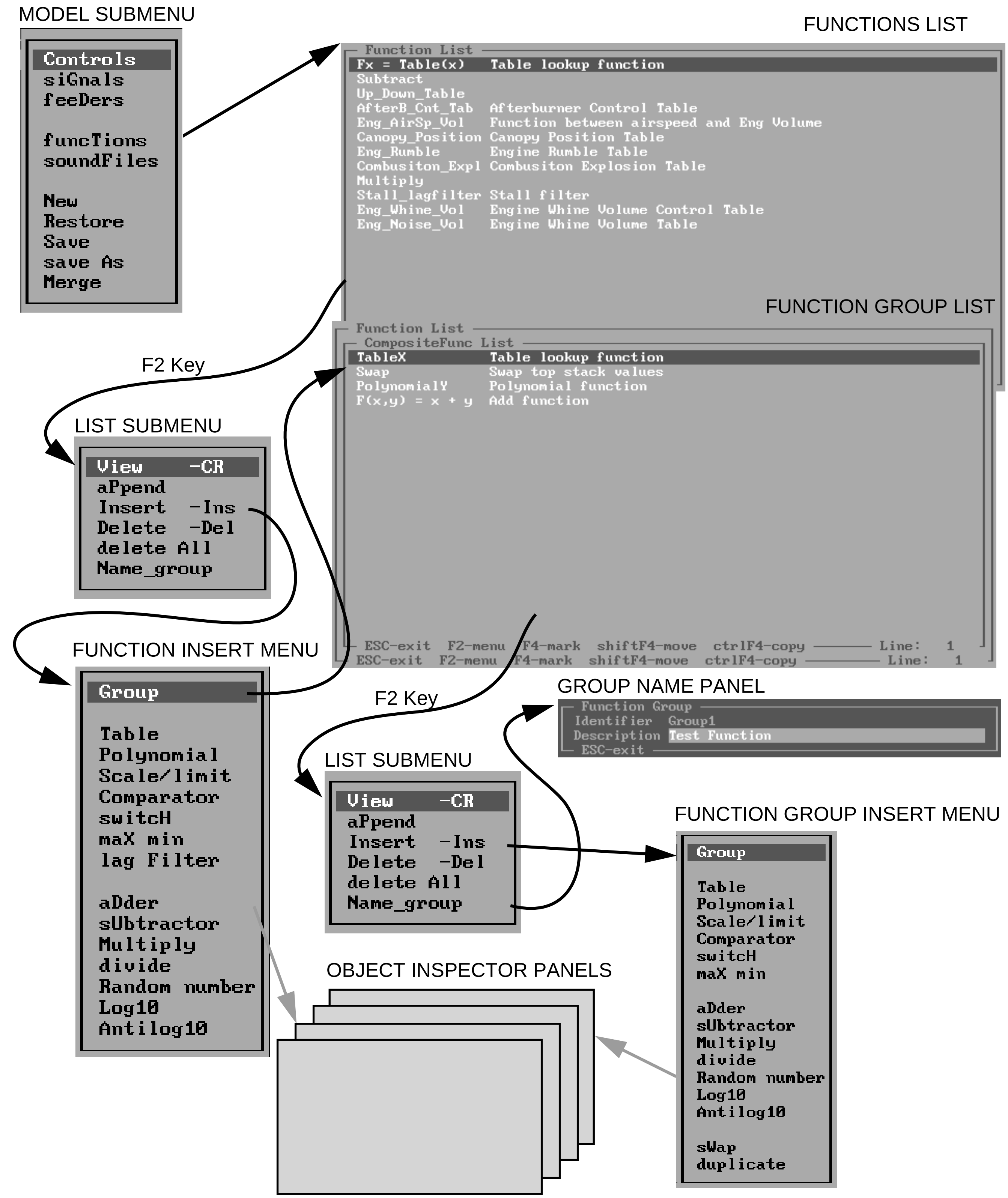
Figure 5 Feeders List Menus



Functions List Menus

The functions list is the set of math function objects in the model, and has the same keystroke access as the control list.

Figure 6 Functions List Menus



Sound Files List Menus

The sound files list consists of all the pre-recorded sounds which are available to be replayed from the hard disk. The list contains individual files or groups of files. Grouping allows the host to dynamically select files from within a predefined subset, using an integer file identifier.

The active keys are the same as the controls list, with the F2 bringing up the soundfile submenu, as shown in the diagram below.

The inspector panel for each sound file allows for modification of:

- Sound file name
- File index (used as part of file group access)
- Replay start position
- Replay finish position
- File Validity
- Flags to control playback formats, such as always playing to completion, looping, random start, and delay.

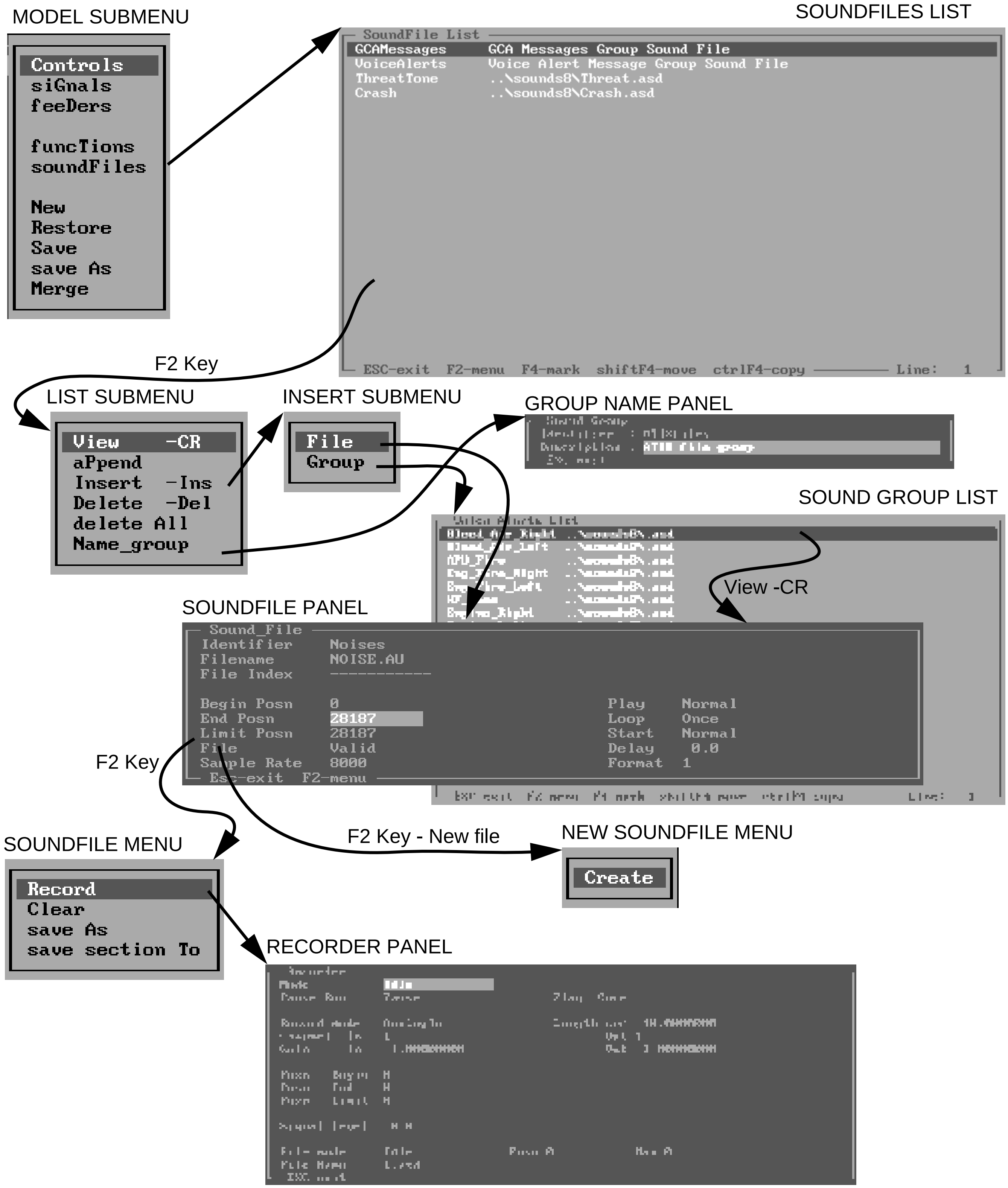
The F2 key when used with the inspector panel allows access to the sound file submenu which:

- provides an “off-line” recording and editing capability
- can save a file
- allows trimming of a file to exact length
- updates limits in file
- can create a new file or clear an existing one.
- can save a section of the sound file to another file

Note: if the selected filename does not exist then only the create option is available.

For more information on recording and playing sounds, see the Sound Files, Record/Replay signals, and Record/Replay Feeders sections.

Figure 7 Sound Files List Menus



Host Interface Menus

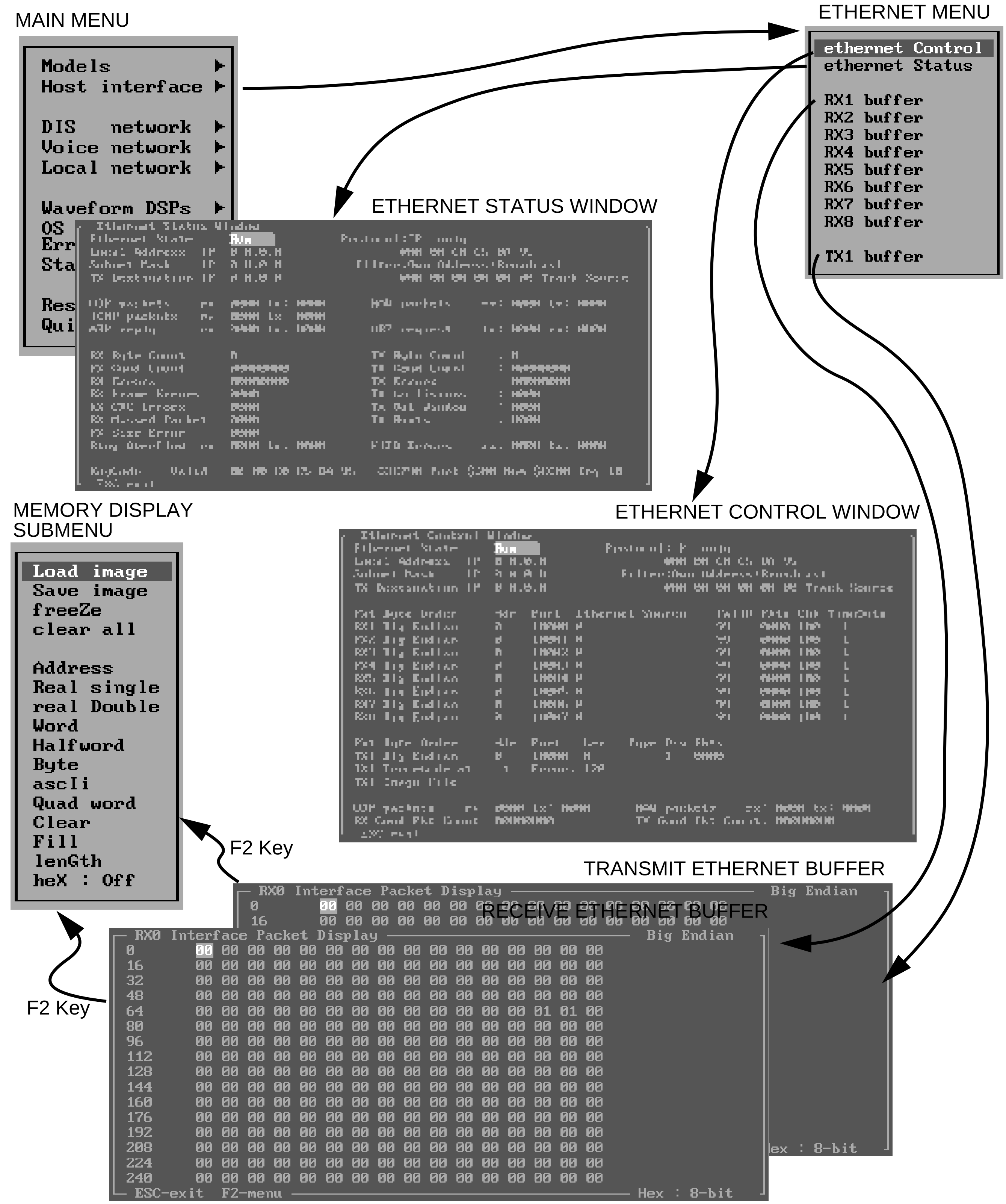
The host interface menu set allows user access to the ethernet data and interface status. For a complete description of the ethernet control and status windows, see the Ethernet Configuration Commands section.

Support is provided for

- Display of current receive data buffer for up to eight ports
- Display of current transmission buffer
- Modification of data in either buffer
- Buffer display formats including floating point, integer (32, 16 and 8 bit), and ASCII string.
- Support for different byte orders, i.e. Little Endian and Big Endian.
- Ability to save a buffer image to a file for latter recall for test purposes.
- Display of current raw and UDP level interface status.

For more information on the ethernet status and control windows, and on host interfacing, see the section entitled “Ethernet Configuration Commands”.

Figure 8   Host interface Menus



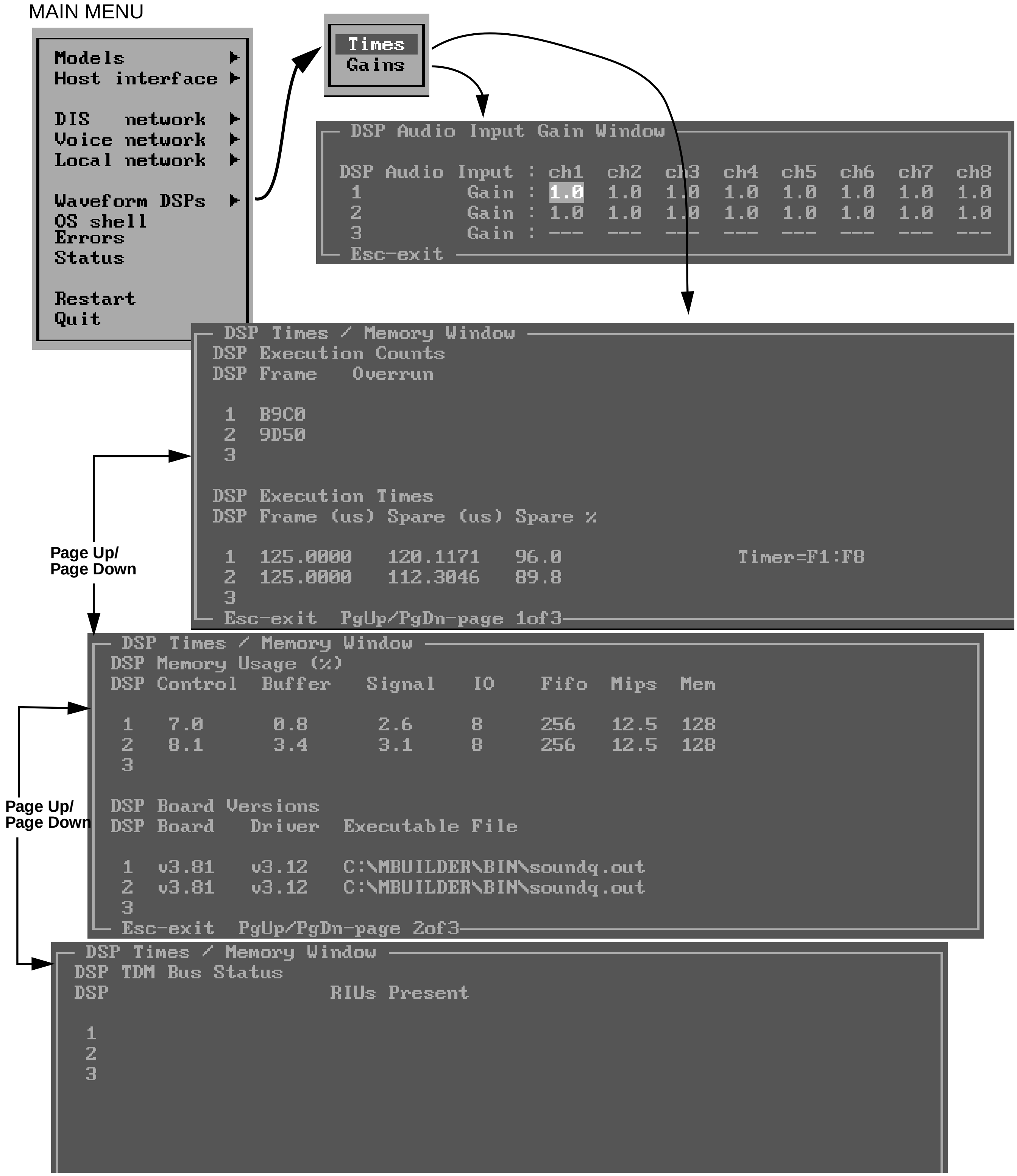
Digital Signal Processor Menu

The digital signal processor (DSP) menu provides access to four pages of information relevant to the current running state of the DSP waveform synthesizers.

Support is provided for

- Display of frame rates and spare time availability
- Display of internal status counters
- Memory buffer usage
- Preamp gain settings
- RIUs present on a TDM ring

Figure 9 DSP Menus



The Gains page shows the Preamp gain settings for the 8 channel waveform synthesizers. If no numbers appear, it means that either the card is not present, or it is a TDM card.

The top of the DSP timing page indicates the amount of the DSP processor time that is being used. The time available is one over the sample rate. The bottom of the DSP timing page indicates the DSP frame information. It includes:

- The frame count,
- The frame overrun (which is the number of frames incorrectly processed to an overload of the DSP processor), and
- The Com OK (which increments when two DSP boards are linked and set up as a master and a slave), and the ComFail (which increments when two DSP boards are set up to communicate over global channels but fail to do so properly).

The second screen indicates the usage of the DSP memory. The memory is divided into

three sections (Control, Buffer, and Signal), each of which can go up to 100% memory usage before problems occur.

The third screen shows, for each TDM card, which RIU’s are present on the TDM ring. The RIU number will only appear if there is an input feeder for the RIU, and the RIU is actually attached to the ring and functioning properly.

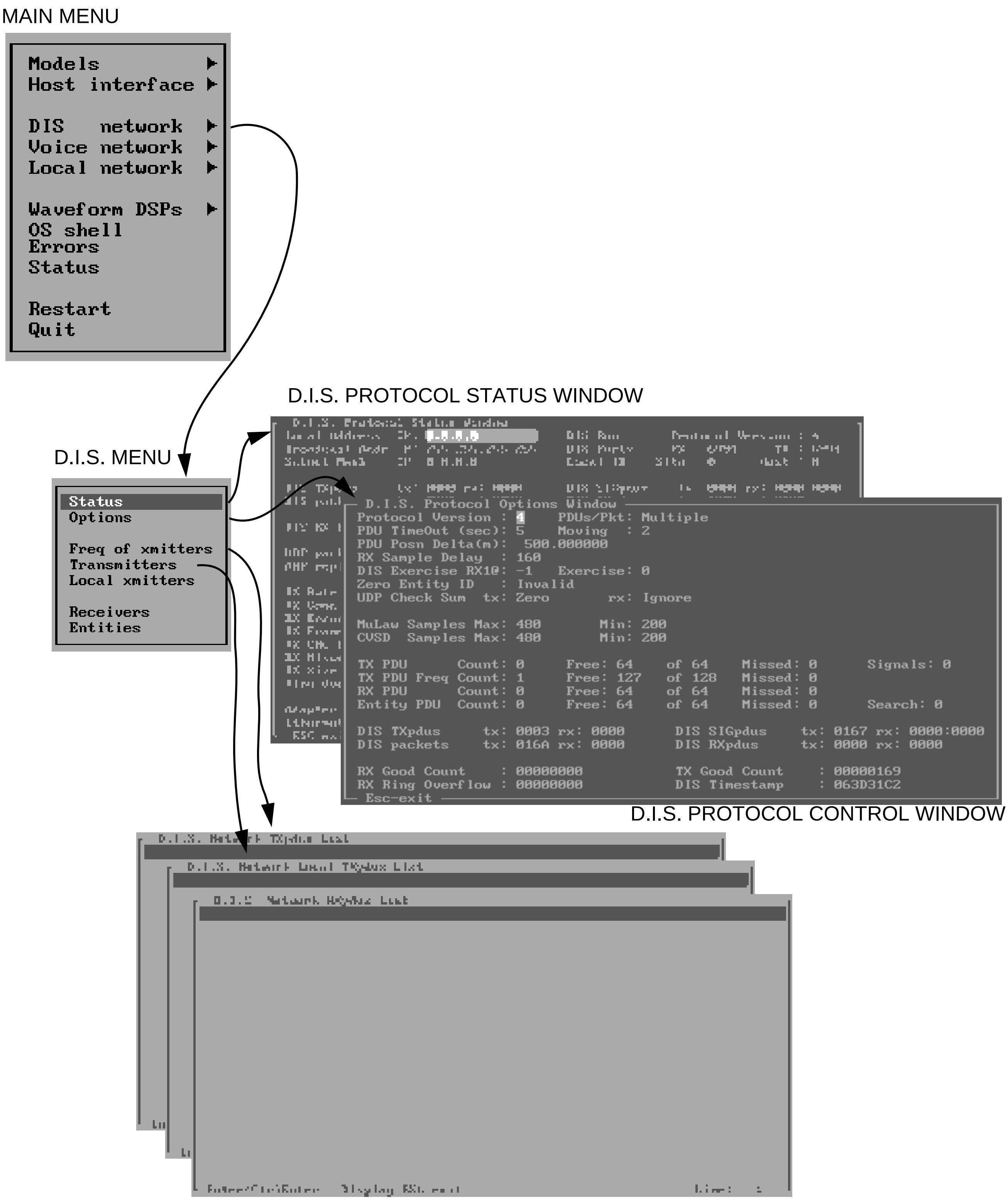
DIS Network Menus

The DIS (Distributed Interactive Simulation) Network menus provide control and monitoring of the DIS network attachment. For a more detailed description of the status and protocol windows, refer to the section entitled “DIS Configuration Commands”.

Support is provided for

- Status of D.I.S. protocols, addresses, and network
- Control of D.I.S. protocol options
- D.I.S. network activity, including simulated radio transmissions received over the D.I.S. network
- List of Entity PDU’s received over D.I.S. Network
- List of transmitters from the DACS that are transmitting over the D.I.S. network

Figure 10 DIS Network Menus



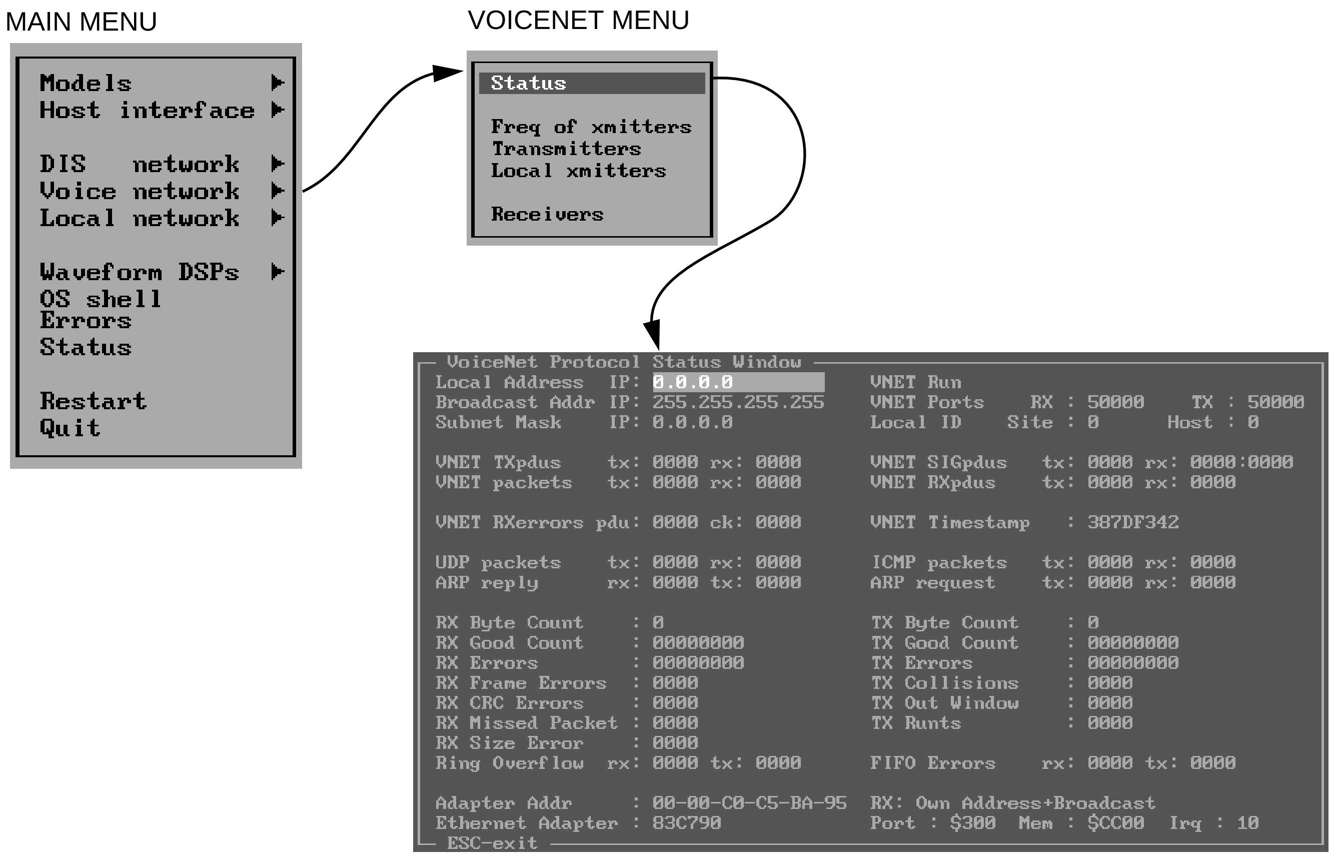
Voice Network Menus

The Voice Network menus provide control and monitoring of the Voice Network, a proprietary ASTi network that allows DACS to communicate with each other over an Ethernet link. For a more detailed description of the status window, refer to the section entitled “VoiceNet Configuration Commands”.

The Voice Network Menus provide

- Status of the network
- Control over local Voice Network address List of objects transmitter
- List of objects transmitting and receiving over the Voice Network

Figure 11 VoiceNet Menu



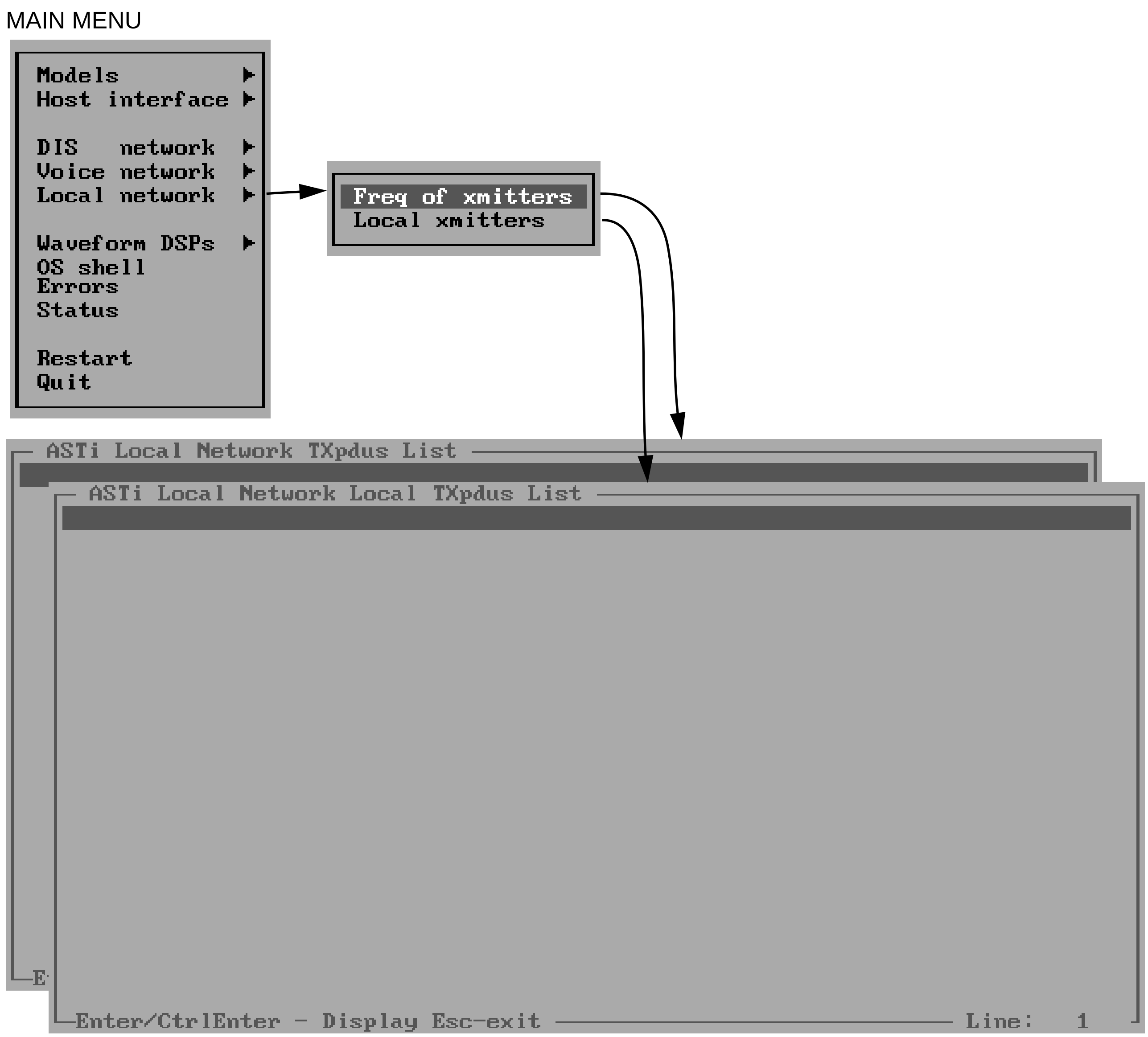
Local Network Menus

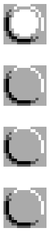
The Local Network menus provide monitoring of radio traffic on the Local Network, which is used to connect DSP cards within a single DACS.

The Local Network Menu provides access to lists of the objects transmitting over the Local Network, along with their frequencies, entity ID’s, and radio ID’s.

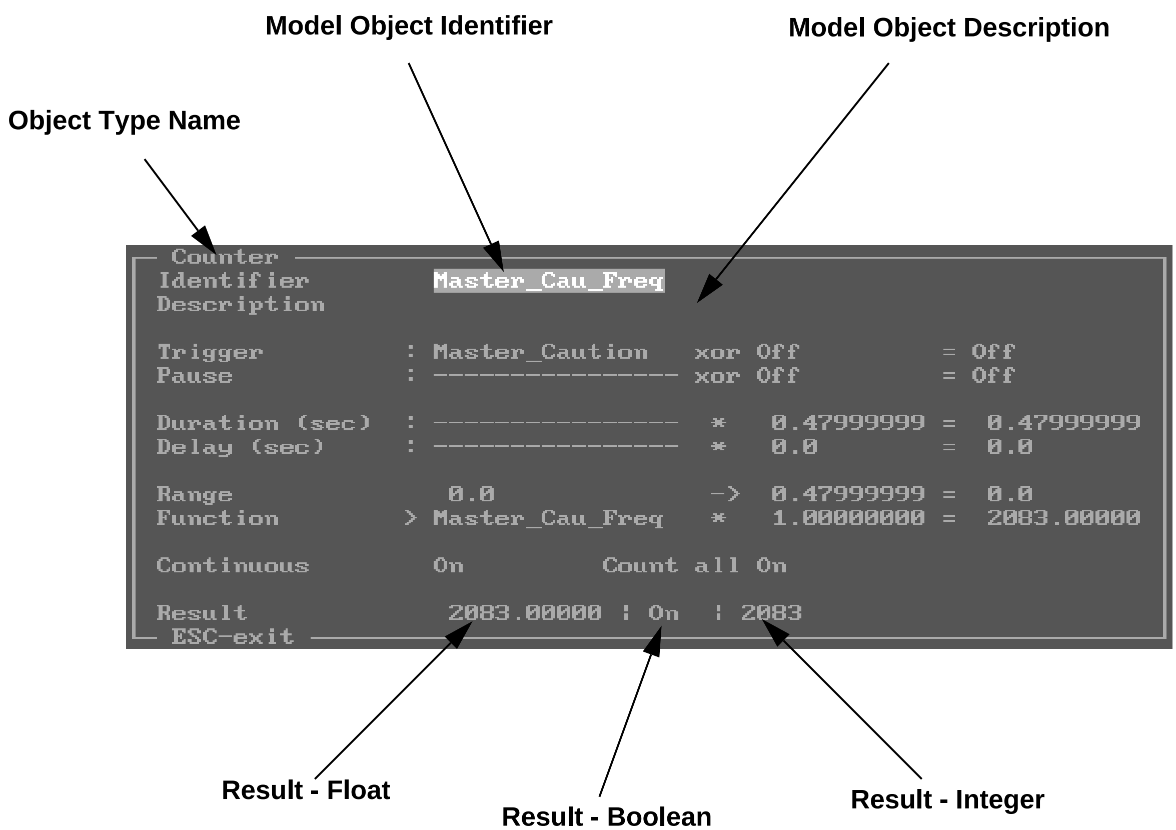
For more information on using the Local Network, see the section entitled “Global Channels”.

Figure 12 Local Network Menu





Object Inspector Panels



Inspector Panel

Identifier

Used to give each object a unique name.

Description

Description field to clarify display.

Object Type Name

Indication of the type of object currently being inspected.

Result

All Control objects have an output result which is picked-up by any other model object that is connected to it.

Result - Float

Output result from control object - a floating point value.

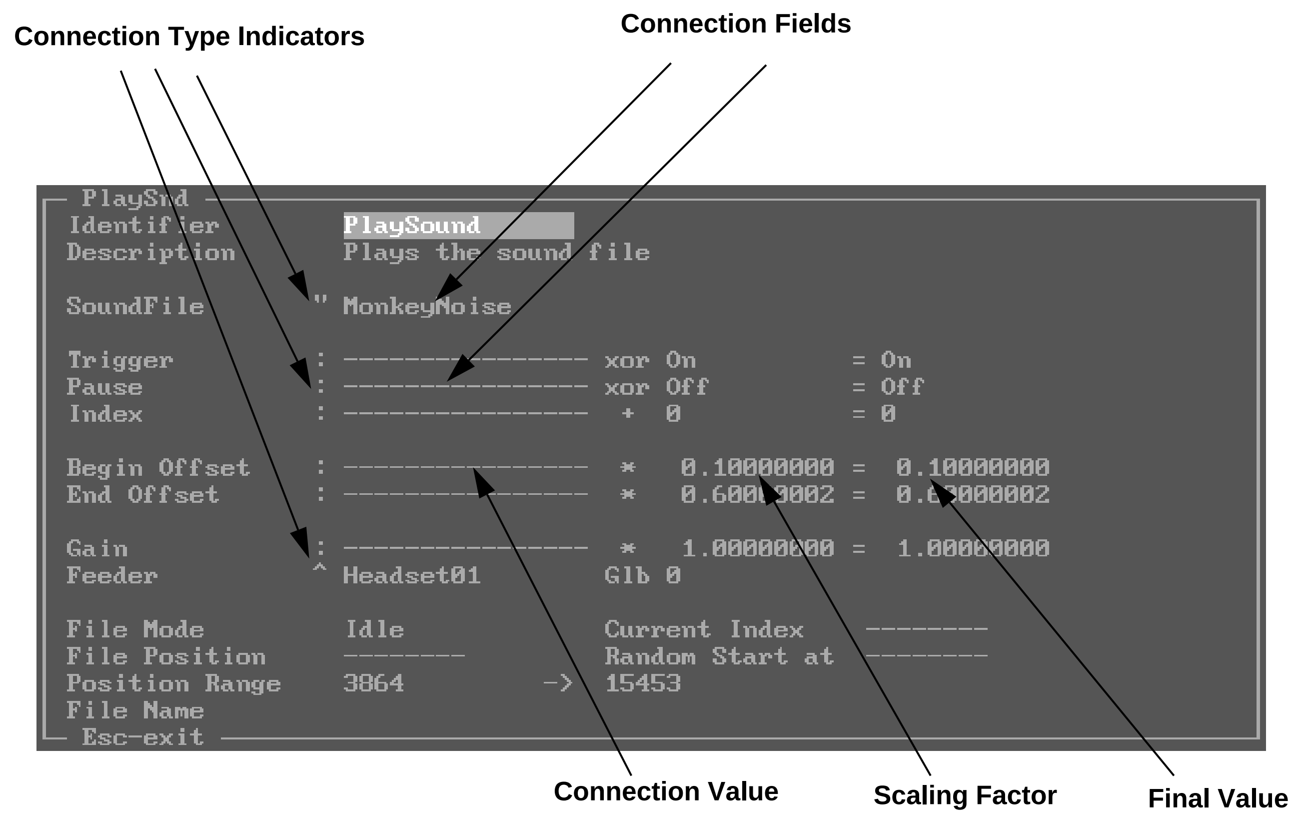
Result - Integer

Output result from control object - Integer (rounded value of floating point result)

Result - Boolean

Output result from control object - Boolean (Binary value of floating point result, with < 0.3 being Off, > 0.7 being On).

Model Connections



Panels within Model Builder have the same basic construction.

Each connection field is preceded by a character to indicate the type of object targeted by the connection.

Colons (:) indicate the field connection is to a control object (e.g. Host input, counter, or Math Function).

Carat (^) indicate the signal output can be connected to a feeder.

Arrow (>) indicates the selection of a math function from the user provided list (Table lookup, add, subtract etc.).

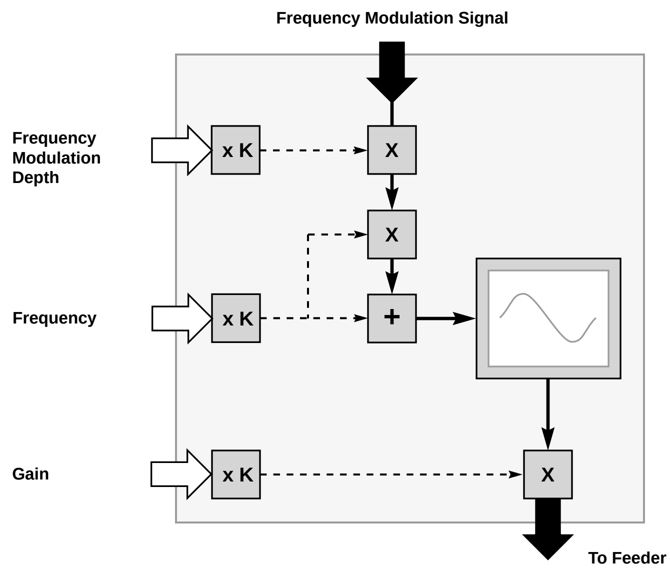
Double Quotes (“) indicate the connection to a soundfile from the list., which can be either individual files or groups of files.

Where a connection is made to a control object, as in the Gain field above, the result value is obtained from the target control, multiplied by the scaling factor (user modifiable) to create the displayed final value.

Signals

Basic Signals

Sine Wave Signal

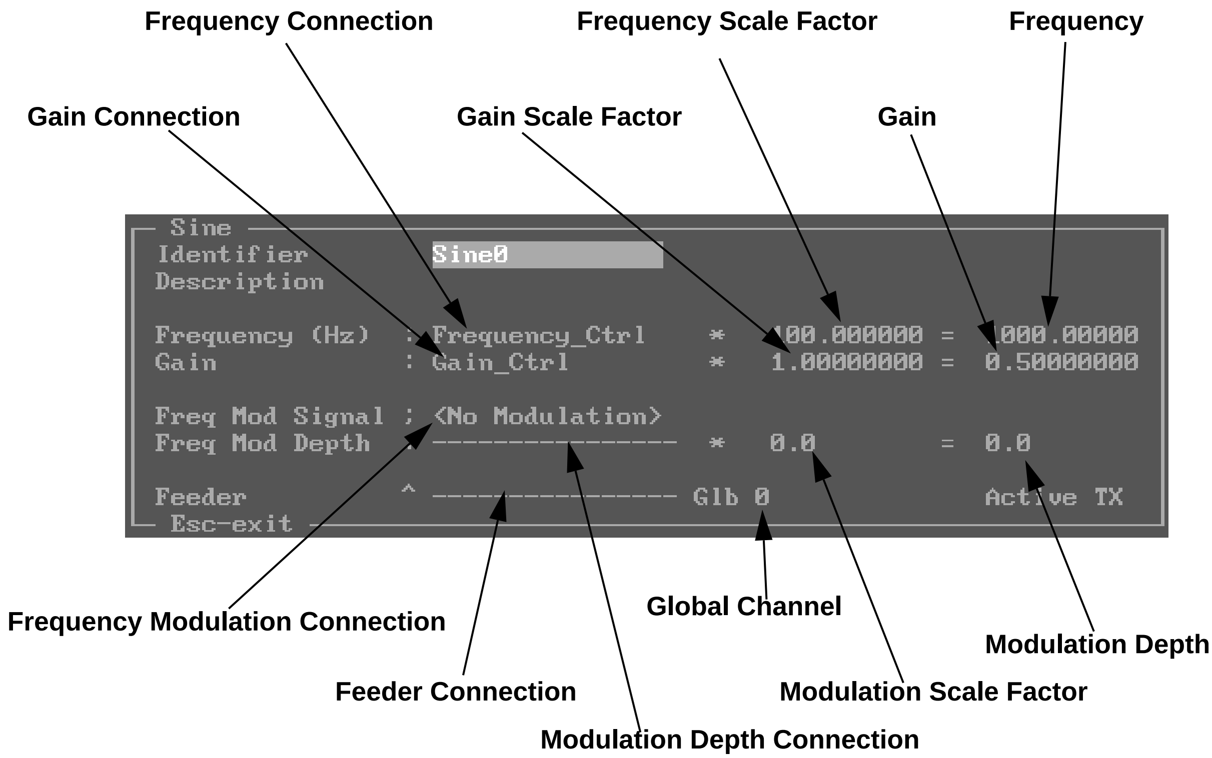


ACTUAL FREQUENCY = FREQUENCY (1 + (FREQ MOD DEPTH x FREQ MOD SIGNAL))

Sine Wave Signal

This signal source produces a sinusoidal wave signal which can be mixed in any proportion on any of the analog output channels. Both amplitude and frequency can be controlled by input variables from elsewhere in the model, or from the host interface. The frequency can also be modulated by another signal within the signal processor, with the model having control over the depth of modulation. This is shown diagrammatically above.

Sine Wave Signal



Frequency Connection

Control object connection to provide overall frequency control from elsewhere in model.

Frequency Scale Factor

Scaling factor for frequency control value.

Frequency

Frequency (in Hertz) of sine wave generated by waveform synthesizer. If the frequency connection is blank then the frequency scale factor is used as the frequency value; otherwise the frequency is the scale factor times the output result of the control object.

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of sine wave. If the gain connection is blank then the gain scale factor is used as the gain value; otherwise the gain is the scale factor times the output result of the control object.

Frequency Modulation Connection

Connection to the frequency modulating signal, which is scaled by the modulation depth and added to the frequency.

Sine Wave Signal

Modulation Depth Connection

Control object connection to provide frequency modulation depth control from elsewhere in model.

Modulation Scale Factor

Scaling factor for frequency modulation depth control value.

Modulation Depth

Frequency modulation depth value, controls the effect of the frequency modulation signal:

$$ActualFreq = Freq \times (1 + (ModDepth \times ModSignal))$$

Usually it falls in the range 0 to 1.0, when used in conjunction with a unity gain modulation signal. Note: To avoid unpredictable behavior care should be taken to ensure that the product of modulation depth and modulation signal does not span a range greater than -1.0 to +1.0.

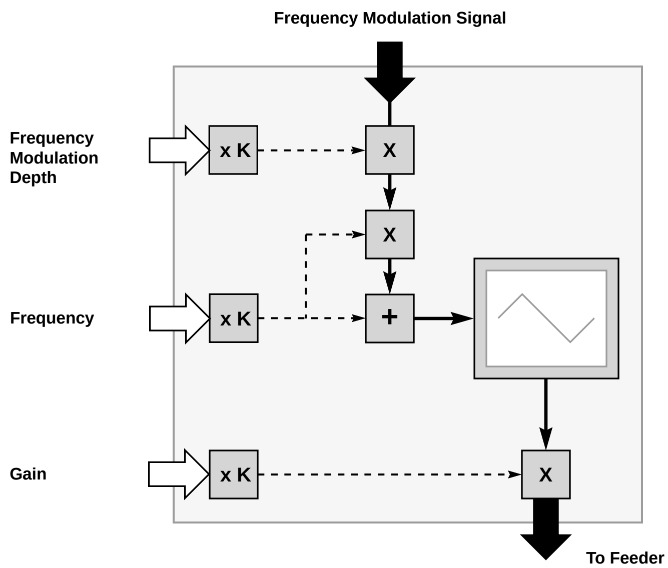
Feeder Connection

Connection to a feeder, which adds the sine wave into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Triangle Wave



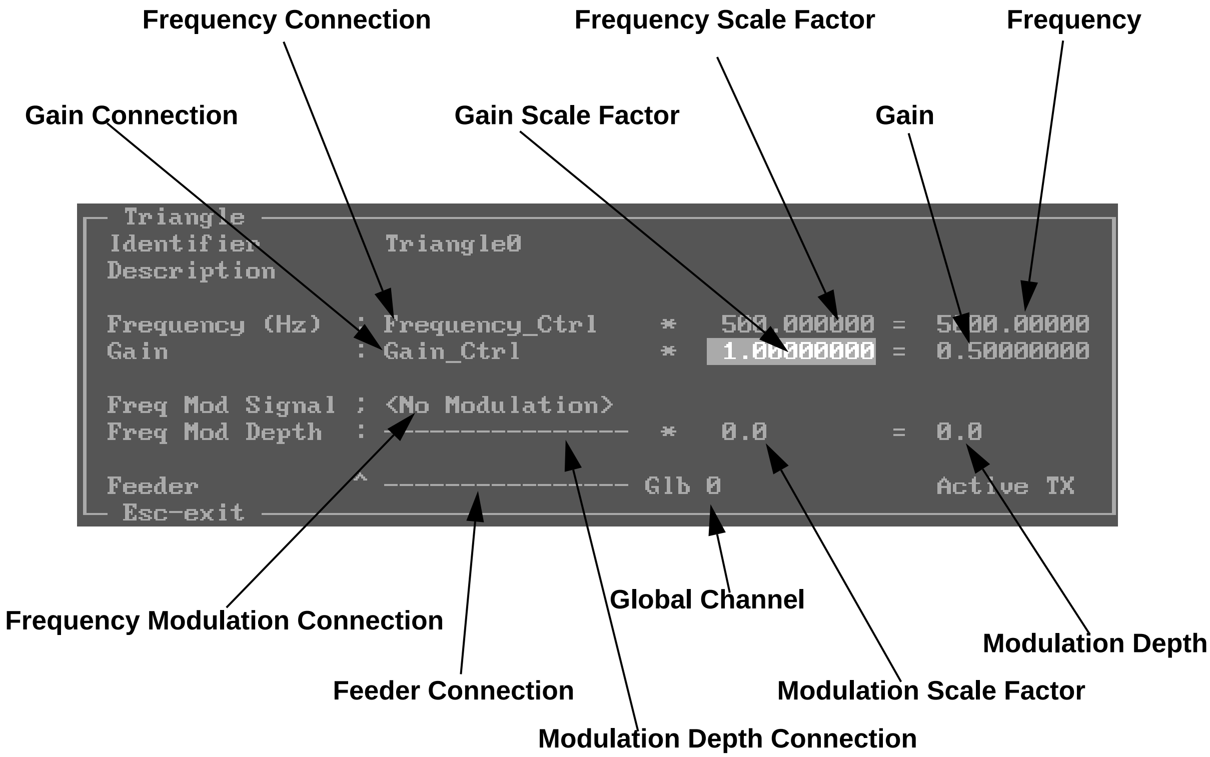
ACTUAL FREQUENCY = FREQUENCY (1 + (FREQ MOD DEPTH x FREQ MOD SIGNAL))

Triangle Wave Signal

This signal source produces a triangle wave signal which can be mixed in any proportion on any of the analog output channels. Both amplitude and frequency can be controlled by input variables from elsewhere in the model, or from the host interface. The frequency can also be modulated by another signal within the signal processor, with the model having control over the depth of modulation.

The triangle wave is identical in operation to the sine wave

Triangle Wave



Frequency Connection

Control object connection to provide overall frequency control from elsewhere in model.

Frequency Scale Factor

Scaling factor for frequency control value.

Frequency

Frequency (in Hertz) of triangle wave generated by waveform synthesizer. If the frequency connection is blank then the frequency scale factor is used as the frequency value; otherwise the frequency is the scale factor times the output result of the control object.

Gain Connection

Control object connection provides amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of triangle wave. If the gain connection is blank then the gain scale factor is used as the gain value; otherwise the gain is the scale factor times the output result of the control object.

Frequency Modulation Connection

Connection to the frequency modulating signal, which is scaled by the modulation depth and added to the frequency.

Triangle Wave

Modulation Depth Connection

Control object connection to provide frequency modulation depth control from elsewhere in model.

Modulation Scale Factor

Scaling factor for frequency modulation depth control value.

Modulation Depth

Frequency modulation depth value, controls the effect of the frequency modulation signal:

$$ActualFreq = Freq \times (1 + (ModDepth \times ModSignal))$$

Usually it falls in the range 0 to 1.0, when used in conjunction with a unity gain modulation signal. Note: To avoid unpredictable behavior care should be taken to ensure that the product of modulation depth and modulation signal does not span a range greater than -1.0 to +1.0.

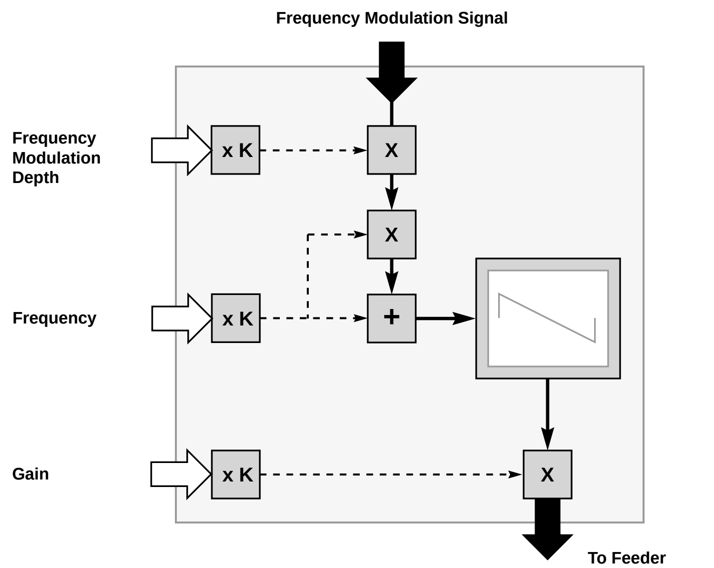
Feeder Connection

Connection to a feeder, which adds the triangle wave into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Sawtooth Wave



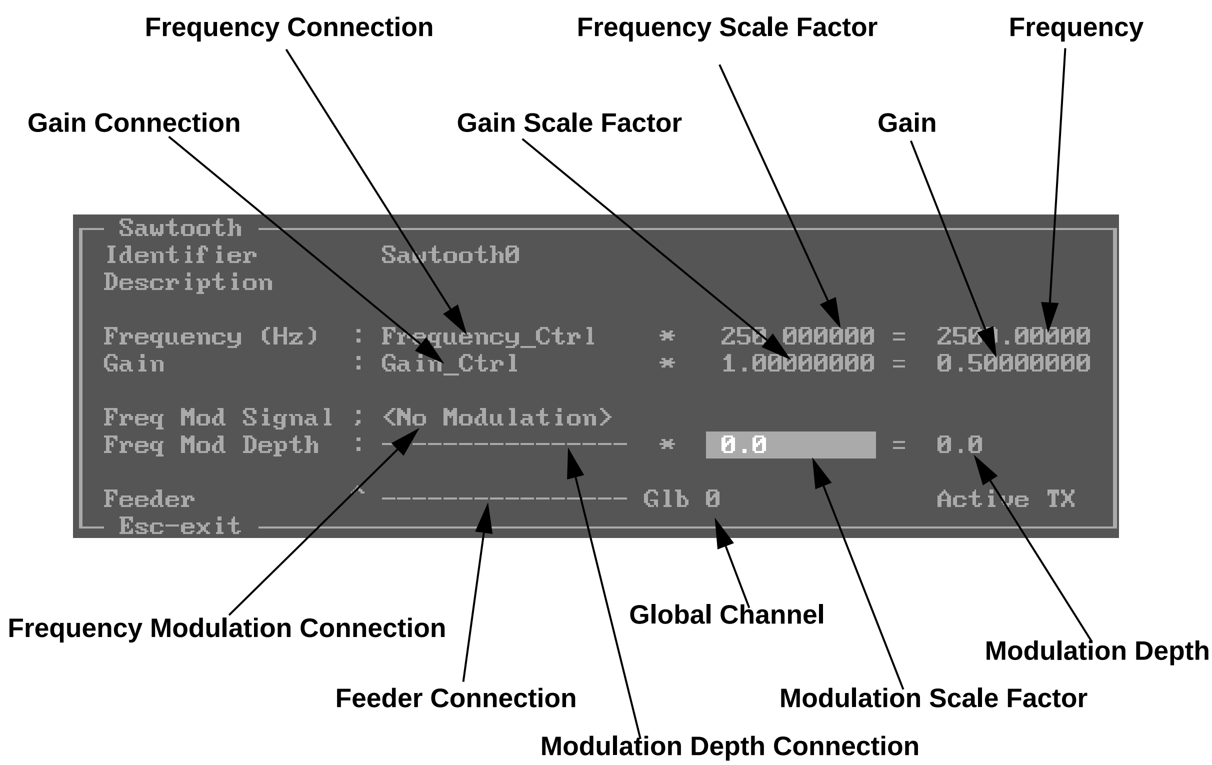
ACTUAL FREQUENCY = FREQUENCY (1 + (FREQ MOD DEPTH x FREQ MOD SIGNAL))

Sawtooth Wave Signal

This signal source produces a sawtooth wave signal which can be mixed in any proportion on any of the analog output channels. Both amplitude and frequency can be controlled by input variables from elsewhere in the model, or from the host interface. The frequency can also be modulated by another signal within the signal processor, with the model having control over the depth of modulation.

The sawtooth wave is identical in operation to the sine wave

Sawtooth Wave



Frequency Connection

Control object connection to provide overall frequency control from elsewhere in model.

Frequency Scale Factor

Scaling factor for frequency control value.

Frequency

Frequency (in Hertz) of sawtooth wave generated by waveform synthesizer. If the frequency connection is blank then the frequency scale factor is used as the frequency value; otherwise the frequency is the scale factor times the output result of the control object.

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of sawtooth wave. If the gain connection is blank then the gain scale factor is used as the gain value; otherwise the gain is the scale factor times the output result of the control object.

Frequency Modulation Connection

Connection to the frequency modulating signal, which is scaled by the modulation depth and added to the frequency.

Sawtooth Wave

Modulation Depth Connection

Control object connection to provide frequency modulation depth control from elsewhere in model.

Modulation Scale Factor

Scaling factor for frequency modulation depth control value.

Modulation Depth

Frequency modulation depth value, controls the effect of the frequency modulation signal:

$$ActualFreq = Freq \times (1 + (ModDepth \times ModSignal))$$

Usually it falls in the range 0 to 1.0, when used in conjunction with a unity gain modulation signal. Note: To avoid unpredictable behavior care should be taken to ensure that the product of modulation depth and modulation signal does not span a range greater than -1.0 to +1.0.

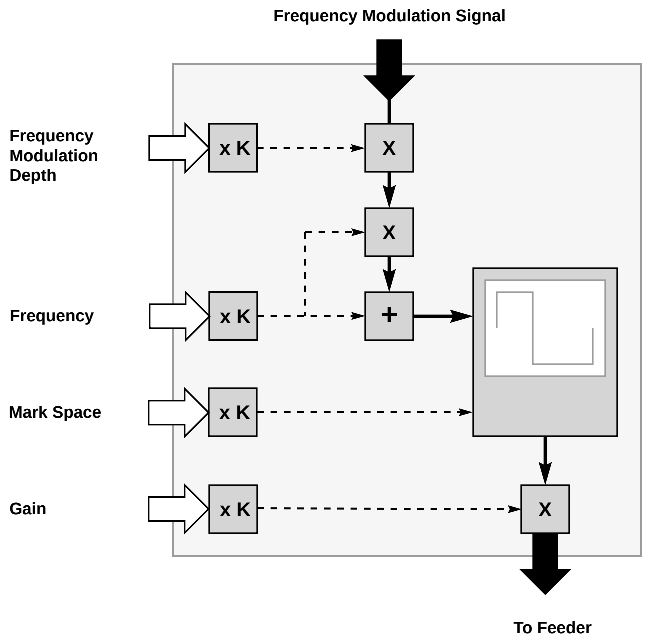
Feeder Connection

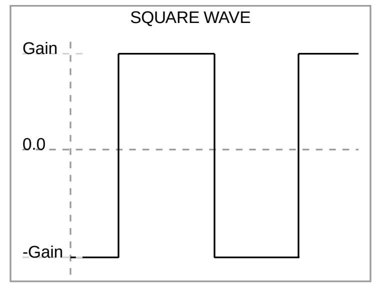
Connection to a feeder, which adds the sawtooth wave into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Square Wave



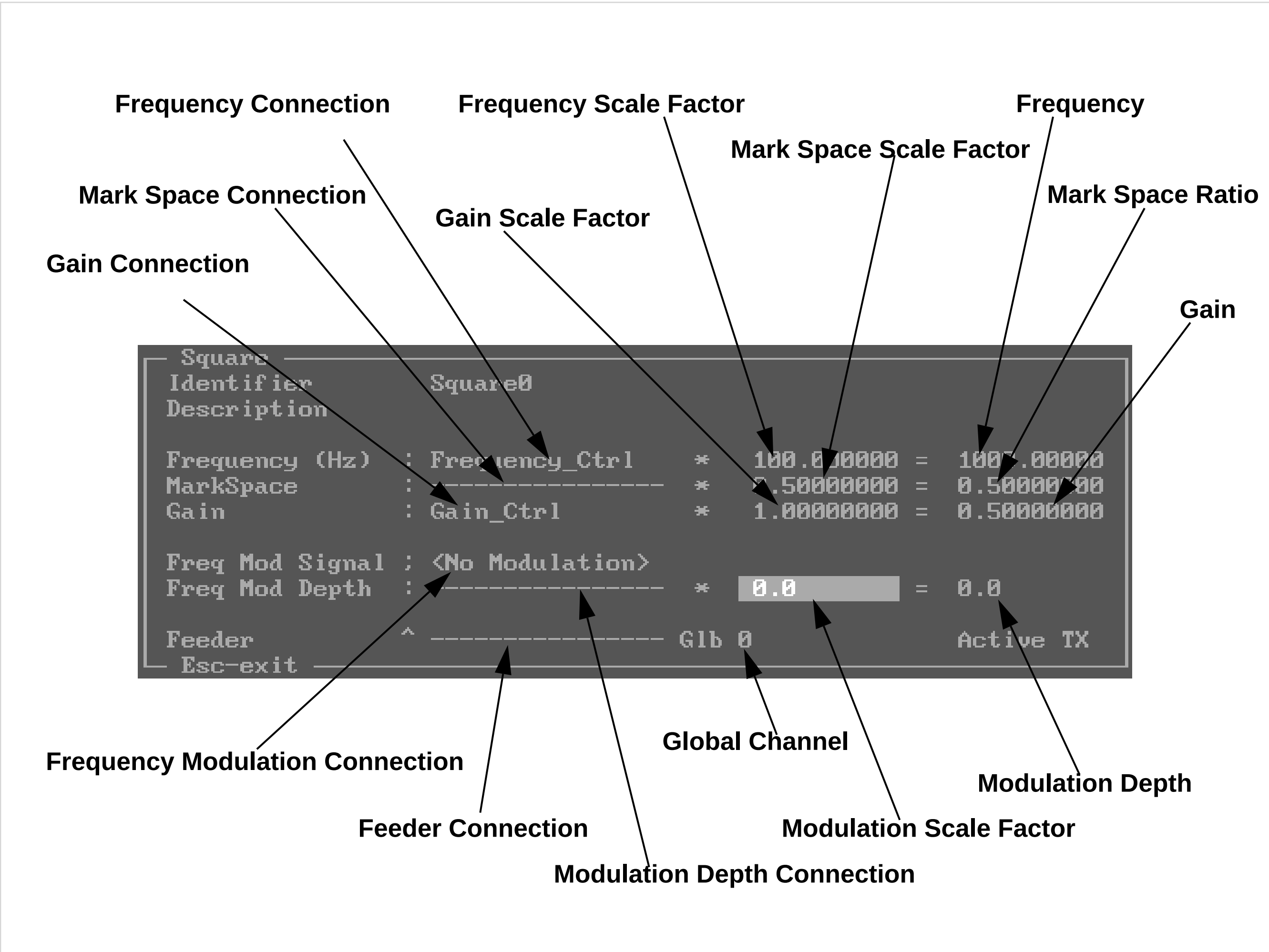


ACTUAL FREQUENCY = FREQUENCY (1 + (FREQ MOD DEPTH x FREQ MOD SIGNAL))

Square Wave Signal

This signal source produces a square wave signal which can be mixed in any proportion on any of the analog output channels. Both amplitude, frequency and mark/space ratio (duty cycle) can be controlled by input variables from elsewhere in the model, or from the host interface. The frequency can also be modulated by another signal within the signal processor, with the model having control over the depth of modulation.

Square Wave



Frequency Connection

Control object connection to provide overall frequency control from elsewhere in model.

Frequency Scale Factor

Scaling factor for frequency control value.

Frequency

Frequency (in Hertz) of square wave generated by waveform synthesizer. If the frequency connection is blank then the frequency scale factor is used as the frequency value; otherwise the frequency is the scale factor times the output result of the control object.

Mark Space Connection

Control object connection to provide mark space control from elsewhere in model.

Mark Space Scale Factor

Scaling factor for mark space ratio control value.

Mark Space Ratio

Mark to space ratio for square wave. Value ranges from 0 to 1.0 with 0 being all space, 1.0 being all mark. If the M/S connection is blank then the M/S scale factor is used as the value; otherwise the M/S is the scale factor times the output result of the control object.

Square Wave

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of square wave. If the gain connection is blank then the gain scale factor is used as the gain value; otherwise the gain is the scale factor times the output result of the control object.

Frequency Modulation Connection

Connection to the frequency modulating signal, which is scaled by the modulation depth and added to the frequency.

Modulation Depth Connection

Control object connection to provide frequency modulation depth control from elsewhere in model.

Modulation Scale Factor

Scaling factor for frequency modulation depth control value.

Modulation Depth

Frequency modulation depth value, controls the effect of the frequency modulation signal:

$$ActualFreq = Freq \times (1 + (ModDepth \times ModSignal))$$

Usually it falls in the range 0 to 1.0, when used in conjunction with a unity gain modulation signal. Note: To avoid unpredictable behavior care should be taken to ensure that the product of modulation depth and modulation signal does not span a range greater than -1.0 to +1.0.

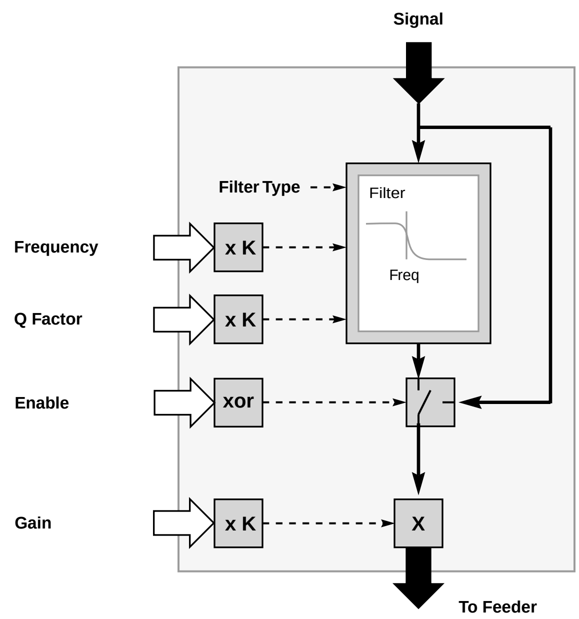
Feeder Connection

Connection to a feeder, which adds the square wave into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

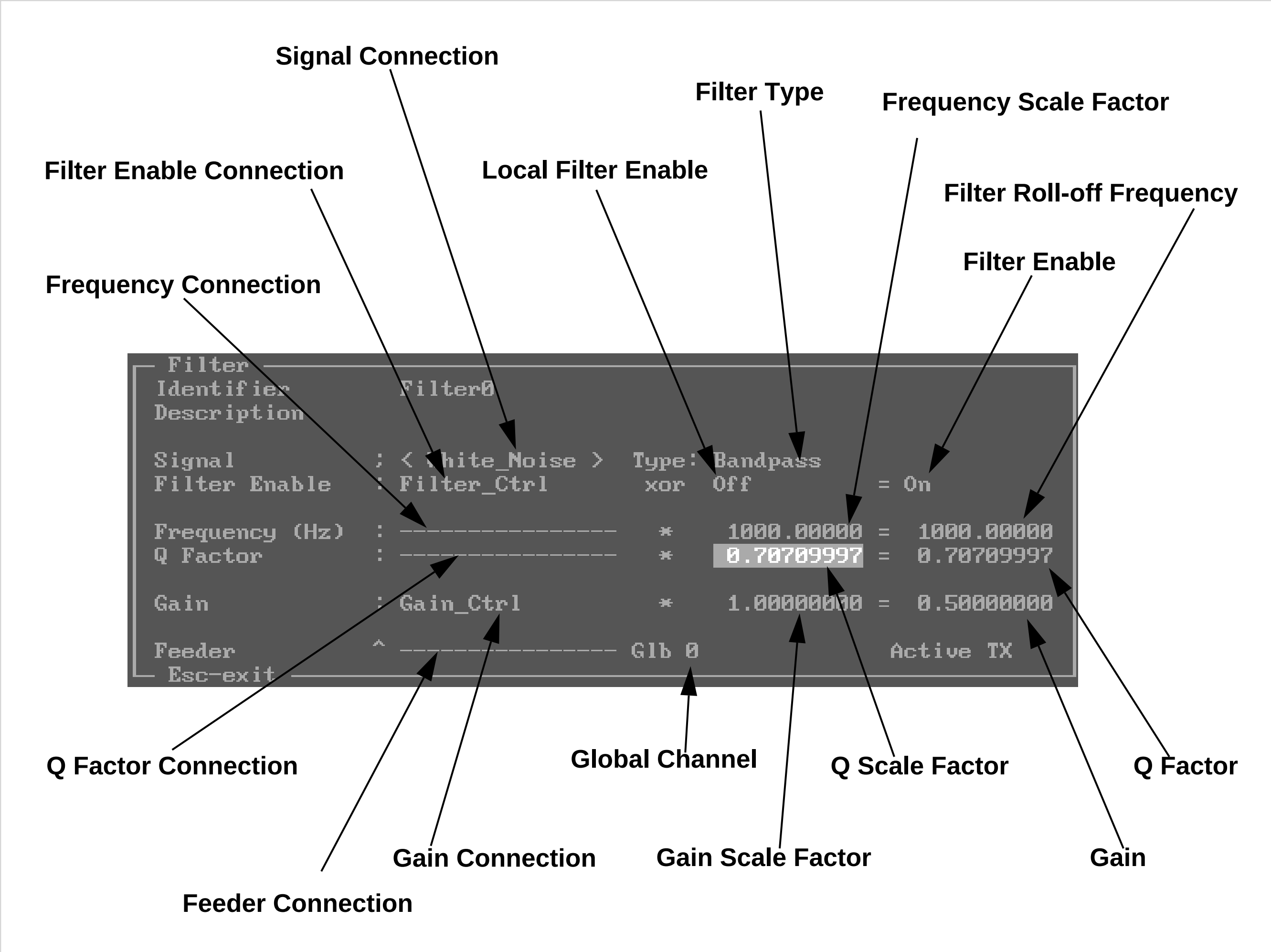
Filter



Filter Signal

This signal source produces a filtered copy of the selected signal which can be mixed in any proportion on any of the analog output channels. The type of filtering can be selected from low-pass, band-pass or high-pass. The filter quality factor, roll-off frequency, and gain can be controlled by input variables from elsewhere in the model, or from the host interface. The default signal is the internal pseudo random noise source, providing an improved noise source with better tunability.

Filter



Signal Connection

Connection to the signal to be used as input to the filter. Deleting this connection selects the default internal white noise generator.

Filter Type

Selects a two pole filter type from; Low-pass, Band-pass, High-pass, Low-passQ, Band-passQ, High-passQ, Notch or AllPass. The three Q filters are amplitude adjusted such that the filter has unity gain at the roll-off frequency, and maintains this gain as the quality factor is increased. The band-pass filters have the low-pass and high-pass poles at the same roll-off frequency.

Filter Enable Connection

Control object connection to provide filter enable control from elsewhere in model.

Local Filter Enable

Local value for the filter enable. If an enable connection is made then this provides an exclusive or function for inverting the enable control state

Filter Enable

The current filter enable state, when On the filter will be active, when Off the signal will be passed through with gain control only.

Frequency Connection

Control object connection to provide overall frequency control from elsewhere in model.

Filter

Frequency Scale Factor

Scaling factor for frequency control value.

Frequency

Roll-off frequency (in Hertz) of filter. If the frequency connection is blank then the frequency scale factor is used as the frequency value; otherwise the frequency is the scale factor times the output result of the control object.

Q Factor Connection

Control object connection to provide quality factor control from elsewhere in model.

Q Scale Factor

Scaling factor for Q factor control value.

Q Factor

Quality factor for filter. If the Q factor connection is blank then the Q scale factor is used as the Q value; otherwise the Q is the scale factor times the output result of the control object.

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of filtered source. If the gain connection is blank then the gain scale factor is used as the gain value; otherwise the gain is the scale factor times the output result of the control object.

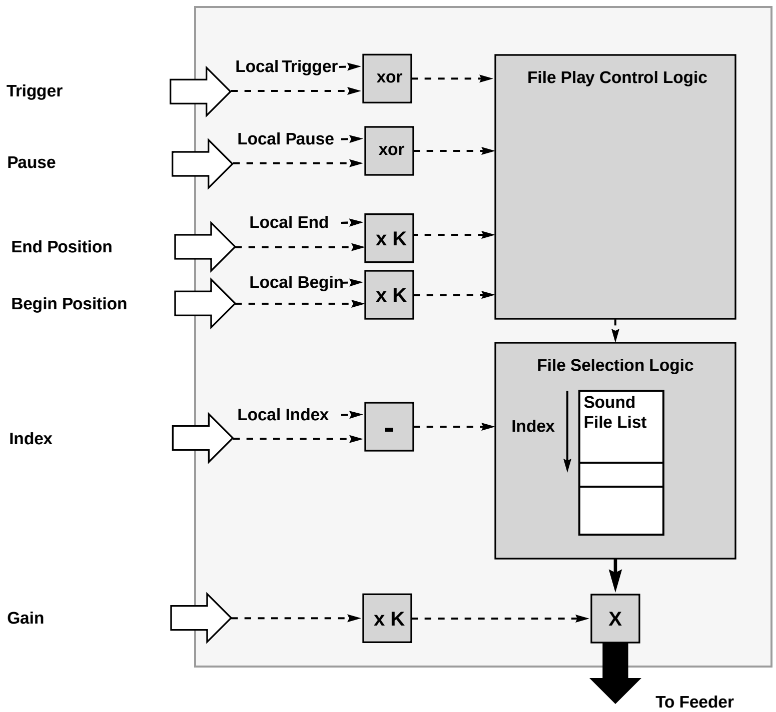
Feeder Connection

Connection to a feeder, which adds the filtered source into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Play Soundfile



Play Soundfile Signal

Sounds which have no dynamically varying elements (except for overall volume level) are best handled as fixed off-line recorded sound files. (E.g. Missile launch)

The output signal from a pre-recorded sound file can be mixed in any proportion on any output channel.

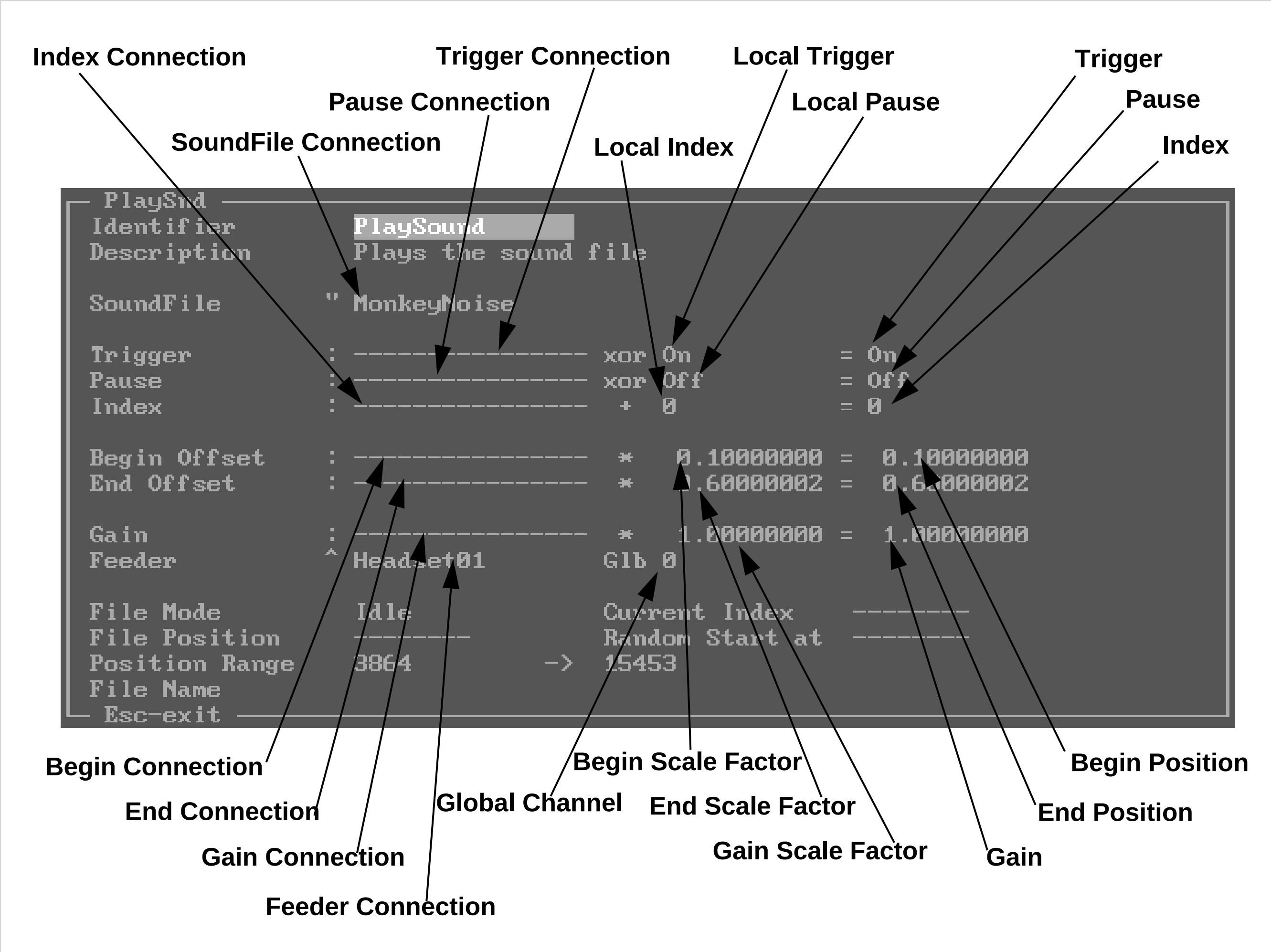
The file to be replayed is selected from the soundfile directory list. If the files are grouped together, then an integer file number can be used to select from a set of mutually exclusive files.

The trigger boolean, the file index number and the overall gain are controllable from elsewhere in the model or directly from the host input variables.

Local mode control is available to determine whether the file is played continuously or single shot, if there is a delay after completion of replay when in continuous mode, and whether the file plays to the end or can be truncated.

Local control of the trigger boolean allows the starting signal to be either an off to on transition, or an on to off.

Play Soundfile



SoundFile Connection

Sound file selection from model's soundfile directory list

Trigger Connection

Control object connection to start/stop control elsewhere in model.

Local Trigger

Local value for the trigger. If no connection is made this allows the trigger to be left permanently in the On position. If a trigger connection is made then this provides an exclusive or function for inverting the trigger control state

Trigger

The current trigger state, a value of On starts the soundfile playing. If in continuous mode the file replays while this trigger is On, else if in one-shot mode the file replays once for each Off to On transition of the trigger.

Pause Connection

Control object connection to provide pause control from elsewhere in model.

Play Soundfile

Local Pause

Local value for the pause. If no connection is made this allows the pause to be left permanently in the off position. If a pause connection is made then this provides an exclusive or function for inverting the pause control state

Pause

The current pause state, a value of on freezes the soundfile playing, a value of off allows the play to continue from the current file position

Index Connection

Control object connection to provide file selection control from elsewhere in model.

Local Index

Local value for the file index. If a connection is made then this value is subtracted from the incoming index value; otherwise it provides a default value for the file index.

Index

The current value of the file index. This index is used to select one of a group of files from the sound files list. It is only active when the soundfile connection is made to a file group, and the index matches that of one file in the list. If no matches are found then no file is replayed.

Begin Connection

Control object connection to provide begin position control from elsewhere in model.

Begin Scale Factor

Local value for the begin position or scaling factor for begin position connection

Begin Position

The current begin position offset as a fraction of the selected files total length, i.e. a value of 0.5 starts file half way into normal play section.

End Connection

Control object connection to provide end position control from elsewhere in model.

End Scale Factor

Local value for the end position or scaling factor for end position connection.

End Position

The current end position offset as a fraction of the selected files total length, i.e. a value of 0.1 stops the file at 90% of it’s normal end point

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of file replay source. If the gain connection is blank then the gain scale factor is used as the gain value; otherwise the gain is the scale factor times the output result of the control object.

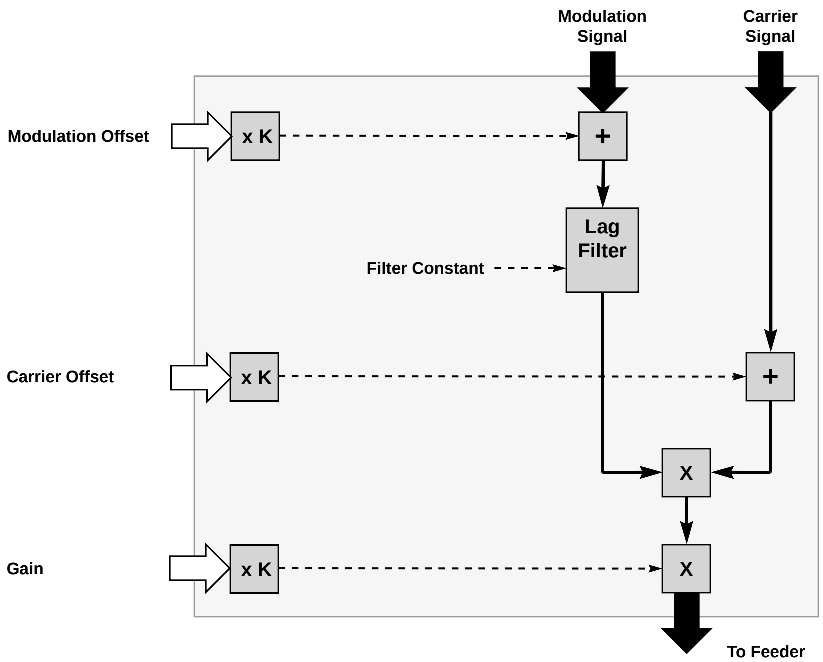
Feeder Connection

Connection to a feeder, which adds the replayed sound source into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Amplitude Modulator



Amplitude Modulator

The amplitude modulator provides a signal multiplication capability between two signals, a carrier waveform and a modulating envelope. This is useful for general warning tones (e.g. Radar Warning Receivers). Complex warning tones can be generated when the amplitude modulator is used with one of the pulse signals described later.

The modulation signal can be offset from zero to allow for control of the modulation depth. A lag filter is also provided to soften the edges which occur when square wave modulating a sine wave. The filter constant determines the effective slew rate of the modulating signal.

The filter function is defined as:

$Y_N = Y_{N-1} + K(X_N - Y_{N-1})$

Where

X_N = new input value

Y_N = new output value

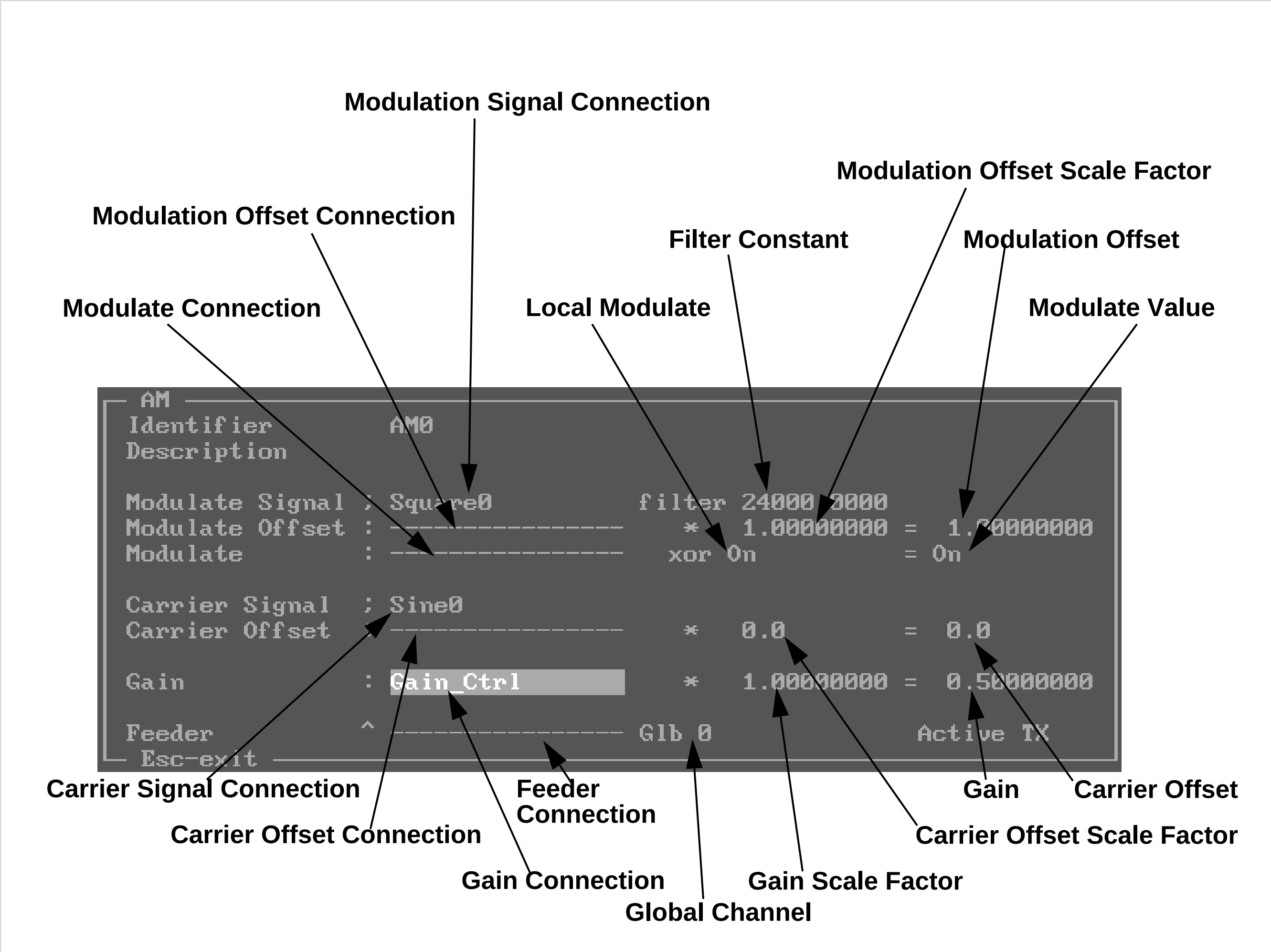
Y_{N-1} = last frame's output value

K = filter constant

where:

K = pi * FilterFreq / SampleRate

Amplitude Modulator



Modulation Signal Connection

Connection to the modulating signal.

Filter Constant

Value for modulation signal lag filter, roll-off frequency (in Hertz).

Modulation Offset Connection

Connection to a control for the offset to be added to the modulation signal.

Modulation Offset Scale Factor

Scaling factor for modulation offset.

Modulation Offset

Value added to modulation signal prior to multiplication by carrier signal. If the offset connection is blank then the offset scale factor is used; otherwise the offset is the scale factor times the output result of the control object.

The modulation offset should be 1.0 to provide a full depth of modulation from a square or sinusoidal source. This assumes the gain of the originating signal is set to 1.0, in which case it will swing between -1.0 and 1.0, hence the need for a 1.0 offset. If a pulse stream is used then this offset should be set to 0.0 for an on/off modulation of the carrier.

Modulate Connection

Connection to a control for the modulation state.

Local Modulate

Local state for modulation.

Amplitude Modulator

Modulate Value

Modulate control, when On carrier is modulated; otherwise, the carrier passes through with no modulation. If the modulate connection is used then the modulate value is the exclusive-or of the connected control value and the local modulate flag.

Carrier Signal Connection

Connection to the carrier signal to be used by the modulator.

Carrier Offset Connection

Connection to a control for the carrier offset to be added to the carrier signal.

Carrier Offset Scale Factor

Scaling factor for carrier offset.

Carrier Offset

Value added to carrier signal prior to multiplication by modulation signal. If the offset connection is blank then the offset scale factor is used as the value; otherwise the offset is the scale factor times the output result of the control object.

The carrier offset can be used to effectively mix a proportion of the modulation signal with the modulated carrier.

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of amplitude modulated source. If the gain connection is blank then the gain scale factor is used as the gain value; otherwise the gain is the scale factor times the output result of the control object.

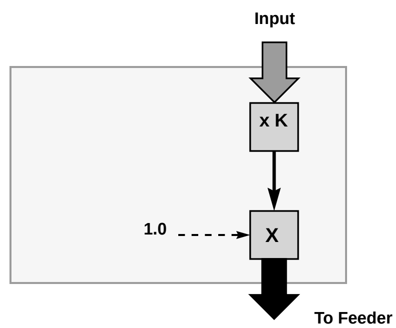
Feeder Connection

Connection to a feeder, which adds the modulated wave into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Analog Input

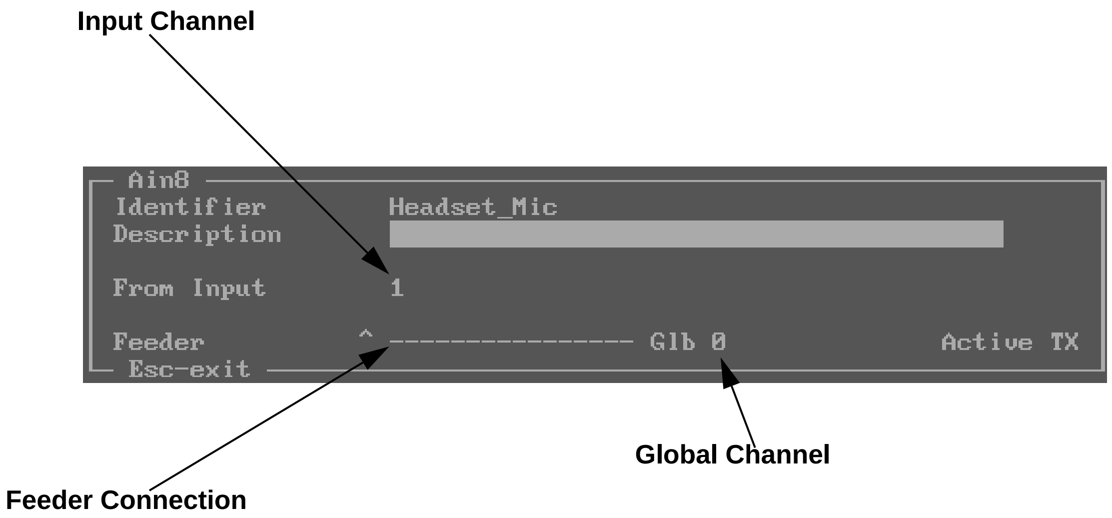


Analog Input

Brings an analog input channel in as a signal which can be mixed, filtered, or added into any combination of highway channels via a feeder connection.

Note: This object replaces the analog input objects of earlier revisions of ModelBuilder, since all three analog input feeders can be replaced by this one object used in conjunction with either a balancer, or a signal mixer.

Analog Input



Input Channel

Input channel on waveform synthesizer.

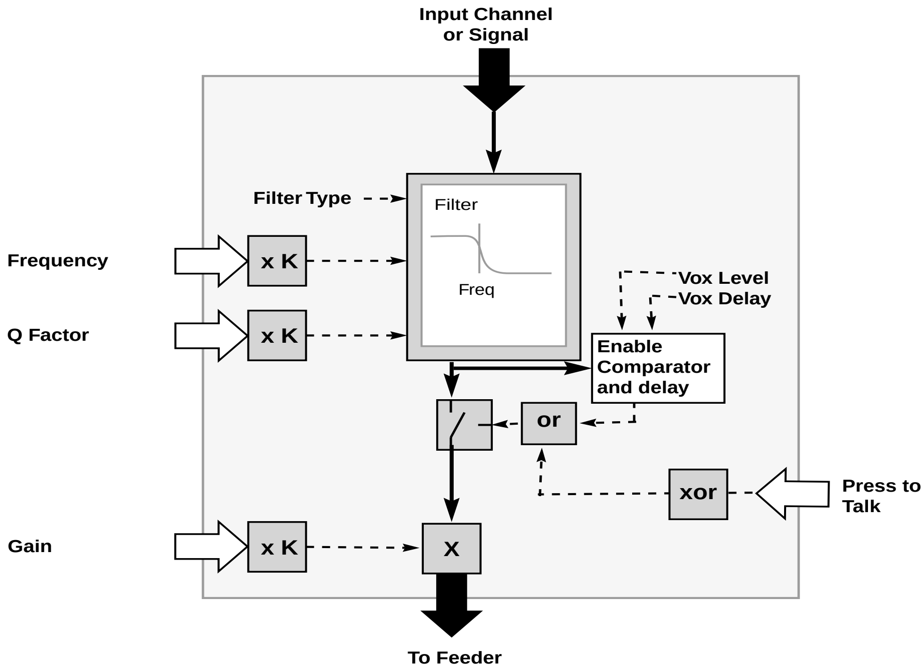
Feeder Connection

Connection to a feeder, which adds the analog input into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

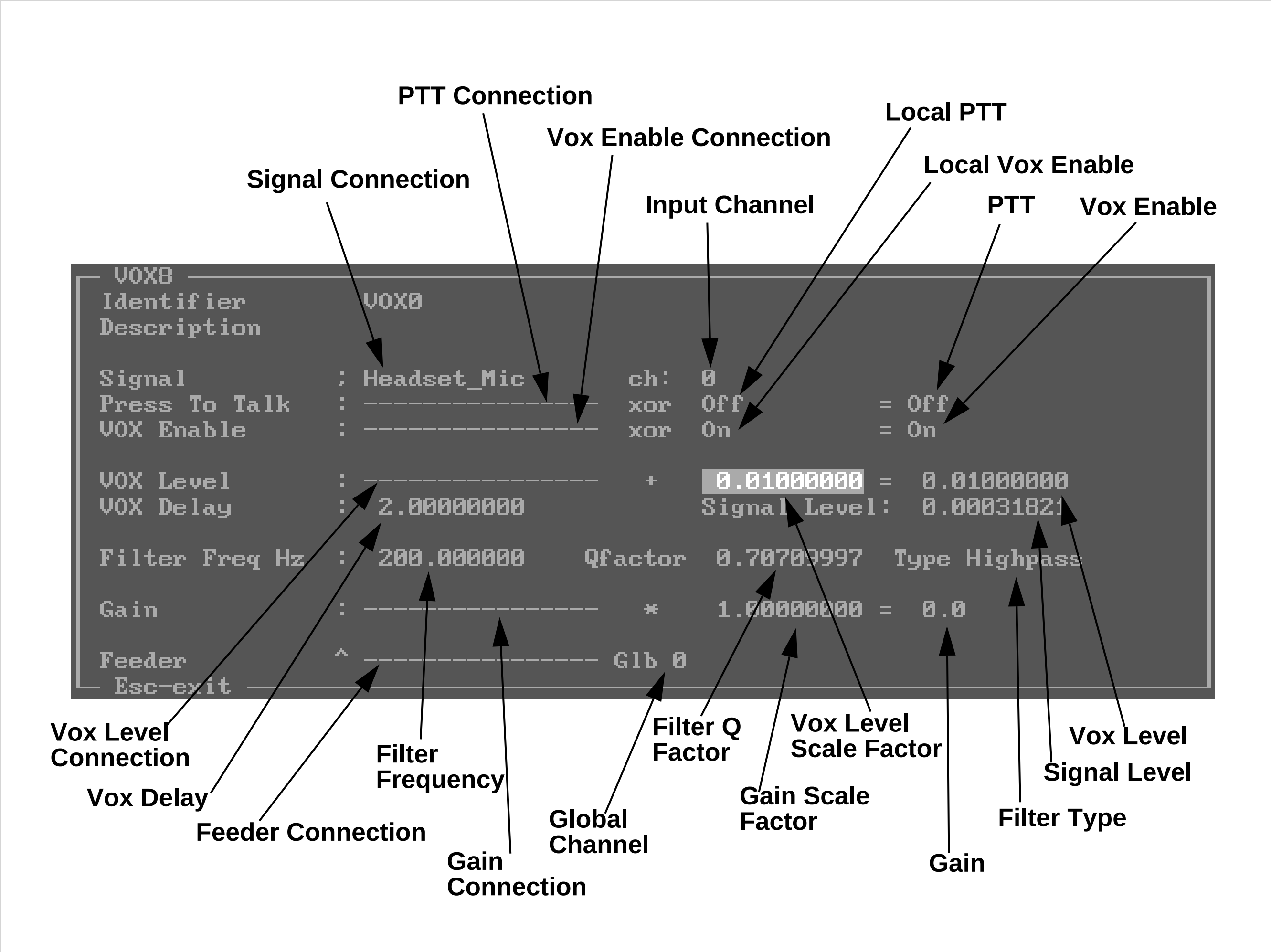
VOX



VOX

The VOX object allows VOX or push to talk control over an input channel or a signal. If the filtered input sound level exceeds the VOX level, the VOX comparator will turn on for an amount of time equal to the VOX delay.

VOX



Input Channel

Analog input channel on waveform synthesizer. This should be set to zero if the input is taken from the Signal Connection.

Signal Connection

Connection to an input signal. If this field is not empty, then the input channel should be set to zero.

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of the output signal. If the gain connection is blank then the gain scale factor is used as the gain value; otherwise the gain is the scale factor times the output result of the control object.

PTT Connection

Connection to a control object that provides a Press to Talk value.

Local PTT

Provides a local Exclusive Or value for the Press to Talk Connection.

VOX

PTT

If the Press To Talk is ON, then the input signal will be passed to the output.

Signal Level

Indicates the sound level of the input signal.

Vox Level Connection

Provides a connection to a control object which allows the host computer to control the Vox level.

Vox Level Scale Factor

Provides an offset value for the Vox level Connection. If the Vox Level Connection Field is empty, then this becomes the Vox Level.

Vox Level

If the Vox enable is on, then the Signal Level is compared to the Vox Level. If the signal level is higher, it enables the input to be fed through for a period of time equal to the Vox Delay. If the signal level exceeds the Vox level while the timer is on, it will reset the timer. (i.e. If the input signal exceeds the Vox level at least once every delay time, the input signal will be continually fed through.).

Vox Delay

The amount of time after the Signal Level falls below the Vox level that the input signal will continue to be fed through.

Filter Type

Type of filter the input signal will be filtered by. The filtering occurs before the Vox compares the signal level to the Vox Level. If no filtering is desired, the filter type should be set to AllPass.

Filter Frequency

Provides the characteristic frequency of the filter.

Filter Q Factor

Provides the Q factor for the filter.

Vox Enable Connection

Connection to a control object that allows host control over the Vox Enable.

Local Vox Enable

Provides an exclusive or value for the Vox enable connection. If the Vox Enable Connection is empty, it provides the value for the Vox Enable.

Vox Enable

If the Vox enable is on, the Vox comparator (whose action is described in Vox Level) can enable the feeding through of the signal. If it is off, the signal can only be fed through by the Press to Talk.

Feeder Connection

Connection to a feeder, which adds the output into the signal highway.

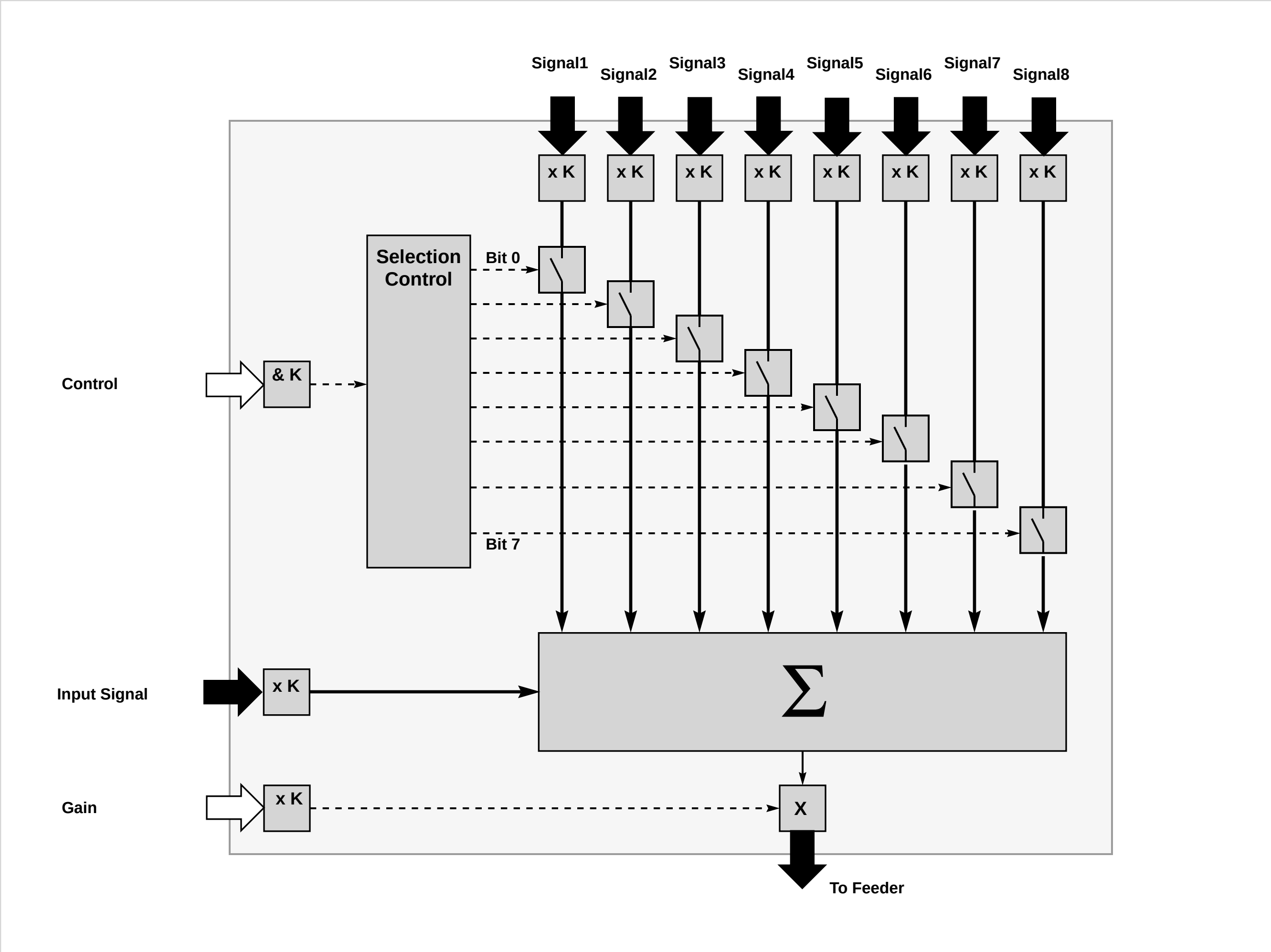
Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Global In

For information on the Global In signal, see the section on global channels.

Signal Selector



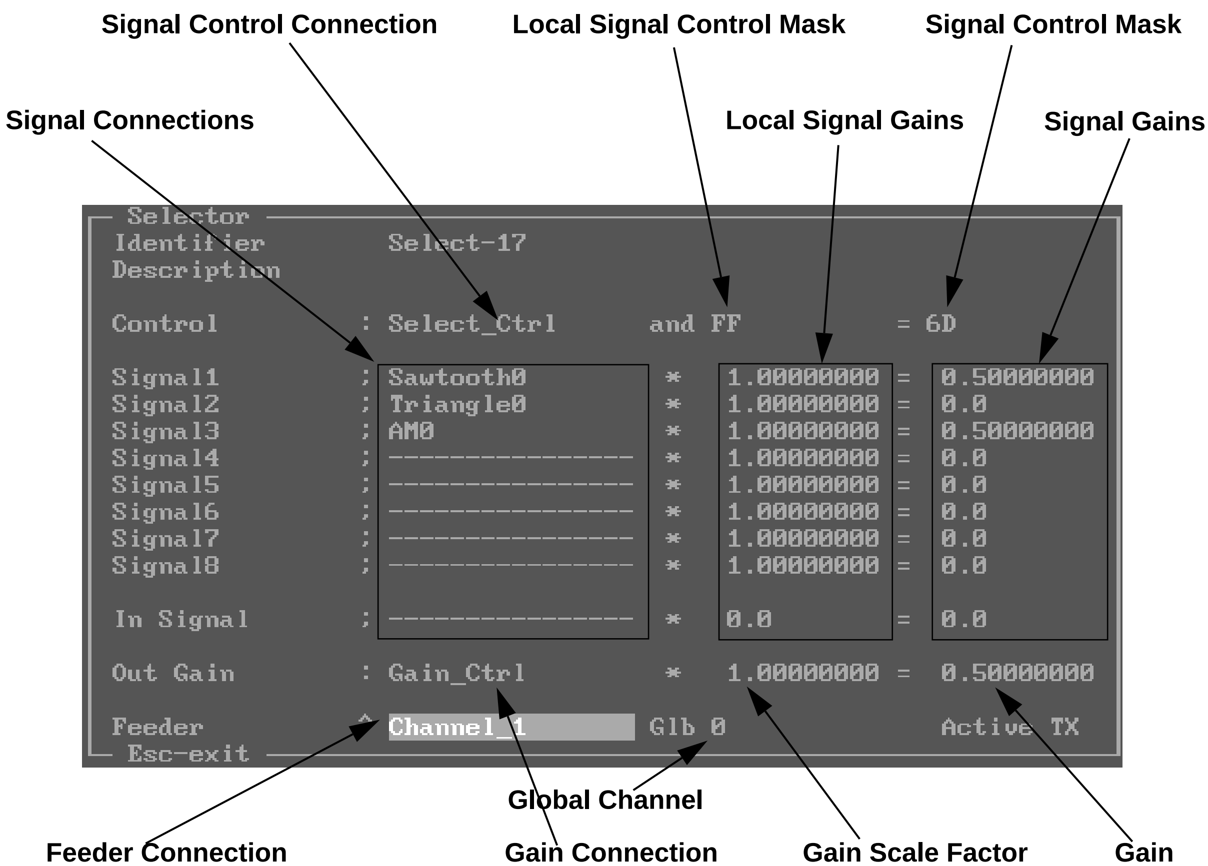
Signal Selector

The signal selector provides a controlled mixing of up to eight signals into a single signal.

Control is provided over which of the eight signals should be mixed and the overall output signal gain.

Note: When connecting the output of the signal selector to any other signal object (e.g. Filter, Frequency or Amplitude Modulator) the input signals should be restricted to the basic signal sets. The complex signals that are in the Radios, Nav-Aids and Intercom packages will not always be mixed into this signal output correctly, but will be mixed into the signal highway correctly. This is a result of the radio and intercom matching being performed at the highway interface and not within this selector object.

Signal Selector



Signal Connections

Signal connection from the signal list, for the eight signals which are mixed into a composite output by the selector. There is also a ninth signal, the In Signal, which is mixed into the output independent of the Control Mask.

Local Signal Gains

Local gain control for each signal.

Signal Gains

Overall gain for each signal, taking into account each signal's control bit (on or off).

Signal Control Connection

Connection to control section of model for switching signals on or off. If connected to a boolean all the signal will be switched together. If connected to an integer, the least significant byte provides a bit mask for each signal, with the l.s.b. controlling signal1 and the m.s.b. signal8.

Local Signal Control Mask

Local control mask for the eight signals. When a connection is present the local mask acts as a bit enable which is the AND of the local mask and the connection value.

Signal Selector

Signal Control Mask

Individual signal control byte. Each bit enables on of the eight signals. When the l.s.b. is 1 the signal is added to the group. The remaining signals are controlled by each bit in sequence, with the m.s.b. being the control for signal8.

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of file replay source. If the gain connection is blank then the gain scale factor is used as the gain value; otherwise the gain is the scale factor times the output result of the control object.

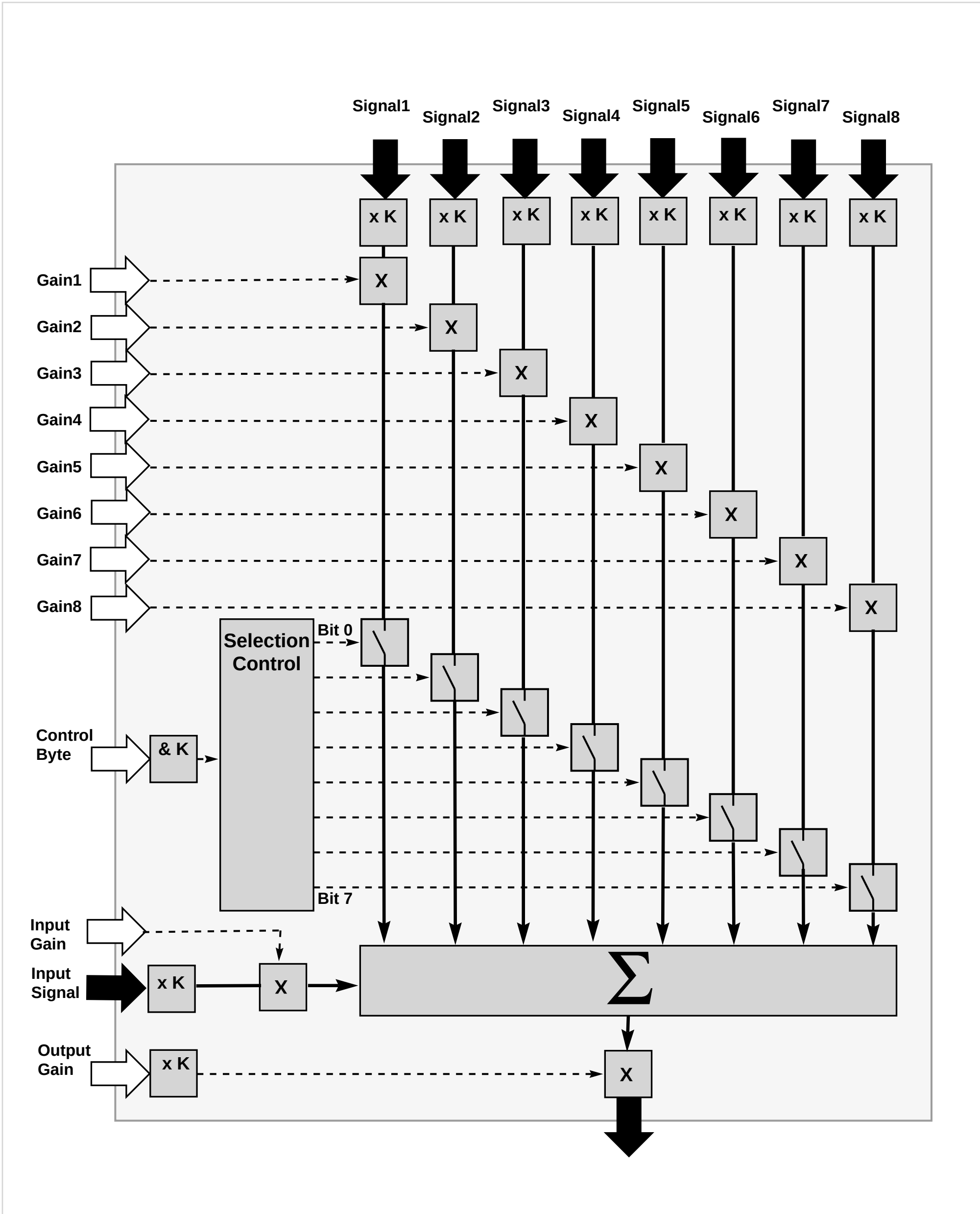
Feeder Connection

Connection to a feeder, which adds the total signal sound source into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Signal Mixer

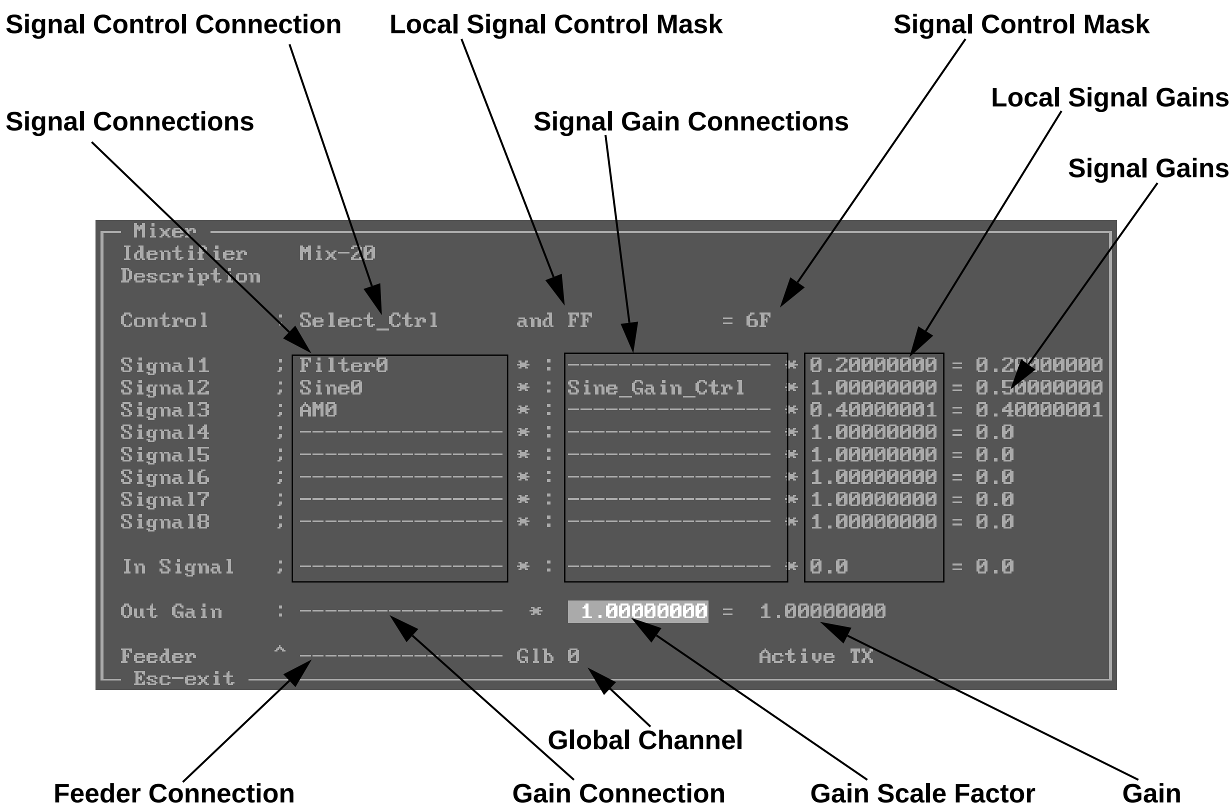


Signal Mixer

The signal mixer provides a controlled mixing of up to eight signals into a single signal. Control is provided over which of the eight signals should be mixed, with both individual signal and overall gain control.

Note: When connecting the output of the signal mixer to any other signal object (e.g. Filter, Frequency or Amplitude Modulator) the input signals should be restricted to the basic signal sets. The complex signals that are in the Radios, Nav-Aids and Intercom packages will not always be mixed into this signal output correctly, but will be mixed into the signal highway correctly. This is a result of the radio and intercom matching being performed at the highway interface and not within this mixer object.

Signal Mixer



Signal Connections

Signal connection from the signal list, for the eight signals which are mixed into a composite output by the selector. There is also a ninth signal, the In Signal, which is mixed into the output independent of the Control Mask.

Local Signal Gains

Local gain control for each signal.

Signal Gains

Overall gain for each signal, taking into account each signal's control bit (on or off).

Signal Control Connection

Connection to control section of model for switching signals on or off. If connected to a boolean all the signal will be switched together. If connected to an integer, the least significant byte provides a bit mask for each signal, with the l.s.b. controlling signal1 and the m.s.b. signal8.

Local Signal Control Mask

Local control mask for the eight signals. When a connection is present the local mask acts as a bit enable which is the and of the local mask and the connection value.

Signal Mixer

Signal Control Mask

Individual signal control byte. Each bit enables on of the eight signals. When the l.s.b. is 1 the signal is added to the group. The remaining signals are controlled by each bit in sequence, with the m.s.b. being the control for signal8.

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of file replay source. If the gain connection is blank then the gain scale factor is used as the gain value; otherwise the gain is the scale factor times the output result of the control object.

Feeder Connection

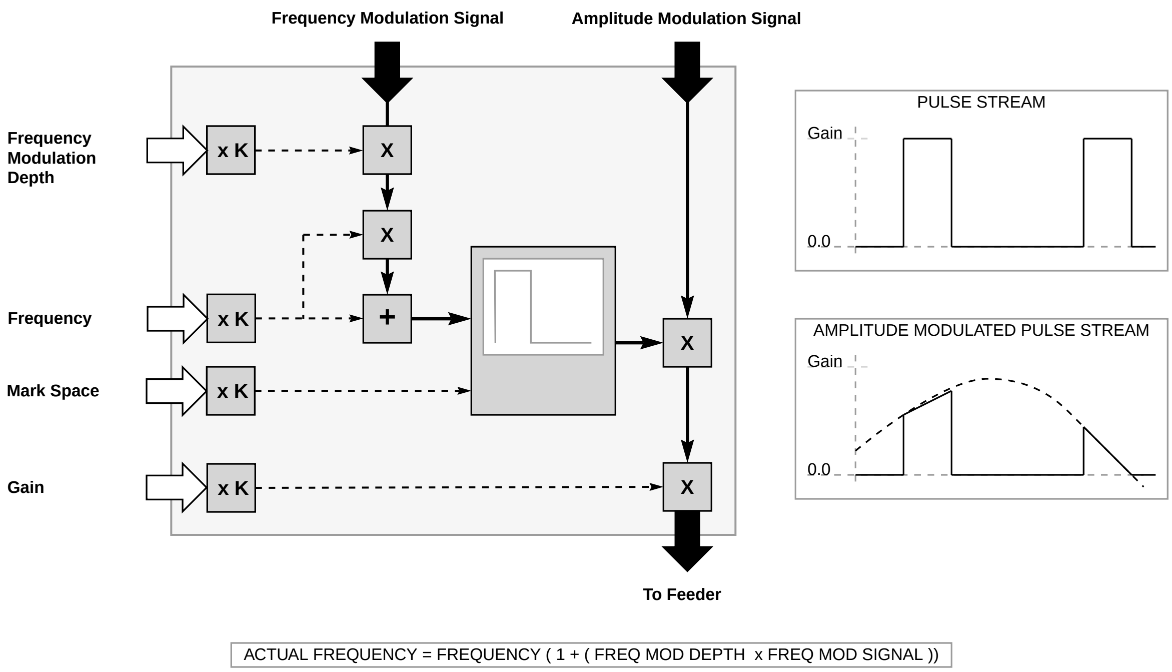
Connection to a feeder, which adds the total signal sound source into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Pulse Stream and Pulse Sequence (Pulse Stream Package)

Pulse



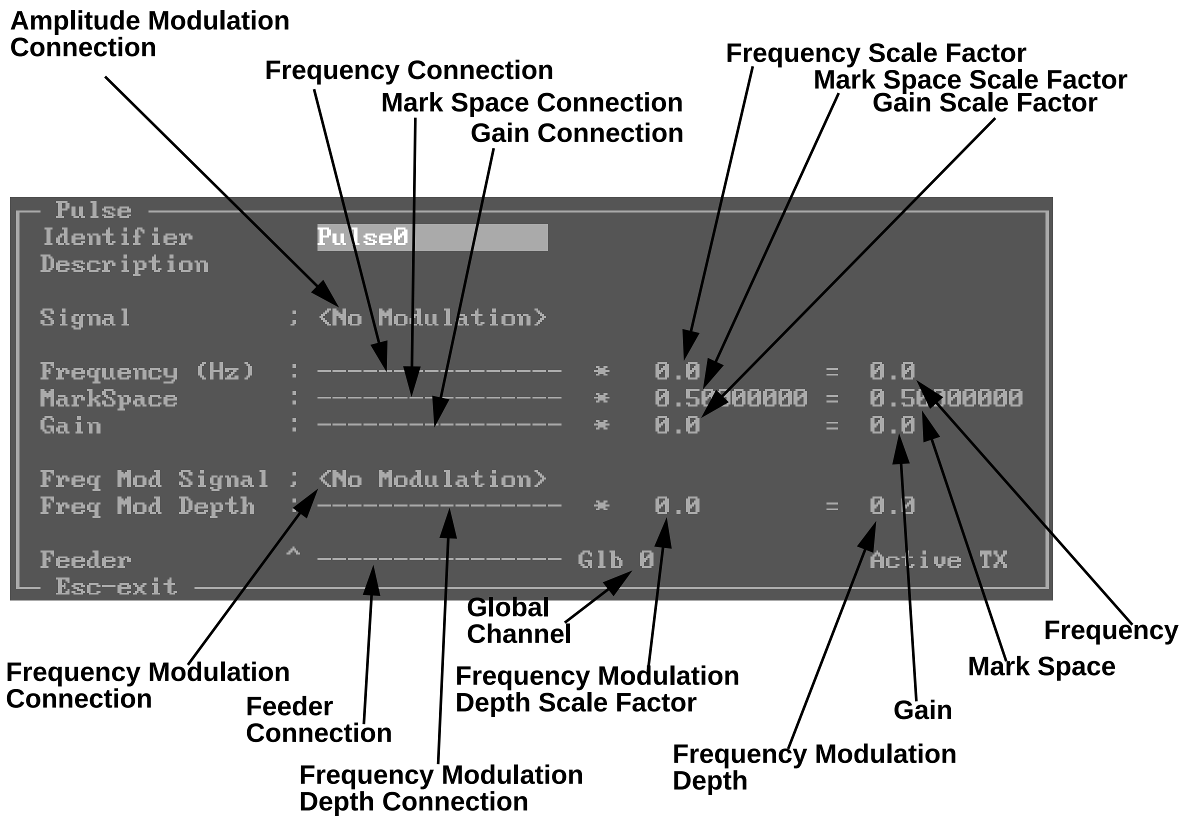
Pulse Signal

This signal source produces a pulse stream signal which can be mixed in any proportion on any of the analog output channels.

NOTE: A pulse signal is similar to the square wave except it is limited to positive amplitudes. Both gain, frequency and mark/space ratio can be controlled by input variables from elsewhere in the model, or from the host interface.

Both the frequency and pulse amplitude can be modulated by other signals within the signal processor, with the pulse signal object having control over the depth of modulation.

Pulse



Amplitude Modulation Connection
Connection to the pulse amplitude modulating signal. No connection defaults to a fixed amplitude pulse stream.

Frequency Connection
Control object connection to provide overall frequency control from elsewhere in model.

Frequency Scale Factor
Scaling factor for frequency control value.

Frequency
Frequency (in Hertz) of the pulse signal generated by waveform synthesizer. If the frequency connection is blank then the frequency scale factor is used as the frequency value, else the frequency is the scale factor times the output result of the control object.

Mark Space Connection
Control object connection to provide mark space control from elsewhere in model.

Mark Space Scale Factor
Scaling factor for mark space ratio control value.

Pulse

Mark Space Ratio

Mark to space ratio for square wave. Value ranges from 0 to 1.0 with 0 being all space, 1.0 being all mark. If the M/S connection is blank then the M/S scale factor is used as the value, else the M/S is the scale factor times the output result of the control object.

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of square wave. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Frequency Modulation Connection

Connection to the frequency modulating signal, which is scaled by the modulation depth and added to the frequency.

Modulation Depth Connection

Control object connection to provide frequency modulation depth control from elsewhere in model.

Modulation Scale Factor

Scaling factor for frequency modulation depth control value.

Modulation Depth

Frequency modulation depth value, controls the effect of the frequency modulation signal:

$$ActualFreq = Freq \times (1 + (ModDepth \times ModSignal))$$

Usually it falls in the range 0 to 1.0, when used in conjunction with a unity gain modulation signal.

CAUTION: To avoid unpredictable behavior care should be taken to ensure that the product of modulation depth and modulation signal does not span a range greater than -1.0 to +1.0.

Feeder Connection

Connection to a feeder, which adds the square wave into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Pulse Stream

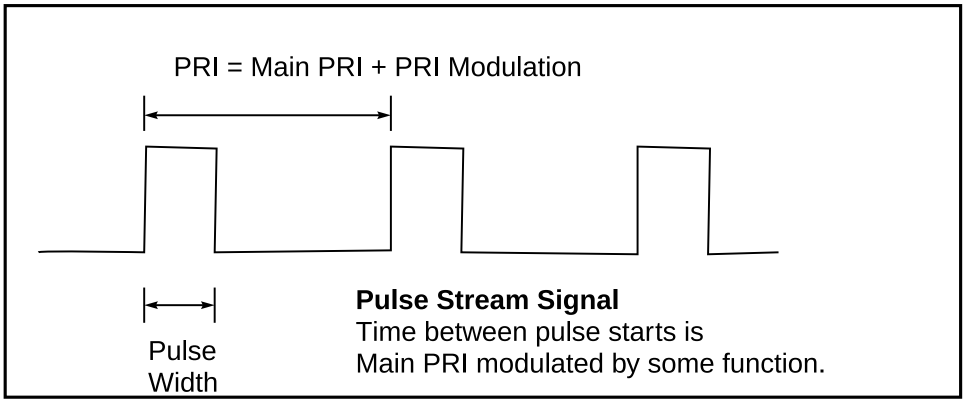


Figure 1

Pulse Stream Signal

The Pulse Stream signal source is a sophisticated signal source which generates a stream of pulses. Like the Pulse signal, the pulses have an amplitude between 0 and 1.

Through the next few pages is a series of diagrams explaining how the Pulse Stream Signal works. The parameters referred to are set in the Pulse Stream Object Inspector screen, which is shown below. You may want to refer to that periodically while reading this section.

The pulse stream is typically used to frequency or amplitude modulate other signals, to provide time varying tones.

Figure 1 shows a Pulse stream. The pulse width and PRI (Pulse Repetition Interval) are shown. The PRI is usually the Main PRI (supplied in the Object Inspector) modulated by some other signal or function.

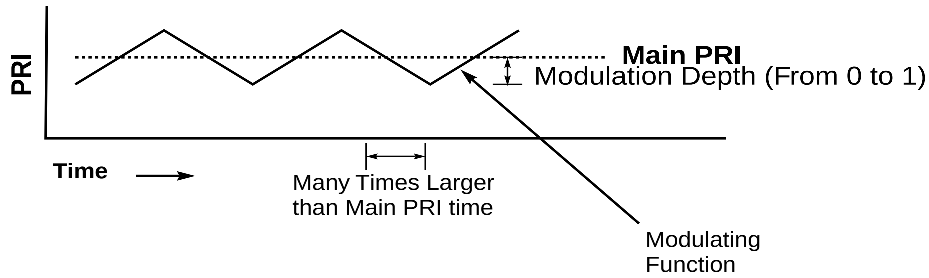
There are several ways to Modulate the timing between pulses. These different modulation methods are called Pulse types, and are specified in the Pulse Type field. Each Pulse type has a number from 1 to 255 and a name.

For each pulse type, some of the parameters are ignored. A table of which pulse types use which parameters is given in figure 6.

The next few pages describe the various pulse types.

Pulse Stream

Signal Modulated



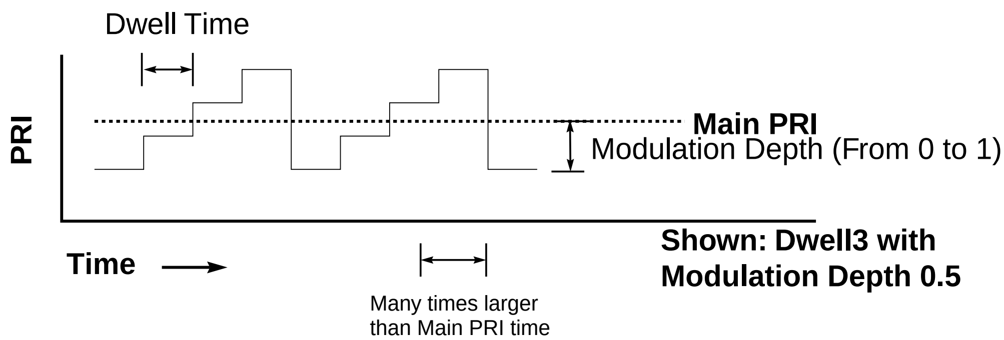
Value of PULSE TYPE field	Modulating Function	Note
2	Sine Wave	Frequency, Mod depth
3	Triangle Wave	Given by Fields in
4	Sawtooth Wave	PULSE STREAM object
5	Square Wave	
9	External Signal	Signal Connection
1,6-8,10-100,220-255	Steady - no modulation	

Figure 2

Figure 2 shows how the first few Pulse Types work. The graph shows the PRI (Pulse Repetition Interval) as a function of time. The specific example is a triangle modulated pulse stream (Pulse type 3, Triangle.) Built into the object are sine, triangle, sawtooth and square wave modulations. You can also use an arbitrary signal to modulate the PRI by selecting Pulse Type 9 (external) and entering a signal in the PRI Modulation field. For the built in modulation signals, you need to specify a modulation frequency and modulation depth. For the external pulse type, the modulation frequency is ignored and the modulation depth becomes a scale factor for the external modulation signal.

Pulse Stream

Dwell



Value of PULSE TYPE field	Name	Note
101-150	Dwell1-Dwell150	Number of steps = Dwell number +1. Dwell Time and Mod Depth specified in Pulse Stream Object

Figure 3

Pulse types 101 to 150 are called dwell1 to dwell 150. The spacing between pulses (PRI) is modulated in a step like fashion. The length of time on each step is given by the dwell time, and the number of steps is the dwell number plus one. (e.g., for dwell5 there will be six steps.) The steps are evenly spaced, and their height is given by the modulation depth, as shown.

The PRI time varies between

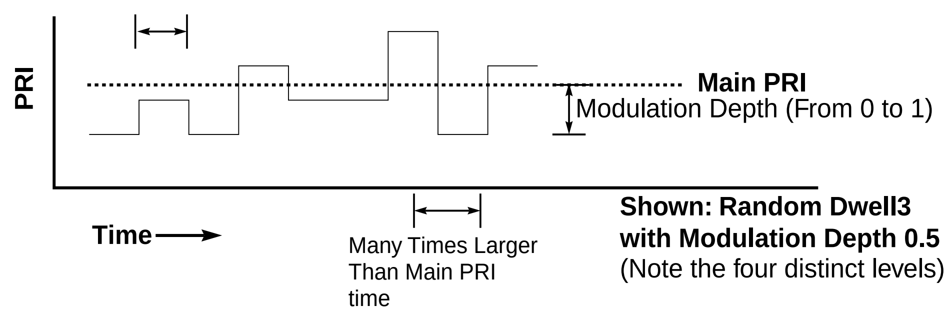
PRI Main x (1 - Modulation Depth)

and

PRI Main x (1 + Modulation Depth)

Pulse Stream

Random Dwell



Value of PULSE TYPE field	Name	Note
151-198	Random Dwell1 to Random Dwell 48	Number of steps = Dwell number +1. Dwell Time, Mod depth specified in Pulse Stream Object
199	Random Dwell	Jumps to any level, up to Modulation Depth

Figure 4

Figure 4 shows the Random Dwell Pulse Type (Pulse types 151-199). The random dwell is the same as the dwell, except that instead of stepping sequentially through the levels it jumps randomly among them. The number of levels is given by the Random Dwell number + 1 - i.e., Random Dwell 5 would jump through 6 different PRI modulation times.

In addition, there is a Random Dwell with no number (Pulse type 199). With this pulse type, the PRI time jumps randomly throughout it's allowed range, staying at each PRI time for a dwell time.

The allowed range for the random dwell is between

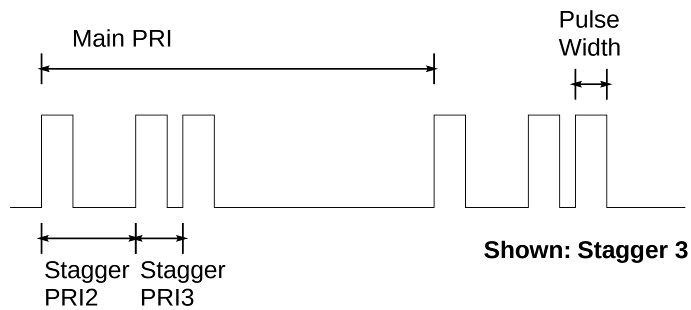
PRI Main x (1 - Modulation Depth)

and

PRI Main x (1 + Modulation Depth)

Pulse Stream

Stagger



Value of PULSE TYPE field	Name	Note
201-208	Stagger 1 to Stagger 8	Number of Pulses = Stagger Number.
200 209-219	Stagger 1 Stagger 8	Frequency modulation fields and dwell time are ignored

Figure 5

The Stagger Pulse types operate differently from the other pulse types mentioned before. Instead of modulating the PRI, up to eight PRI values are given which define the spacing of the pulses. The number of pulses in the stream is given by the stagger number (e.g. Stagger 4 has four pulses per cycle.)

The length from cycle to cycle is given by the Main PRI value, while the placement of the intermediate pulses is shown in the diagram.

The stagger pulse type is the only one to use the Stagger PRI values. It ignores all of the PRI modulation fields.

Pulse Stream

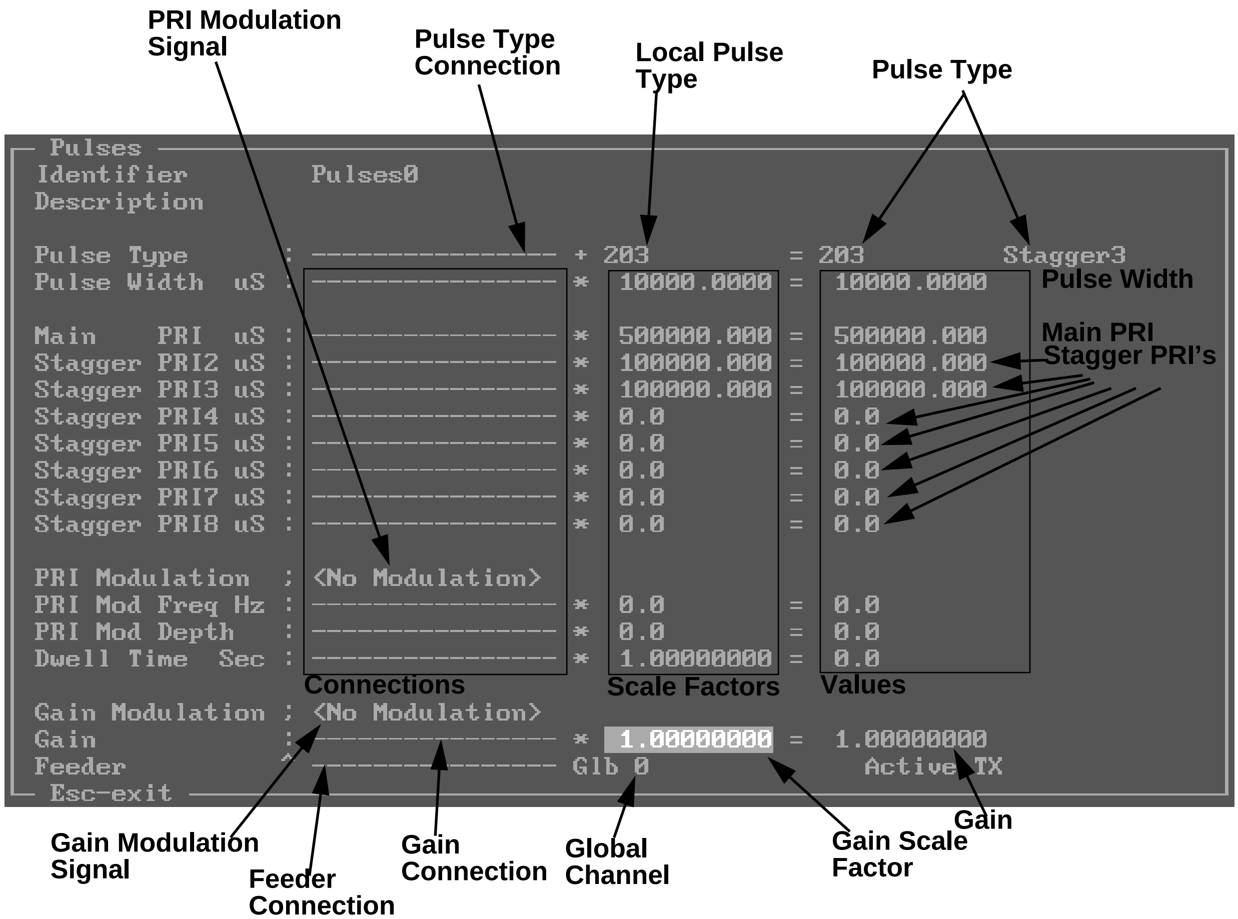
Pulse Stream Pulse Type Table

Pulse Type Number	Pulse Type Name	Parameters Used: Stagger PRIs	PRI Modulation Freq	PRI Modulation Depth	Dwell Time
1	steady	N	N	N	N
2	sine	N	Y	Y	N
3	triangle	N	Y	Y	N
4	sawtooth	N	Y	Y	N
5	square	N	Y	Y	N
6-8	steady	N	N	N	N
9	external	N	N	Y	N
10-100	steady	N	N	N	N
101-150	dwell1-dwell50	N	N	Y	Y
151-198	random dwell1 to random dwell 48	N	N	Y	Y
199	random dwell	N	N	Y	Y
200	stagger1	Y	N	N	N
201-208	stagger1-stagger8	Y	N	N	N
209-219	stagger8	Y	N	N	N
220-255	steady	N	N	N	N

Figure 6

This pulse stream type table summarizes which parameters are used by which pulse types.

Pulse Stream



Pulse Type Connection

Provides a connection to a control object to specify the pulse type.

Local Pulse Type

Provides an offset for the pulse type connection. If the pulse type connection field is empty, it provides the pulse type.

Pulse Type

This field determines the type of pulse stream. A detailed discussion of the different types of pulse streams are given in the immediately preceding pages. The pulse type is a number between 0 and 255. Each number has an associated name which appears next to the number. A pulse type of zero turns off the pulse stream.

Pulse Width Connection

Provides a connection to a control object which specifies the pulse width.

Pulse Width Scale Factor

Provides a scale factor for the Pulse Width Connection. If the Pulse Width Connection field is empty, this field provides the Pulse Width.

Pulse Width

Provides the width of the pulses in the pulse stream, in microseconds. See figure 1, above.

NOTE: the resolution of the pulse width is limited to the sample rate of the DSP. If the DSP is running at 8000 samples/sec, the pulse width resolution will be 125 microseconds.

Pulse Stream

Main PRI Connection

Provides connection to a control object which provides the Main PRI value.

Main PRI Scale Factor

Provides a scale factor for the Main PRI connection. If the Main PRI connection is empty, this field provides the Main PRI value.

Main PRI

Provides the basic spacing between pulses, as measured from the beginning of successive pulses (see fig 1, above). This spacing can be modulated by the different pulse types. The Main PRI is measured in microseconds.

NOTE: the timing resolution of the Main PRI is limited to the sample rate of the DSP. If the DSP is running at 8000 samples/sec, the pulse width resolution will be 125 microseconds.

Stagger PRI Connections

Provide connections to control objects which provide the Stagger PRI values.

Stagger PRI Scale Factors

Provide scale factors for the Stagger PRI Connections. If any of the connection fields are empty, this provides the Stagger PRI value.

Stagger PRI’s

Gives the stagger time, in microseconds, for the stagger pulse types. These values are only used by the stagger pulse types (pulse types 200 to 219). See figure 5 above for a detailed explanation.

NOTE: the timing resolution of the Stagger PRI’s is limited to the sample rate of the DSP. If the DSP is running at 8000 samples/sec, the pulse width resolution will be 125 microseconds.

PRI Modulation Connection

Provides a connection to a signal which modulates the Main PRI value. This field is only used by Pulse Type 9 (external).

PRI Mod Freq Connection

Provides a connection to a control objects which provide the PRI Modulation Frequency.

PRI Mod Freq Scale Factor

Provide scale factors for the PRI Modulation Frequency Connection. If any of the connection fields are empty, this provides the PRI Modulation Frequency.

PRI Mod Freq

For pulse types 2 through 5 (sine, triangle, sawtooth, square), this field provides the frequency of the signal modulating the Main PRI. It is ignored for other pulse types.

PRI Mod Depth Connection

Provides a connection to a control objects which provide the PRI Modulation Depth.

PRI Mod Depth Scale Factor

Provide scale factors for the PRI Modulation Frequency Connection. If any of the connection fields are empty, this provides the PRI Modulation Depth.

PRI Mod Depth

A number between 0 and 1 which determines the modulation depth for the Main PRI modulation. A 0 indicates no modulation. The range of PRI values will be

from

PRI Main x (1 - Modulation Depth)

to

PRI Main x (1 + Modulation Depth)

This field is ignored for the Steady and Stagger Pulse types.

Pulse Stream

Dwell Time Connection

Provides a connection to a control object which provides the dwell time.

Dwell Time Scale Factor

Provides a scale factor for the Dwell Time Connection. If the connection field is empty, this provides the Dwell Time.

Dwell Time

This is used by the dwell and random dwell pulse types (numbers 101 to 199). It gives the value, in seconds, that the PRI stays on a particular value. See figures 3 and 4, above, for a detailed explanation.

Gain Modulation Signal

Connection to a signal which will modulate the amplitude of the pulse stream.

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of square wave. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

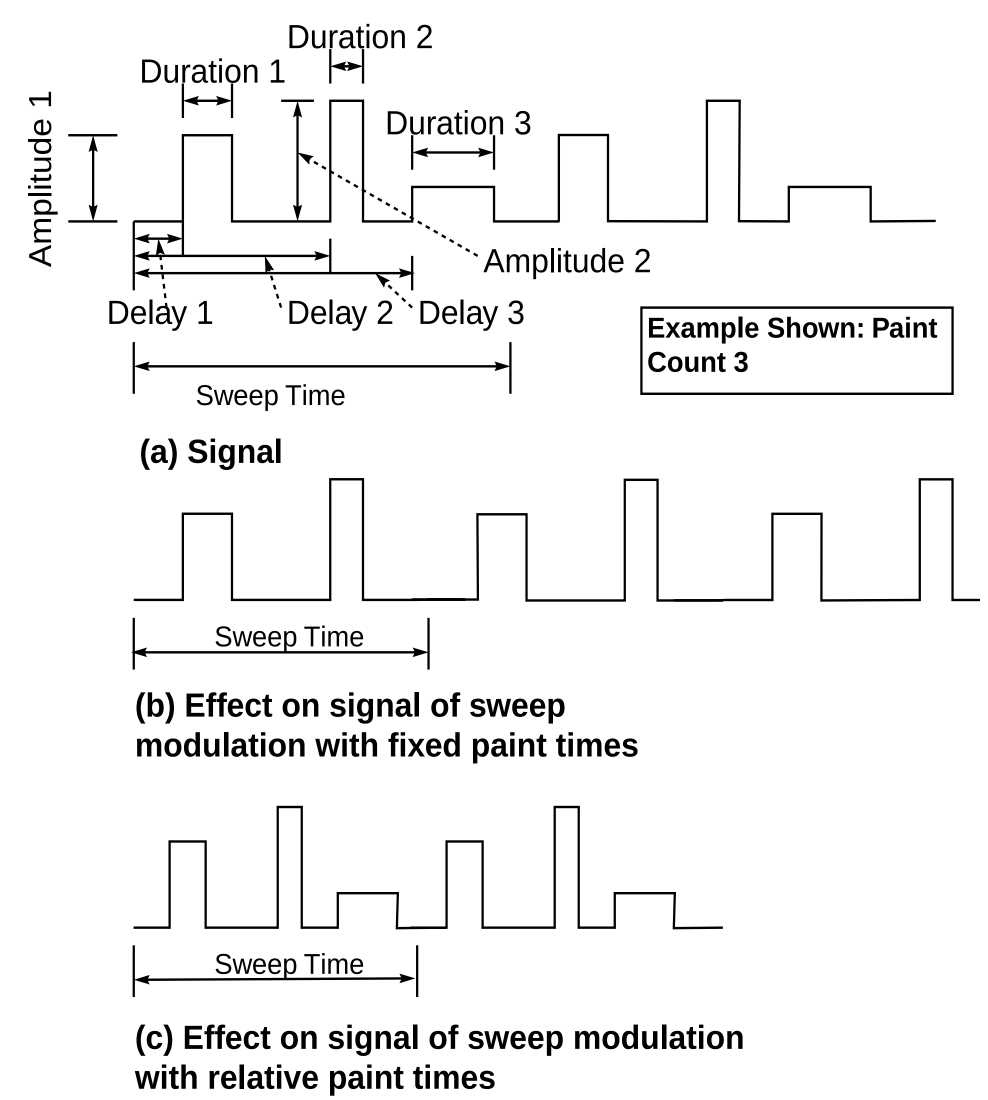
Feeder Connection

Connection to a feeder, which adds the square wave into the signal highway.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same signal that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Pulse Sequence



Pulse Sequence Signal

The Pulse Sequence signal is a signal object that can generate a repeating series of up to eight pulses of arbitrary pulse width, pulse amplitude, and timing. Typically, this signal is used to frequency or amplitude modulate other signals.

Part (a) of the figure above shows what the various parameters in the Pulse Sequence specify. The paint count specifies the number of pulses, while the delays, amplitudes, and durations are specified as shown.

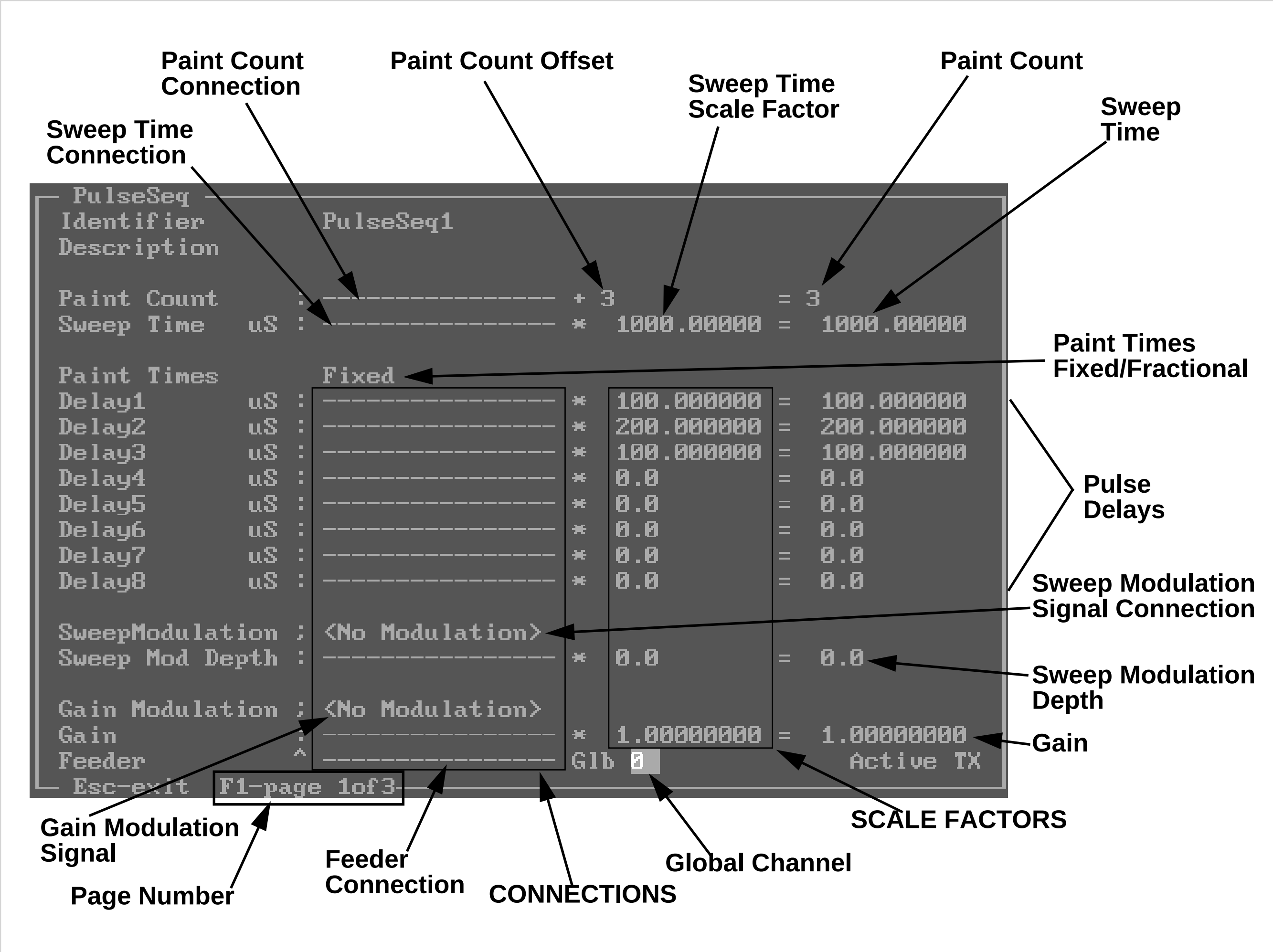
Pulse Sequence

The pulse sweep time can be modulated by an external signal. How the pulses within the sweep act depends on whether the Paint Times are defined as “Fixed” or “Fractional”.

For fixed paint times, the paint times and durations are defined in terms of the number of microseconds after the initiation of the sweep. Modulating the sweep time will not affect the time of the pulses, and if the sweep time cuts off a pulse it will not be generated. Part (b) of the figure shows the same pulse sequence as part (a), but with the sweep time shortened in fixed paint time mode.

For fractional paint times, the paint times and durations are defined in terms of the fraction of the total sweep time. Modulating the sweep time will compress (or extend) the pulses and move them closer together (or farther apart). Part (c) of the figure shows the same pulse sequence as part (a), but with the sweep time shortened in fractional paint time mode.

Pulse Sequence



PAGE 1 of 3

Paint Count Connection

Provides connection to a control object which supplies the paint count.

Paint Count Offset

Provides an offset for the paint count connection. If the paint count connection is empty, the offset provides the paint count value.

Paint Count

The paint count is the number of pulses in a sweep. Allowable values are from 1 to 8. A value of 0 disables the pulse sequence.

Sweep Time Connection

Provides connection to a control object which supplies the sweep time.

Sweep Time Scale Factor

Provides a scale factor for the sweep time connection. If the sweep time connection is empty, the scale factor becomes the sweep time, in uS.

Sweep Time

Provides the length of the sweep that the pulses are contained in, in microseconds. See the figure above for details.

NOTE 1: If the Paint Times flag is set to fractional, this field will be a frequency in hertz.

NOTE 2: The timing resolution of the sweep time is 1/sample rate. For a sample rate of 8kHz, the resolution is 125 microseconds.

Pulse Sequence

Paint Times Fixed/Fractional

This is a flag which is set either to Fixed or Fractional. If set to Fixed, the sweep time, delay times, and durations are all measured in microseconds.

If set to Fractional, the sweep is a frequency in hertz, and the Durations and delays are specified as a fraction of the sweep time.

Pulse Delay Connections

Provide connections to control objects which determine the pulse delays.

Pulse Delay Scale Factors

Provide scale factors for the Pulse delay connections. If the connections are empty, the scale factors become the delays.

Pulse Delays

Provide the delay, for each pulse, between the beginning of the sweep and the beginning of each pulse. If a pulse delay puts a pulse outside the sweep, that pulse will not be generated.

NOTE: The resolution of the Delays is 1/sample rate. For a sample rate of 8kHz, the resolution is 125 microseconds.

Sweep Modulation Signal Connection

Provides connection to a signal which modulates the sweep time (Fixed mode) or the sweep frequency (Fractional mode).

Sweep Modulation Depth Connection

Provides a connection to a control object which gives the sweep modulation depth.

Sweep Modulation Depth Scale Factor

Provides a scale factor for the Sweep Modulation Depth Connection. If the connection field is empty, this field provides the Sweep Modulation Depth.

Sweep Modulation Depth

Provides the modulation depth for the sweep modulation signal. This should be between 0 and 1. A one means that if the modulation signal has a gain of 1, then the sweep time (or frequency) will get modulated between zero and twice it’s normal value. A modulation depth of zero means that no sweep modulation will occur.

Gain Modulation Signal

Connection to a signal which will modulated the amplitude of the pulse sequence.

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of square wave. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Feeder Connection

Connection to a feeder, which adds the square wave into the signal highway.

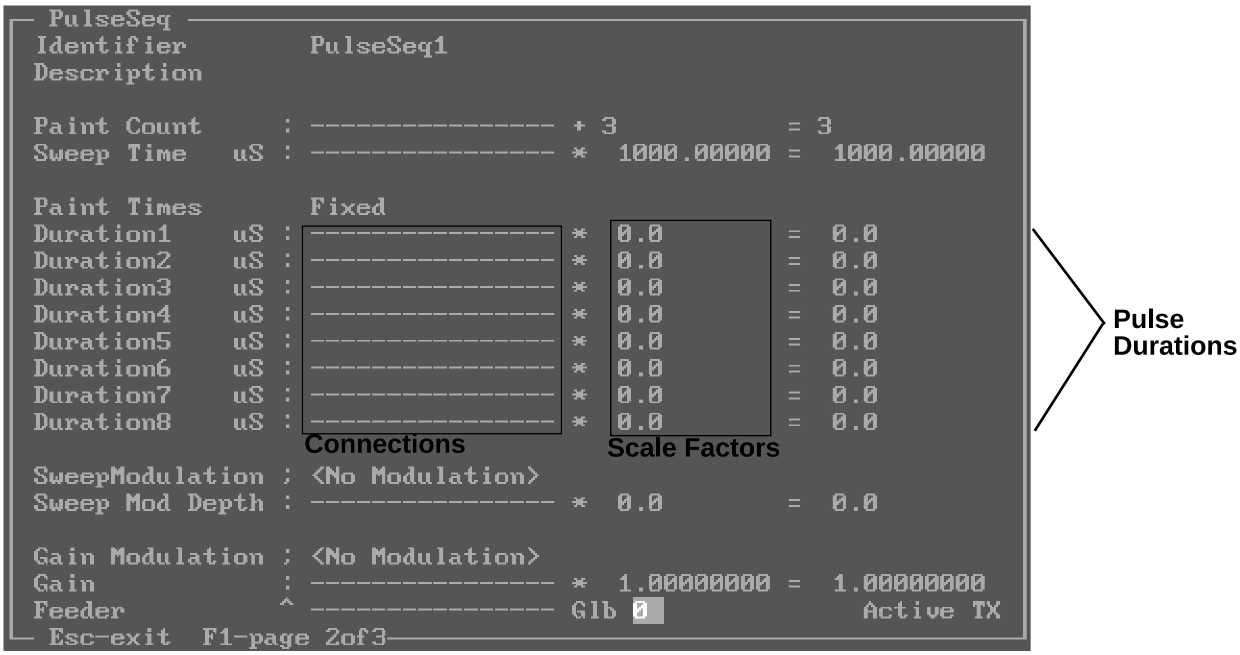
Page Number

This object has three screens. The page number indicates what screen is being displayed. F1 toggles between the screens.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same signal that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Pulse Sequence



PAGE 2 of 3

Pulse Duration Connections

Provide connections to control objects which determine the pulse durations.

Pulse Duration Scale Factors

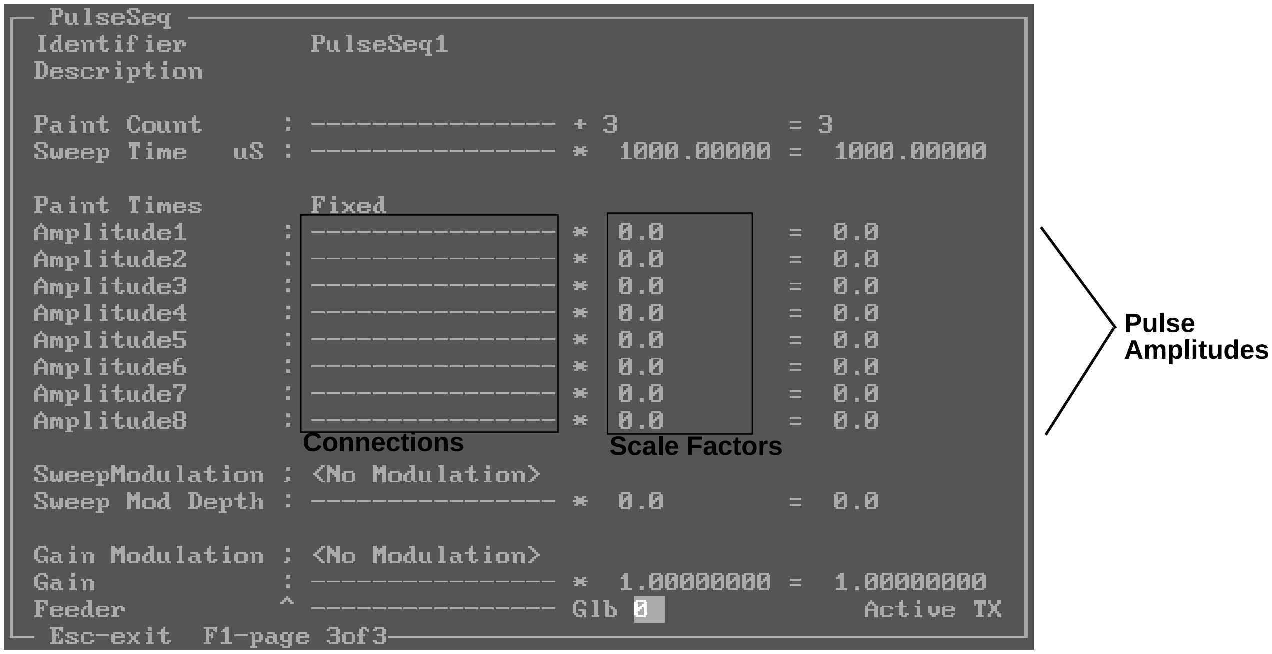
Provide scale factors for the Pulse Duration connections. If the connections are empty, the scale factors become the pulse durations.

Pulse Durations

Provide the width of each pulse, in microseconds if in Fixed mode, or as a fraction of the sweep time in Fractional mode.

NOTE: The resolution of the pulse durations is 1/sample rate. For a sample rate of 8kHz, the resolution is 125 microseconds.

Pulse Sequence



PAGE 3 of 3

Pulse Amplitude Connections

Provide connections to control objects which determine the pulse Amplitudes.

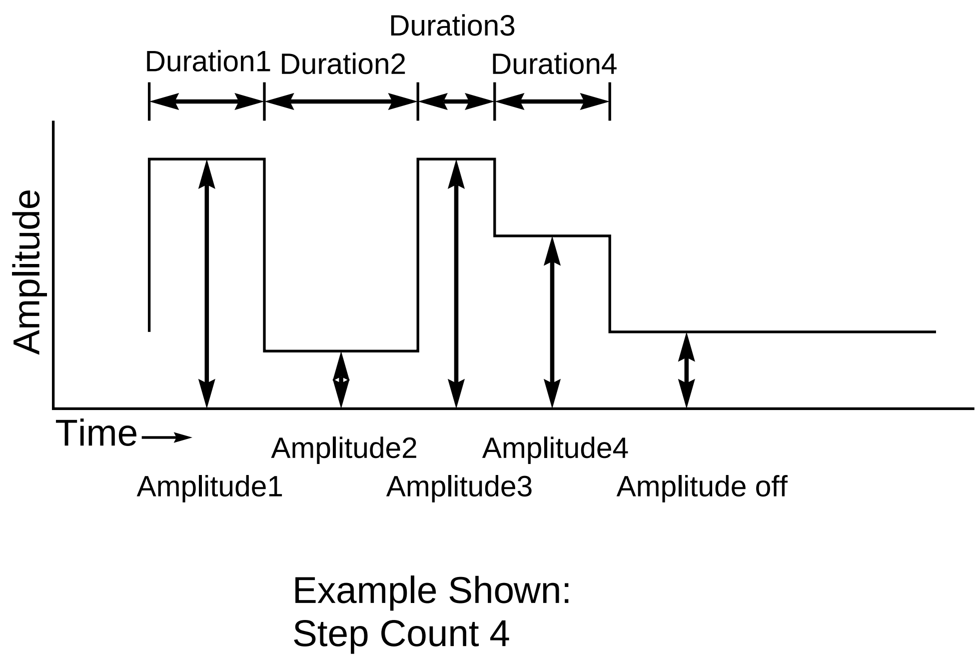
Pulse Amplitude Scale Factors

Provide scale factors for the Pulse Amplitude connections. If the connections are empty, the scale factors become the pulse Amplitudes.

Pulse Amplitudes

Provides the height of each pulse. Each pulse height should be between 0 and 1.

Pulse Step



Pulse Step Signal

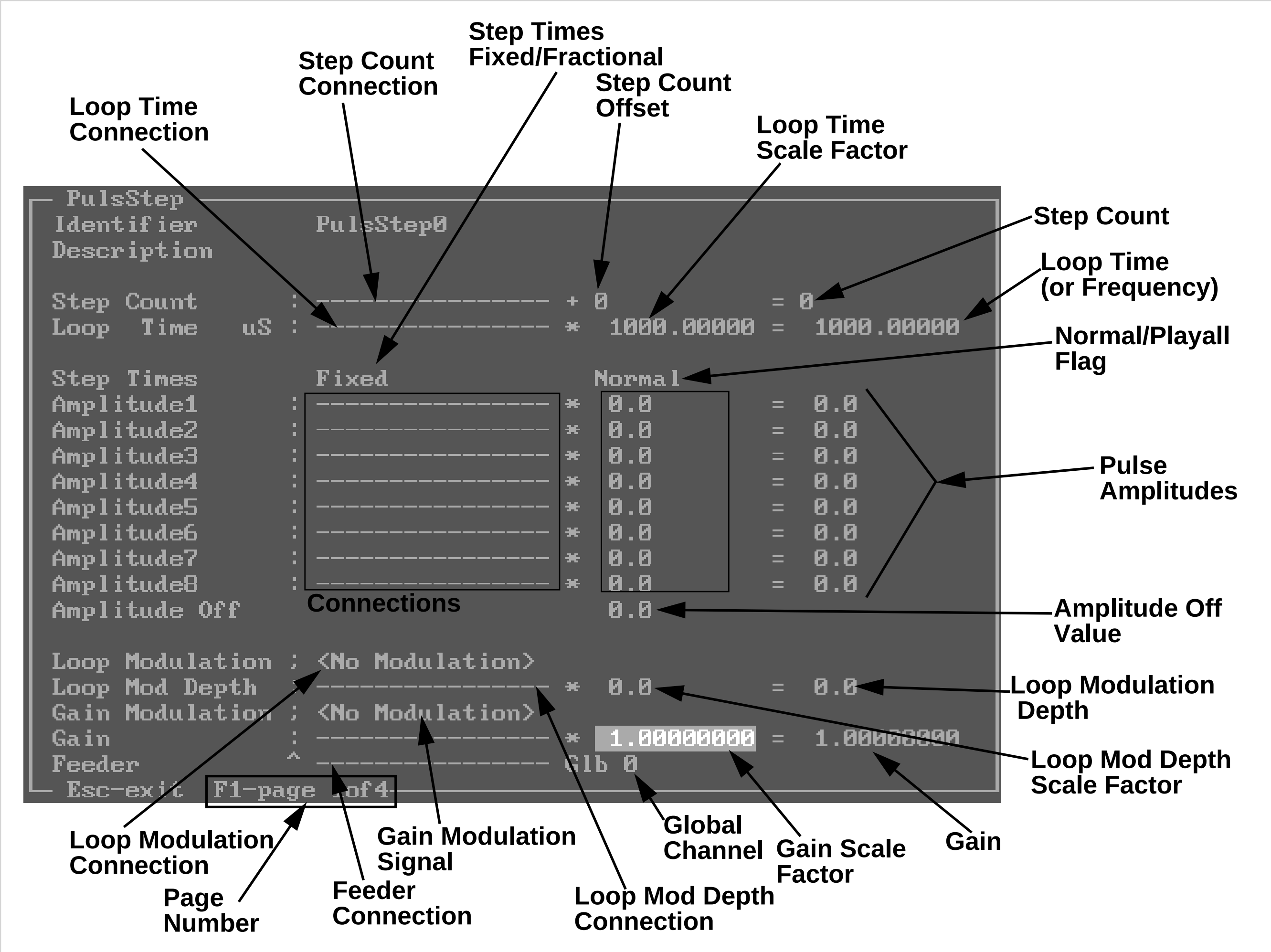
The Pulse Sequence signal is a signal object that can generate a repeating series of up to sixteen sequential pulses of arbitrary pulse width and pulse amplitude. The pulses follow immediately one after the other - there is no space between them. Typically, this signal is used to frequency or amplitude modulate other signals.

The string of pulses can be looped repeatedly with a specified loop time, or can be triggered as a “one-shot” sequence. When looped, the loop time can be modulated with another signal. If the loop time is modulated, the pulses can be shortened and lengthened with the loop time (Fractional Step times) or kept at a constant length (Fixed step times.)

At the end of the pulses, the output of the signal goes to a constant amplitude which is specified in the “amplitude off” field (i.e. the amplitude when all the pulses are off.)

In addition, the gain of the signal can be modulated by an external signal.

Pulse Step



PAGE 1 OF 4

Step Count Connection

Connection to a control object which provides the step count.

Step Count Offset

Provides an offset for the Step Count Connection. If the step count connection is empty, the offset value provides the step count.

Step Count

This number is an integer which provides the number of pulses. It must lie between 1 and 16. Setting the value to zero turns the pulse stream off, so the signal output is just the "amplitude off" value. If the pulse step signal is in One-Shot mode, then changing the Step count from zero to a non zero value will trigger the pulse stream to start.

Loop Time Connection

Connection to a control object which provides the Loop Time.

Loop Time Scale Factor

Provides a scale factor for the Loop Time Connection. If the connection is empty, the scale factor becomes the loop time.

Pulse Step

Loop Time (or Frequency)

The Loop Time (Fixed Mode) or Loop Frequency (Fractional Mode) determines the rate at which the pulses repeat themselves. If the loop time is zero, the Pulse Step object goes into One Shot Mode.

In One Shot mode, the pulses are generated only when triggered by the Step Count changing from 0 to a non-zero value. The rest of the time the output is given by the Amplitude Off value.

Step Times Fixed/Fractional

This flag determines whether the pulse durations are given in absolute times in microseconds (fixed mode) or as a fraction of the loop time (fractional mode).

When the loop time is modulated, fractional mode causes the pulses to shorten and lengthen in proportion to the loop time. In fixed mode, the pulses remain at their fixed durations when the loop time is varied. If the loop time becomes less then the sum of the pulse durations in fixed mode, the end pulses will get cut off.

Normal/Playall Flag

This flag is used in One Shot mode. In normal mode, if the step count is toggled from 0 to a nonzero value and back to zero, the pulses will stop playing immediately when the step count goes back to zero. In PlayAll mode, the pulses will finish playing to the end of the sequence.

Pulse Amplitude Connections

Connections to control objects which provide the Pulse Amplitudes.

Pulse Amplitude Scale Factors

Provide scale factors for the Pulse Amplitude Connections. If any of the connections are empty, the Scale factor becomes the Pulse amplitude.

Pulse Amplitudes

Provides the amplitudes for the pulses. See the figure (above).

Amplitude Off Value

Provides the amplitude of the signal when no pulses are being generated. This occurs either because the step count is zero or in the “dead time” at the end of a loop when there are no more pulses.

Loop Modulation Connection

Provides connection to a signal which modulates the loop time (Fixed mode) or the loop frequency (Fractional mode).

Loop Modulation Depth Connection

Provides a connection to a control object which gives the loop modulation depth.

Loop Modulation Depth Scale Factor

Provides a scale factor for the loop Modulation Depth Connection. If the connection field is empty, this field provides the loop Modulation Depth.

Loop Modulation Depth

Provides the modulation depth for the loop modulation signal. This should be between 0 and 1. A one means that if the modulation signal has a gain of 1, then the loop time (or frequency) will get modulated between zero and twice it’s normal value. A modulation depth of zero means that no loop modulation will occur.

Gain Modulation Signal

Connection to a signal which will modulated the amplitude of the pulse sequence.

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for gain control value.

Gain

Amplitude gain of the signal. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Pulse Step

Feeder Connection

Connection to a feeder, which adds the signal into the signal highway.

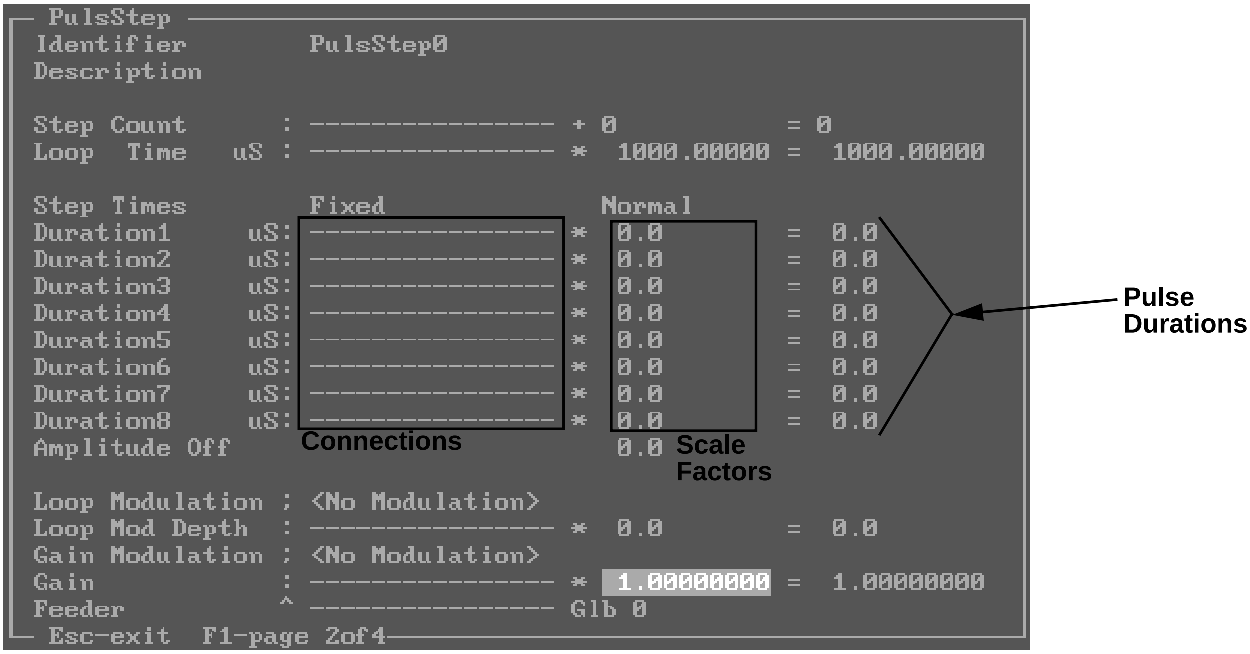
Page Number

This object has three screens. The page number indicates what screen is being displayed. F1 toggles between the screens.

Global Channel

Connection to a Global Channel. If this field is non-zero, the same signal that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

Pulse Step



PAGE 2 OF 4

NOTE: Pages 3 and 4 are identical to pages 1 and 2, except they specify Durations and Amplitudes for Pulses 9 to 16.

Pulse Duration Connections

Connections to control objects which provide the Pulse Durations.

Pulse Duration Scale Factors

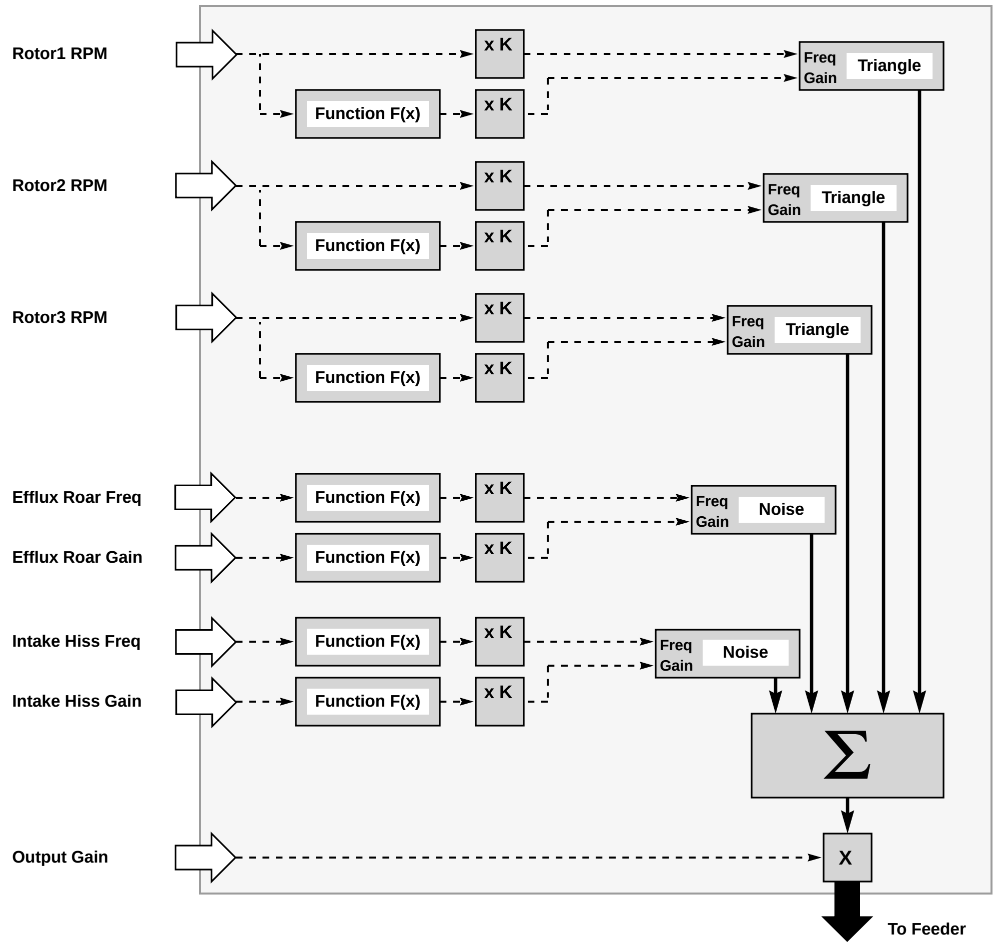
Provide scale factors for the Pulse Duration Connections. If any of the connections are empty, the Scale factor becomes the Pulse Duration.

Pulse Durations

Provides the Durations for the pulses in microseconds (Fixed Times) or as a fraction of the Sweep Time (Fractional Times). See the figure (above).

Engine Signals (Engine Package)

Engine

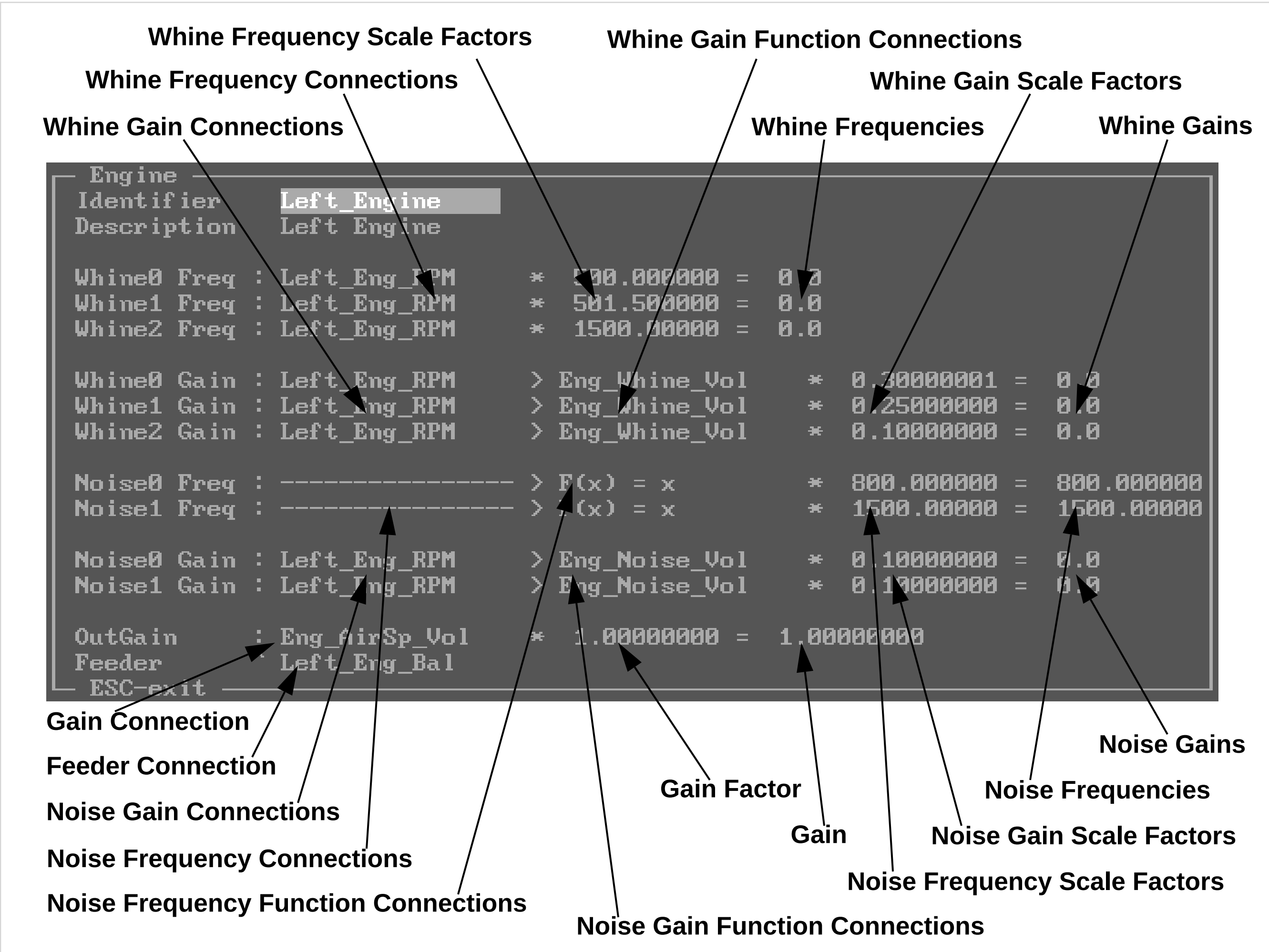


Engine

The engine signal provides a composite source suitable for simulating a single jet engine. It uses three triangle waves and two band-limited white noise sources.

Control is provided over triangle wave frequencies and amplitudes, noise bandwidth and gain, and overall gain for the composite engine sound.

Engine



Whine Frequency Connections
Provide control connections for the three whine frequencies.

Whine Frequency Scale Factors
Scaling factors for each whine frequency control.

Whine Frequencies
The frequency (in Hertz) of each of the three triangle waves used to produce the engine whines.

Whine Gain Connections
Provide control connections for the three whine gains.

Whine Gain Function Connections
Connections to a selected table or function for controlling the whine gains based upon the input control connection.

Whine Gain Scale Factors
Scaling factors for each whine gain control.

Whine Gains
The final gain factor for each of the three triangle waves used to produce the engine whines.

Noise Frequency Connections
Provide frequency control connections for the two bandwidth limited noise sources.

Noise Gain Function Connections
Connections to a selected table or function for controlling the noise roll-off frequency based upon the input control connection.

Engine

Noise Frequency Scale Factors

Scaling factors for each noise frequency control.

Noise Frequencies

The roll-off frequency (in Hertz) of each of the two bandwidth limited noise sources used to produce the engine hiss/roar.

Noise Gain Connections

Provide control connections for the two noise gains.

Noise Gain Function Connections

Connections to a selected table or function for controlling the noise gains based upon the input control connection.

Noise Gain Scale Factors

Scaling factors for each noise gain control.

Noise Gains

The final gain for each of the two noise sources.

Gain Connection

Control object connection to provide overall amplitude gain control of the composite engine sound from elsewhere in model.

Gain Scale Factor

Scaling factor for overall gain control value.

Gain

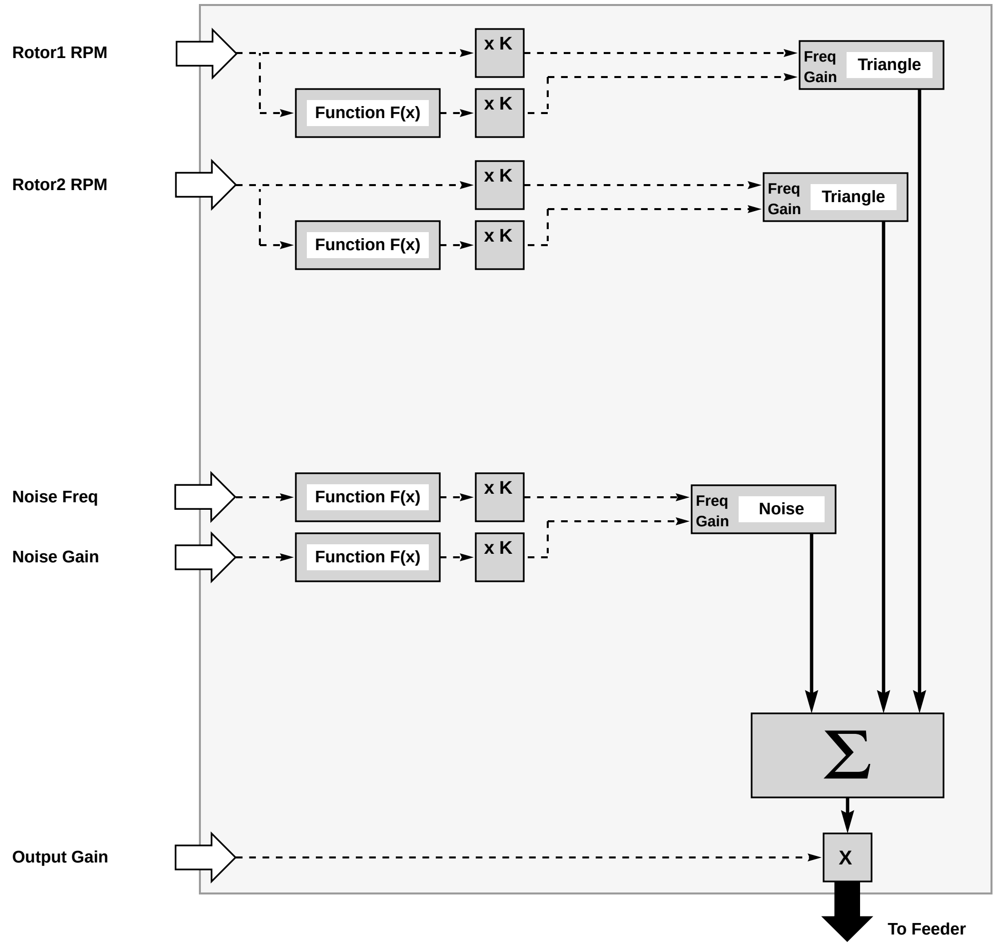
Overall amplitude gain of the engine source. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Feeder Connection

Connection to a feeder, which adds the engine sound into the signal highway.

Note: The whine frequency and gain controls are the same connections, hence the noise sources can have control of frequency and gain from separate parts of the model, whereas the whines are always frequency and gain controlled from the same input connection.

Small Engine

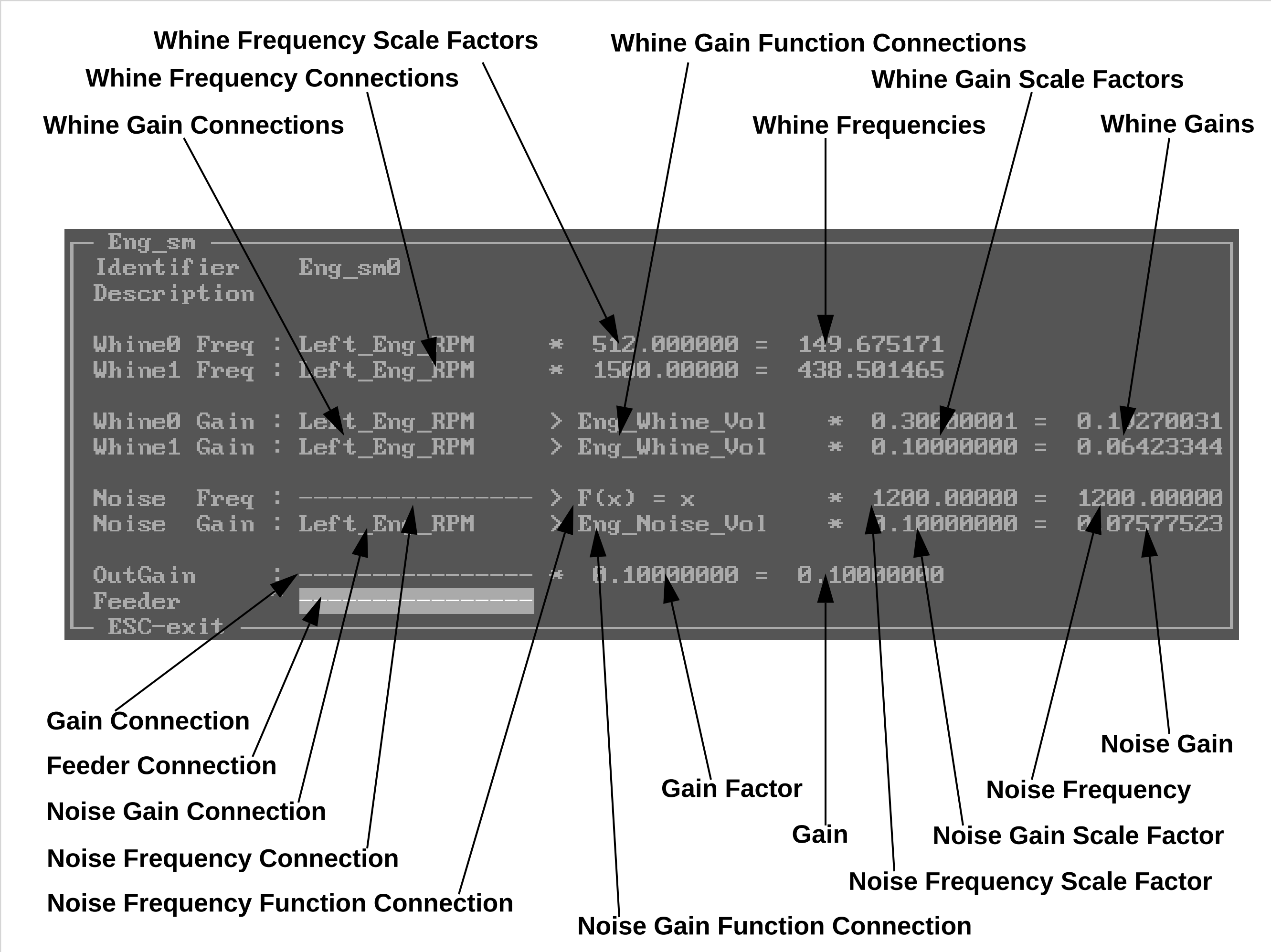


Small Engine

The small engine signal provides a simplified composite source suitable for simulating a single jet engine. It uses two triangle waves and one band-limited white noise sources.

Control is provided over triangle wave frequencies and amplitudes, noise bandwidth and gain, and overall gain for the composite engine sound.

Small Engine



Whine Frequency Connections
: Provide control connections for the three whine frequencies.

Whine Frequency Scale Factors
: Scaling factors for each whine frequency control.

Whine Frequencies
: The frequency (in Hertz) of each of the three triangle waves used to produce the engine whines.

Whine Gain Connections
: Provide control connections for the three whine gains.

Whine Gain Function Connections
: Connections to a selected table or function for controlling the whine gains based upon the input control connection.

Whine Gain Scale Factors
: Scaling factors for each whine gain control.

Whine Gains
: The final gain factor for each of the three triangle waves used to produce the engine whines.

Noise Frequency Connection
: Provide frequency control connection for the bandwidth limited noise source.

Noise Gain Function Connection
: Connection to a selected table or function for controlling the noise roll-off frequency based upon the input control connection.

Small Engine

Noise Frequency Scale Factor
Scaling factors for noise frequency control.

Noise Frequencies
The roll-off frequency (in Hertz) of the bandwidth limited noise source used to produce the engine hiss/roar.

Noise Gain Connection
Provide control connection for the two noise gain.

Noise Gain Function Connection
Connection to a selected table or function for controlling the noise gain based upon the input control connection.

Noise Gain Scale Factors
Scaling factor for noise gain control.

Noise Gains
The final gain for the noise source.

Gain Connection
Control object connection to provide overall amplitude gain control of the composite engine sound from elsewhere in model.

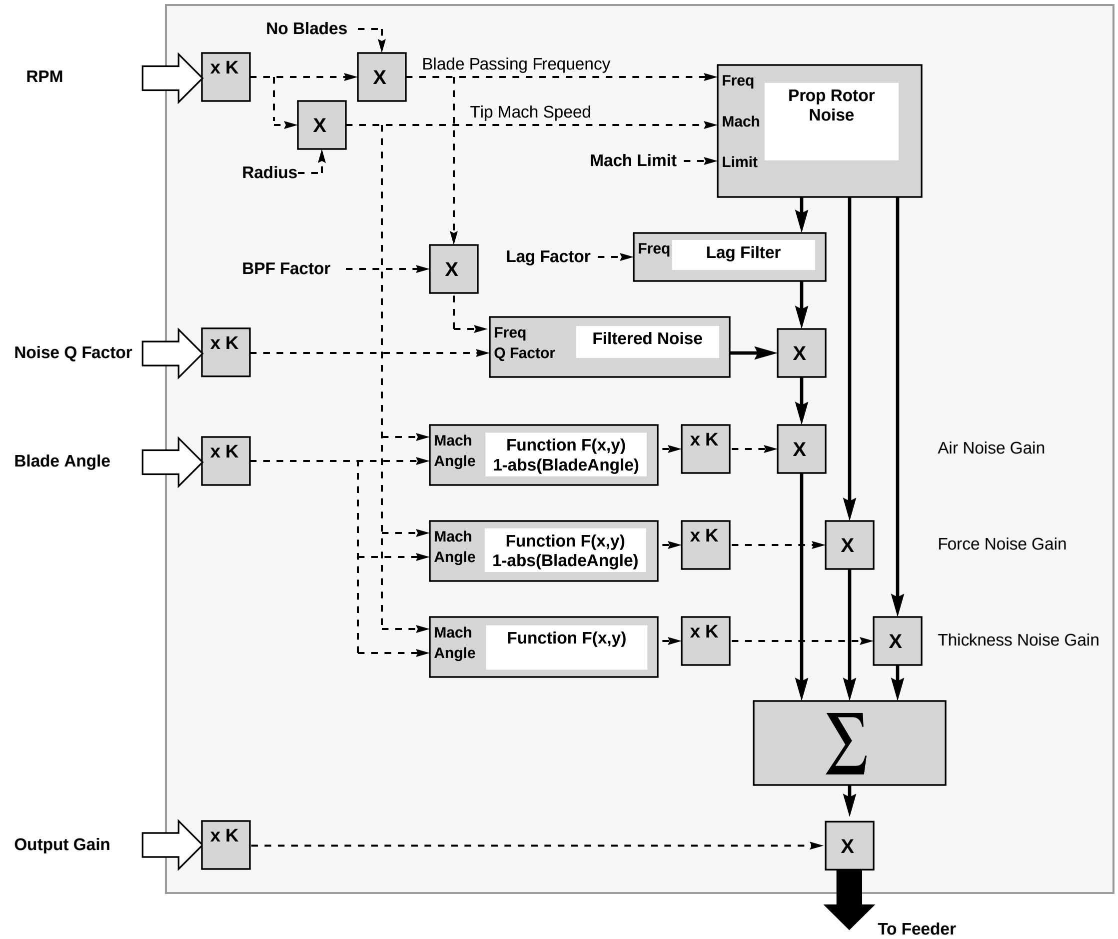
Gain Scale Factor
Scaling factor for overall gain control value.

Gain
Overall amplitude gain of the engine source. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Feeder Connection
Connection to a feeder, which adds the engine sound into the signal highway.

Note: The whine frequency and gain controls are the same connections, hence the noise sources can have control of frequency and gain from separate parts of the model, whereas the whines are always frequency and gain controlled from the same input connection.

Propeller

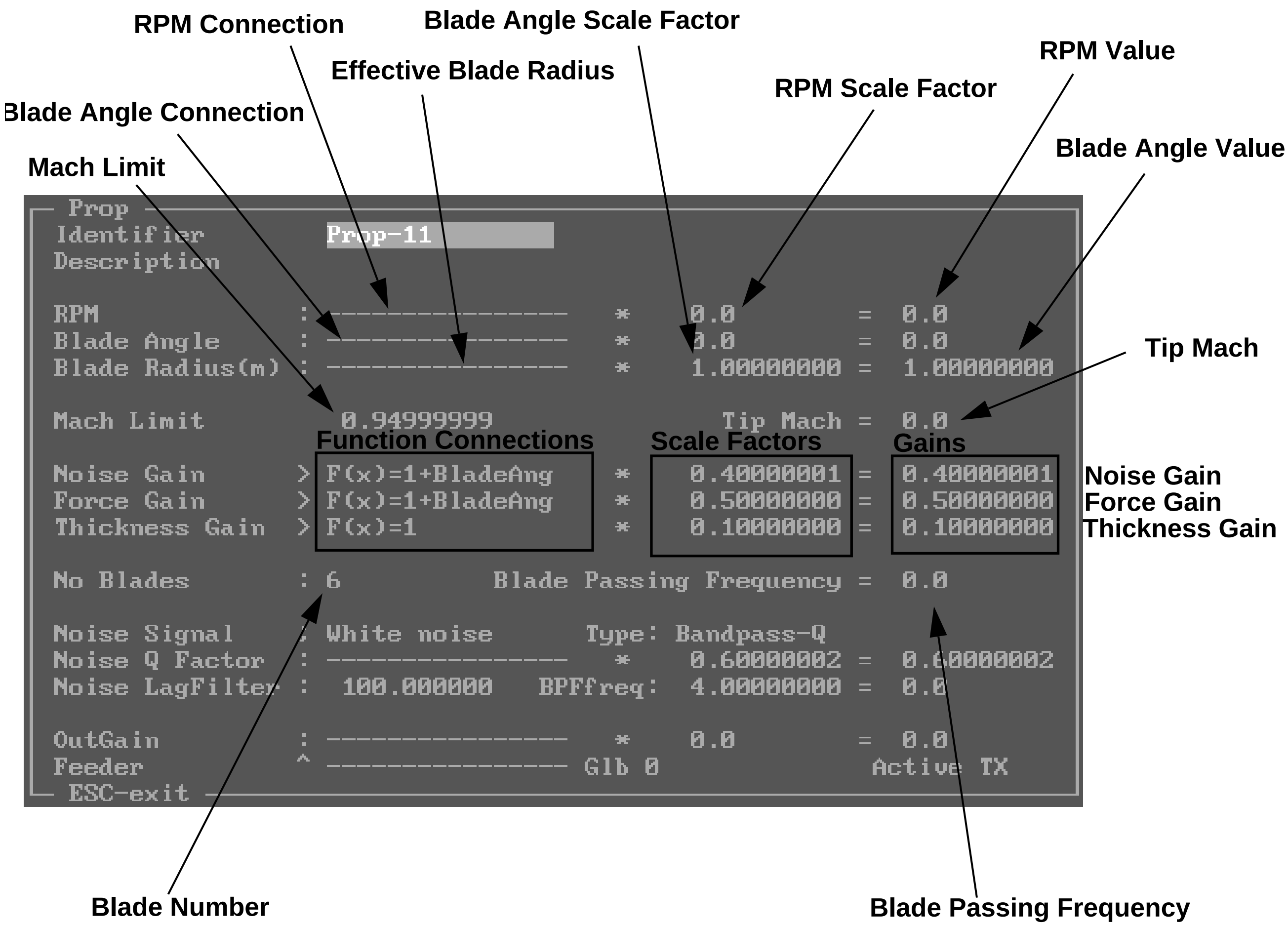


Propeller

The Propeller object provides a composite sound for a rotating propeller blade. It includes the three principal sources of noise; Air noise from the movement of air over the blades, Force noise from the impact of the blade with the air medium, and thickness noise due to the dual edge sound sources on a blade.

The overall sound can be tuned based upon blade parameters such as radius and blade count, with overall gain control based on both RPM and blade angle.

Propeller



RPM Connection

Provides control connection for the blade shaft RPM (Revolutions per Minute).

RPM Scale Factor

Scaling factors for blade RPM.

RPM Value

The frequency (in Revolutions per Minute) of the blade shaft.

Blade Angle Connection

Provides control connection for the blade angle.

Blade Angle Scale Factor

Scale factor for blade angle.

Blade Angle Value

The Blade angle value.

Noise Gain Function Connection

Connections to a selected table or function for controlling the noise gains based upon RPM and Blade Angle. The RPM is the top of stack value, while the Blade Angle is second.

Noise Gain Scale Factor

Scaling factor for noise gain.

Noise Gain

The gain for the air noise component of the blade sound. If the no function connection is made then the noise gain equals the scale factor, else it is the scale factor times the result of the selected function of RPM and Blade Angle.

Propeller

Force Gain Function Connection

Connections to a selected table or function for controlling the force gain based upon RPM and Blade Angle. The RPM is the top of stack value, while the Blade Angle is second.

Force Gain Scale Factor

Scaling factor for force gain.

Force Gain

The gain for the principle component of the blade sound due to the air force on the blade. If the no function connection is made then this gain equals the scale factor, else it is the scale factor times the result of the selected function of RPM and Blade Angle

Thickness Gain Function Connection

Connections to a selected table or function for controlling the thickness gain based upon RPM and Blade Angle. The RPM is the top of stack value, while the Blade Angle is second.

Thickness Gain Scale Factor

Scaling factor for thickness gain.

Thickness Gain

The gain for the thickness noise component of the blade sound. If the no function connection is made then the noise gain equals the scale factor, else it is the scale factor times the result of the selected function of RPM and Blade Angle. This component is the impulsive sound associated with transonic tip speeds characteristic of helicopter rotor blades, and increases significantly in volume as the tip mach speed approaches 1.0.

Tip Mach

The calculated tip mach speed, which is:

$$TipMach = \frac{2 \times \pi \times EffectiveBladeRadius}{60 \times SpeedOfSound}$$

Note: Speed of Sound is assumed to be constant for this model.

Mach Limit

A limit for the calculated tip mach speed. This keeps the maximum mach speed to a predetermined maximum. Usually between 0.95 and 0.99, depending how dominant the thickness noise is required. The sound model is not accurate above 0.99 since supersonic effects start to dominate the sound spectrum.

Effective Blade Radius

Constant used to scale tip mach speed based on blade radius (in meters).

Blade Number

Number of blades on shaft

Blade Passing Frequency

Frequency (in Hertz) of blades passing a stationary point.

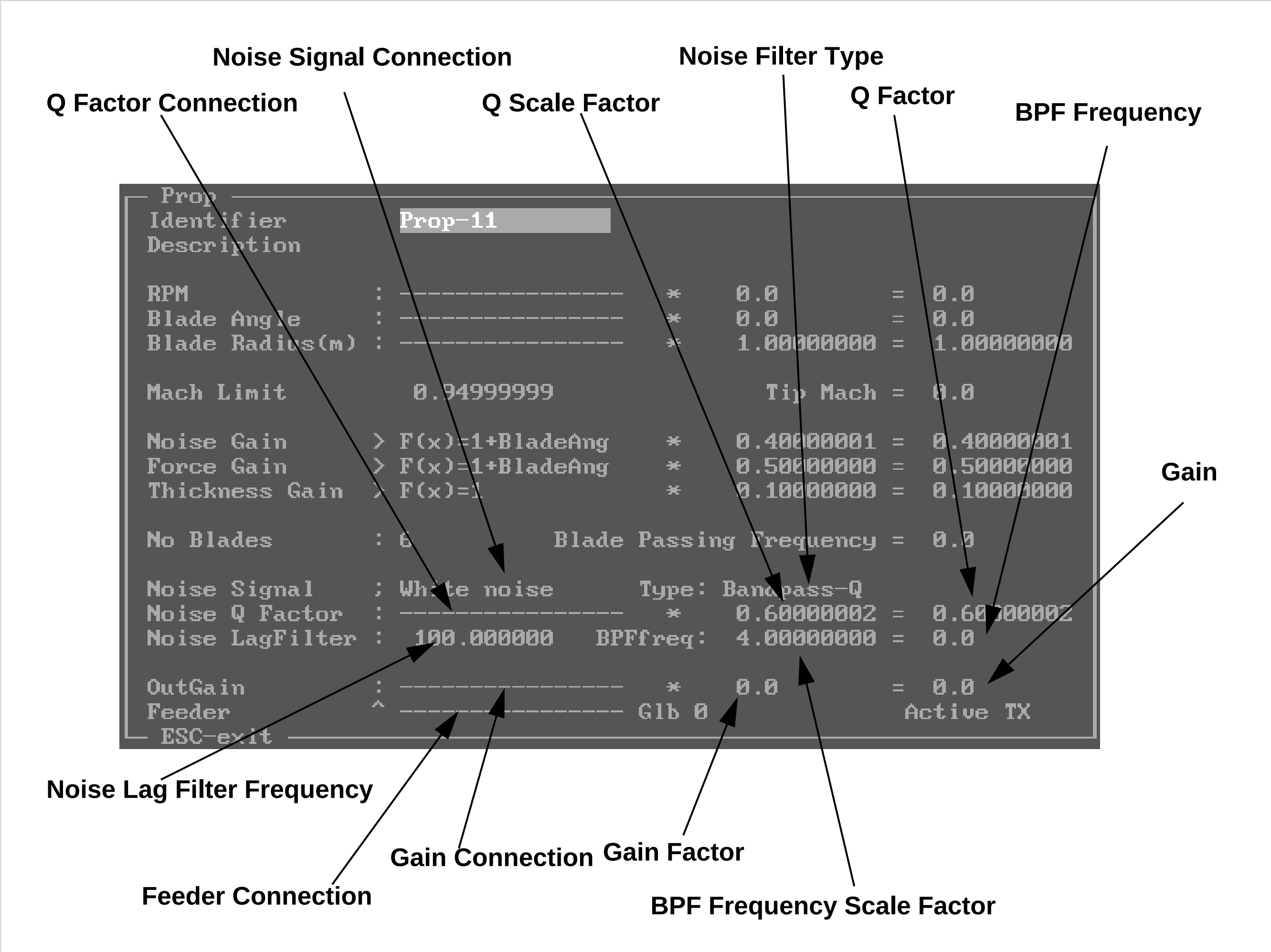
$$BPF = \frac{BladeNumber \times ShaftRPM}{60}$$

Note: The default functions for the noise and force gains are dependent on the blade angle.

$$Gain = 1 + abs(BladeAngle)$$

The effect of tip mach speed is already accounted for in the sound effect model. This default was selected since the effect and range of Blade Angle is aircraft dependent, with no clear default function derived from the underlying physics of sound production from a moving blade. The above equation provides a simple gain effect based on a linear increase in sound level as the blade moves away from a flat (zero angle) position. The thickness noise is not usually dependent on blade angle, hence a unity function has been selected for it’s default. If however it is necessary to change these overall gains the stack setup for these functions has the tip mach speed at the top and the blade angle second.

Propeller



Noise Signal Connection

Connection to the signal to be used as the air noise input to the filter. Deleting this connection selects the default internal white noise generator.

Noise Filter Type

Selects a two pole filter type from; Low-pass, Band-pass, High-pass, Low-passQ, Band-passQ, High-passQ. The latter three are amplitude adjusted such that the filter has unity gain at the roll-off frequency, and maintains this gain as the quality factor is increased. The band-pass filters have the low-pass and high-pass poles at the same roll-off frequency.

Q Factor Connection

Control object connection to provide quality factor control from elsewhere in model.

Q Scale Factor

Scaling factor for Q factor control value.

Q Factor

Quality factor for air noise filter. If the Q factor connection is blank then the Q scale factor is used as the Q value, else the Q is the scale factor times the output result of the control object.

BPF Frequency Scale Factor

Scale factor for noise filter roll-off frequency.

Propeller

BPF Frequency

Noise filter roll-off frequency, which is:.

$$Freq = ScaleFactor \times BladePassingFreq$$

Noise Lag Filter Frequency

Roll-off frequency for noise envelope lag filter (in Hertz).

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Factor

Scaling factor for gain control value.

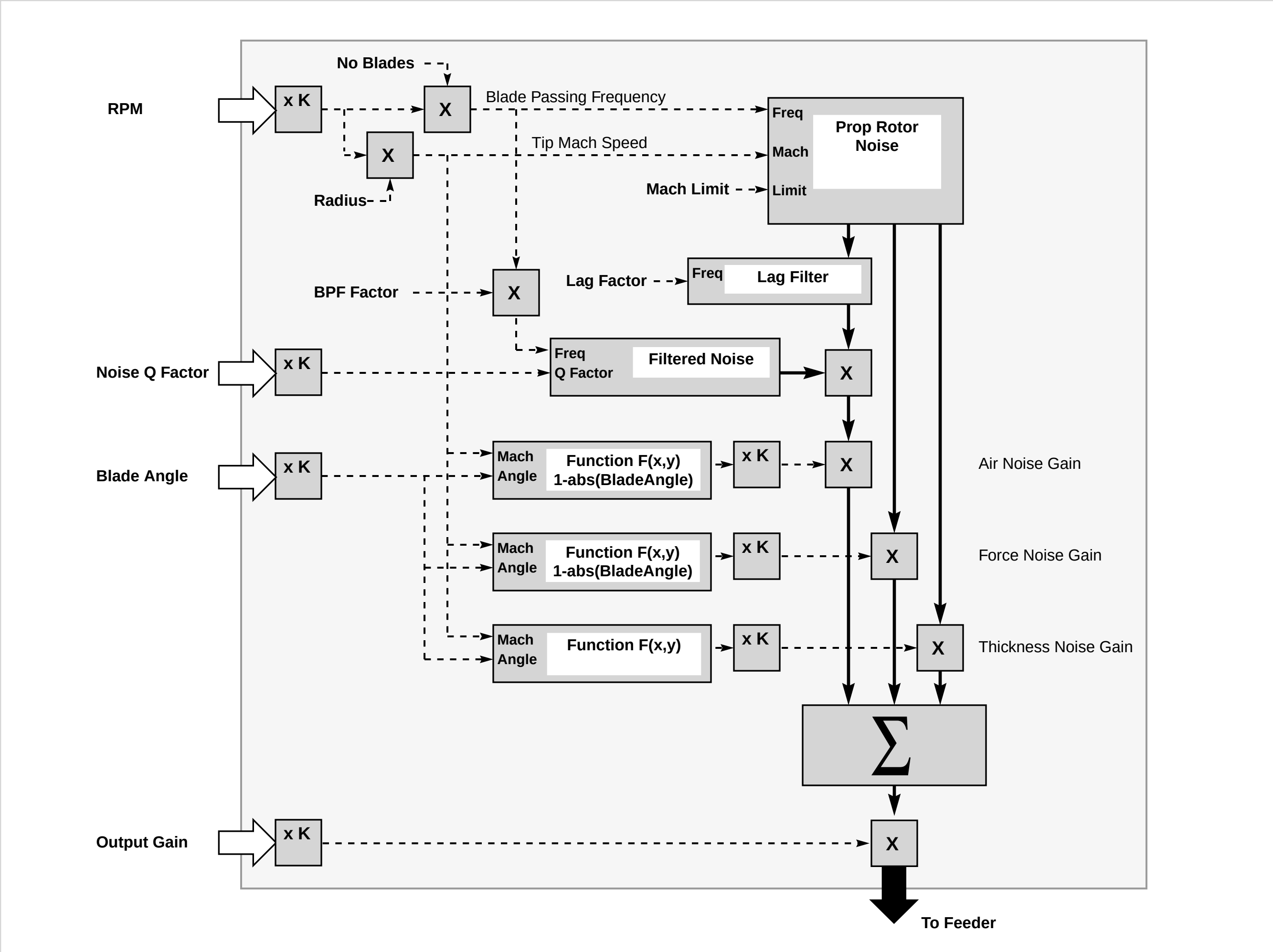
Gain

Amplitude gain of prop/rotor composite sound source. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Feeder Connection

Connection to a feeder, which adds the prop/rotor sound into the signal highway.

Rotor

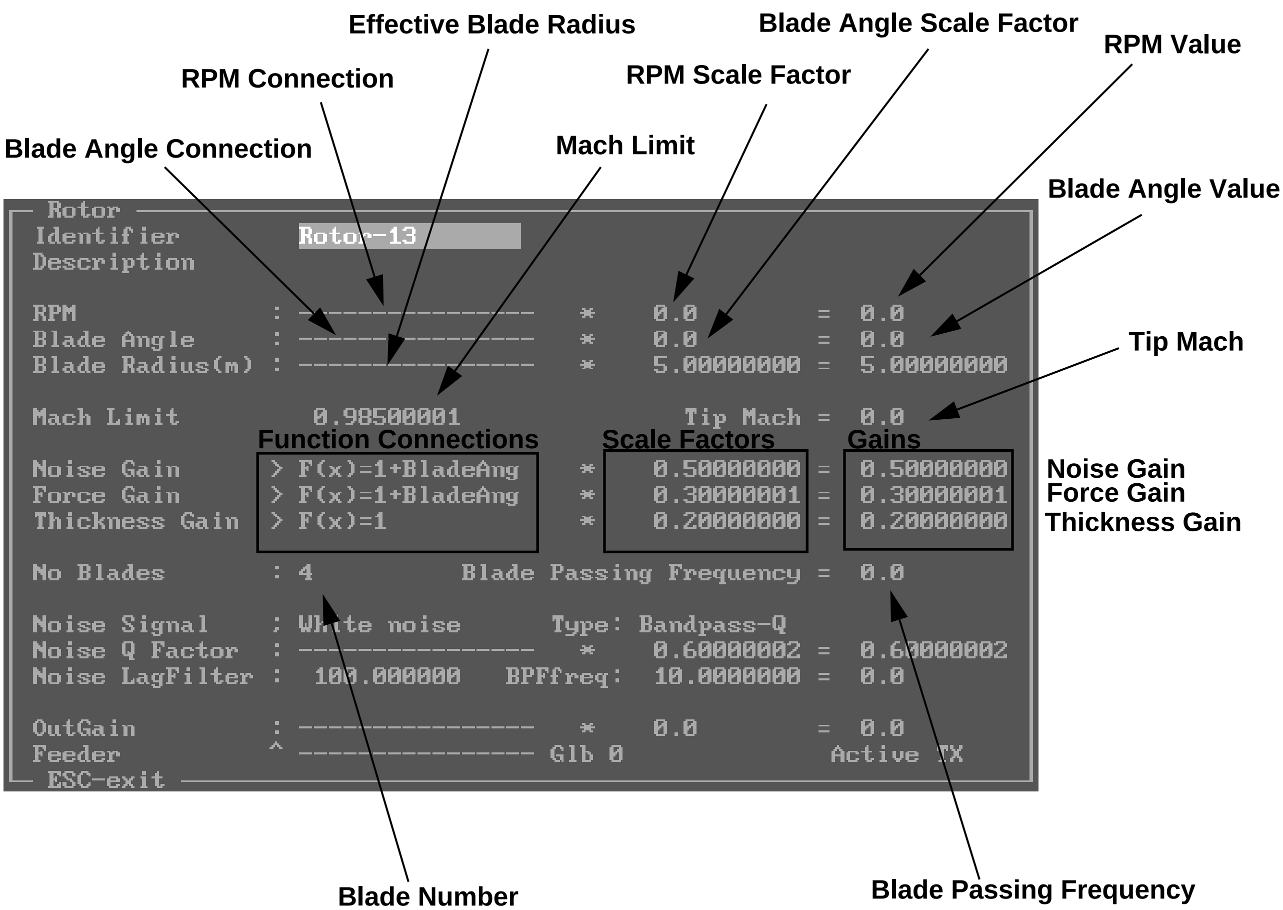


Rotor

The Rotor object provides a composite sound for a rotating helicopter blade. It includes the three principal sources of noise; Air noise from the movement of air over the blades, Force noise from the impact of the blade with the air medium, and thickness noise due to the dual edge sound sources on a blade.

The overall sound can be tuned based upon blade parameters such as radius and blade count, with overall gain control based on both RPM and blade angle.

Rotor



RPM Connection

Provides control connection for the blade shaft RPM (Revolutions per Minute).

RPM Scale Factor

Scaling factors for blade RPM.

RPM Value

The frequency (in Revolutions per Minute) of the blade shaft.

Blade Angle Connection

Provides control connection for the blade angle.

Blade Angle Scale Factor

Scale factor for blade angle.

Blade Angle Value

The Blade angle value.

Noise Gain Function Connection

Connections to a selected table or function for controlling the noise gains based upon RPM and Blade Angle. The RPM is the top of stack value, while the Blade Angle is second.

Noise Gain Scale Factor

Scaling factor for noise gain.

Noise Gain

The gain for the air noise component of the blade sound. If the no function connection is made then the noise gain equals the scale factor, else it is the scale factor times the result of the selected function of RPM and Blade Angle.

Rotor

Force Gain Function Connection

Connections to a selected table or function for controlling the force gain based upon RPM and Blade Angle. The RPM is the top of stack value, while the Blade Angle is second.

Force Gain Scale Factor

Scaling factor for force gain.

Force Gain

The gain for the principle component of the blade sound due to the air force on the blade. If the no function connection is made then this gain equals the scale factor, else it is the scale factor times the result of the selected function of RPM and Blade Angle.

Thickness Gain Function Connection

Connections to a selected table or function for controlling the thickness gain based upon RPM and Blade Angle. The RPM is the top of stack value, while the Blade Angle is second.

Thickness Gain Scale Factor

Scaling factor for thickness gain.

Thickness Gain

The gain for the thickness noise component of the blade sound. If the no function connection is made then the noise gain equals the scale factor, else it is the scale factor times the result of the selected function of RPM and Blade Angle. This component is the impulsive sound associated with transonic tip speeds characteristic of helicopter rotor blades, and increases significantly in volume as the tip mach speed approaches 1.0.

Tip Mach

The calculated tip mach speed, which is:

$$TipMach = \frac{2 \times \pi \times EffectiveBladeRadius}{60 \times SpeedOfSound}$$

Note: Speed of Sound is assumed to be constant for this model.

Mach Limit

A limit for the calculated tip mach speed. This keeps the maximum mach speed to a predetermined maximum. Usually between 0.95 and 0.99, depending how dominant the thickness noise is required. The sound model is not accurate above 0.99 since supersonic effects start to dominate the sound spectrum.

Effective Blade Radius

Constant used to scale tip mach speed based on blade radius (in meters).

Blade Number

Number of blades on shaft.

Blade Passing Frequency

Frequency (in Hertz) of blades passing a stationary point.

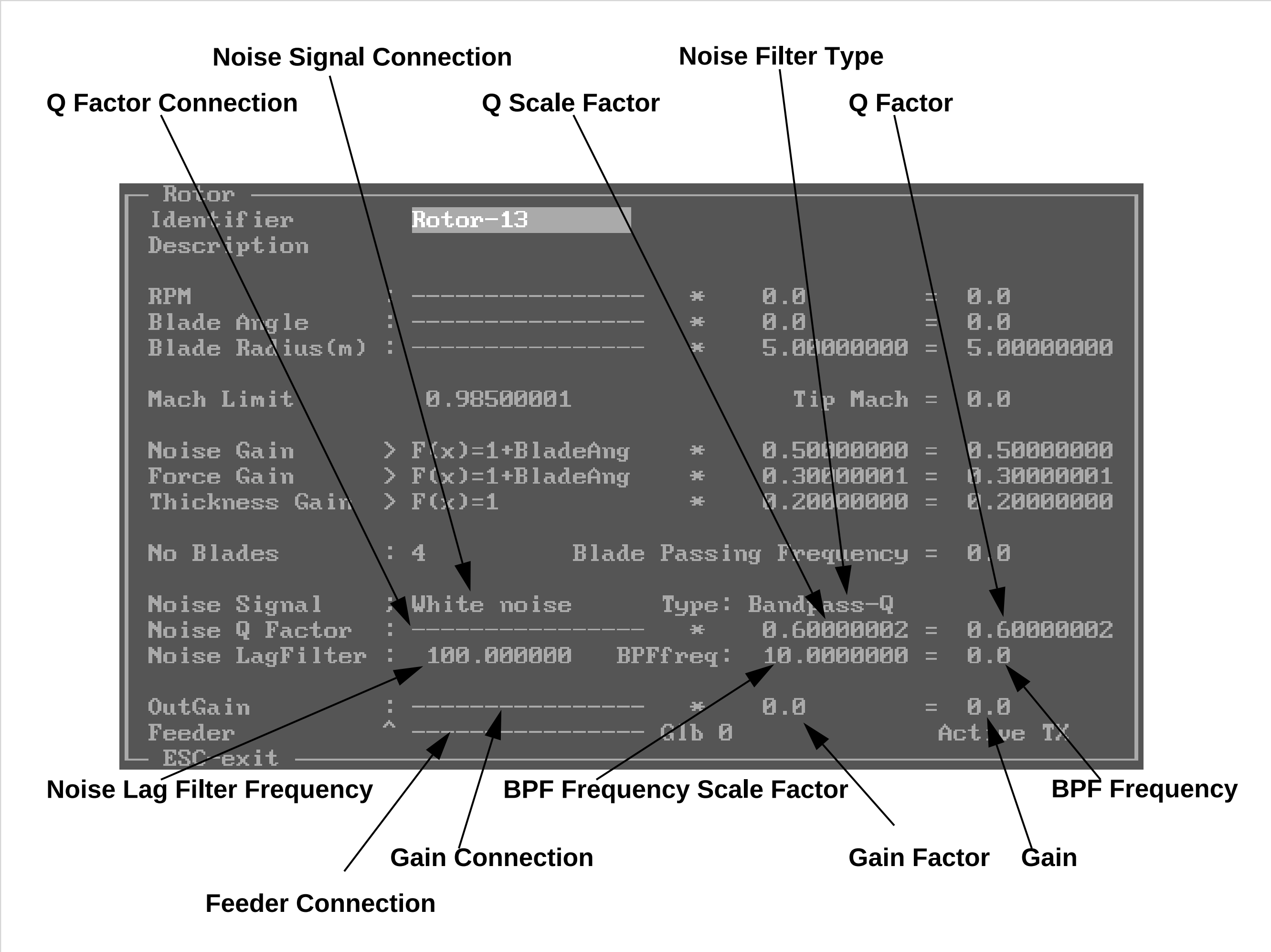
$$BPF = \frac{BladeNumber \times ShaftRPM}{60}$$

Note: The default functions for the noise and force gains are dependent on the blade angle.

$$Gain = 1 + abs(BladeAngle)$$

The effect of tip mach speed is already accounted for in the sound effect model. This default was selected since the effect and range of Blade Angle is aircraft dependent, with no clear default function derived from the underlying physics of sound production from a moving blade. The above equation provides a simple gain effect based on a linear increase in sound level as the blade moves away from a flat (zero angle) position. The thickness noise is not usually dependent on blade angle, hence a unity function has been selected for it’s default. If however it is necessary to change these overall gains the stack setup for these functions has the tip mach speed at the top and the blade angle second.

Rotor



Noise Signal Connection

Connection to the signal to be used as the air noise input to the filter. Deleting this connection selects the default internal white noise generator.

Noise Filter Type

Selects a two pole filter type from; Low-pass, Band-pass, High-pass, Low-passQ, Band-passQ, High-passQ. The latter three are amplitude adjusted such that the filter has unity gain at the roll-off frequency, and maintains this gain as the quality factor is increased. The band-pass filters have the low-pass and high-pass poles at the same roll-off frequency.

Q Factor Connection

Control object connection to provide quality factor control from elsewhere in model.

Q Scale Factor

Scaling factor for Q factor control value.

Q Factor

Quality factor for air noise filter. If the Q factor connection is blank then the Q scale factor is used as the Q value, else the Q is the scale factor times the output result of the control object.

BPF Frequency Scale Factor

Scale factor for noise filter roll-off frequency.

Rotor

BPF Frequency

Noise filter roll-off frequency, which is:.

$$Freq = ScaleFactor \times BladePas\sin gFreq$$

Noise Lag Filter Frequency

Roll-off frequency for noise envelope lag filter (in Hertz).

Gain Connection

Control object connection to provide amplitude gain control from elsewhere in model.

Gain Factor

Scaling factor for gain control value.

Gain

Amplitude gain of prop/rotor composite sound source. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Feeder Connection

Connection to a feeder, which adds the prop/rotor sound into the signal highway.

Radio Signals (Radio Package)

The radio object simulates a receiver/transmitter unit, while the receiver objects provide a simulation of radio signal reception only. Each frame, the signal list is scanned to determine which radios or receivers are tuned to the same frequency as the various transmitters (see next section), then any signals attached to the matching transmitters (including the TX part of the radio) are passed across to each radio receiver.

The radio object provides a simulation of the radio environment, which includes a discrimination for AM and FM band radios, and full background noise and signal strength effects.

Many radios can be handled at the same time, each with its own position. Positions are specified as world positions in X,Y, and Z or latitude, longitude, and altitude, with all of the range calculations being done automatically.

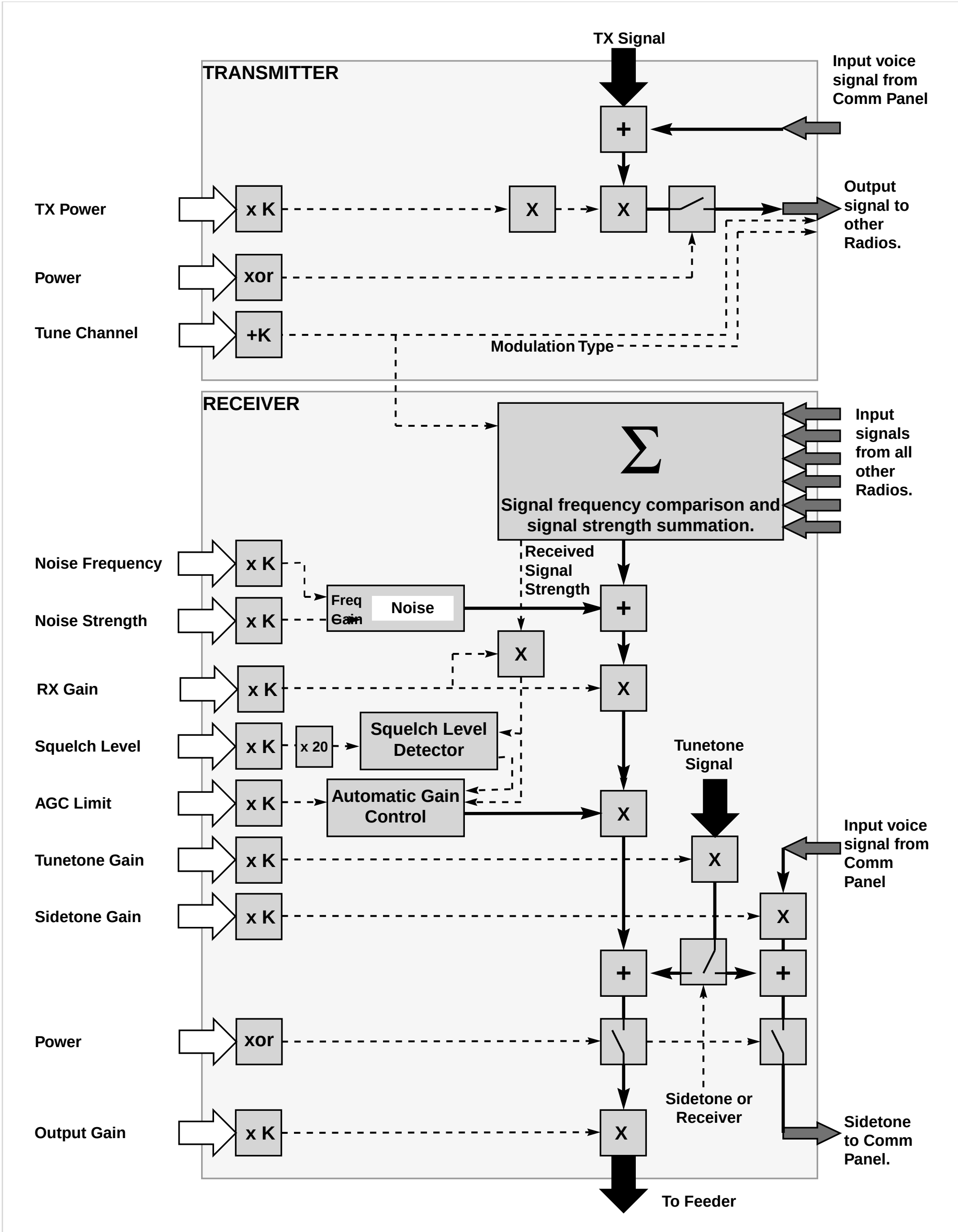
A radio can be switched between up to 8 modes, each of which can be custom tailored to provide control over parameters such as AGC, antenna gain, internal radio noise, and other parameters. The default settings for these modes are commonly used radios used, including UHF, VHF and SINCGARS. This allows the user to get started quickly, while retaining the flexibility to further fine tune the simulation.

Extending the radios for DIS mode simply involves changing a flag in the attached World Position object and assigning appropriate DIS ID numbers. (See the World Position object for more information on this.) DIS radios can also be attached to other entities on the DIS network, through the Entity object.

The DIS features of this radio includes, in addition to high fidelity simulation, compatibility with all known DIS radio implementations. In particular, type 3 and type 4 SINCGARS can both be simulated, with CCTT compatible CVSD or CECOM compatible CVSD. The radio will receive audio data with either CVSD type or mulaw automatically. Reception does not need to be limited to a single compression type.

Also, the ability to receive multiple voice streams (for AM radios) over the DIS network is supported.

Radio



Radio

The Radio Object provides a generic, high level radio simulation, which includes transmit and receive operations, frequency tuning effects, AM or FM modes, signal strength variation due to range, sidetone, background noise, squelch, and AGC.

As with real world radio equipment, the Radio Object can both transmit and receive signals. Typically, a Radio Object is attached to a Communications Selector Panel, which provides the audio for transmission (usually a microphone), and which routes the received radio audio. Transmitted signals form the output of the Radio Object and are broadcast to all other Radio Objects in the model. With Local Net, the radio

Radio

can broadcast to radios in other models running on different DSP cards in the DACS. Similarly, with VoiceNet or DIS, the radio can transmit to other radios on a network.

Every frame, Model Builder scans all of the radio transmitters to determine which radios are in tune to the receiver. The received signal strength is computed for all in tune radios based on the power of the transmitter, the antenna gains of the transmitter and receivers, and the relative world positions. (The frequency space is divided into 25KHz sections. Frequencies in the same band are considered the same, while frequencies outside the range are considered different. There is no interference between bands.)

If frequency hopping or encryption is enabled, the parameters of the transmitter and receiver are compared to see if the audio can be received. (In frequency hopping mode, the frequency field is ignored. The frequency is implied in the selected hopset).

If multiple transmitters are broadcasting on the same frequency, Model Builder will do one of two things. For AM signals, the received RF power will be combined and the received audio will be a sum of the transmitted signals in proportion to their signal strength. For FM signals, only the strongest received signal will be included.

Once the received power is determined, the RF signal/noise ratio is calculated. The noise level is determined by thermal noise, internal radio noise, and other parameters which can be set in the radio object. (The default values are set to give some common generic radios, and can be adjusted.) The signal/noise ratio is then compared to the squelch level. If the ratio is less than the squelch level, the signal will not be received. Setting the squelch to zero disables the squelch.

After the signal is determined to be received, the signal power and noise power are multiplied by the AGC. This simulates the AGC operation in a real radio, so even with high signal to noise ratio, a signal will be very low if the maximum AGC value is set low. Additionally, when the squelch is off, the maximum AGC will determine the background noise when no signal is being received.

Once the RF signal/noise is computed and Model Builder determines the audio is to be received, it computes the audio signal to noise ratio. This is based on the simulated RF signal/noise ratio, the major and minor modulation type, noise bandwidth of the radio, whether the voice is analog or digital, and other parameters.

The received audio is then routed out. If the radio is attached to one or more communications select panels, they receive the audio. Additionally, if the output feeder in the radio is specified, the audio is sent there as well. (This allows monitoring of a particular radio.) If the radio is connected to an intercom bus (by plugging the intercom bus signal into the transmit signal connection of the radio), the received sound is also put onto the intercom bus.

The radio object can transmit as well as receive, just like a real radio. When the radio receives audio to transmit (either through the transmit signal connection or from a communications select panel), reception is cut off and the radio transmits.

Radio

There are also receiver and transmitter objects that are subsets of the radio object. The transmitter will only transmit and the receiver will only receive, but otherwise they function exactly the same way as the full radio described here.

NOTES:

1. For frequencies between 1 and 100,000, no background noise or signal attenuation effects are simulated. These frequencies provide a clear channel of communication, regardless of transmission power, world position, etc. Frequencies in this range need to be exactly the same in order to be received.

2. Any radio put at the world position 0,0,0 (that is, the center of the earth) will receive any radios at the same frequency without any signal loss or occulting. This allows the model to include a radio that monitors a particular radio band, without regard to position or transmit power.

3. Although the radio object has a “Full Duplex” flag to allow simultaneous transmission and reception, this is only meant to be used in the “Intercom Mode”, described in the intercom section of the manual.

If full duplex operation is needed in other situations, separate transmitters and receivers should be used.

NOTE FOR BEGINNERS:

The ASTi simulated radio, like a real radio, is a complex thing with many parameters. When you first get started, these are the fields you should pay attention to (you can leave the rest to the default settings):

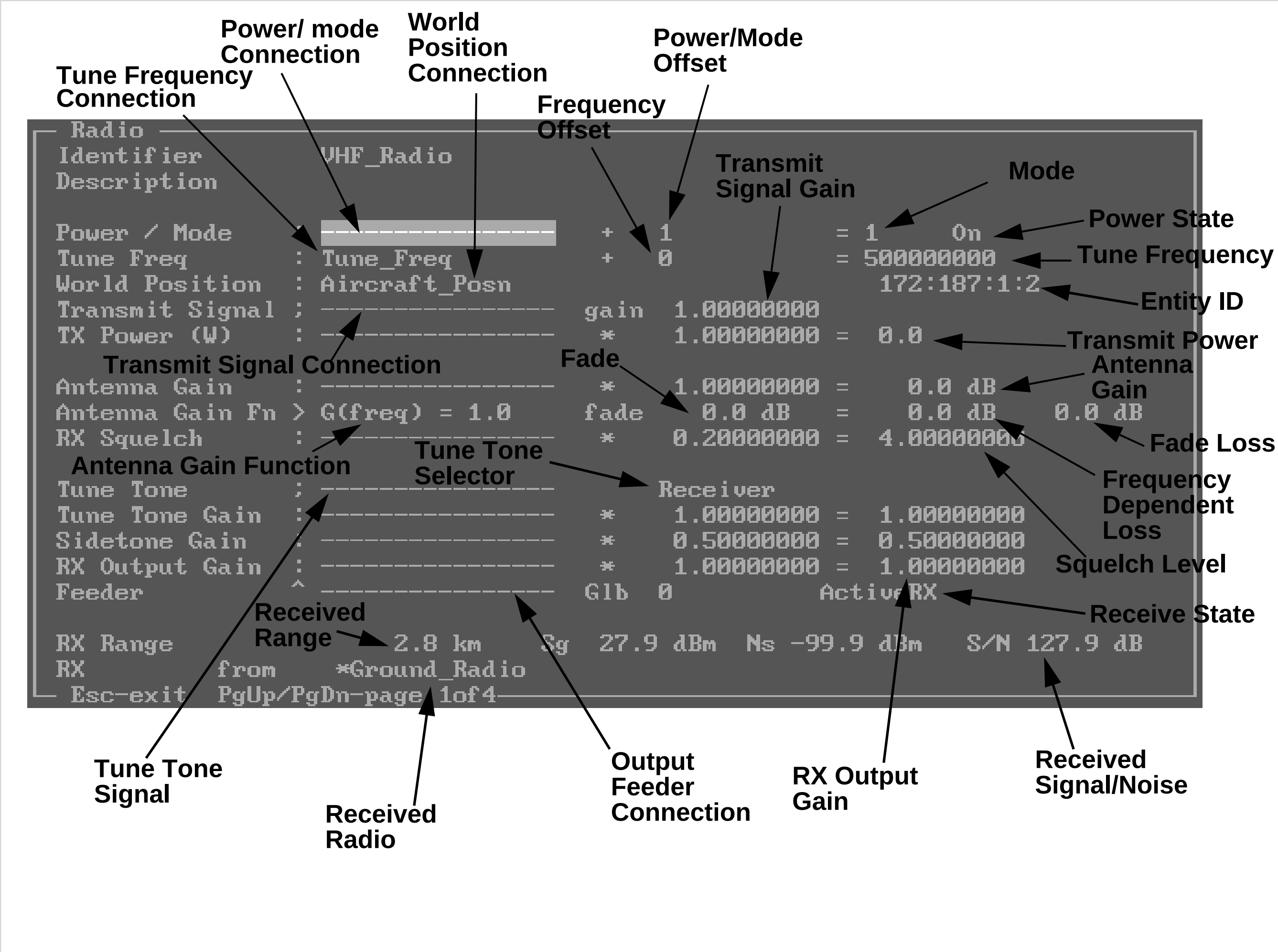
POWER/MODE - should be set from 1 to 8 to select one of the radio modes.

TUNE FREQUENCY - should be set to some non-zero number. Remember the frequency is in Hertz.

WORLD POSITION CONNECTION- The radio must be connected to a world position control object, so it has a position to do ranging calculations with.

The above three parameters are the minimum that need to be set to get a radio working. The radio will also need to be attached to a Communications Select Panel.

Radio



Tune Frequency Connection

Connection to a control object which provides the current radio tune frequency, in Hz.

Tune Frequency Offset

Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the tune frequency connection. All frequencies are in Hz.

Tune Frequency

Frequency value in Hz used for comparison to determine which radios are connected to each other.

A zero value of frequency disables both transmission and reception, providing an easy equivalent to a power-off condition. Any frequency below 100,000 hz will give a clear channel of communication (that is, the ranging effects are ignored).

Frequencies above 100 KHz are divided into bands of 25000 hz. Transmissions in the same band are assumed to have the same frequency - there is no adjacent band interference. This feature allows floating point numbers to be used in the frequency without having to worry about rounding effects.

Radio

Power/ Mode Connection

Provides a connection to a control object that selects the operating mode of the radio. The radio has eight modes of operation, and each mode can be configured by the user. A mode value of zero powers off the radio.

Power/Mode Offset

Provides an offset for the Power/Mode connection.

Power State

Indicates whether the radio is powered on or off. A powered off radio will not send or receive any audio.

The power state is set to off by selecting a mode of 0 or a tune frequency of 0.

Mode

Indicates which of the eight operating modes the radio is in. The mode characteristics are set on pages three and four of the radio object.

World Position Connection

Connection to a control object that specifies the radio's world position, or attaches it to a DIS entity. The world position attachment also supplies the site, host, and entity information required by DIS if DIS is being used.

Squelch Level

When the received RF signal/noise ratio is less than the squelch value given in this field, the AGC gain is set to zero, providing the normal background noise suppression. To disable the squelch, set the squelch level to zero.

The squelch shown is given in dB's and is the squelch level connection times the squelch level scale factor time 20.

Transmit Power

Indicates the transmission power of the radio, in watts. The total power is the power specified in the TX Power connection, reduced by the power losses specified in the individual radio mode and the antenna gain.

Antenna Gain

Provides a linea gain for the power of the transmitted and received "radio signals". This field simulates the size and radiative efficiency of the antenna. Note that all modeled antennae are isotropic.

Fade

Provides a random loss in received and transmitted RF power from 0 to the specified amount, in dB. The loss varies with time in a random way that mimics the loss found in measurements of antenna gain variation on actual aircraft due to multipath interference with the plane's skin.

Antenna Gain Function

Provides a connection to a function which specifies the antenna gain as a function of frequency. The normal antenna gain as a function of frequency for a fixed length antenna is already built into the model. This can be used to modify it, or to block out some portions of the radio spectrum.

This gain value is linear.

Received Signal/Noise

Indicates the received signal to noise ratio of the radio signal currently being received.

Receive State

If on, this indicates that the radio is receiving a signal. An "Active TX" appears when the radio is transmitting.

Received Radio

Indicates the name of the radio or radios being received.

Radio

Received Range

Indicates the distance from the radio to the transmitter it is receiving, in kilometers. If the reception stops, the last received range will be displayed.

Frequency Dependent Loss

Indicates the power loss, in dB, from the antenna gain function.

Fade Loss

Indicates the power loss, in dB, from the Fade AND from the Antenna gain function.

Transmit Signal Connection

Provides a connection to an audio transmit signal.

This can be used in three ways.

1. If a simple sound is attached, such as a sine wave or recorded playsound, the radio will transmit that sound.

2. If an intercom bus object is attached to the transmit signal, any sound from the intercom bus will be transmitted on the radio. Additionally, any sound received by the radio will be put on the intercom bus. This happens automatically when the intercom bus is attached - no other connections need to be made. This provides a mechanism for several operators to share a single radio. Because the radio is half duplex, when external sound is put on the intercom bus, reception stops and the radio will transmit. (For the special case of using the radio in "Intercom Mode", see the section on intercoms.)

3. A filter object can be inserted into the transmit signal. When this filter is then attached to a comm panel, the comm panel will be attached to the radio, but the received sound will be filtered by the filter object. This can be used to simulate radios that have a "hollow" sound, or other sound distortion.

Transmit Signal Gain

Provides a volume gain for any audio signal attached to the transmit signal connection.

TuneTone Signal

The signal connection for any tones that need to be mixed into the receiver or sidetone signal path, such as local tuning tones, or weapon system cues.

TuneTone Selector

Selector for tunetone signal path, either Receiver or Sidetone, when the former the tunetone is added to the received signal path, when the latter it is added to the sidetone signal.

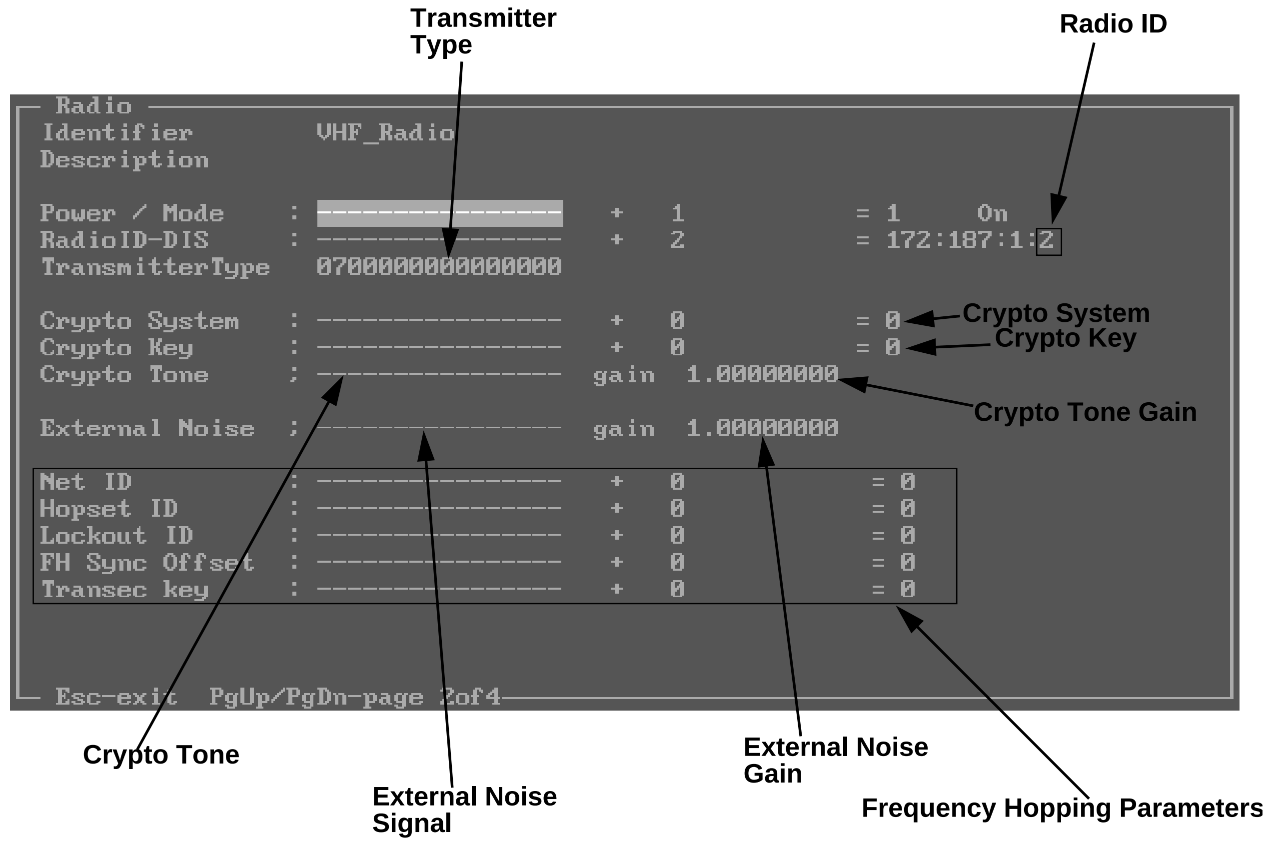
RX Output Gain

Overall volume gain for received sound.

Output Feeder Connection

Connection to a feeder, which provides a mechanism for outputting the radio onto the signal highway. This field can be left blank if the signal is to be picked up by a comm panel feeder object, or a selector or mixer.

Radio



Radio ID

ID number of transmitter that must be assigned to use DIS, Voice Net, or Local Net. The host number, site number, entity number and Radio ID must form a unique set. All numbers but the radio ID are set through the World Position Connection.

If a radio ID is not assigned (that is, if it would be set to zero), the software will automatically assign a value that is unique. Assigning a value through the connection field or the offset field will override the Model Builder assigned value. (In previous versions of model builder, this feature is not provided and the radio ID must be set explicitly for each radio.)

Transmitter Type

The transmitter type is a field required for all DIS radios. It has no effect on the operation of the radio.

Crypto System/Crypto Key

The Crypto System and Crypto Key numbers turn the radio into an encrypted radio. The crypto system only works with radio modes that have a 'digital' voice type (see below). For analog modes, the numbers in yellow will always be zero, regardless of the values in the connection or offset fields.

Radio

If the radio does have a digital voice type, than in addition to being in tune and in range, the Crypto Key and Crypto System fields need to match for the receiver to hear the transmitter. If they don't match, the signal plugged into the Crypto Tone field will be heard. This allows a sound to be heard for an improperly decrypted signal.

The exception to the matching criteria is that if the transmitter is set with a Crypto System or Crypto Key of zero, then the receiver will be able to hear the audio properly, regardless of its Crypto Key and Crypto System fields.

Crypto Tone

The Crypto Tone is played to the output feeder whenever the radio is receiving from a transmitter that is in tune and in range, but has different Crypto System or Crypto key values.

Crypto Tone Gain

Provides an audio gain for the Crypto tone.

External Noise

Provides a connection to a signal which can be used for the audio noise heard on the radio if the standard white noise is not wanted. In order for the connected signal to be used as audio noise, the NoiseBW (on page four of the radio object) must be set to zero. Otherwise, the standard white audio noise will be used.

External Noise Gain

An audio gain for the external noise signal. This value will also increase the volume of the standard white noise, as well.

Frequency Hopping Parameters

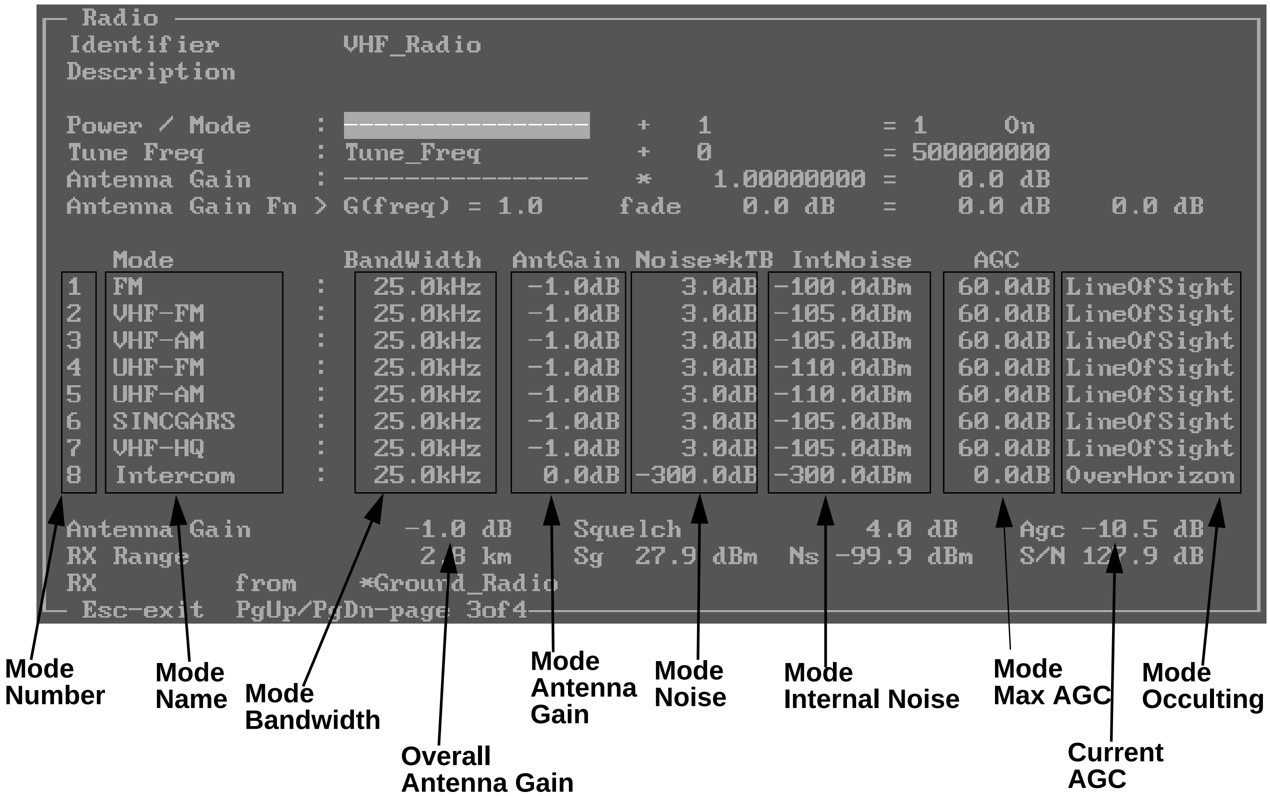
The frequency hopping parameters provide a way to simulate the frequency hopping mode of a SINCGARS radio. Frequency hopping will only be enabled if the Mode System type is either “scgar” (SINCGARS type 3) or “SCGAR” (SINCGARS type 4). Otherwise all the frequency hopping parameters will be set to zero.

When the radio Mode System Type is a Sincgars radio, and the Net ID is nonzero, the word “FreqHop” will appear on this screen to indicate the radio is in frequency hopping mode.

In frequency hopping mode, the tune frequency is ignored. (Actually this is not quite true - it is still used to determine the frequency dependent part of the antenna gain and the RF noise level). Radios will be considered in tune only if all of the Frequency Hopping parameters are the same.

The exception to this is that a value of zero, either in the transmitter or the receiver, acts as a wildcard. If either the transmitter or the receiver has one of the parameters (except Net ID) set to zero, the software will consider them to match.

Radio



Mode Number

The radio object has eight user configurable operating modes, which are determined by the Power/Mode connection. The mode number is the number that the mode selection refers to.

Mode Name

This field provides a name for the operating mode. The mode name has no effect on the simulation.

Mode Bandwidth

The mode bandwidth is used to determine the amount of audio noise mixed into the received audio, based on the simulated bandwidth of the radio band. This parameter does NOT affect the "in tune" calculation, which is based on clearly demarcated bands of 25 Khz.

Mode Antenna Gain

Gives an antenna gain for the operating mode. This value is in dB.

Overall Antenna Gain

Indicates the total antenna gain, in dB, computed from the mode antenna gain and the overall antenna gain specified on page on of the radio object.

Mode Noise

Gives the simulated thermal noise for the radio mode. This affects the simulated RF noise.

Radio

Mode Internal Noise

Gives the simulated internal noise for the radio receiver for the mode. This affects the simulated RF noise.

Mode Max AGC

Determines the maximum AGC for the radio operating mode. The AGC is set based on the simulated received signal strength.

Current AGC

Indicates the current value of the AGC. This is determined by the simulated received signal strength and the Mod Max AGC value.

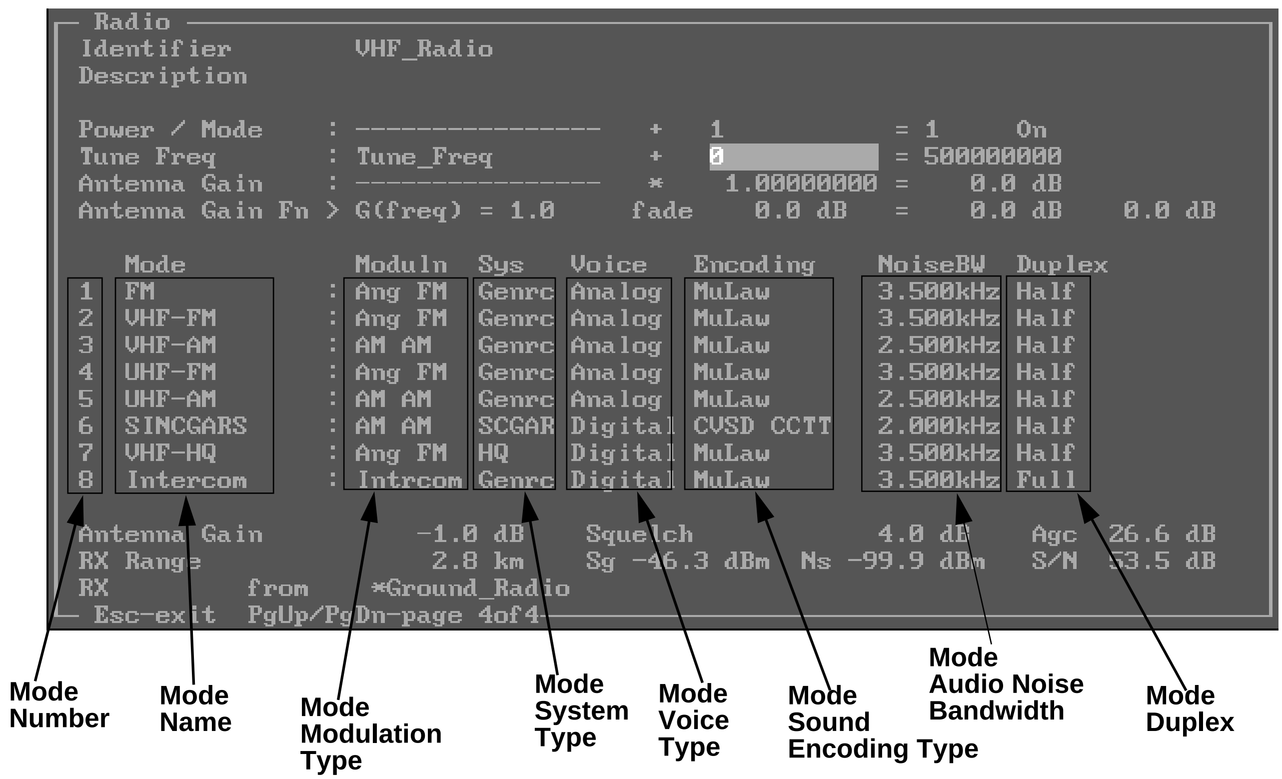
Mode Occulting

Determines whether the radio mode will receive a transmitter that is below the horizon, or whether that receiver would be occulted.

The overhorizon determination is based on the world positions, and the smooth ellipsoidal earth model WGS84.

For local terrain occulting, see the App Note entitled “Using the Terrain Database Interface”.

Radio



Mode Number

Described on the previous page.

Mode Name

Described on the previous page.

Mode Modulation Type

Indicates the modulation type of the radio mode. There are three major modulation types - FM, AM, and Intercom. In addition, there are several minor modulation types.

The FM modulation type receives only the audio stream of the signal with the strongest received RF power. The AM modulation type will mix the audio from multiple transmitters that are tuned to the same frequency. The audio volume of each signal will be mixed in proportion to the received RF power.

The intercom modulation type should only be used to attach intercom buses over DIS. It provides communication with no ranging or signal loss. The method for attaching intercom buses over DIS using this feature is described in the “Intercoms” chapter of this manual.

The minor modulation types affect the audio noise of the received signal - the same RF signal/noise ratio will give different audio signal/noise ratios, based on the minor and major modulation type.

Radio

A radio in AM mode will only receive signals from other AM radios, and an FM radio will only receive other FM radios. The minor modulation types are not matched, however.

If an AM transmission is received by an FM receiver, or vice versa, the received signal will add to the noise power of the receiver and degrade the RF signal/noise ratio.

Mode System Type

Describes the radio type. “SCGAR” is a type 4 SINCGARS radio, and “scgar” is a type 3 SINCGARS radio. They are both provided to allow compatibility with other systems. The other system types (Generic, HQ, HQII) have no effect on the simulation, but are included in the DIS transmitter PDU.

Mode Voice Type

Determines whether the voice is being sent as analog or digital.

Analog voice will have audio noise mixed in at the received end, based on the received RF signal/noise ratio and the modulation type.

Digital voice will be heard with no noise if the signal is received, and not be heard at all if it is not. The radio must be set to digital voice to use the crypto features on page two of the radio object.

There is a third voice type called “No Audio”. It is used in DIS simulations. It will cause the radio to broadcast DIS transmitter PDU packets for the radio, but not any signal PDU packets.

Mode Sound Encoding Type.

For broadcasts over DIS, the analog sound can be compressed using either Mulaw, CECOM compatible CVSD, or CCTT compatible CVSD. This field affects broadcast only - received audio will be decoded based on the audio type. Thus, a receiver can receive audio that is compressed either through mulaw or through one of the two CVSD types.

Mode Audio Noise Bandwidth

This affects the bandwidth of the audio noise heard on the receiver. The noise heard is bandwidth limited white noise.

If this field is set to zero, the audio noise heard will come from the External Noise signal connection on page two of the radio object.

Mode Duplex

This should be left to half duplex for any radio mode except the intercom mode.

Receiver Signals (Radio and Nav-Aids Packages)

The receiver signals contain several receiver objects. The receiver objects provide a simulation of radio signal reception only. One receiver has a beat frequency oscillator capability. Each frame the signal list is scanned to determine which radios or receivers are tuned to the same frequency as the various transmitters (see next section), then any signals attached to the transmitters (including the TX part of the radio) are passed across to each radio receiver. This provides a simulation of the radio environment, which includes a discrimination for AM and FM band radios, and full background noise and signal strength effects.

The DME receiver and TACAN receiver function identically, the only difference being that the TACAN and DME receivers can be tuned automatically to the standard TACAN and DME frequencies.

The Receiver object is the same as the radio object with all the fields relating to transmission removed. This will not be described here. For a description of how the reception works, see the section on the Radio object.

There is one special receiver object (the receive signal PDU object) which receives sound streams from a DIS or VoiceNet network without regard to frequency, world position, or other effects.

Receiver

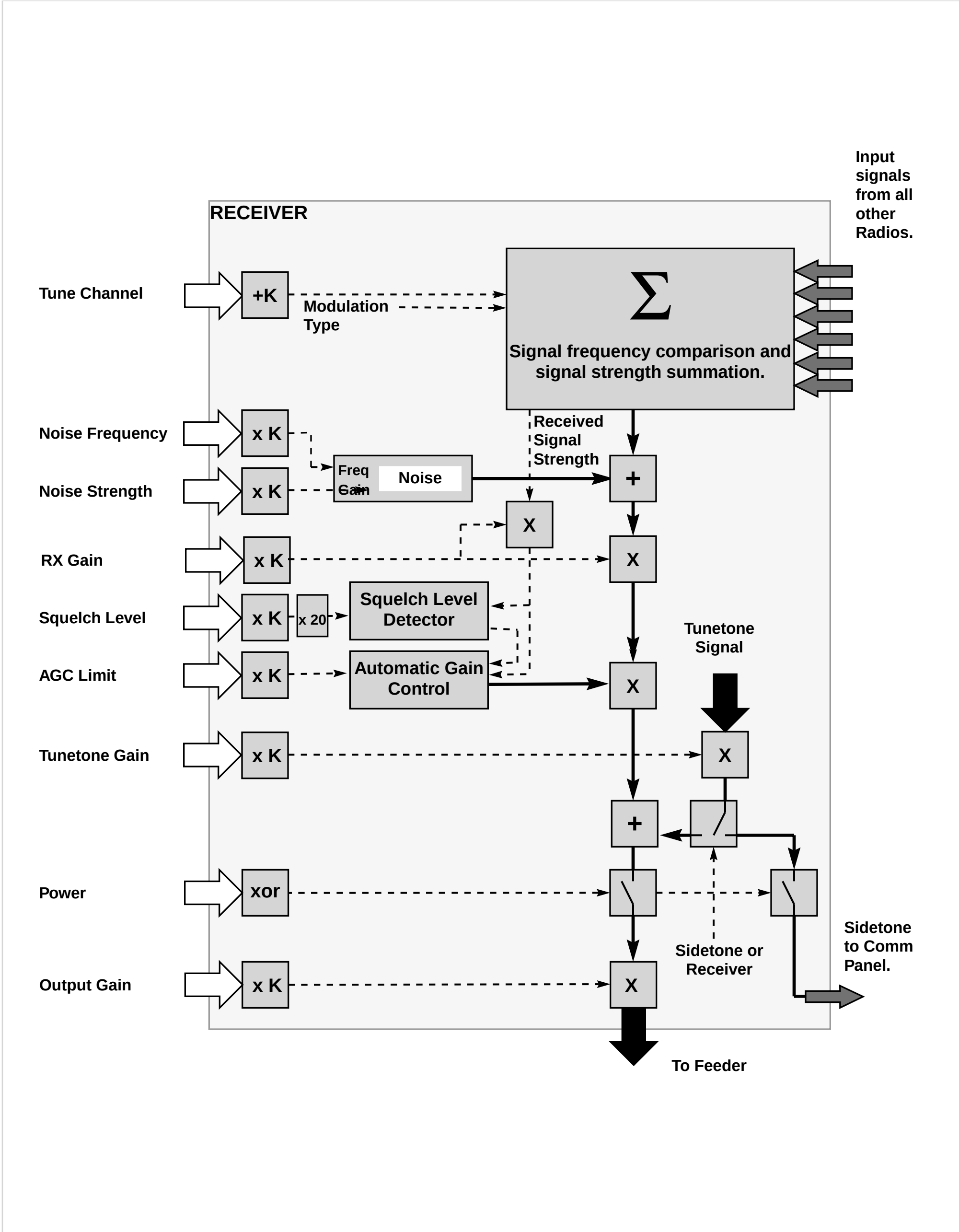
No Picture

Receiver

The Receiver Object provides a generic, high level radio receiver simulation, which includes frequency tuning effects, AM or FM bands, signal strength variation due to range, sidetone, background noise, squelch, and AGC.

The receiver object is the same as the Radio object, with the fields relating to transmission removed. For a description of the fields in the receiver object, refer to the section on the Radio object.

DME Receiver / TACAN Receiver



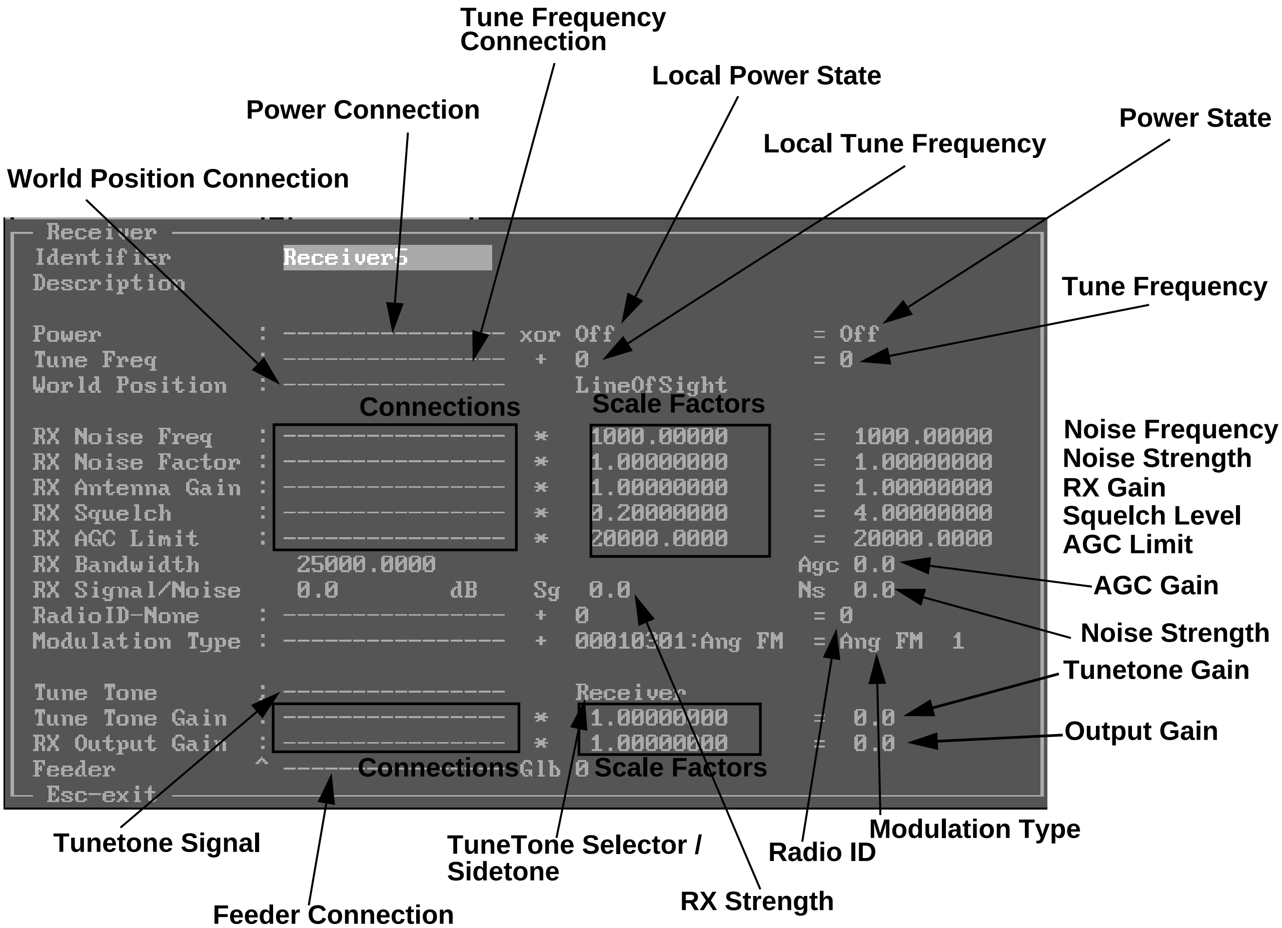
Receiver

The Receiver Object provides a generic, high level radio receiver simulation, which includes frequency tuning effects, AM or FM bands, signal strength variation due to range, sidetone, background noise, squelch, and AGC.

It functions identically to the receiver part of the Radio object, described on the previous pages.

For tune channels from 1 to 10000 Hz, all background noise, power loss, interference and occulting effects are ignored to provide clear channels of communication.

DME Receiver / TACAN Receiver



Tune Frequency Connection

Connection to a control object which provides the current radio tune frequency in Hz.

Local Tune Frequency

Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the tune frequency connection. All frequencies are in Hz.

Tune Frequency

Frequency value in Hz used for comparison to determine which radios are connected to each other. Note: A zero value of frequency disables reception, providing an easy equivalent to a power-off condition.

A tune frequency of 1 to 10000 causes the model to ignore occulting, power loss, background noise and other modeled radio effects to provide a clear channel of communication.

For the TACAN and DME receivers, tune channels 1 through 126 correspond to the standard TACAN and DME frequencies of 102.5 to 115 MHz.

Power Connection

Control object connection to provide power control from elsewhere in model.

DME Receiver / TACAN Receiver

Local Power State

Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Power State

The receiver power state, when on and the tune frequency is non-zero the receiver is active, when off it is inactive, simulating a power off condition.

Noise Frequency Connection

Background noise frequency connection field for host control of noise roll-off.

Noise Frequency Scale Factor

Scaling factor, or local value for background noise frequency.

Noise Frequency

Roll-off frequency for background noise used on receive side of radio

Noise Strength Connection

Background noise strength connection field for host control of noise level.

Noise Strength Scale Factor

Scaling factor, or local value for background noise strength.

Noise Strength

Signal strength of background noise on radio. This should range between 0 and 1.0. It is treated as another signal to be added with the other received signals, and hence goes through the squelch and AGC circuits correctly.

RX Gain Connection

Receiver antenna gain connection.

RX Gain Scale Factor

Receiver antenna gain factor.

RX Gain

Gain of the antenna, can be used to simulate receiver failures.

Squelch Level Connection

Squelch level connection for host control of adjustable squelch in dB.

Squelch Level Scale Factor

Local value or connection scale factor for squelch level in dB.

Squelch Level

The squelch level is the squelch connection times the Scale factor times 20.

When the total received signal strength (including background noise) is less than the squelch level then the AGC gain is set to zero, providing the normal background noise suppression. To disable the squelch, set the scale factor to zero.

AGC Limit Connection

Host connection for adjustment of the AGC limit

AGC Limit Scale Factor

Local value or connection scale factor for AGC limit.

AGC Limit

The Automatic Gain Control (AGC) limit. In radio equipment, low signal strength incoming signals are boosted by an AGC circuit to provide even reception of all signals regardless of signal strength. The AGC gain is calculated as 1/(total incoming signal strength), this limit is the maximum AGC gain allowable. The AGC gain is zero if the signal strength is below the squelch level, else it ranges between 1.0 and this AGC limit, which should be kept above 1.0. To disable the AGC set the local AGC limit to 1.0.

RX Strength

The current total signal strength for all matching radio sources.

AGC Gain

The current AGC gain value limited between 1.0 and the AGC limit.

TuneTone Signal

The signal connection for any tones that need to be mixed into the receiver or sidetone signal path, such as local tuning tones, or weapon system cues.

DME Receiver / TACAN Receiver

TuneTone Selector

Selector for tunetone signal path, either Receiver or Sidetone, when the former the tunetone is added to the received signal path, when the latter it is added to the sidetone signal.

TuneTone Gain Connection

Host control of tunetone gain.

TuneTone Gain Scale Factor

Local value or connection scale factor for tunetone gain.

TuneTone Gain

Gain for the tunetone signal

Output Gain Connection

Overall gain connection for host control of receiver volume or on/off capability.

Output Gain Scale Factor

Local value or connection scale factor for output overall gain.

Output Gain

Overall gain for receiver. Usually this is unused, since it is simpler to use the tune frequency = 0 to simulate a power off condition, and the receiver volume level would normally be provided by the Communications Selector Panel on the feeder list.

Feeder Connection

Connection to a feeder, which provides a mechanism for outputting the receiver onto the signal highway. This field can be left blank if the signal is to be picked up by a Communications Selector Panel.

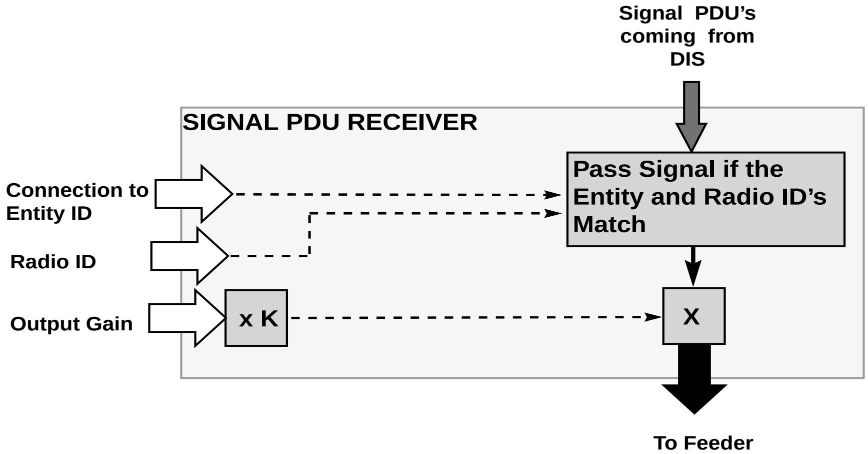
World Position Connection

Connection to a control object that specifies the receiver's world position, or attaches it to a DIS entity.

Radio ID

ID number of receiver that must be assigned to use DIS, Voice Net, or Local Net. The host number, site number, entity number and Radio ID must form a unique set. All numbers but the radio ID are set through the World Position Connection.

Signal PDU Receiver

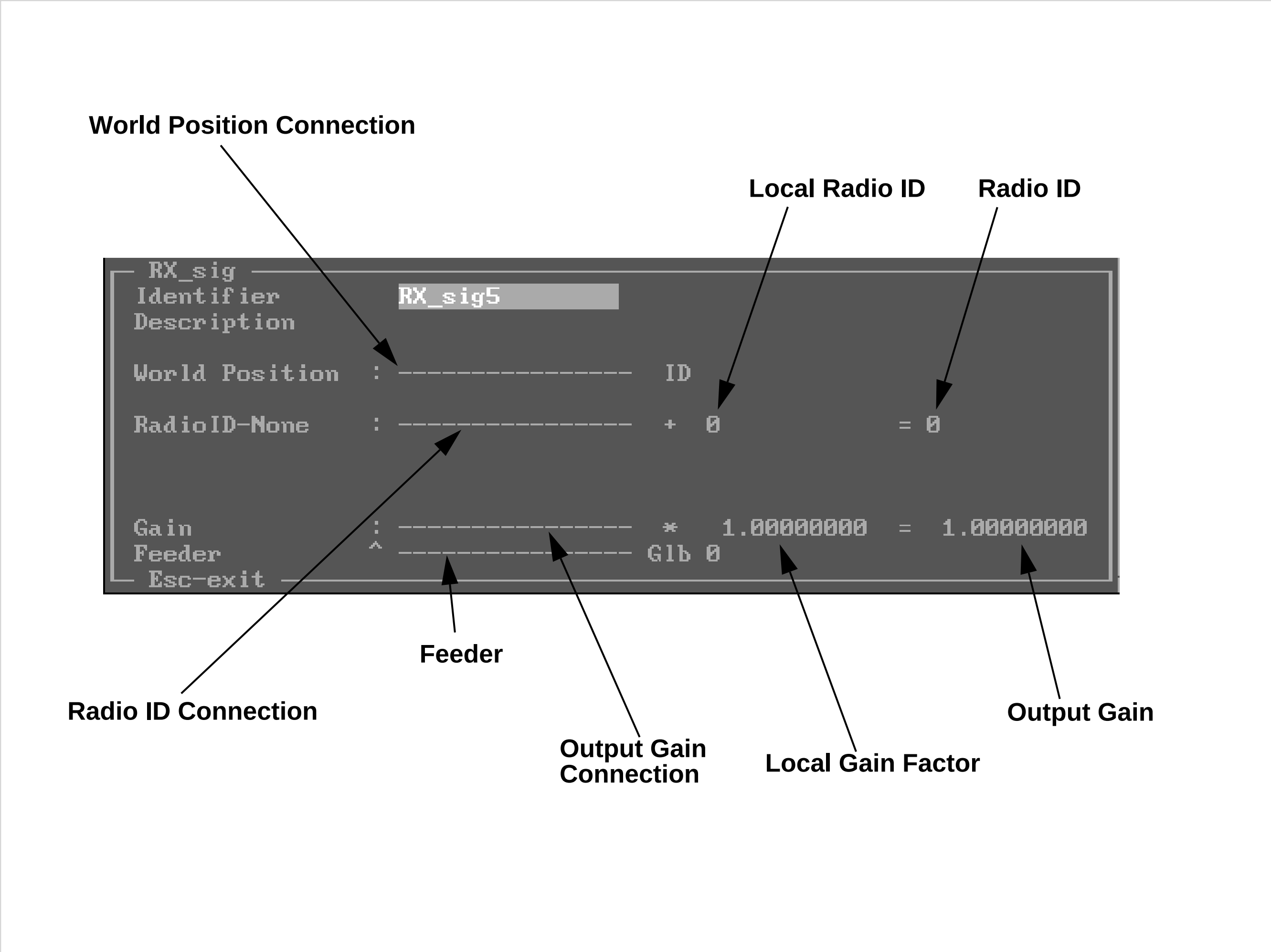


Signal PDU Receiver

The signal PDU receiver picks up the signal PDU's coming in over Ethernet on DIS and sends it to an output feeder. It will pick up the signal PDU's from a given radio, without regard to transmitter frequency, world position, or any other of the normal radio receiving parameters.

It is primarily for use as a debugging tool when setting up a DIS system.

Signal PDU Receiver



World Position Connection

Provides connection to an Entity control object, which specifies the entity whose sound stream you want. For example, to hear a radio transmitting over DIS with entity ID 234:123:7, this field would attach to an entity control object which specified the ID 234:123:7.

Radio ID Connection

Provides a connection to a control object that will give an offset for the Radio ID.

Local Radio ID

Provides either an offset for the Radio ID number, or, if the connection field is empty, provides the Radio ID.

Radio ID

The ID number of the Radio on the Entity that you want to pick up.

Output Gain Connection

Connection to a control object which determines the Output Gain.

Local Gain Factor

Local value or connection scale factor for output overall gain.

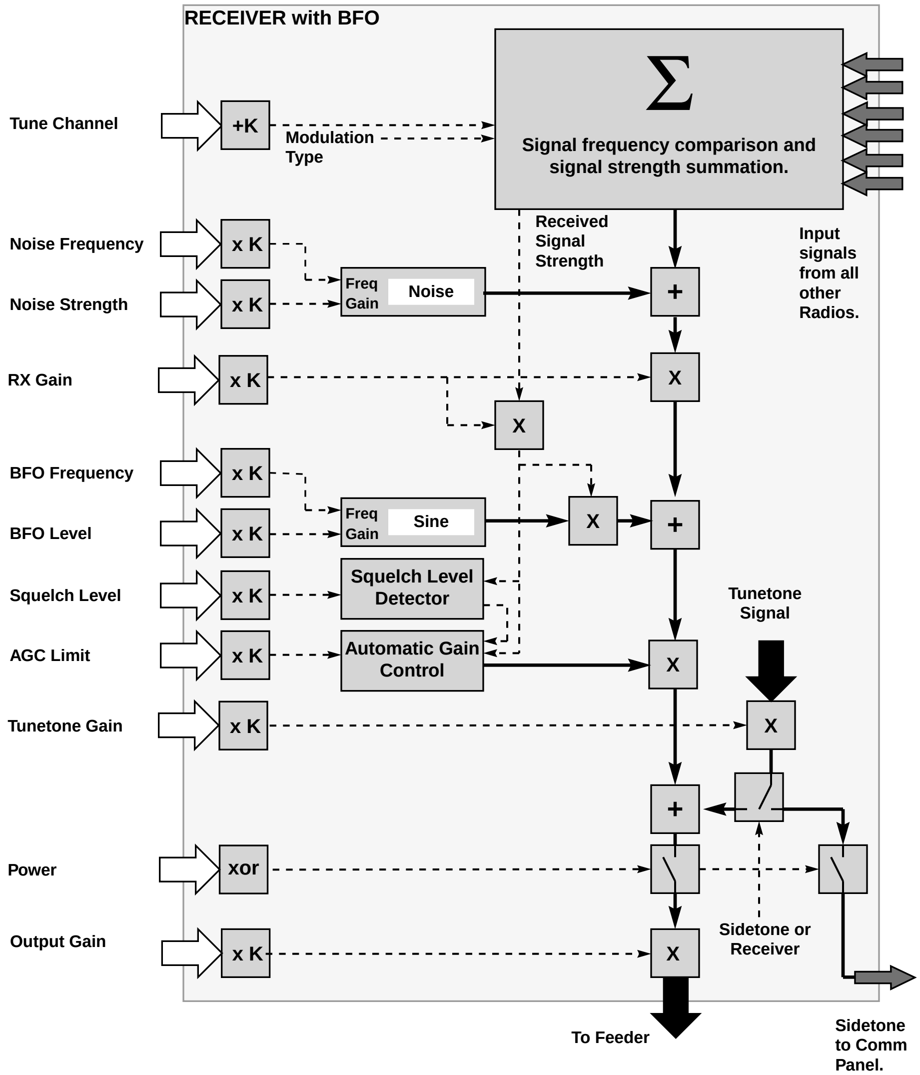
Output Gain

Overall gain for receiver.

Feeder Connection

Connection to a feeder, which provides a mechanism for outputting the received signal onto the signal highway.

Receiver with BFO

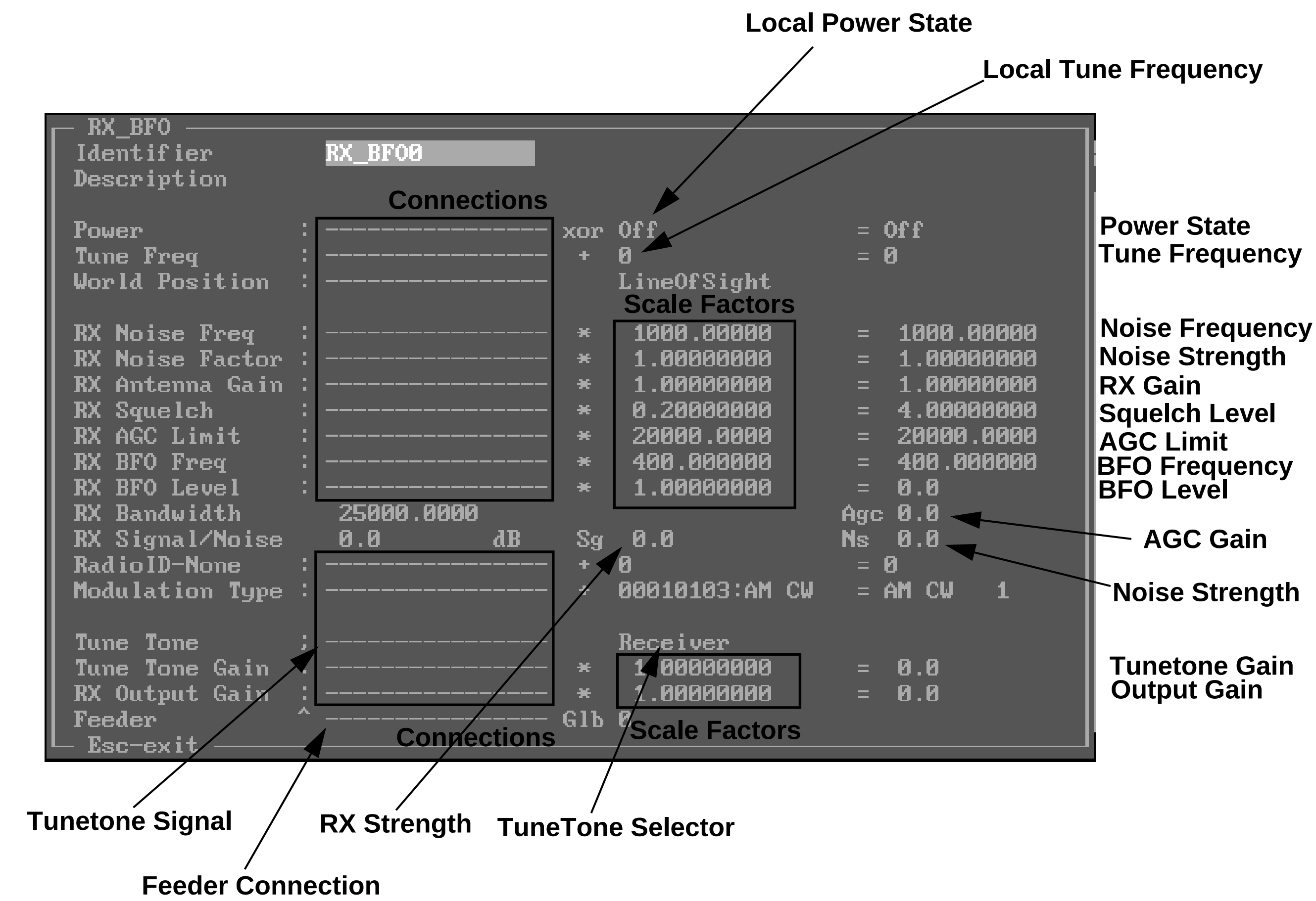


Receiver with BFO

This Object is a receiver with added Beat Frequency Oscillator, it provides all of the features of the receiver described in the previous pages.

The BFO is an additional oscillator whose tone strength is proportional to the received carrier strength. It is generally used for detecting the morse code keying present on a continuous wave beacon, which has very basic carrier keying, with no tone modulation.

Receiver with BFO



BFO Frequency Connection
: Beat frequency oscillator connection field for host control of BFO tone.

BFO Frequency Scale Factor
: Scaling factor, or local value for BFO frequency.

BFO Frequency
: Beat frequency oscillator tone frequency.

BFO Level Connection
: Beat frequency oscillator connection field for host control of BFO tone level.

BFO Level Scale Factor
: Scaling factor, or local value for BFO tone level.

BFO Level
: Beat frequency oscillator tone level. The actual tone level is a product of this value and the received signal strength.

Tune Frequency Connection
: Connection to a control object which provides the current radio tune frequency.

Local Tune Frequency
: Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the tune frequency connection.

Receiver with BFO

Tune Frequency

Frequency value used for comparison to determine which radios are connected to each other. Note: A zero value of frequency disables reception, providing an easy equivalent to a power-off condition.

Power Connection

Control object connection to provide power control from elsewhere in model.

Local Power State

Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Power State

The receiver power state, when on and the tune frequency is non-zero the receiver is active, when off it is inactive, simulating a power off condition.

Noise Frequency Connection

Background noise frequency connection field for host control of noise roll-off.

Noise Frequency Scale Factor

Scaling factor, or local value for background noise frequency.

Noise Frequency

Roll-off frequency for background noise used on receive side of radio

Noise Strength Connection

Background noise strength connection field for host control of noise level.

Noise Strength Scale Factor

Scaling factor, or local value for background noise strength.

Noise Strength

Signal strength of background noise on radio. This should range between 0 and 1.0. It is treated as another signal to be added with the other received signals, and hence goes through the squelch and AGC circuits correctly.

RX Gain Connection

Receiver antenna gain connection.

RX Gain Scale Factor

Receiver antenna gain factor.

RX Gain

Gain of the antenna, can be used to simulate receiver failures.

Squelch Level Connection

Squelch level connection for host control of adjustable squelch.

Squelch Level Scale Factor

Local value or connection scale factor for squelch level.

Squelch Level

When the total received signal strength (including background noise) is less than the squelch level then the AGC gain is set to zero, providing the normal background noise suppression. To disable the squelch, set the scale factor to zero.

AGC Limit Connection

Host connection for adjustment of the AGC limit

AGC Limit Scale Factor

Local value or connection scale factor for AGC limit.

AGC Limit

The Automatic Gain Control (AGC) limit. In radio equipment, low signal strength incoming signals are boosted by an AGC circuit to provide even reception of all signals regardless of signal strength. The AGC gain is calculated as 1/(total incoming signal strength), this limit is the maximum AGC gain allowable. The AGC gain is zero if the signal strength is below the squelch level, else it ranges between 1.0 and this AGC limit, which should be kept above 1.0. To disable the AGC set the local AGC limit to 1.0.

RX Strength

The current total signal strength for all matching radio sources.

AGC Gain

The current AGC gain value limited between 1.0 and the AGC limit.

Receiver with BFO

TuneTone Signal

The signal connection for any tones that need to be mixed into the receiver or sidetone signal path, such as local tuning tones, or weapon system cues.

TuneTone Selector

Selector for tunetone signal path, either Receiver or Sidetone, when the former the tunetone is added to the received signal path, when the latter it is added to the sidetone signal.

TuneTone Gain Connection

Host control of tunetone gain.

TuneTone Gain Scale Factor

Local value or connection scale factor for tunetone gain.

TuneTone Gain

Gain for the tunetone signal

Output Gain Connection

Overall gain connection for host control of receiver volume or on/off capability.

Output Gain Scale Factor

Local value or connection scale factor for output overall gain.

Output Gain

Overall gain for receiver. Usually this is unused, since it is simpler to use the tune frequency = 0 to simulate a power off condition, and the receiver volume level would normally be provided by the Communications Selector Panel on the feeder list.

Feeder Connection

Connection to a feeder, which provides a mechanism for outputting the receiver onto the signal highway. This field can be left blank if the signal is to be picked up by a Communications Selector Panel.

World Position Connection

Connection to a control object that specifies the receiver’s world position, or attaches it to a DIS entity.

Radio ID

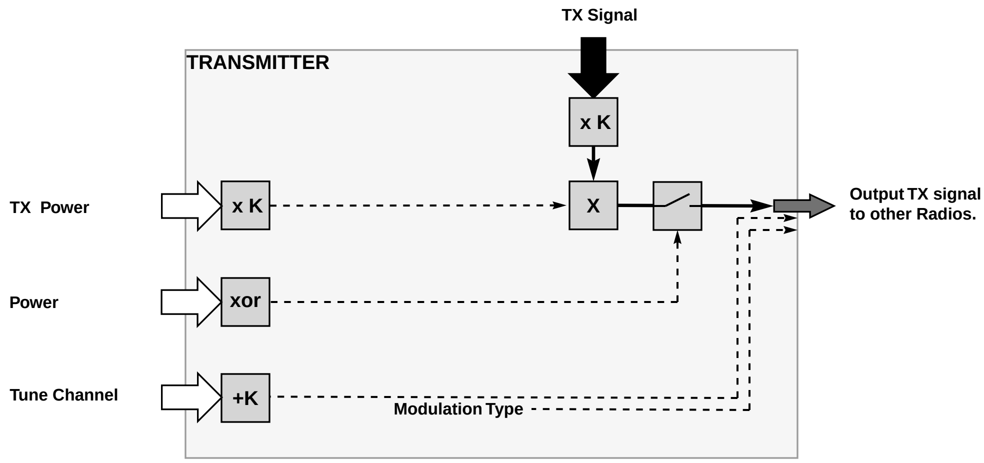
ID number of receiver that must be assigned to use DIS, Voice Net, or Local Net. The host number, site number, entity number and Radio ID must form a unique set. All numbers but the radio ID are set through the World Position Connection.

Transmitter Signals (Radios & Nav-Aids Packages)

The Transmitter signals contain special objects for transmitting the various signals needed for a radio environment simulation. They work with the radio and receiver objects described in the previous section. Each frame the signal list is scanned to determines which radios and transmitters, or jammers are tuned to the same frequency, then any signals attached to the transmitters (including the TX part of the radio) are passed across to each radio receiver. This provides a simulation of the radio environment, which includes a discrimination for AM and FM band radios, and full background noise and signal strength effects.

Many of the transmitter parameters are the same as the ones described in the Radio object.

Transmitter



Transmitter

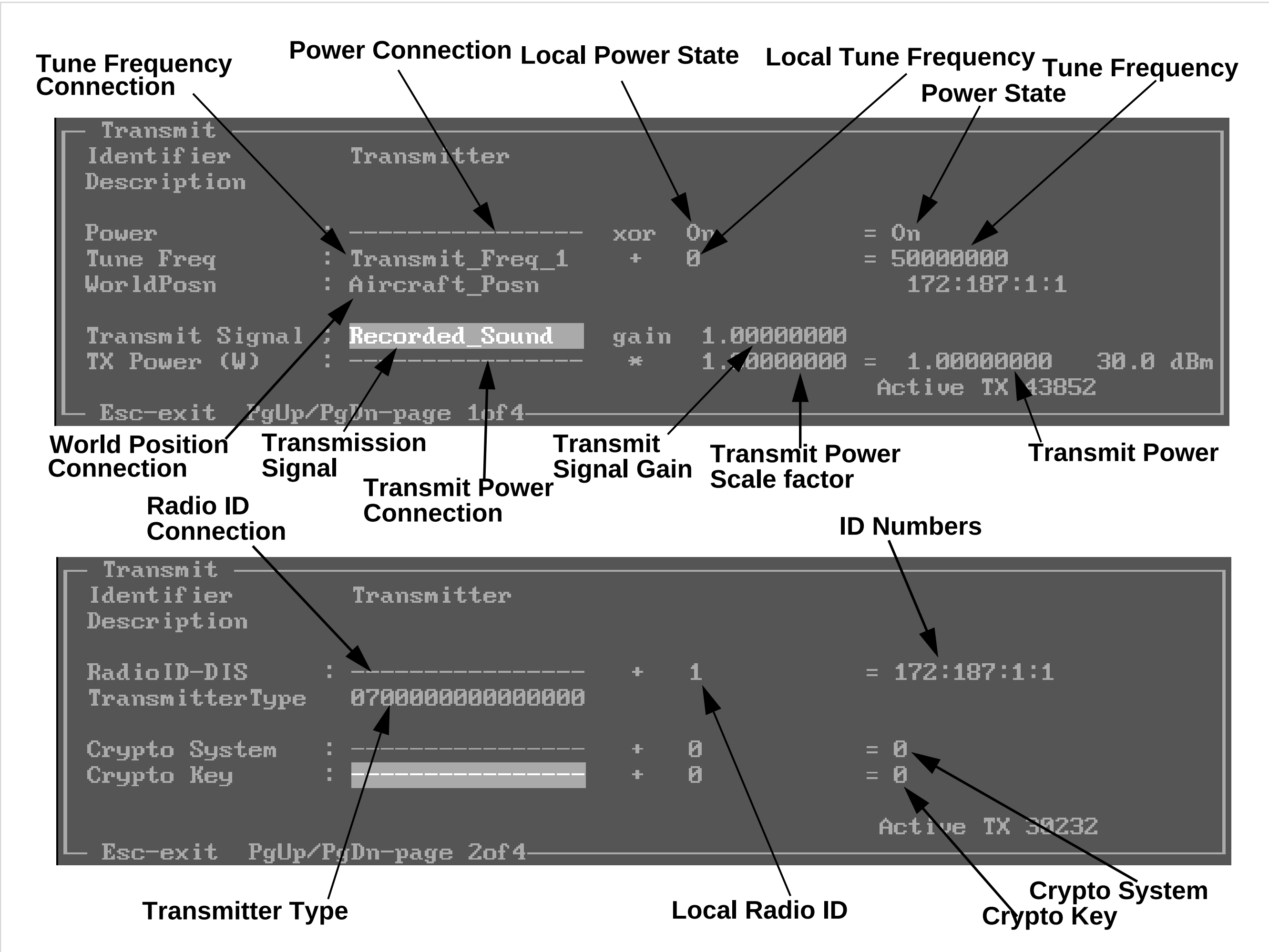
The Transmitter Object provides a generic, high level radio transmitter simulation including frequency tuning effects, AM or FM bands, and signal strength variation due to range.

The transmission signals are connected through the ‘Transmit Signal’ connection field and are typically navigation station identifier tones, marker tones or voice messages; but since the Radio Object uses any signal source, combinations of sounds can be transmitted. The resulting transmit signals are broadcast to all Radio Objects in your model, and are received by those tuned to the same frequency and on the same AM or FM band.

The world position connection determines the transmitter’s position on the globe. Occulting effects and diminishing power with distance according to an inverse square law are computed based on the world positions.

A frequency value of 1 to 10,000 provides a clear communication channel, with no attenuation, power loss, or occulting effects.

Transmitter



Tune Frequency Connection

Connection to a control object which provides the current transmitter tune frequency in Hz.

Local Tune Frequency

Provides either a local fixed tune frequency, in Hz, or an offset for the tune frequency provided by the tune frequency connection.

Tune Frequency

Frequency value (in Hz) used for comparison to determine which radios will receive the transmission signal. Note: A zero value of frequency disables transmission.

A frequency value of 1 to 100000 provides a clear communication channel, with no attenuation, power loss, or occulting effects.

Power Connection

Control object connection to provide power control from elsewhere in model.

Local Power State

Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Transmitter

Power State

The transmitter power state, when on and the tune frequency is non-zero the transmitter is active, when off it is inactive, simulating a power off condition.

Transmission Signal

Signal used for transmission to all other radios on matching frequencies.

Transmitter Type

This field indicates a transmitter type. The transmitter type information is sent out on DIS when DIS is being used, but otherwise is not currently used in the simulation.

Transmit Signal Gain

Amplitude gain for transmission signal.

Transmit Power Connection

Connection to a control object which provides the current transmit power.

Transmit Power Scale Factor

Provides a scaling factor for the transmit power connection.

Transmit Power

Transmission Power, in Watts.

World Position Connection

Connection to a control object that specifies the beacon's world position, or attaches it to a DIS entity.

IDNumbers

Indicates the Site, Host, Entity, and Radio IDs for the transmitter. The Radio ID is determined by the Radio ID connection and the Radio ID offset. The site, host, and entity ID's are determined by the world position object in the world postion connection.

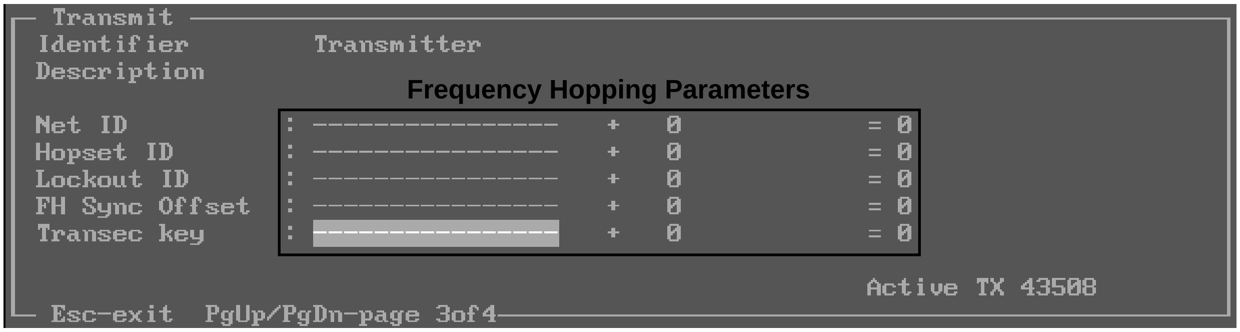
Crypto System

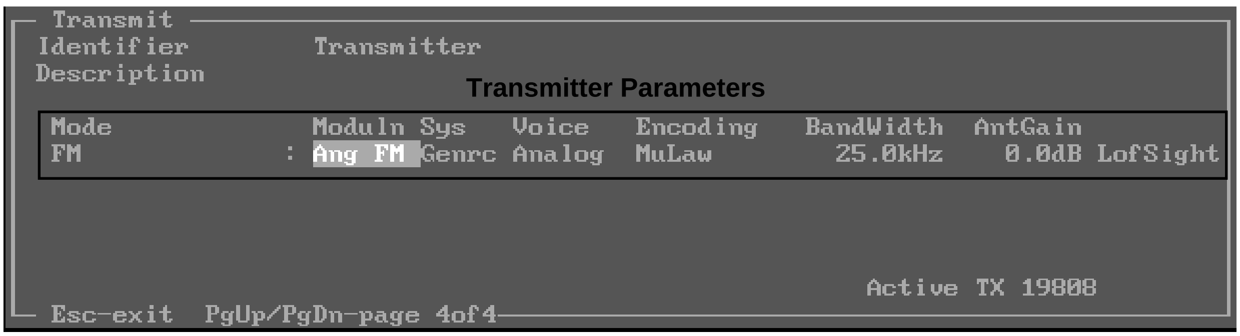
This is used for simulating encrypted communications. It will only be nonzero when the radio voice type is digital. For an explanation of how encrypted comminications work, see the Radio object.

Crypto Key

This is used for simulating encrypted communications. It will only be nonzero when the radio voice type is digital. For an explanation of how encrypted comminications work, see the Radio object.

Transmitter





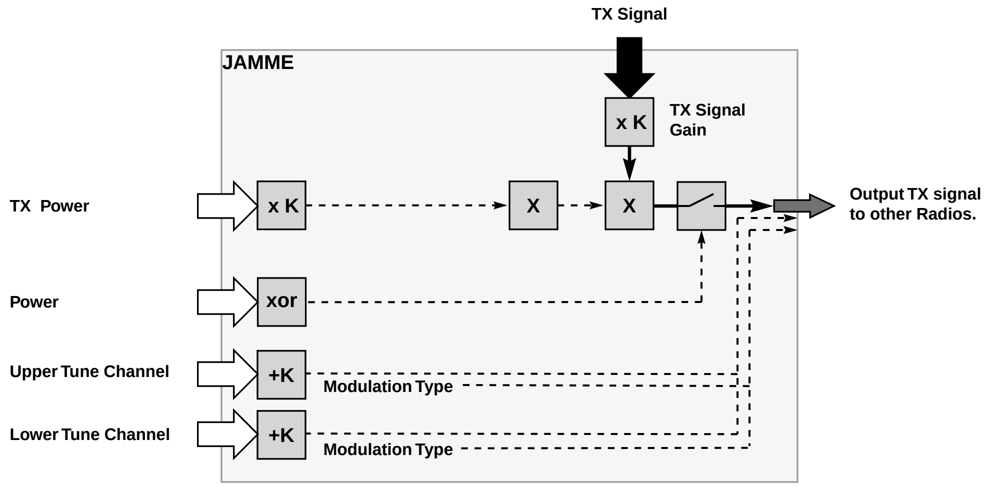
Frequency Hopping Parameters

These fields provide the parameters used to simulate the frequency hopping of a SINCGARS radio. See the Radio object for an explanation.

Transmitter Parameters

These fields determine the modulation type, system type, voice compression, and other parameters for the transmitter. For a detailed explanation of these objects, see the Radio object.

Jammer

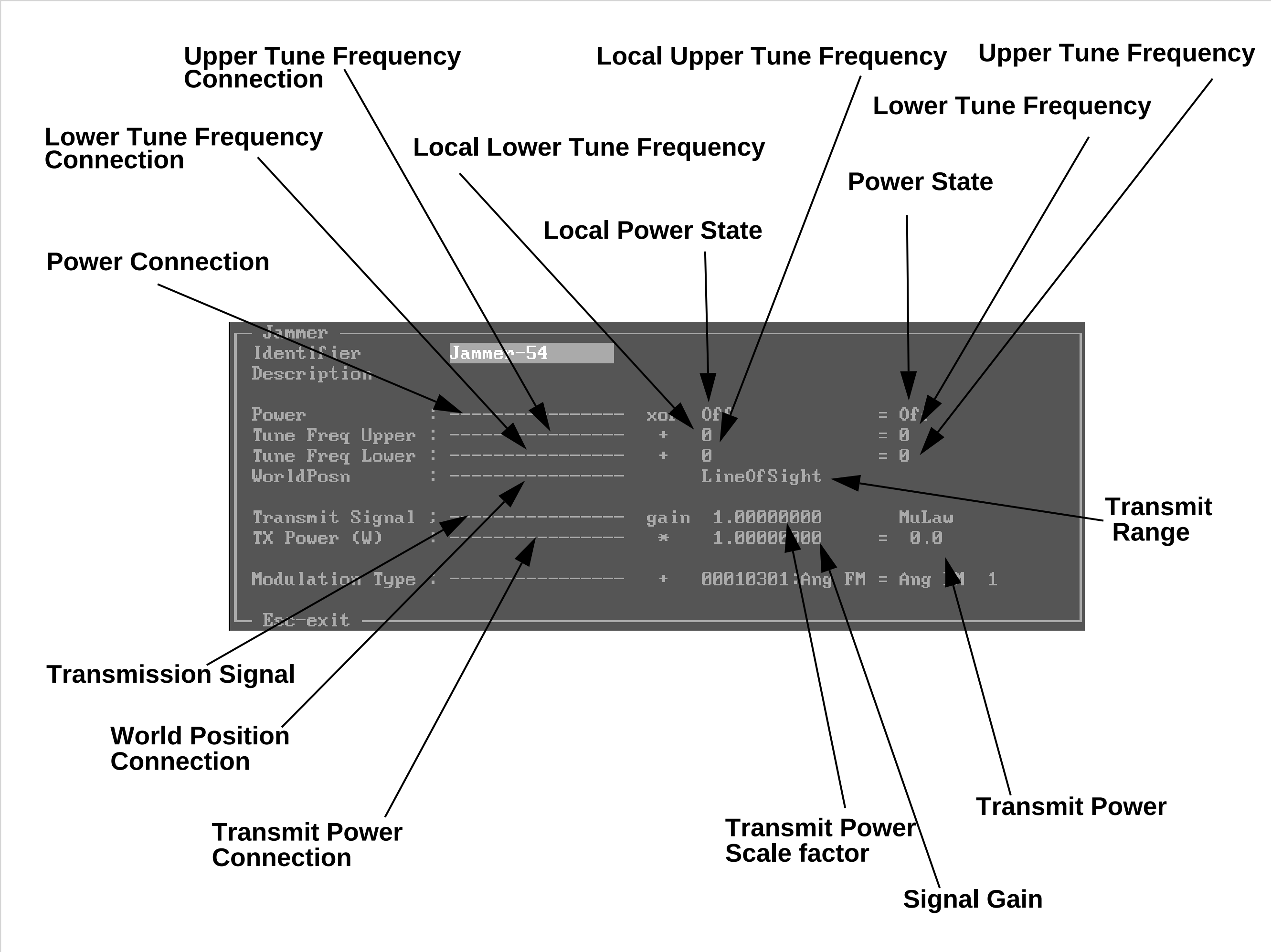


Jammer

The Jammer Object provides a specialized transmitter for broad band signal sources. It functions the same as a transmitter, except it has a range of channels over which it can transmit its signal.

The world position connection determines the jammer's position on the globe. Occulting effects and diminishing power with distance according to an inverse square law are computed based on the world positions.

Jammer



Upper Tune Frequency Connection

Connection to a control object which provides the upper transmitter tune frequency in Hz.

Upper Local Tune Frequency

Provides either a local fixed tune frequency in Hz, or an offset for the tune frequency provided by the upper tune frequency connection.

Upper Tune Frequency

Frequency value, in Hz, used for upper range of comparison to determine which radios will receive the jammer's signal.

Lower Tune Frequency Connection

Connection to a control object which provides the lower transmitter tune frequency in Hz.

Lower Local Tune Frequency

Provides either a local fixed tune frequency, in Hz, or an offset for the tune frequency provided by the lower tune frequency connection.

Lower Tune Frequency

Frequency value, in Hz used for lower range of comparison to determine which radios will receive the jammer's signal.

Power Connection

Control object connection to provide power control from elsewhere in model.

Jammer

Local Power State

Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Power State

The transmitter power state, when on the transmitter is active, when off it is inactive, simulating a power off condition.

Transmission Signal

Signal used for transmission to all other radios on matching frequencies.

Signal Gain

Amplitude gain for transmission signal.

Transmit Range

Indicates whether the jammer is capable of transmitting over the horizon or only in a line of sight. This is used by a receiver in conjunction with a world position control. If the transmitter can only operate on a line of sight then the altitude of the receiver will be taken into account such that the reception will be cut off by the curvature of the earth.

Transmit Power Connection

Connection to a control object which provides the current transmit power.

Transmit Power Scale Factor

Provides a scaling factor for the transmit power connection.

Transmit Power

Transmission Power, in Watts.

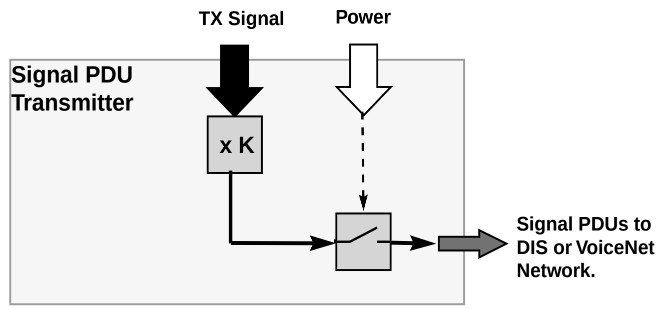
Transmit Range

Indicates whether the transmitter is capable of transmitting over the horizon or only in a line of sight. This is used by a receiver in conjunction with a world position control. If the transmitter can only operate on a line of sight then the altitude of the receiver will be taken into account such that the reception will be cut off by the curvature of the earth.

World Position Connection

Connection to a control object that specifies the beacon’s world position. Note: The jammer can <u>not</u> be used over DIS, Voice Net, or Local Net. To set up a Jammer over Local net, you can set up identical jammer objects in each model, and feed them all the same signal via a global channel.

Signal PDU Transmitter

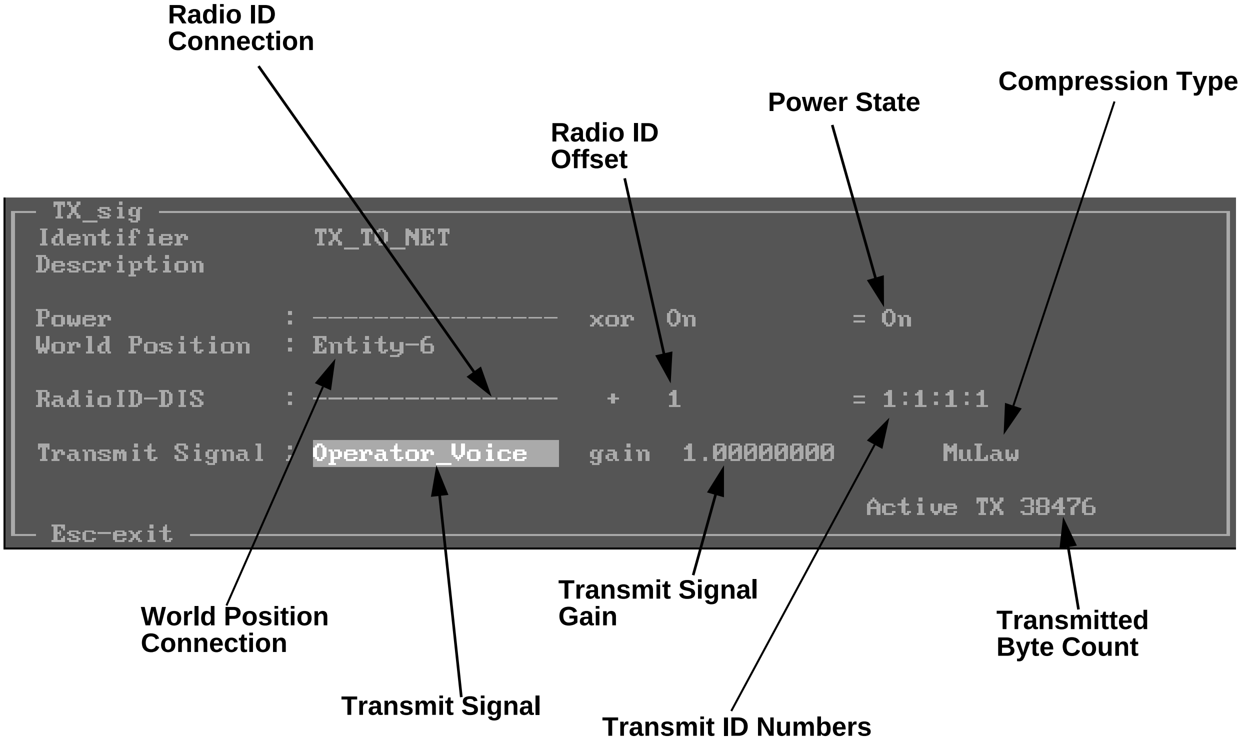


Signal PDU Transmitter

The Signal PDU Transmitter is used to broadcast signal PDUs, without any associated transmitter PDUs, on a DIS or VoiceNet network.

This can be used in conjunction with the Signal PDU Receiver to establish simple point to point contact across a network.

Signal PDU Transmitter



Power State

The transmitter power state, when on the transmitter is active, when off it is inactive.

Transmit Signal

Audio signal used transmitted. If no transmit signal is connected, no signal pdus will be generated.

Transmit Signal Gain

Amplitude gain for transmission signal.

World Position

Connects the PDU transmitter to a world position or entity object. The connected object serves only to assign the Site, Host, and Entity ID numbers to the Signal PDU Transmitter. The actual world position information is ignored.

If an entity object is connected, a world position object must be put in the “Default World Position” field of the entity object. Otherwise, no signal PDUs will be transmitted.

Compression Type

Determines whether the transmitted signal pdus will be mulaw compressed or CVSD compressed.

Signal PDU Transmitter

Transmit ID Numbers

Indicates the site, host, entity, and Radio ID numbers which will be used in the signal pdu packets. The site, host, and entity ID are taken from the world position connection, and the radio ID is determined by the Radio ID offset and connection.

Transmitted Byte Count

Indicates the number of bytes of sound information broadcast. When the transmitter is active, this number will be incrementing.

Radio ID Connection

Provides a connection to a control object to provide the radio ID.

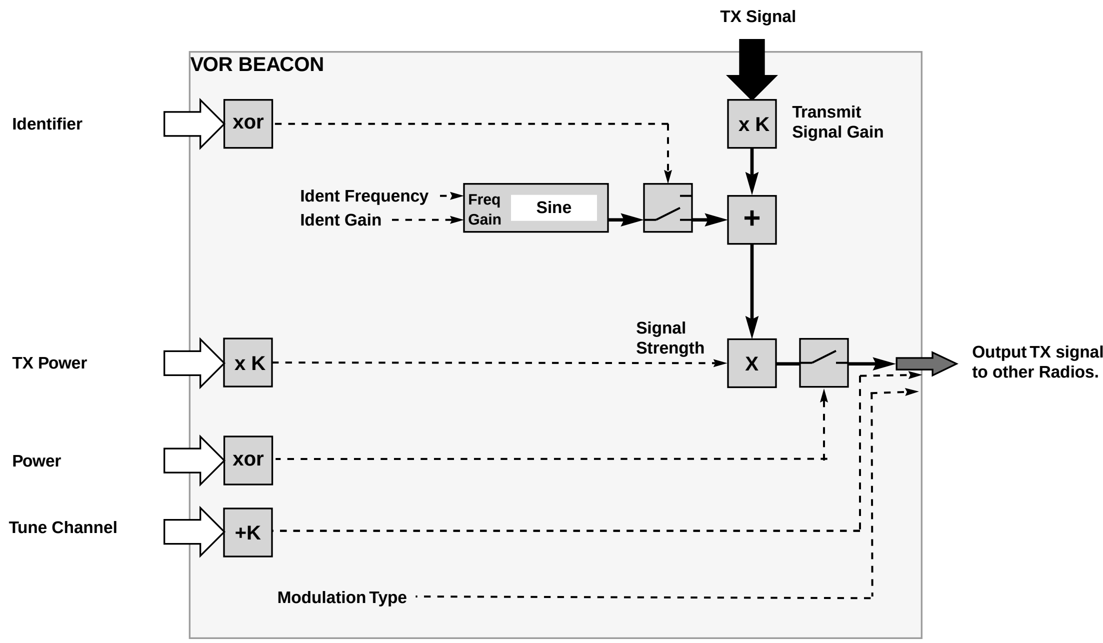
Radio ID offset

Provides an offset for the radio ID connection, or in the absence of a connection, provides the radio ID.

Navigation Aids Signals (Nav-Aids Package)

The Navigation Aids signals are special objects for transmitting the various signals needed in a full navigation beacon simulation. They work with the radio and receiver objects described in the previous section. Each frame, the signal list is scanned to determine which radios, transmitters, beacons, or jammers are tuned to the same frequency. Any signals attached to the transmitters (including the TX part of the radio) are then passed to each radio receiver. This provides a simulation of the radio environment, including a discrimination for AM and FM modulation radios, and full background noise and signal strength effects, with appropriate signal degradation and background noise effects based on range, frequency, modulation type, etc.

VOR Beacon

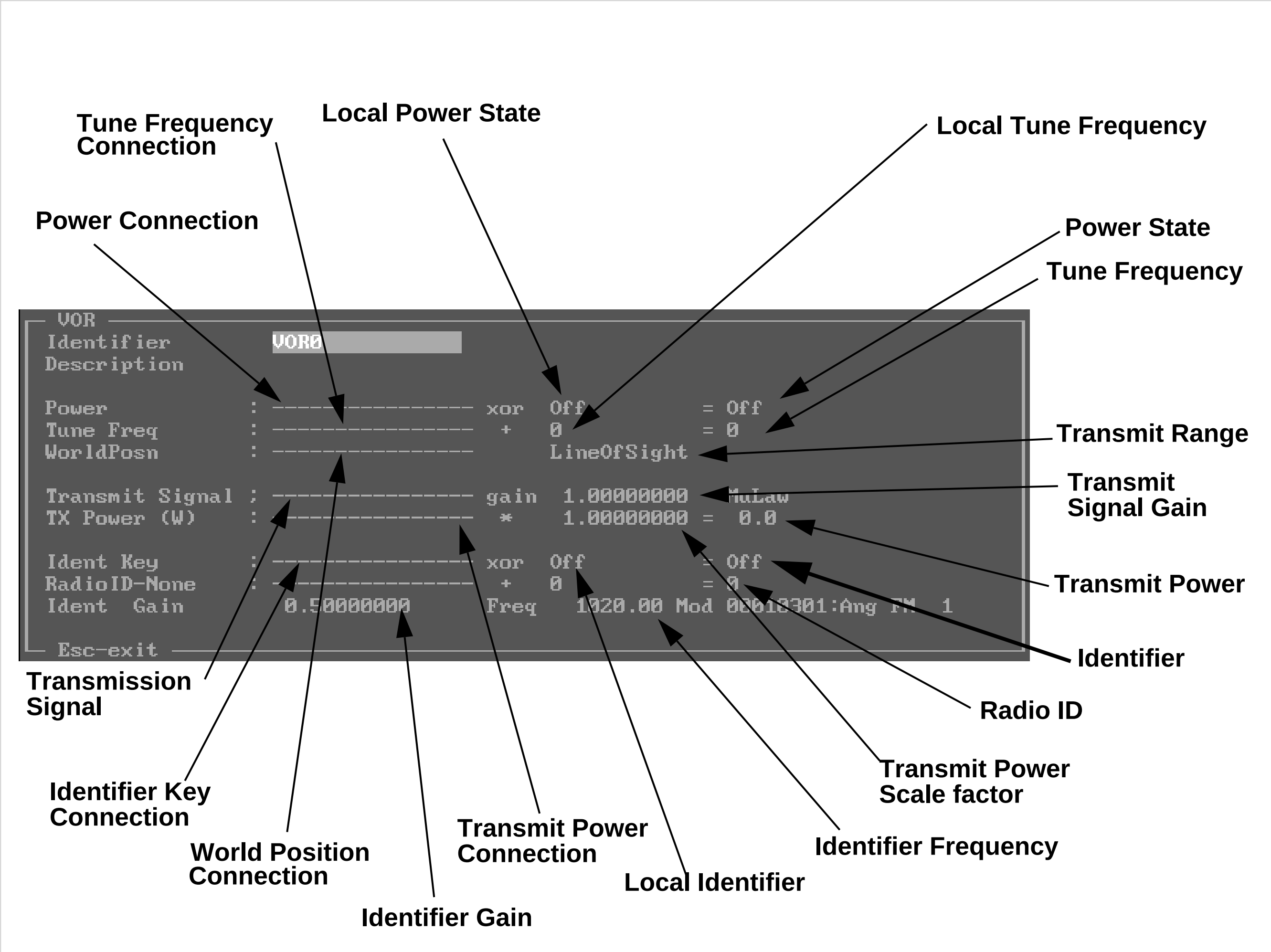


VOR Beacon

The VOR Beacon object is a transmitter with an embedded identifier tone element. It mixes an identifier keyed tone with an optional transmission signal to produce a simulation of a VOR beacon which can be heard over a suitably tuned radio or receiver.

The world position connection determines the beacon’s position on the globe. Occulting effects and diminishing power with distance according to an inverse square law are computed based on the world positions.

VOR Beacon



Tune Frequency Connection

Connection to a control object which provides the current transmitter tune frequency.

Local Tune Frequency

Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the tune frequency connection.

Tune Frequency

Frequency value used for comparison to determine which radios and transmitters are connected to each other. Note: A zero value of frequency disables transmission.

Power Connection

Control object connection to provide power control from elsewhere in model.

Local Power State

Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Power State

The transmitter power state, when on and the tune frequency is non-zero the transmitter is active, when off it is inactive, simulating a power off condition.

Identifier Key Connection

Connection to a control object which provides the identifier keying.

VOR Beacon

Local Identifier

Local Identifier state, used to invert identifier signal.

Identifier

Identifier state, On state produces ident tone, Off state produces no tone.

Transmission Signal

Signal used for transmission to all other radios on matching frequencies.

Transmit Range

Indicates whether the VOR is capable of transmitting over the horizon or only in a line of sight. This is used by a receiver in conjunction with a world position control. If the transmitter can only operate on a line of sight then the altitude of the receiver will be taken into account such that the reception will be cut off by the curvature of the earth.

Transmit Power Connection

Connection to a control object which provides the current transmit power.

Transmit Power Scale Factor

Scale factor for the transmission power.

Transmit Power

Transmission power, in Watts.

Transmit Signal Gain

Signal strength scale factor for transmission.

Identifier Gain

Signal Gain factor for identifier amplitude.

Identifier Frequency

Identifier tone frequency (in Hertz).

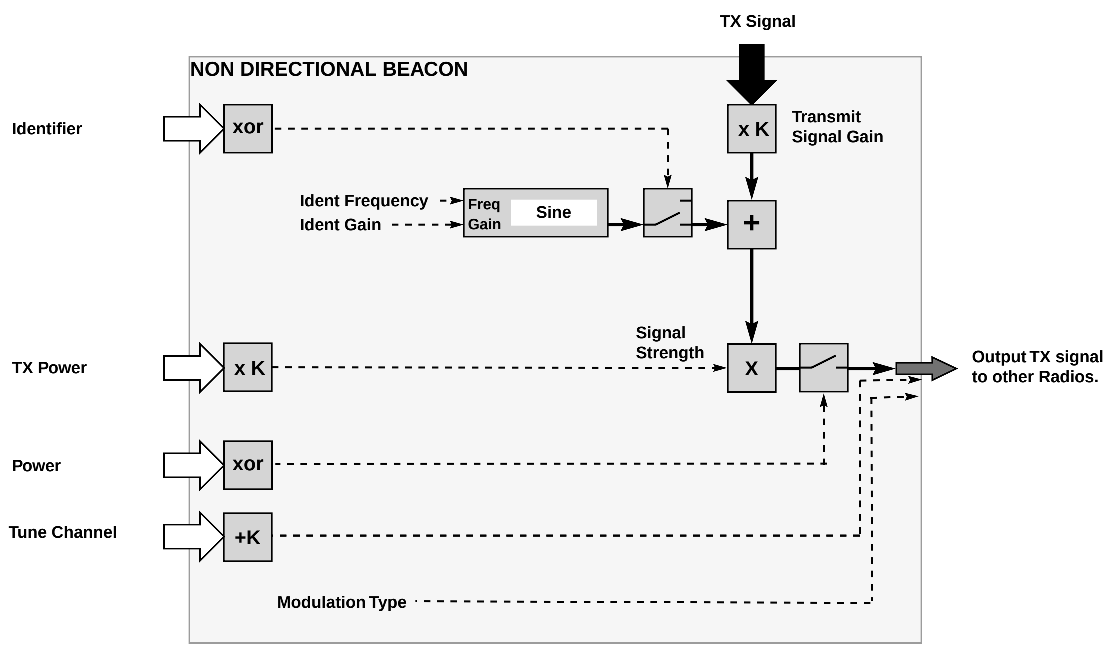
World Position Connection

Connection to a control object that specifies the beacon’s world position, or attaches it to a DIS entity.

Radio ID

ID number of transmitter that must be assigned to use DIS, Voice Net, or Local Net. The host number, site number, entity number and Radio ID must form a unique set. All numbers but the radio ID are set through the World Position Connection.

NDB Beacon

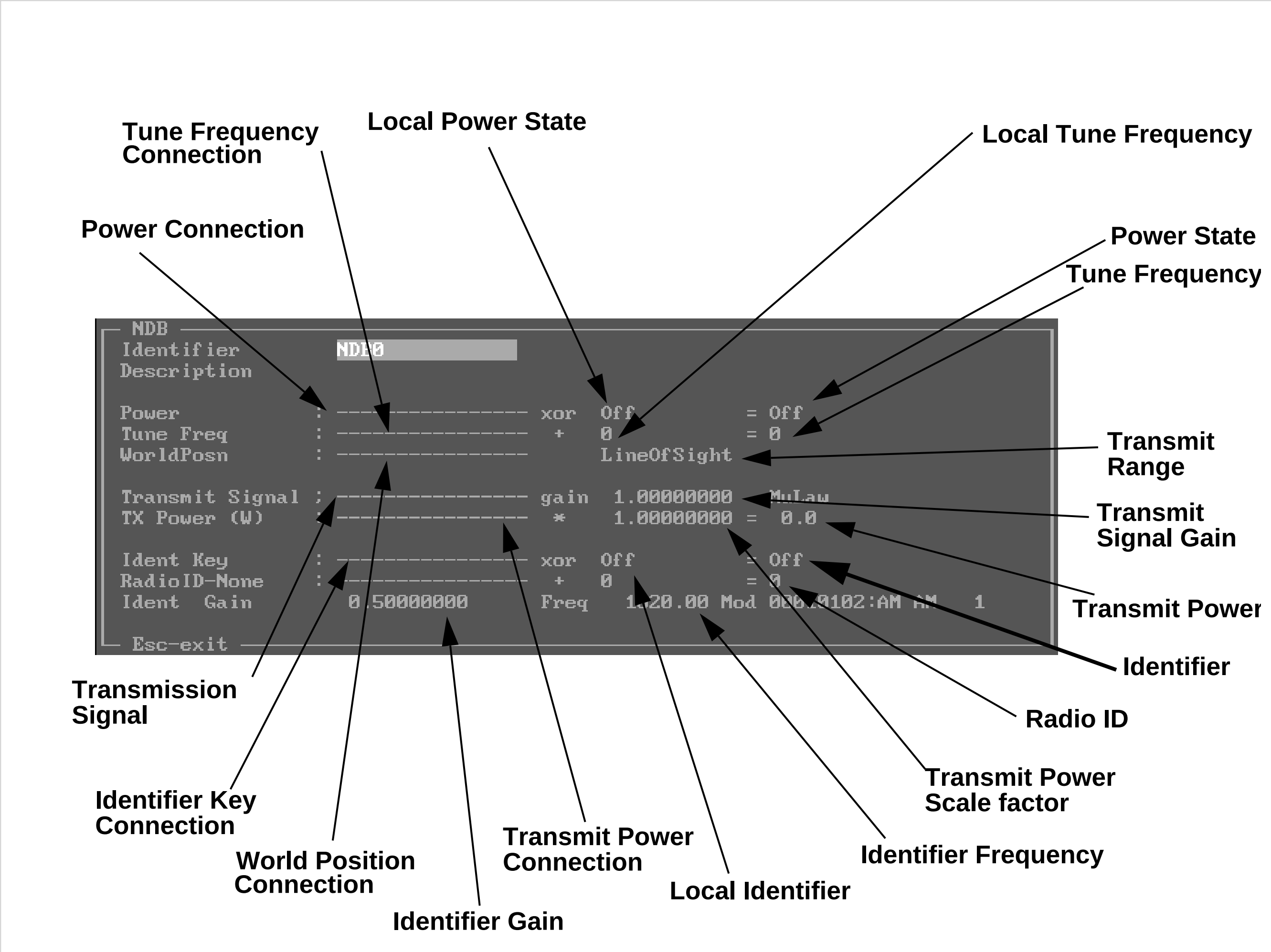


NDB Beacon

The Non Directional Beacon (NDB) object is a transmitter with an embedded identifier tone element. It mixes an identifier keyed tone with an optional transmission signal to produce a simulation of an NDB which can be heard over a suitably tuned radio or receiver.

The world position connection determines the beacon’s position on the globe. Occulting effects and diminishing power with distance according to an inverse square law are computed based on the world positions.

NDB Beacon



Tune Frequency Connection
Connection to a control object which provides the current transmitter tune frequency.

Local Tune Frequency
Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the tune frequency connection.

Tune Frequency
Frequency value used for comparison to determine which radios and transmitters are connected to each other. Note: A zero value of frequency disables transmission.

Power Connection
Control object connection to provide power control from elsewhere in model.

Local Power State
Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Power State
The transmitter power state, when on and the tune frequency is non-zero the transmitter is active, when off it is inactive, simulating a power off condition.

Identifier Key Connection
Connection to a control object which provides the identifier keying.

NDB Beacon

Local Identifier

Local Identifier state, used to invert identifier.

Identifier

Identifier state, On state produces ident tone, Off state produces no tone.

Transmission Signal

Signal used for transmission to all other radios on matching frequencies.

Transmit Signal Gain

Amplitude gain for transmission signal.

Transmit Range

Indicates whether the beacon is capable of transmitting over the horizon or only in a line of sight. This is used by a receiver in conjunction with a world position control. If the transmitter can only operate on a line of sight then the altitude of the receiver will be taken into account such that the reception will be cut off by the curvature of the earth.

Transmit Power Connection

Connection to a control object which provides the current transmit power.

Transmit Power Scale Factor

Scale factor for the transmission power.

Transmit Power

Transmission power, in Watts.

Identifier Gain

Signal Gain factor for identifier amplitude.

Identifier Frequency

Identifier tone frequency (in Hertz).

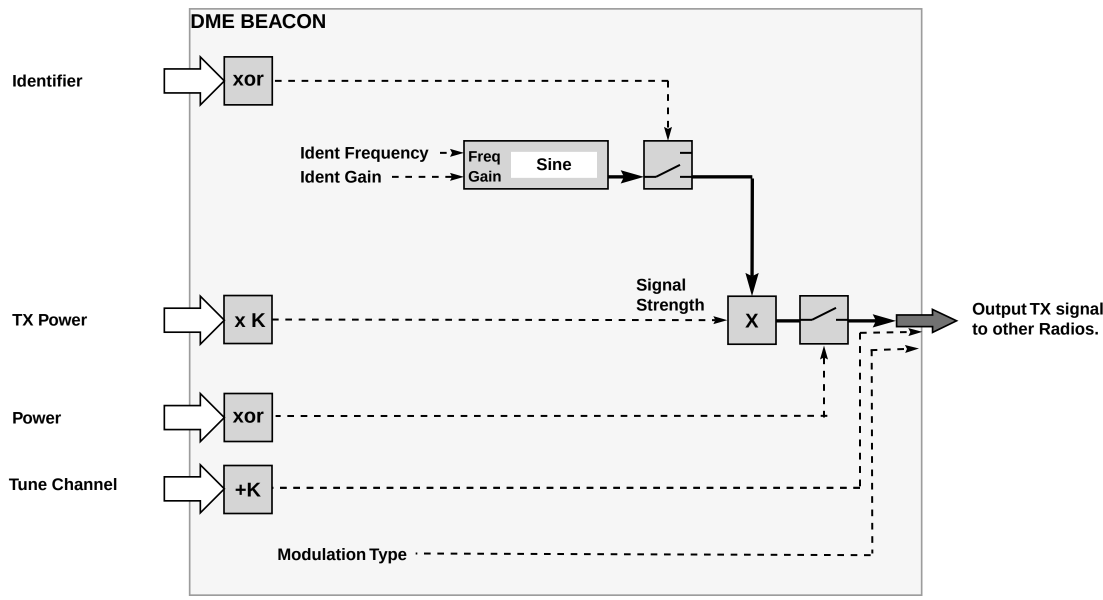
World Position Connection

Connection to a control object that specifies the beacon’s world position, or attaches it to a DIS entity.

Radio ID

ID number of transmitter that must be assigned to use DIS, Voice Net, or Local Net. The host number, site number, entity number and Radio ID must form a unique set. All numbers but the radio ID are set through the World Position Connection.

DME Beacon

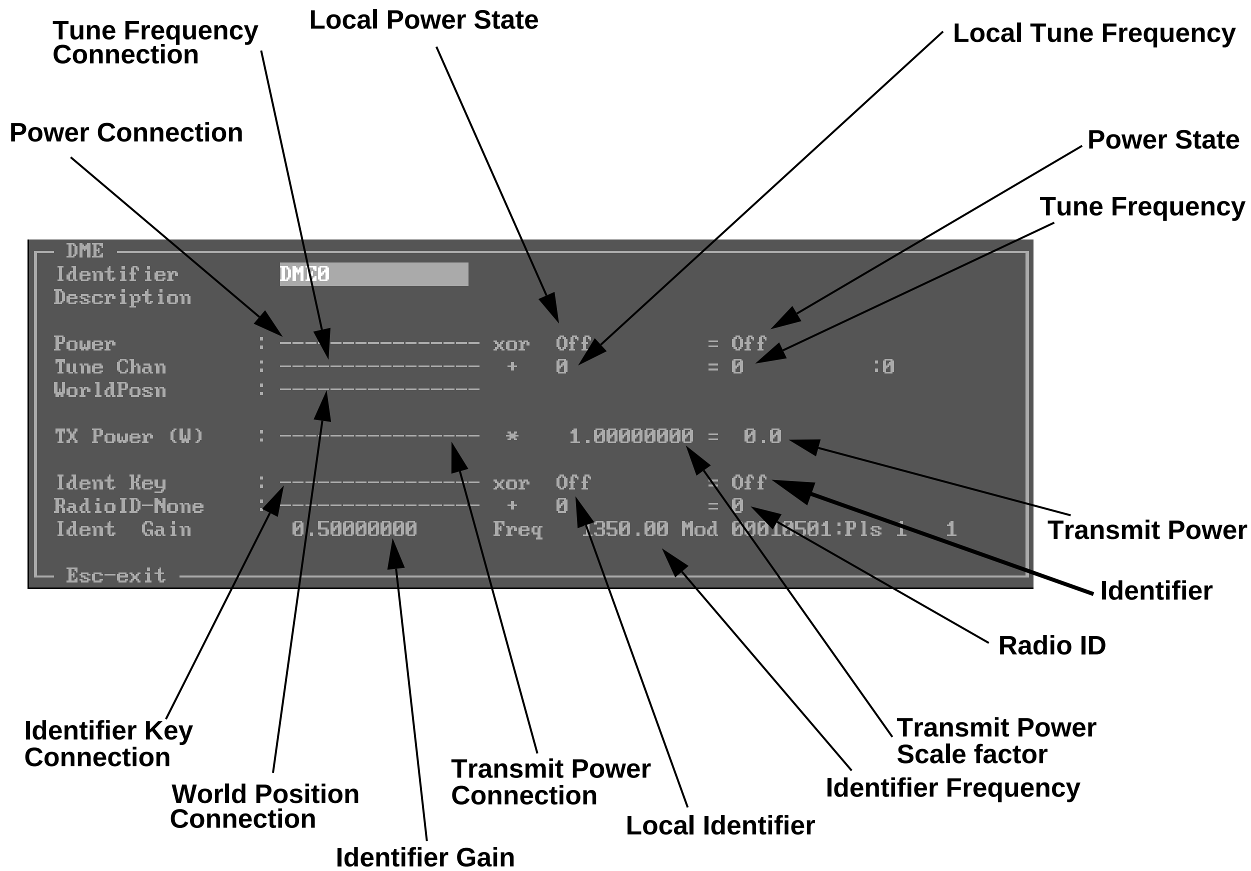


DME Beacon

The DME Beacon object is a transmitter with an embedded tone element. It produces an identifier keyed tone simulation of a DME beacon which can be heard over a suitably tuned radio or receiver.

The world position connection determines the beacon's position on the globe. Occulting effects and diminishing power with distance according to an inverse square law are computed based on the world positions.

DME Beacon



Tune Frequency Connection

Connection to a control object which provides the current transmitter tune frequency.

Local Tune Frequency

Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the tune frequency connection.

Tune Frequency

Frequency value used for comparison to determine which radios and transmitters are connected to each other. Note: A zero value of frequency disables transmission.

Power Connection

Control object connection to provide power control from elsewhere in model.

Local Power State

Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Power State

The transmitter power state, when on and the tune frequency is non-zero the transmitter is active, when off it is inactive, simulating a power off condition.

Identifier Key Connection

Connection to a control object which provides the identifier keying.

DME Beacon

Local Identifier

Local Identifier state, used to invert identifier.

Identifier

Identifier state, On state produces ident tone, Off state produces no tone.

Transmit Power Connection

Connection to a control object which provides the current transmit power.

Transmit Power Scale Factor

Provides either a local fixed transmission power, or a scaling factor for the transmit power connection.

Transmit Power

Transmission power, in Watts.

Identifier Gain

Signal Gain factor for identifier amplitude.

Identifier Frequency

Identifier tone frequency (in Hertz).

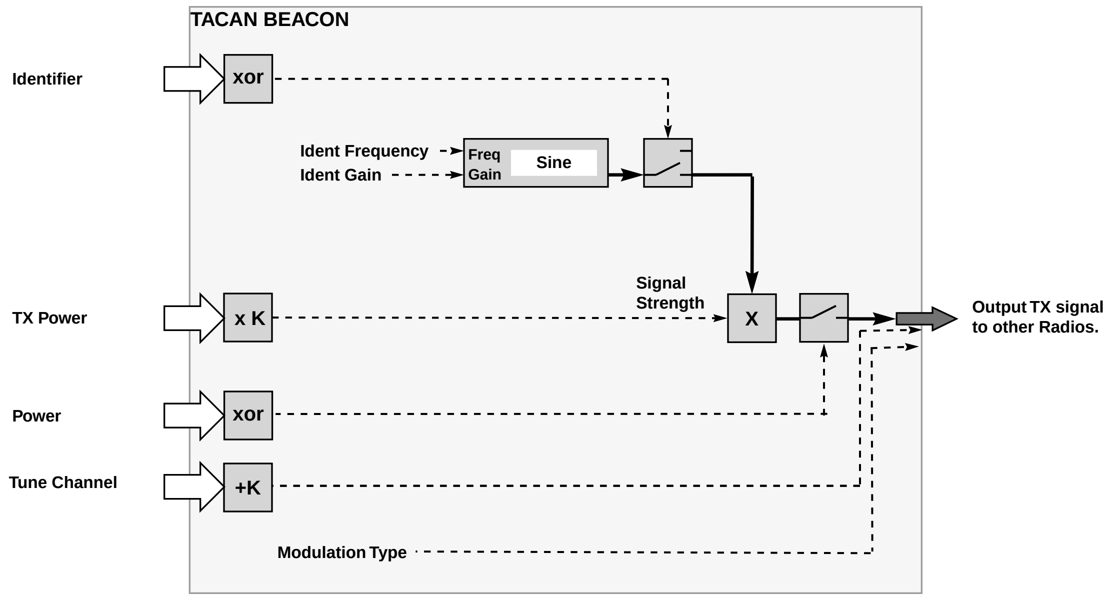
World Position Connection

Connection to a control object that specifies the beacon’s world position, or attaches it to a DIS entity.

Radio ID

ID number of transmitter that must be assigned to use DIS, Voice Net, or Local Net. The host number, site number, entity number and Radio ID must form a unique set. All numbers but the radio ID are set through the World Position Connection.

TACAN Beacon

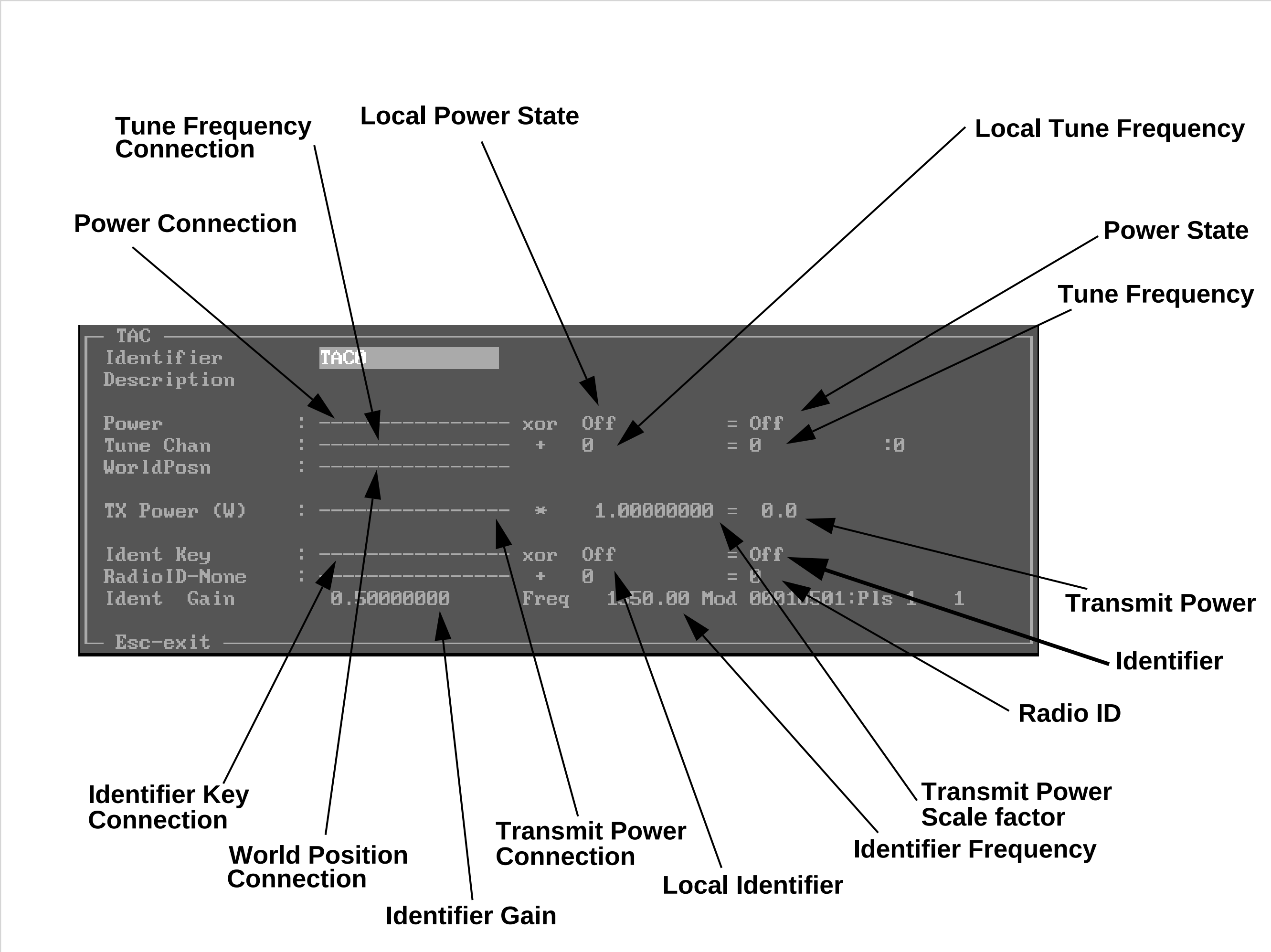


TACAN Beacon

The Tacan Beacon object is a transmitter with an embedded tone element. It produces an identifier keyed tone simulation of a Tacan beacon which can be heard over a suitably tuned radio or receiver.

The world position connection determines the beacon’s position on the globe. Occulting effects and diminishing power with distance according to an inverse square law are computed based on the world positions.

TACAN Beacon



Tune Frequency Connection

Connection to a control object which provides the current transmitter tune frequency.

Local Tune Frequency

Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the tune frequency connection.

Tune Frequency

Frequency value used for comparison to determine which radios and transmitters are connected to each other. Note: A zero value of frequency disables transmission.

Power Connection

Control object connection to provide power control from elsewhere in model.

Local Power State

Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Power State

The transmitter power state, when on and the tune frequency is non-zero the transmitter is active, when off it is inactive, simulating a power off condition.

Identifier Key Connection

Connection to a control object which provides the identifier keying.

TACAN Beacon

Local Identifier

Local Identifier state, used to invert identifier.

Identifier

Identifier state, On state produces ident tone, Off state produces no tone.

Transmit Power Connection

Connection to a control object which provides the current transmit power.

Transmit Power Scale Factor

Provides a scaling factor for the transmit power connection.

Transmit Power

Transmission Power, in Watts.

Identifier Gain

Signal Gain factor for identifier amplitude.

Identifier Frequency

Identifier tone frequency (in Hertz).

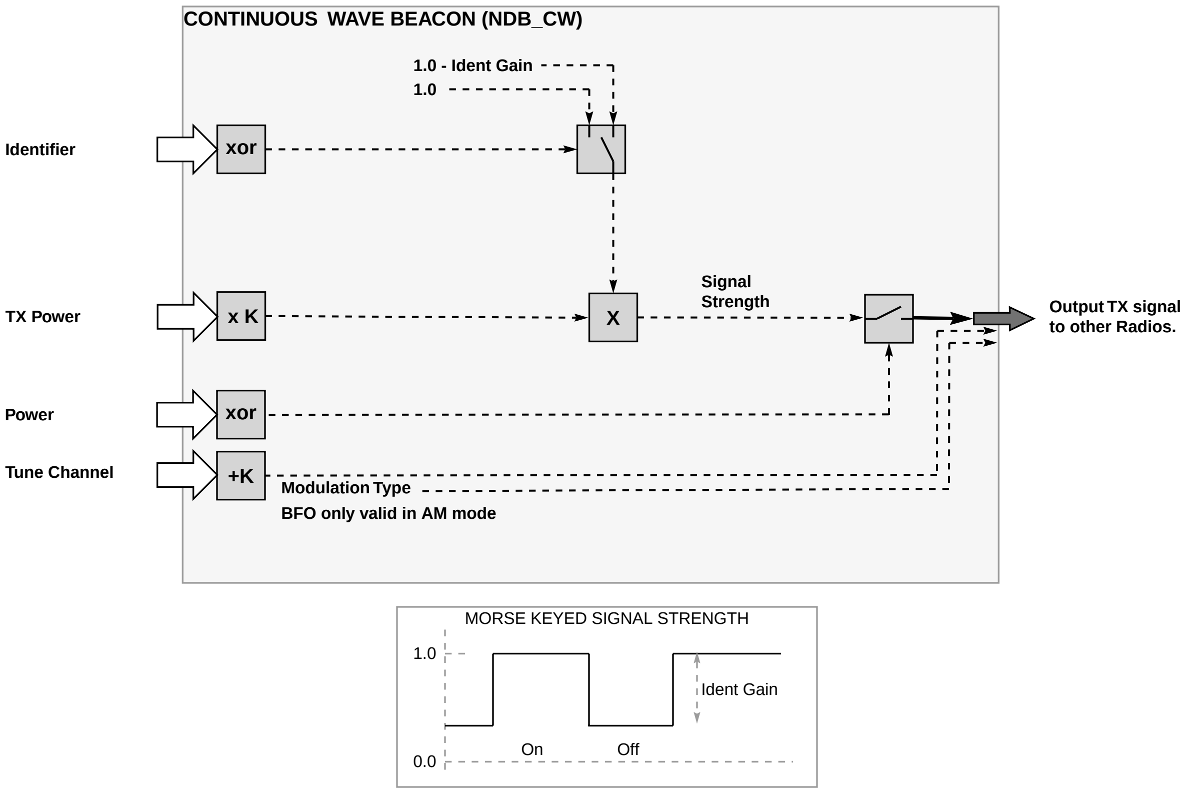
World Position Connection

Connection to a control object that specifies the beacon’s world position, or attaches it to a DIS entity.

Radio ID

ID number of transmitter that must be assigned to use DIS, Voice Net, or Local Net. The host number, site number, entity number and Radio ID must form a unique set. All numbers but the radio ID are set through the World Position Connection.

Continuous Wave Beacon

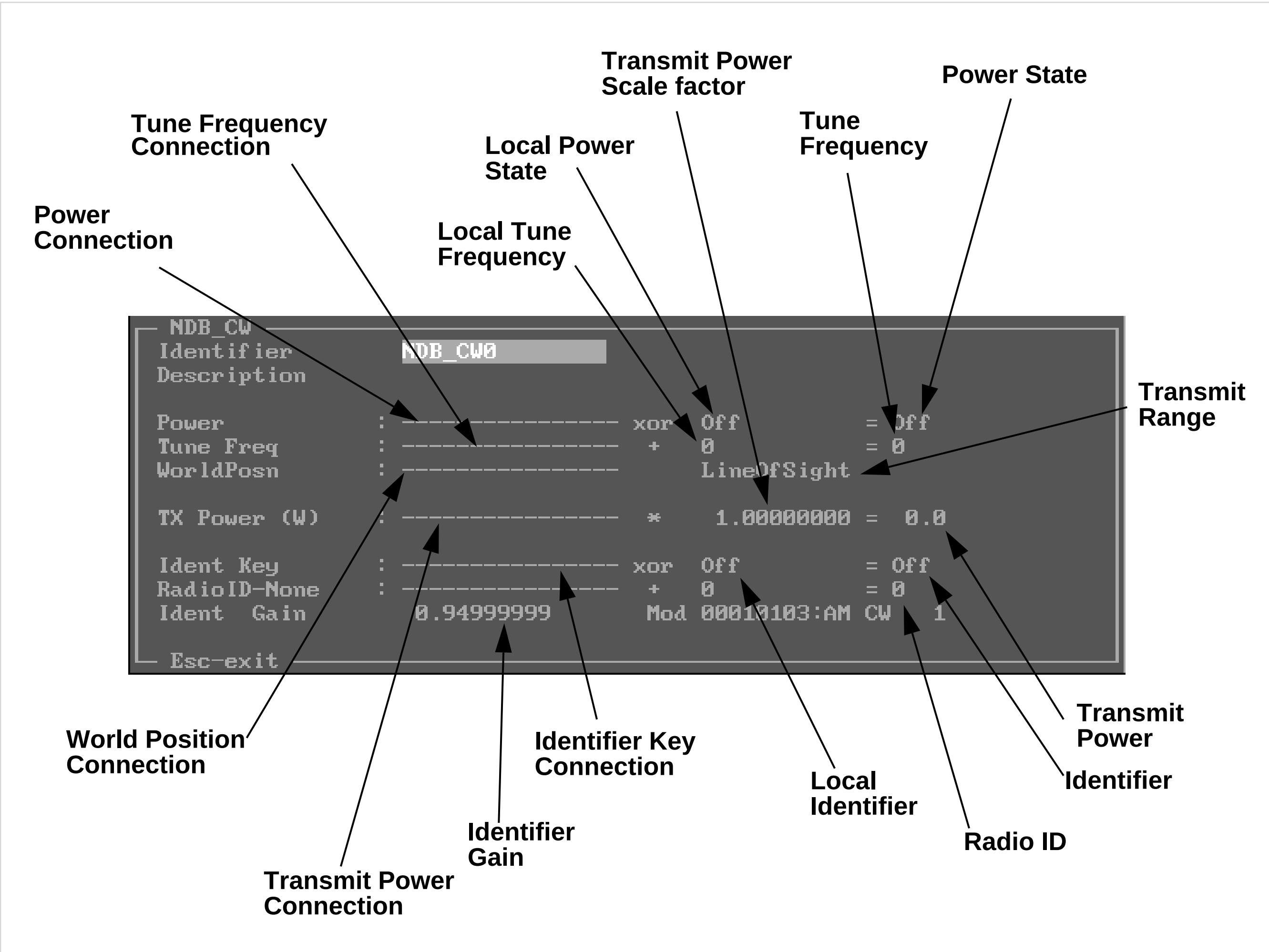


Continuous Wave Beacon

The Continuous Wave NDB is an older style of beacon that uses basic carrier modulation, with no identifier tone, the keying can only be heard via a receiver's beat frequency oscillator. This object provides a simple signal strength keyed transmitter object, which has adjustable depth of modulation and an optional additional carrier signal for any voice transmissions.

The world position connection determines the beacon's position on the globe. Occulting effects and diminishing power with distance according to an inverse square law are computed based on the world positions.

Continuous Wave Beacon



Tune Frequency Connection

Connection to a control object which provides the current transmitter tune frequency.

Local Tune Frequency

Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the tune frequency connection.

Tune Frequency

Frequency value used for comparison to determine which radios and transmitters are connected to each other. Note: A zero value of frequency disables transmission.

Power Connection

Control object connection to provide power control from elsewhere in model.

Local Power State

Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Power State

The transmitter power state, when on and the tune frequency is non-zero the transmitter is active, when off it is inactive, simulating a power off condition.

Identifier Key Connection

Connection to a control object which provides the identifier keying.

Continuous Wave Beacon

Local Identifier

Local Identifier state, used to invert identifier.

Identifier

Identifier state, On state produces ident tone, Off state produces no tone.

Transmit Range

Indicates whether the beacon is capable of transmitting over the horizon or only in a line of sight. This is used by a receiver in conjunction with a world position control. If the transmitter can only operate on a line of sight then the altitude of the receiver will be taken into account such that the reception will be cut off by the curvature of the earth.

Transmit Power Connection

Connection to a control object which provides the current transmit power.

Transmit Power Scale Factor

Provides a scaling factor for the transmit power connection.

Transmit Power

Transmission Power, in Watts.

Identifier Gain

Signal Gain factor for identifier keyed signal strength. This is the modulation depth for the continuous wave keying.

Note: The transmission signal does not amplitude modulate the carrier, and hence does not fully simulate a voice modulated continuous wave NDB with BFO on.

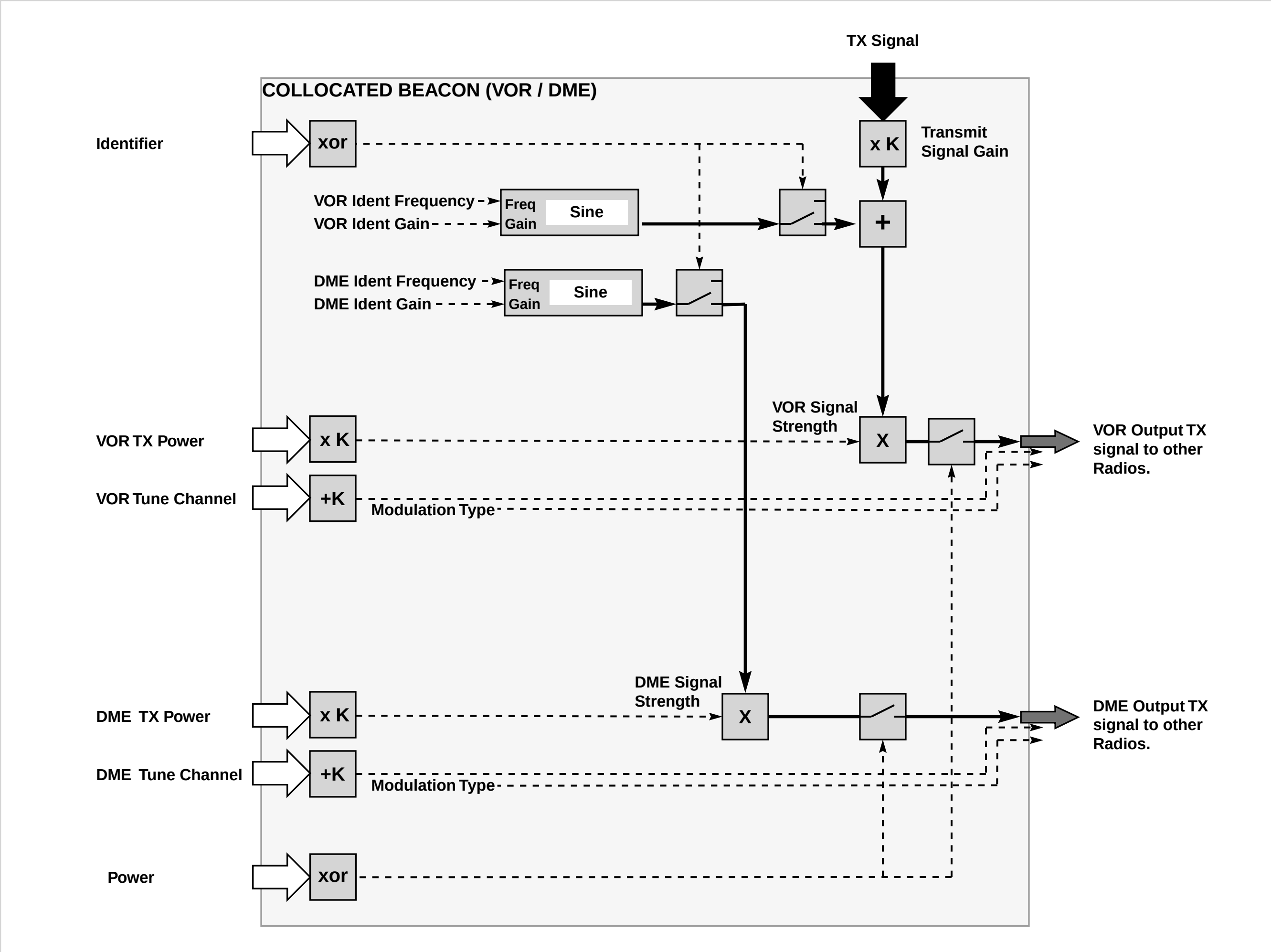
World Position Connection

Connection to a control object that specifies the beacon’s world position, or attaches it to a DIS entity.

Radio ID

ID number of transmitter that must be assigned to use DIS, Voice Net, or Local Net. The host number, site number, entity number and Radio ID must form a unique set. All numbers but the radio ID are set through the World Position Connection.

Collocated VOR/DME



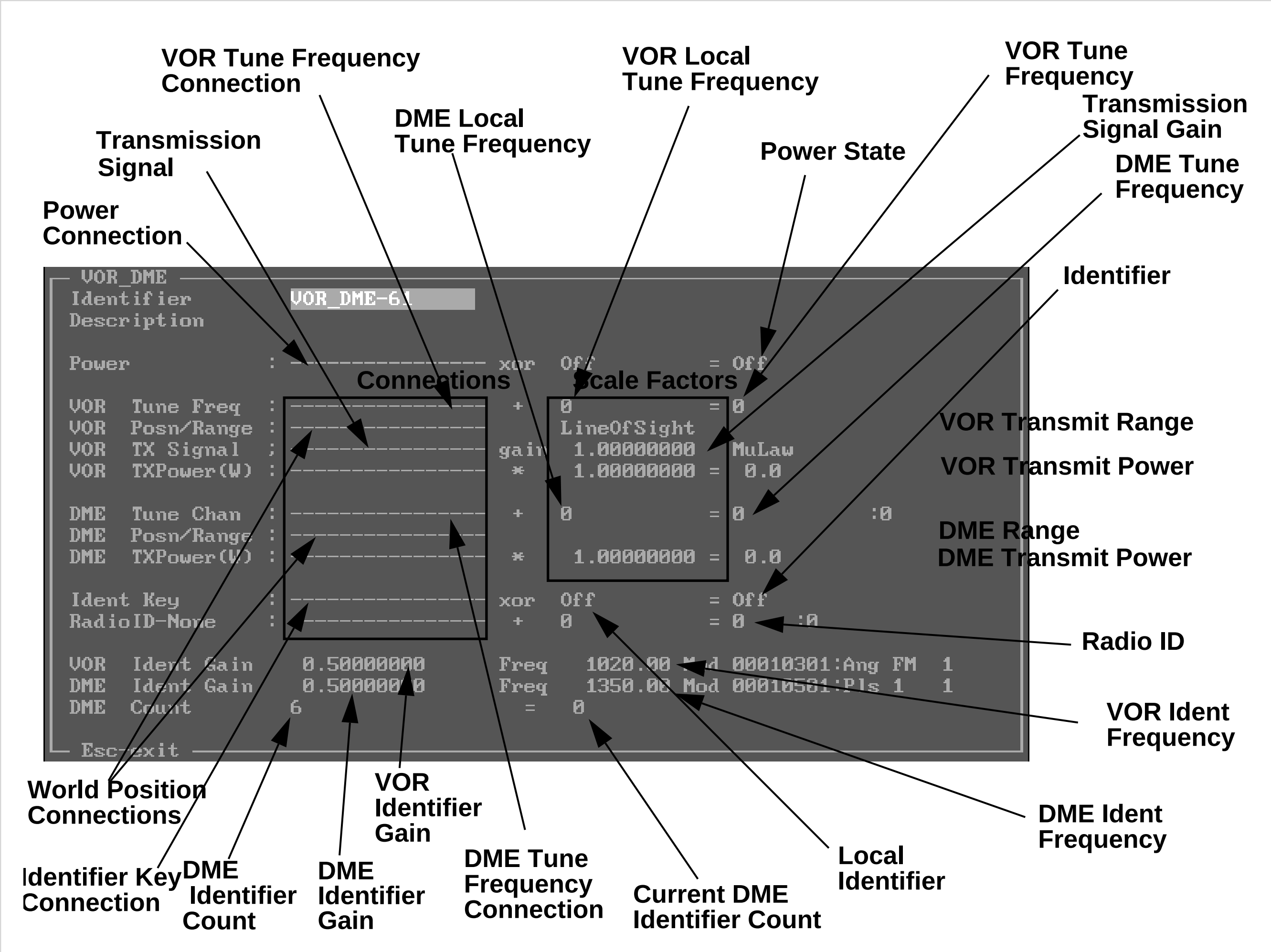
Collocated VOR/DME

The Transmitter Object provides a generic, high level radio transmitter simulation including frequency tuning effects, AM or FM bands, and signal strength variation due to range.

The transmission signals are connected through the 'Transmit Signal' connection field and are typically navigation station identifier tones, marker tones or voice messages; but since the Transmitter Object uses any signal source, combinations of sounds can be transmitted. The resulting transmit signals are broadcast to all Radio Objects in your model, and are received by those tuned to the same frequency and on the same AM or FM band.

The world position connection determines the beacon's position on the globe. Occulting effects and diminishing power with distance according to an inverse square law are computed based on the world positions.

Collocated VOR/DME



VOR Tune Frequency Connection

Connection to a control object which provides the VOR transmitter tune frequency.

VOR Local Tune Frequency

Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the VOR tune frequency connection.

VOR Tune Frequency

Frequency value used for comparison to determine which radios will receive the VOR transmissions. Note: A zero value of frequency disables transmission.

DME Tune Frequency Connection

Connection to a control object which provides the DME transmitter tune frequency.

DME Local Tune Frequency

Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the DME tune frequency connection.

DME Tune Frequency

Frequency value used for comparison to determine which radios will receive the DME transmissions. Note: A zero value of frequency disables transmission.

Collocated VOR/DME

Power Connection
Control object connection to provide power control from elsewhere in model.

Local Power State
Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Power State
The transmitter power state, when on and the tune frequency is non-zero the transmitter is active, when off it is inactive, simulating a power off condition.

Identifier Key Connection
Connection to a control object which provides the identifier keying for both VOR and DME transmitters.

Local Identifier
Local Identifier state, used to invert identifier.

Identifier
Identifier state, On state produces ident tone, Off state produces no tone.

Transmission Signal
Signal used for transmission to all other radios on matching frequencies the VOR source.

Transmission Signal Gain
Amplitude gain for VOR transmission signal.

VOR Transmit Range
Indicates whether the VOR is capable of transmitting over the horizon or only in a line of sight. This is used by a receiver in conjunction with a world position control. If the transmitter can only operate on a line of sight then the altitude of the receiver will be taken into account such that the reception will be cut off by the curvature of the earth.

VOR Transmit Power Connection
Connection to a control object which provides the VOR transmit power.

VOR Transmit Power Scale Factor
Provides either a local fixed transmission power, or a scaling factor for the VOR transmit power connection.

VOR Transmit Power
Signal power, in watts, of VOR transmission.

DME Transmit Power Connection
Connection to a control object which provides the DME transmission power.

DME Transmit Power Scale Factor
Provides either a local fixed transmission power, or a scaling factor for the DME transmit power connection.

DME Transmit Power
DME transmission power, in Watts.

VOR Identifier Gain
Signal Gain factor for VOR identifier amplitude.

VOR Identifier Frequency
VOR Identifier tone frequency (in Hertz).

DME Identifier Gain
Signal Gain factor for DME identifier amplitude.

DME Identifier Frequency
DME Identifier tone frequency (in Hertz).

DME Identifier Count
Number of Keyed identifiers to skip before Keying DME identifier. The VOR and DME identifier tones are mutually exclusive.

Current DME Identifier Count
Local count of identifiers skipped.

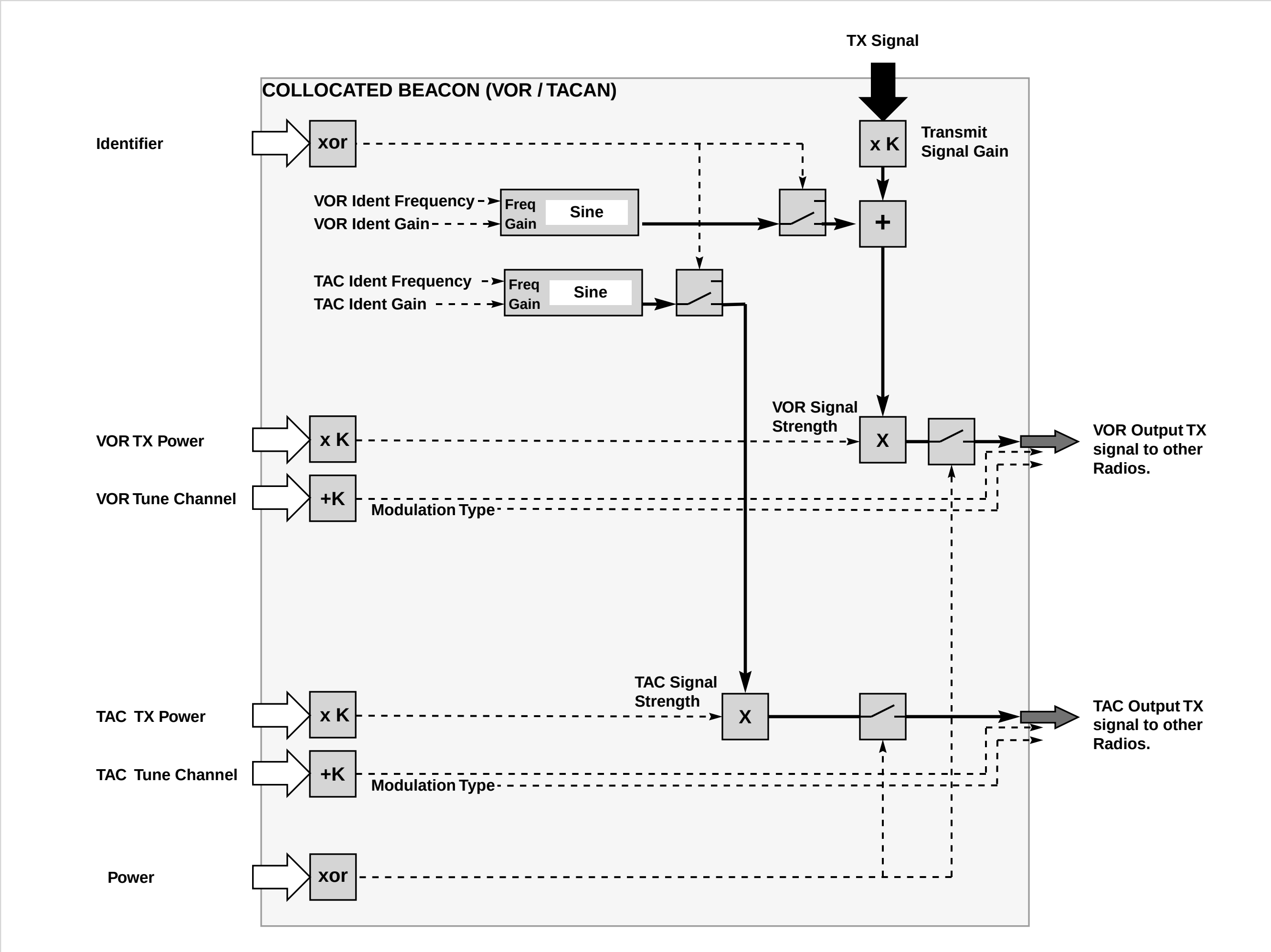
World Position Connection
Connection to a control object that specifies the beacon's world position, or attaches it to a DIS entity.

Collocated VOR/DME

Radio ID

ID number of transmitter that must be assigned to use DIS, Voice Net, or Local Net. The host number, site number, entity number and Radio ID must form a unique set. All numbers but the radio ID are set through the World Position Connection.

Collocated VOR/TACAN



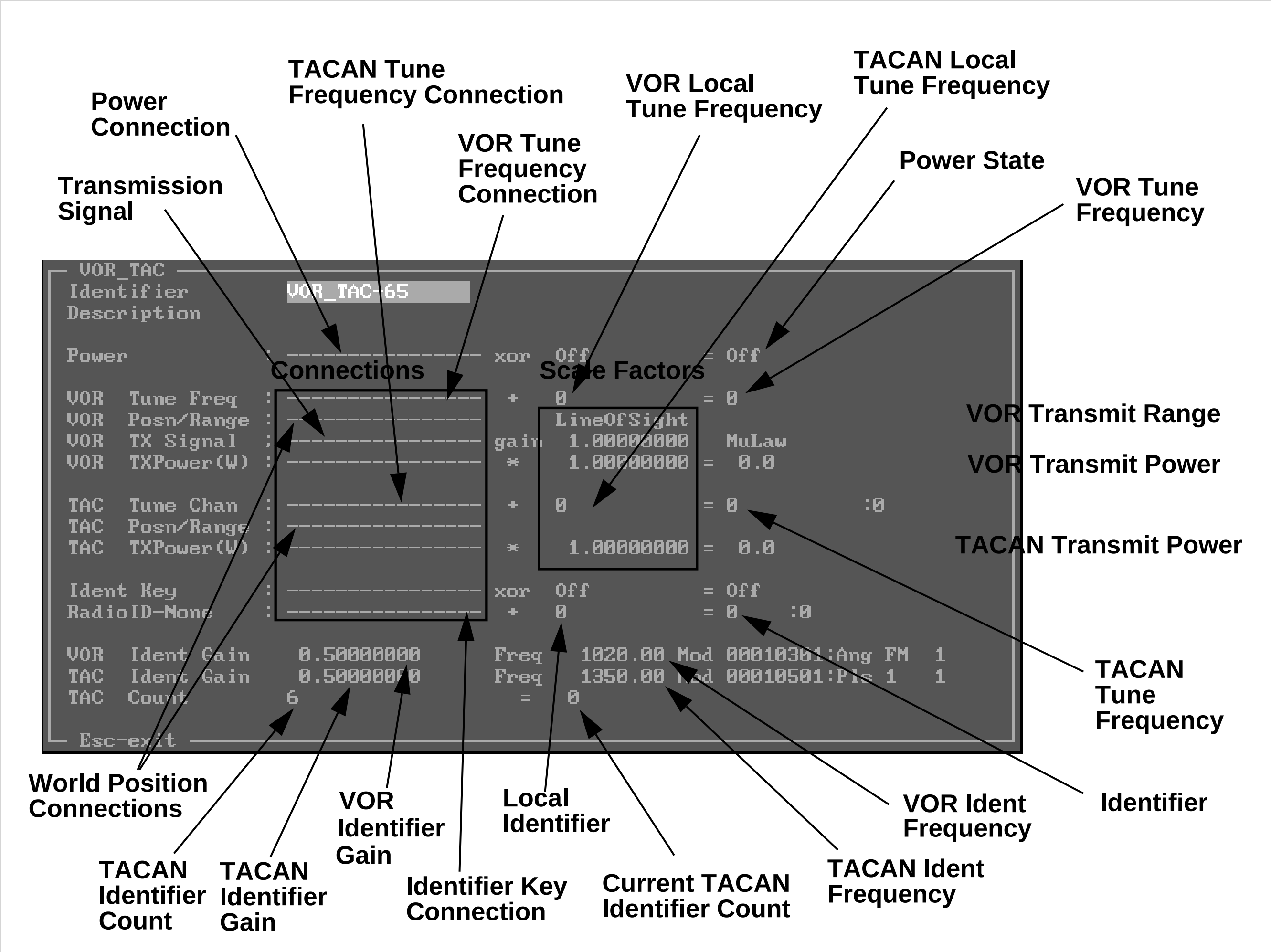
Collocated VOR/TACAN

The Transmitter Object provides a generic, high level radio transmitter simulation including frequency tuning effects, AM or FM bands, and signal strength variation due to range.

The transmission signals are connected through the ‘Transmit Signal’ connection field and are typically navigation station identifier tones, marker tones or voice messages; but since the Transmitter Object uses any signal source combinations of sounds can be transmitted. The resulting transmit signals are broadcast to all Radio Objects in your model, and are received by those tuned to the same frequency and on the same AM or FM band.

The world position connection determines the beacon’s position on the globe. Occulting effects and diminishing power with distance according to an inverse square law are computed based on the world positions.

Collocated VOR/TACAN



VOR Tune Frequency Connection
Connection to a control object which provides the VOR transmitter tune frequency.

VOR Local Tune Frequency
Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the VOR tune frequency connection.

VOR Tune Frequency
Frequency value used for comparison to determine which radios will receive the VOR transmissions. Note: A zero value of frequency disables transmission.

TACAN Tune Frequency Connection
Connection to a control object which provides the TACAN transmitter tune frequency.

TACAN Local Tune Frequency
Provides either a local fixed tune frequency, or an offset for the tune frequency provided by the TACAN tune frequency connection.

TACAN Tune Frequency
Frequency value used for comparison to determine which radios will receive the TACAN transmissions. Note: A zero value of frequency disables transmission.

Collocated VOR/TACAN

Power Connection
Control object connection to provide power control from elsewhere in model.

Local Power State
Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Power State
The transmitter power state, when on and the tune frequency is non-zero the transmitter is active, when off it is inactive, simulating a power off condition.

Identifier Key Connection
Connection to a control object which provides the identifier keying for both VOR and TACAN transmitters.

Local Identifier
Local Identifier state, used to invert identifier.

Identifier
Identifier state, On state produces ident tone, Off state produces no tone.

Transmission Signal
Signal used for transmission on the VOR frequency along with the keyed tone.

VOR Transmit Range
Indicates whether the VOR is capable of transmitting over the horizon or only in a line of sight. This is used by a receiver in conjunction with a world position control. If the transmitter can only operate on a line of sight then the altitude of the receiver will be taken into account such that the reception will be cut off by the curvature of the earth.

VOR Transmit Power Connection
Connection to a control object which provides the VOR transmit power.

VOR Transmit Power Scale Factor
Provides either a local fixed transmission power, or a scaling factor for the VOR transmit power connection.

VOR Transmit Power
Signal strength scale factor for VOR transmission.

TACAN Transmit Power Connection
Connection to a control object which provides the TACAN transmit power.

TACAN Transmit Power Scale Factor
Provides either a local fixed transmission power, or a scaling factor for the TACAN transmit power connection.

TACAN Transmit Power
TACAN transmission power, in Watts.

VOR Identifier Gain
Signal Gain factor for VOR identifier amplitude.

VOR Identifier Frequency
VOR Identifier tone frequency (in Hertz).

TACAN Identifier Gain
Signal Gain factor for TACAN identifier amplitude.

TACAN Identifier Frequency
TACAN Identifier tone frequency (in Hertz).

TACAN Identifier Count
Number of Keyed identifiers to skip before Keying TACAN identifier. The VOR and TACAN identifier tones are mutually exclusive.

Current TACAN Identifier Count
Local count of identifiers skipped.

World Position Connection
Connection to a control object that specifies the beacon’s world position, or attaches it to a DIS entity.

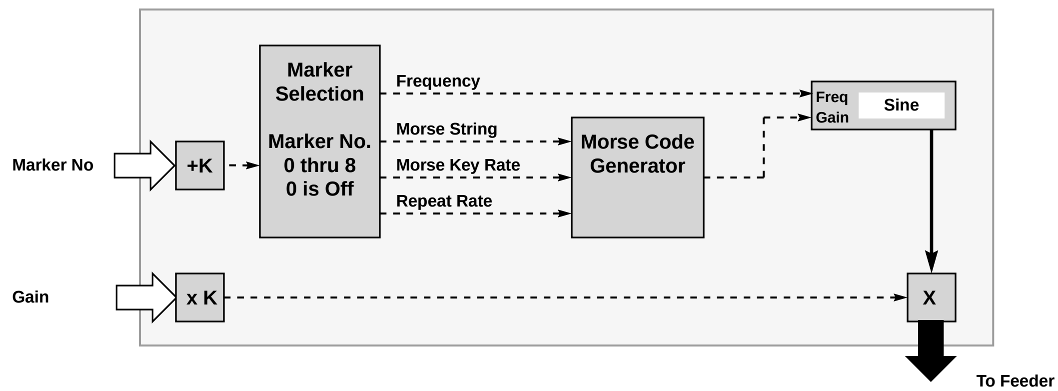
Collocated VOR/TACAN

Radio ID

ID number of transmitter that must be assigned to use DIS, Voice Net, or Local Net. The host number, site number, entity number and Radio ID must form a unique set. All numbers but the radio ID are set through the World Position Connection.

.

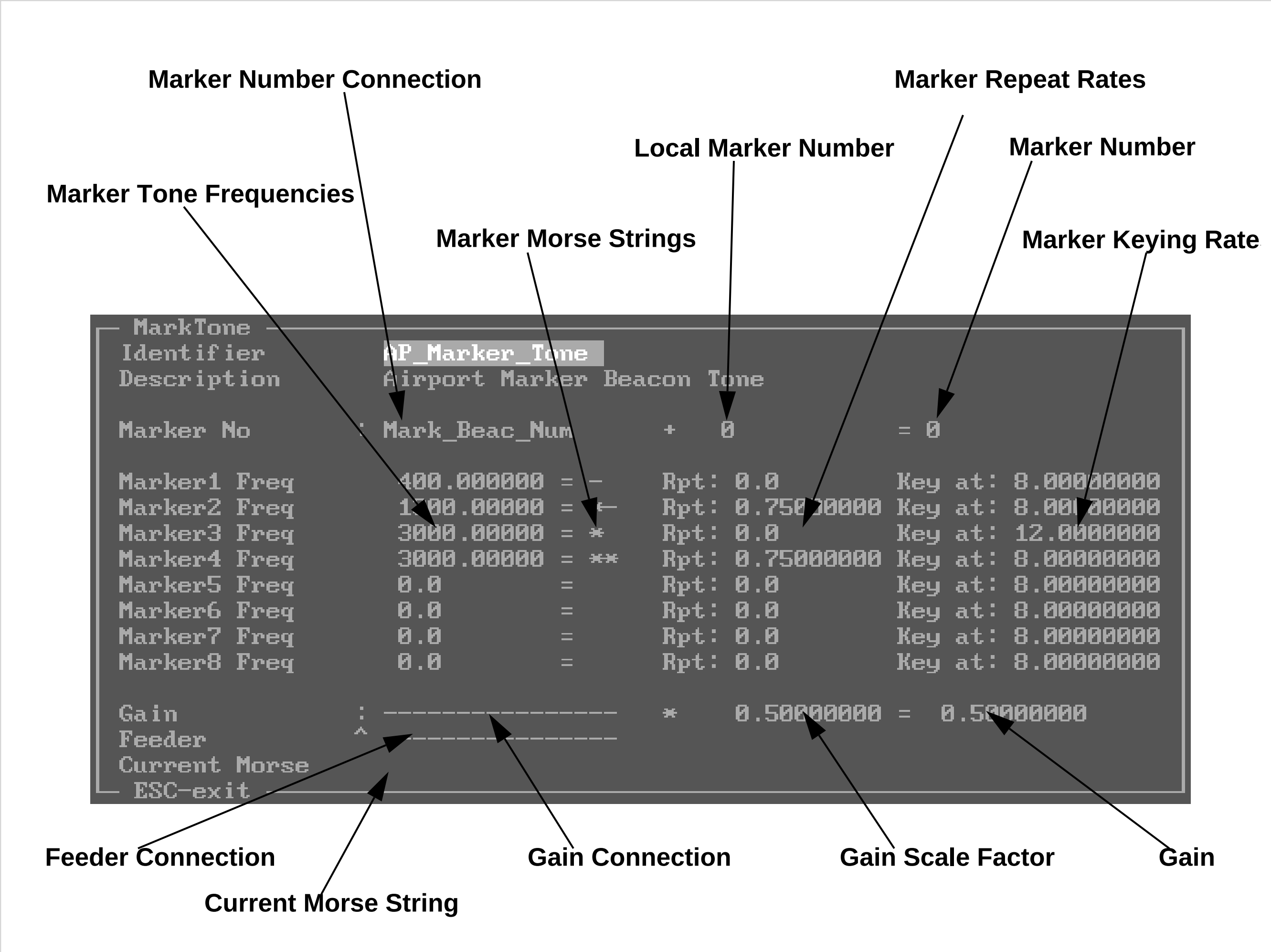
Marker Tone



Marker Tone

This provides a simulation of the tones for a standard marker beacon with support for outer, middle, inner and back course markers. The marker is controlled by an input marker number (0 thru 8) which selects which marker characteristic to generate. Each of the eight types of marker are user definable with separate control of tone frequency, morse key string, morse dot rate and repeat rate.

Marker Tone



Marker Number Connection

Connection to a control object which provides the current marker number (1 thru 8).

Local Marker Number

Provides either a local marker number, or a bit mask of the marker no. provided by the marker connection.

Marker Number

Marker number currently selected (1 thru 8). A marker value of 0 disables the tone generator.

Marker Tone Frequency

Sine wave frequency used for each marker.

Marker Keying Rate

Dot keying rate (in dots per second) to be used for each marker.

Marker Morse String

Four character morse code string used for each of the eight markers. (* and - provide dot and dash capability respectively.

The default configurations are:

Marker 1 is the outer marker (one dash)

Marker 2: middle (dot-dash)

Marker3: inner (all dots)

Marker4: Backcourse (double dots).

Marker Tone

Marker Repeat Rate

Repeat rate (in seconds) for transmission of morse string. Different for each marker. A value of zero means continuous repetition, with no gaps at the end of the string.

Gain Connection

Connection to a control object which provides the sine wave amplitude gain.

Local Gain Value

Value for gain, or scaling factor for gain connection value.

Gain

Amplitude gain of waveform generated by synthesizer.

Current Morse

Morse string in use currently.

Feeder Connection

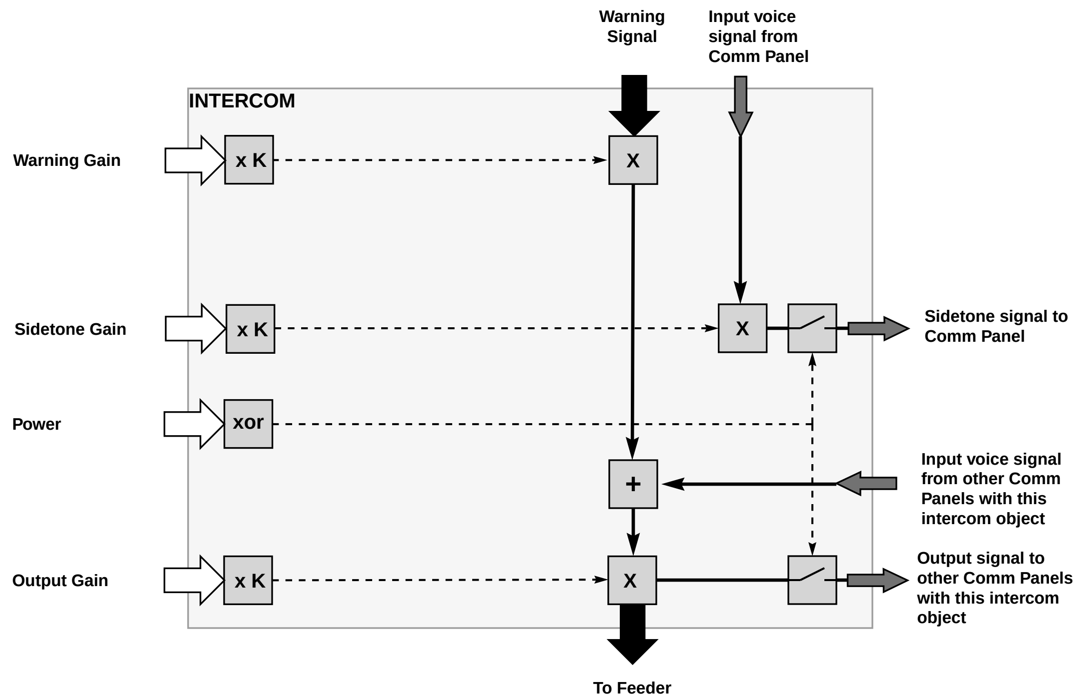
Connection to a feeder, which provides a mechanism for outputting the marker tone onto the signal highway. This field can be left blank if the signal is to be picked up by a selector or mixer.

Intercom Signals (Intercom Package)

The Intercom signal is a special object for intercom, interphone and Public Address communication simulation. It works with the Comm Panel objects described later. The Intercom signal acts as an intercom bus.

It is possible to connect intercom buses up over a DIS or VoiceNet network. This is described at the end of the section.

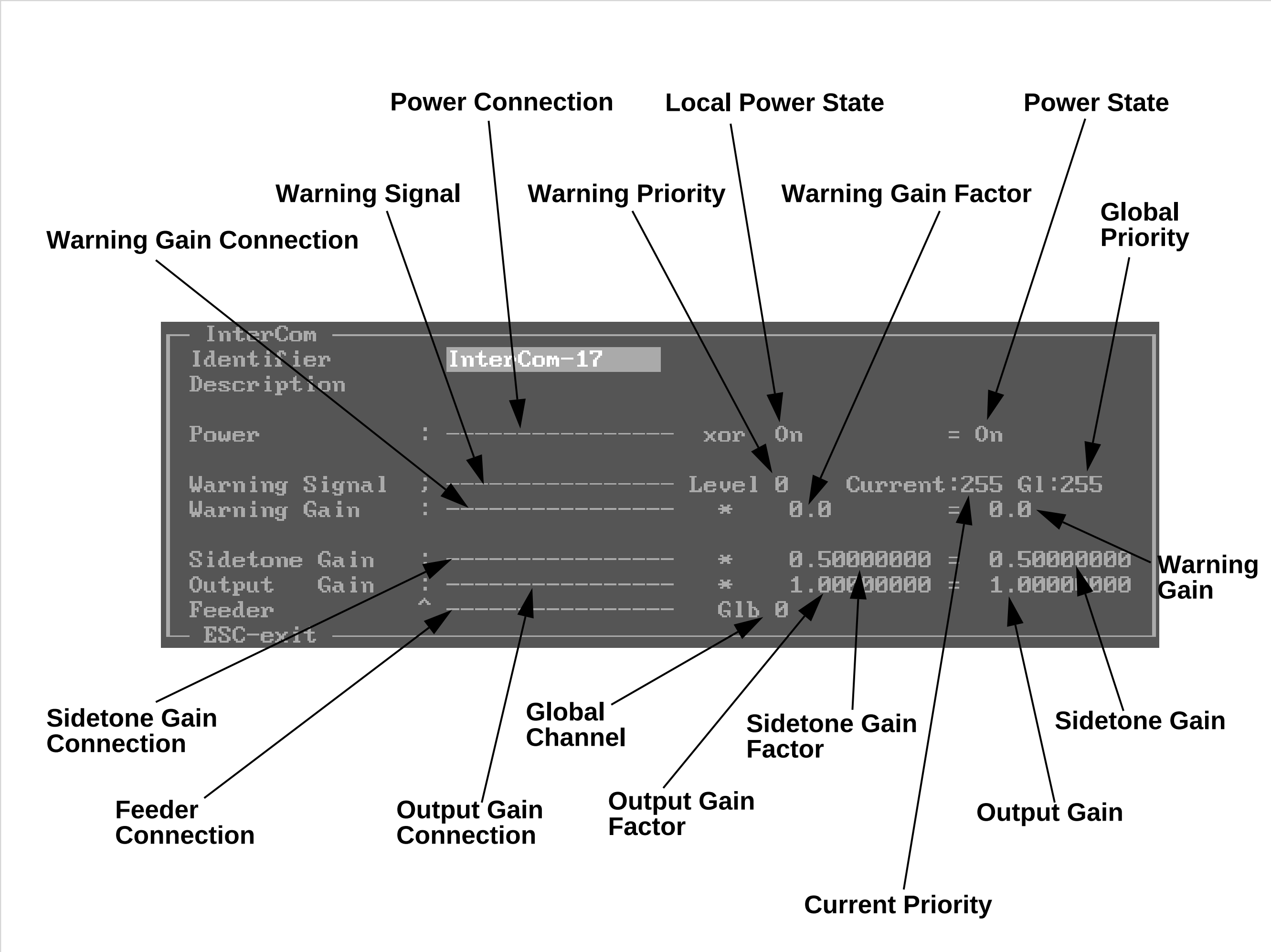
Intercom



Intercom

The Intercom Object provides a simple simulation of an intercom bus between communication panels. Any input signal from a Comm Panel object will be mixed with an optional warning signal and routed to any other selected Comm Panel or feeder output.

Intercom



Power Connection

Control object connection to provide power control from elsewhere in model.

Local Power State

Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state

Power State

The intercom power state, when on the intercom bus is active, when off it is inactive, simulating a power off condition.

Warning Signal

The signal connection for any tones that need to be mixed into the intercom, such as voice alerts, or warning tones.

Warning Priority

The priority level for the warning signal. When used with a comm select panel feeder that supports priority levels, all voices and warning signals can be prioritized such that only the highest priority signals can be heard (1 is the highest priority and 255 is the lowest.) If two or more signals have the same priority, they will be mixed together. Signals with lower priorities will be muted.

Intercom

If the priority is set to zero, then the warning signal will be heard by everyone but will not override any other signals.

Current Priority

Priority level currently asserting itself on the intercom bus. This is the highest priority signal the Intercom object is receiving. All signals of this priority will be heard. 1 is the highest priority, 255 is the lowest.

Warning Gain Connection

Host control of warning gain.

Warning Gain Scale Factor

Local value or connection scale factor for warning gain.

Warning Gain

Warning signal gain.

SideTone Gain Connection

Host control of sidetone gain.

SideTone Gain Scale Factor

Local value or connection scale factor for sidetone gain. Note: If there is no sidetone connection and this gain factor is zero then the sidetone capability is disabled.

SideTone Gain

If the comm select panel that is attached to the Intercom is not using local sidetones, then this sidetone gain will control the volume of the sidetone coming from the intercom.

Output Gain Connection

Overall gain connection for host control of intercom volume or on/off capability.

Output Gain Scale Factor

Local value or connection scale factor for output overall gain.

Output Gain

Overall output gain for intercom.

Global Channel

Used to connect to intercom buses on different DSPs. To use this feature, set up an Intercom object in each model. Assign each of them the same number in this field. The intercoms will act as a single intercom bus.

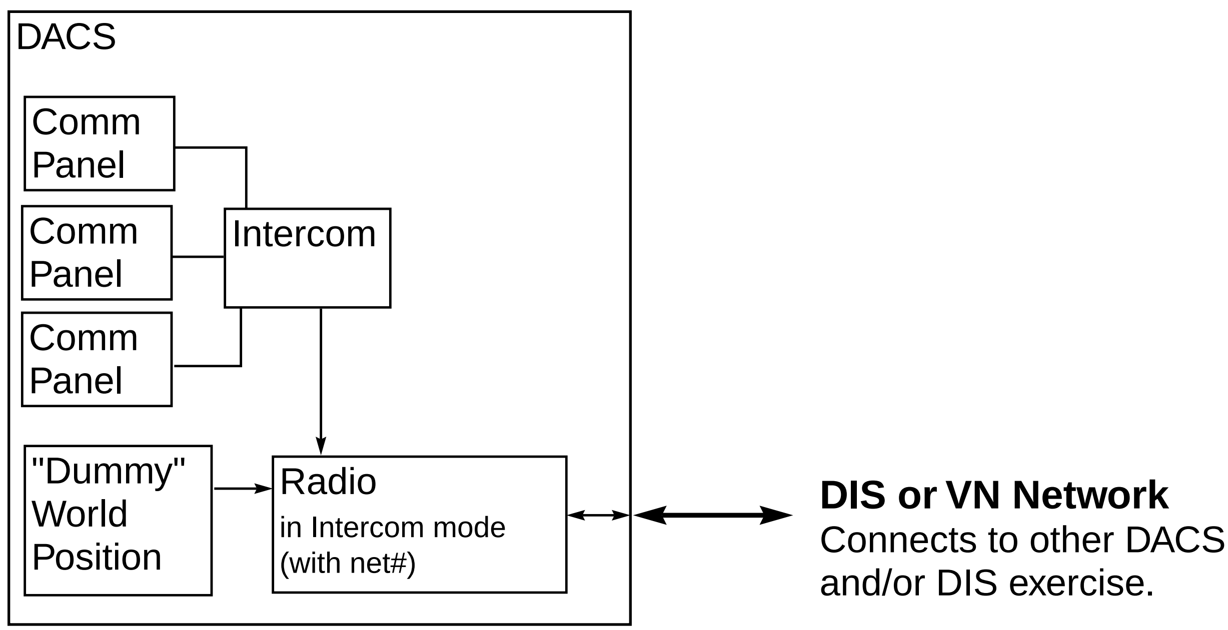
Global Priority

For intercom objects on separate DSPs connected by a global channel, this number indicates the highest priority level coming from the other DSP boards.

Feeder Connection

Connection to a feeder, which provides a mechanism for outputting the intercom onto the signal highway. This field can be left blank if the signal is to be picked up by a Comm Panel

Intercom over DIS or VoiceNet



Intercom over DIS or VoiceNet

Local intercom buses can be connected together over a DIS or VoiceNet network. It requires the following software options:

MB-Intercom

MB-Radio

MB-DIS or MB-VoiceNet

The basic idea is that the local intercom buses are attached to Radios which are set to a special “Intercom” mode. The radios connect intercom buses on different systems, based on assigned net numbers.

For each intercom bus to be connected to the DIS network, create a radio object. Set the following radio parameters:

Power/Mode = 8 (Selects the Intercom mode)

- **Tune Frequency** = net #. (This can be set by the offset or dynamically through the tune frequency connection. It should be between 1 and 100000). Intercom buses on the network with the same net numbers will be linked together. They should all have the same exercise number as well.
- Connect the desired intercom object into the **Transmit Signal** field of the radio.
- Insert a world position into the **World Position** connection of the Radio. The world position object should be set to DIS (or VoiceNet), and given a valid entity and exercise ID. The actual position is ignored.
- The rest of the radio parameters should be left at the default values. Most of them are ignored in the intercom mode.

As a final step, it's necessary to go into the model configuration window and change the "DIS RXs / Radio" Field. This determines the number of simultaneous audio streams the intercom can pick up from the network. Each DACS broadcasts one audio stream per intercom bus, so this number can be set to one less than the number of DACS on the network.

IMPORTANT NOTE: An intercom bus that is attached to an "intercom radio" CANNOT also have a global channel attachment. Only local intercom buses can be attached over DIS as is described in this section.

If you wish to have an intercom bus that connects across the network AND accross DSP cards in one DACS, you should set up local intercom buses in each model, and attach EACH of them to their own "intercom radio". This will automatically link up the buses on the different DSP cards.

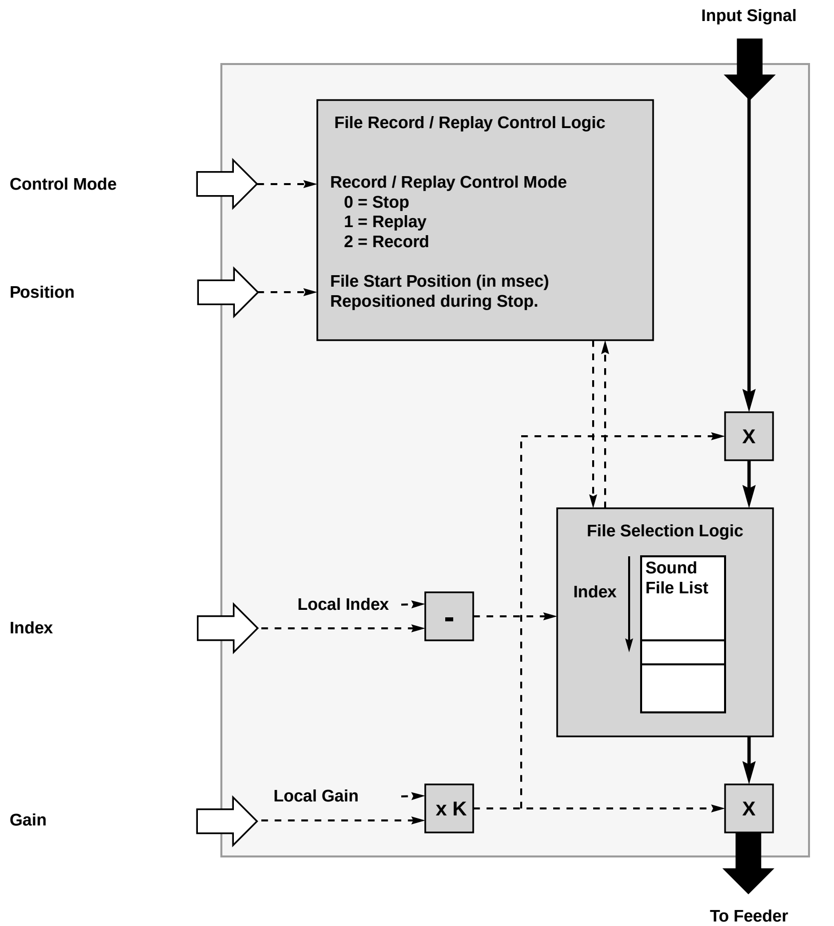
NOTE TO ADVANCED USERS:

The DIS RXs / Radio field determines the amount of buffer space allocated for the radios in the model. The trade off is between the number of audio streams that can be received and the total number of radios in the model.

Because this field affects ALL the radios in the model, it is a good idea to have all of the "Intercom" radios in a separate model from the other DIS (or VoiceNet) radios. Otherwise, the non-intercom radios will have more buffer memory allocated to them than they need.

Record Replay Signals (Record/Replay Package)

Record / Replay



Record / Replay Signal

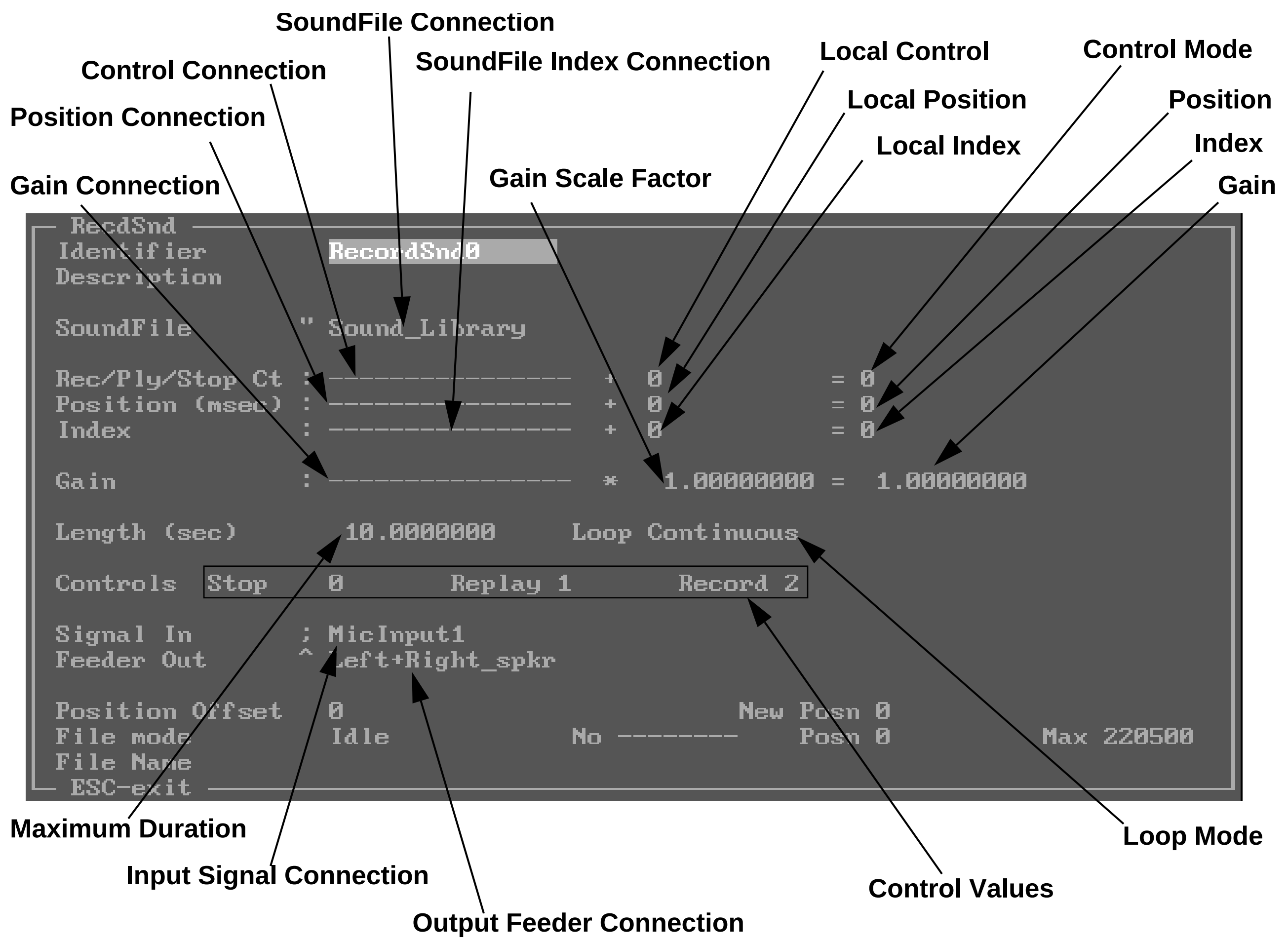
The record replay object provides the ability to record or replay any signal to or from a selected file on the hard disk, under host computer control. It is anticipated that this will be used with analog input or mixer signals to provide for any combination of input voice recording.

For replay, control is provided over which combination of the highway channels receive the composite sound on replay. This allows different combinations to be used for recording and replaying, e.g. a single channel record such as pilot's headset output to be replayed to all of the headsets.

A position input provides the host with a repositioning capability for both record and replay.

The record object can be linked to a file group such that a different series of files can be used for the recording process. This is useful for demonstration or debrief purposes.

Record / Replay



SoundFile Connection

Sound file selection from model's soundfile directory list

Control Connection

Record / replay mode connection to elsewhere in model.

Local Control

Provides local offset for control mode value (useful for testing only).

Control Mode

The record/replay control variable, with defaults of 0 = Stop, 1 = Replay, 2 = Record.

Position Connection

File position connection to elsewhere in model.

Local Position

Provides local offset for position value (useful for testing only).

Position

File start position (32 bit integer in milliseconds) from start or end of file. This value is used to reposition the file while in the stop mode. When in continuous loop mode, a positive value provides an offset forward in the file, while a negative offset repositions the file back from the last stop position. When not in loop mode, a positive position provides an offset from the start of the file, while a negative number is an offset from the end of the file. A value of zero leaves file pointer at it's current position.

Record / Replay

Soundfile Index Connection

Control object connection to provide file selection control from elsewhere in model.

Local Index

Local value for the file index. If a connection is made then this value is subtracted from the incoming index value, else it provides a default value for the file index.

Index

The current value of the file index. This index is used to select one of a group of files from the sound files list. It is only active when the soundfile connection is made to a file group, and the index matches that of one file in the list. If no matches are found then no file is recorded or replayed.

Gain Connection

Control object connection to provide input/output amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for input/output gain control value.

Gain

Input/output amplitude gain of file record source. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Continuous Loop Mode

The continuous loop mode sets the recording and replay such that the effective end of recording marker moves in a continuous loop. This allows the file to act as a recorder for the last few minutes of sound which is continuously overwritten until a stop command is issued. If this mode is Off the file is recorded once only, completing when the maximum duration is reached.

Control Values

The record/replay control variable, uses default values of 0 = Stop, 1 = Replay, 2 = Record. These fields allow the user to define other more convenient mappings of these three commands.

Maximum Duration

The maximum record/replay time allocated to the soundfile (in seconds).

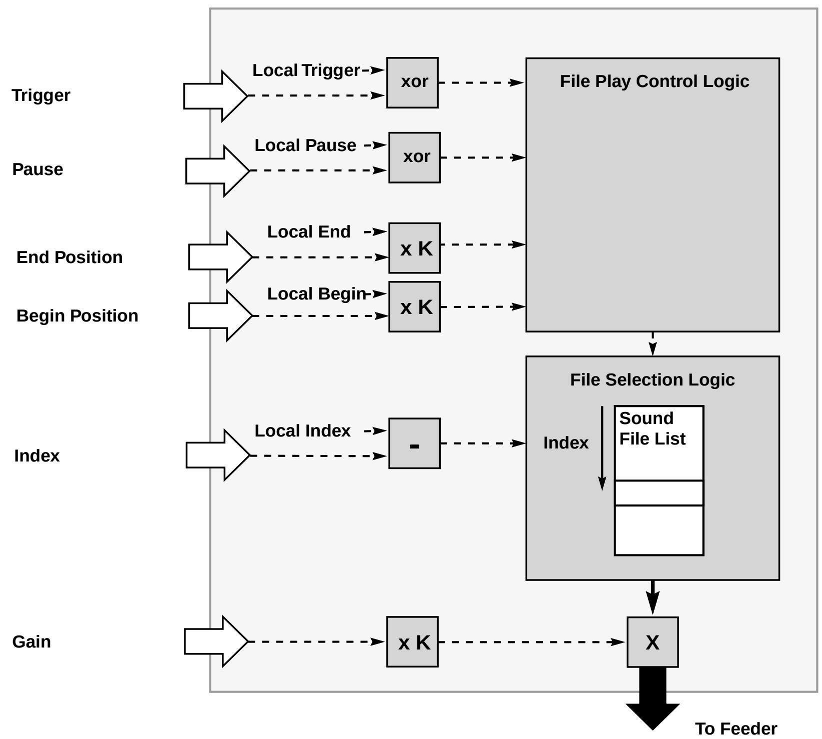
Input Signal Connection

Connection to a signal, which provides the source for the recorder.

Output Feeder Connection

Connection to a feeder, which adds the replayed sound source into the signal highway.

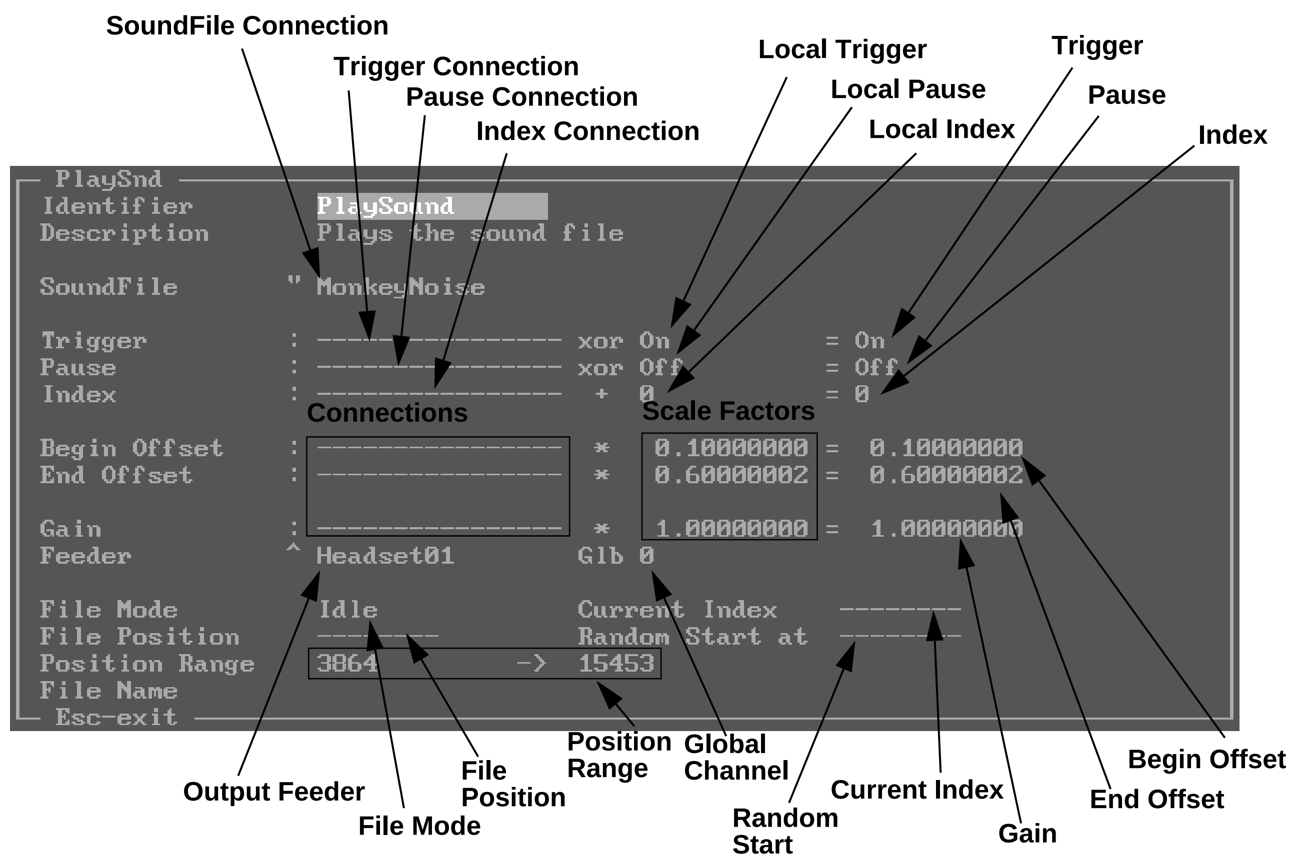
Play Sound



Play Sound Signal

The play sound object provides the ability to play back sound files recorded which have been recorded to disk. It is designed to provide flexibility in replaying of sounds, featuring begin and end offsets which can be used to trim off the beginning and end of the sound files. In addition, it should be used when the sound file to be played is set to start in a random position.

Play Sound



SoundFile Connection

Connection to a Soundfile or Sound Group (a group of sound files.) See the Sound File Section later in the manual.

Gain Connection

Control object connection to provide input/output amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for output gain control value, or gain value if there is no Gain Connection.

Gain

Output amplitude gain of file record source. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Output Feeder

Connection to a feeder, which adds the replayed sound source into the signal highway.

Begin Offset Connection

Connection to a control object that provides the Begin Offset.

Begin Offset Scale Factor

Scale factor for the Begin offset connection. If the Connection field is empty, this provides the Begin Offset.

Play Sound

Begin Offset

Indicates the position from the beginning of the sound file that the sound will start playing. This number is between 0 and 1, and is the fraction of the file length. For example, a 5 second file with a Begin Offset of 0.1 would start one half second into the file.

End Offset Connection

Connection to a control object that provides the End Offset.

End Offset Scale Factor

Scale factor for the End offset connection. If the Connection field is empty, this provides the End Offset.

End Offset

Indicates the position from the end of the sound file that the sound will stop playing. This number is between 0 and 1, and is the fraction of the file length. For example, a 5 second file with an End Offset of 0.1 would end one half second before the end of the file.

Trigger Connection

Connection to a control object that provides the Trigger.

Local Trigger

Provides an exclusive or for the trigger connection object.

Trigger

When the trigger value goes from off to on, the sound begins playing. If the trigger is turned off while the sound is playing, it will terminate the sound if the Play field in the Sound file is set to Normal. (See the Sound File section later in the manual.) Retriggering after that will cause the sound to start again at the beginning.

Pause Connection

Connection to a control object that provides the Pause.

Local Pause

Provides an exclusive or for the Pause connection object.

Pause

While the sound is playing, if the pause goes on the sound stops playing. Turning Pause off causes the sound to resume playing from where it was paused.

Index Connection

Control object connection to provide file selection control from elsewhere in model.

Local Index

Local value for the file index. If a connection is made then this value is subtracted from the incoming index value, else it provides a default value for the file index.

Index

The current value of the file index. This index is used to select one of a group of files from the sound files list. It is only active when the soundfile connection is made to a file group, and the index matches that of one file in the list. If no matches are found then no file is replayed. If the index is changed when the Trigger is on, it will initiate the playing of the indexed soundfile from the beginning.

Current Index

Indicates the index of the file in the soundgroup that is currently playing.

File Mode

Indicates whether the sound file is IDLE or PLAYING.

File Position

Current position in the file. This number is only relevant when the file is playing. The position is given in terms of the number of sound samples - e.g., if the DSP is running at an 8 kHz sample rate, when the play is 1 second into the file the position will be 8000.

Position Range

Indicates the range in the file that is currently being played. This is determined by the Begin and End Offsets, and is given in terms of the number of samples.

Play Sound

Random Start

If the sound file is set to Random Start, when the sound is triggered it will take it's starting position from the counter in this field. If the file is not designated random start, this field is empty. (See the Sound Files Section.)

Global Channel

Connection to a Global Channel. If this field is non-zero, the same sound that gets sent to the feeder connection will be sent to the global channel, where it can be picked up by other models running on other DSP boards. See the section on Global Channels for details.

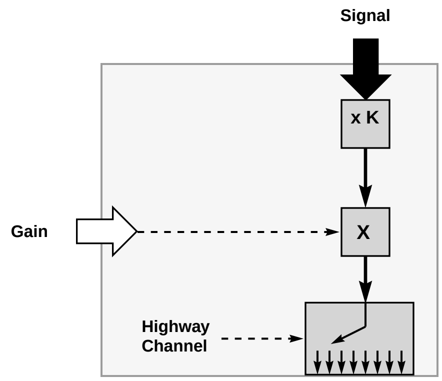
Feeders

Feeders are the objects that connect either a signal source or analog input through to the mixing highway. They are divided into:

- Basic feeders, which provide connection for one signal onto one or more channels of the highway
- Mixing feeders which allow for simple selection of multiple signals onto a single highway channel
- Analog input feeders, which provide summation of one or more analog input sources onto a single highway channel
- Analog output feeder, which connects a series of highway channels to the analog outputs

Basic Feeders

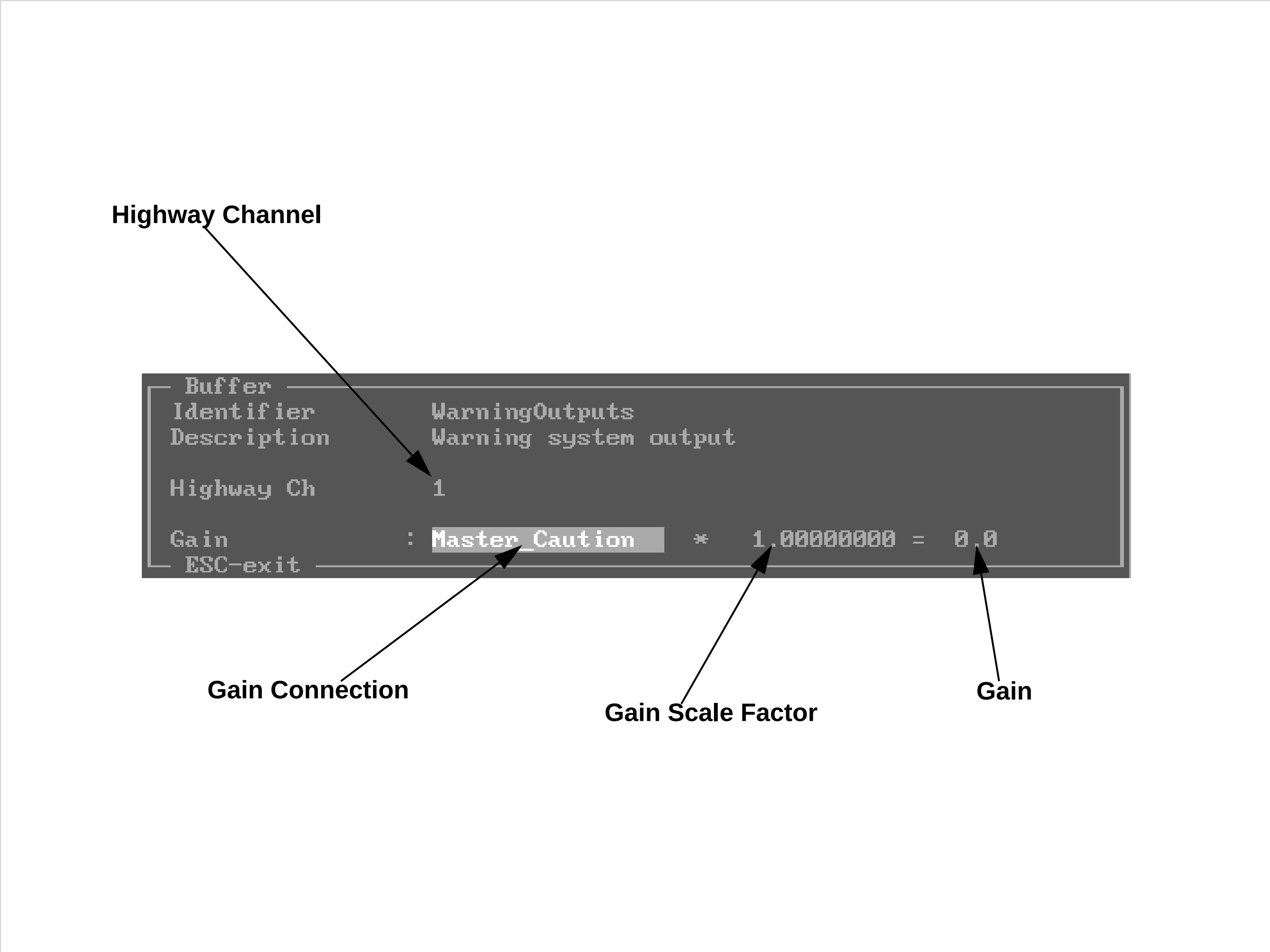
Buffer



Buffer

The buffer is a feeder which allows a signal source to be mixed onto a single channel on the mixing highway

Buffer



Highway Channel

Highway channel to which signal source will be added (1 thru 8).

Gain Connection

Connection to a control object which provides the overall amplitude gain.

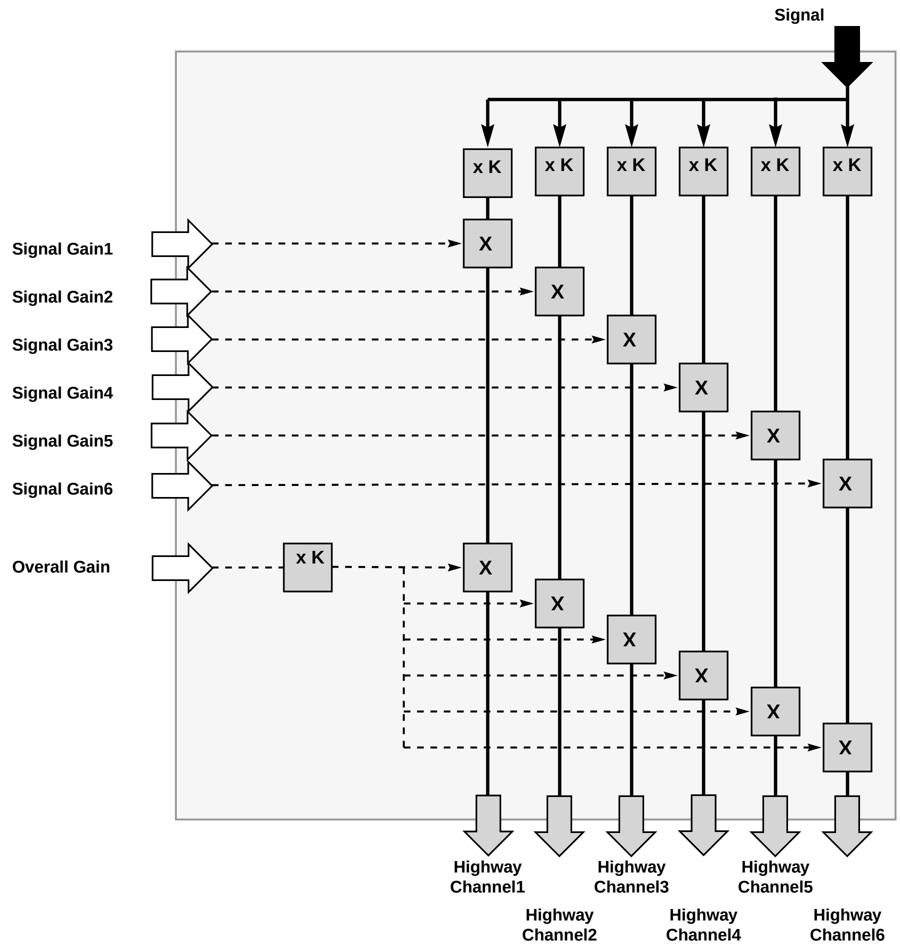
Local Gain Value

Value for gain, or scaling factor for gain connection value.

Gain

Amplitude gain of waveform generated by synthesizer.

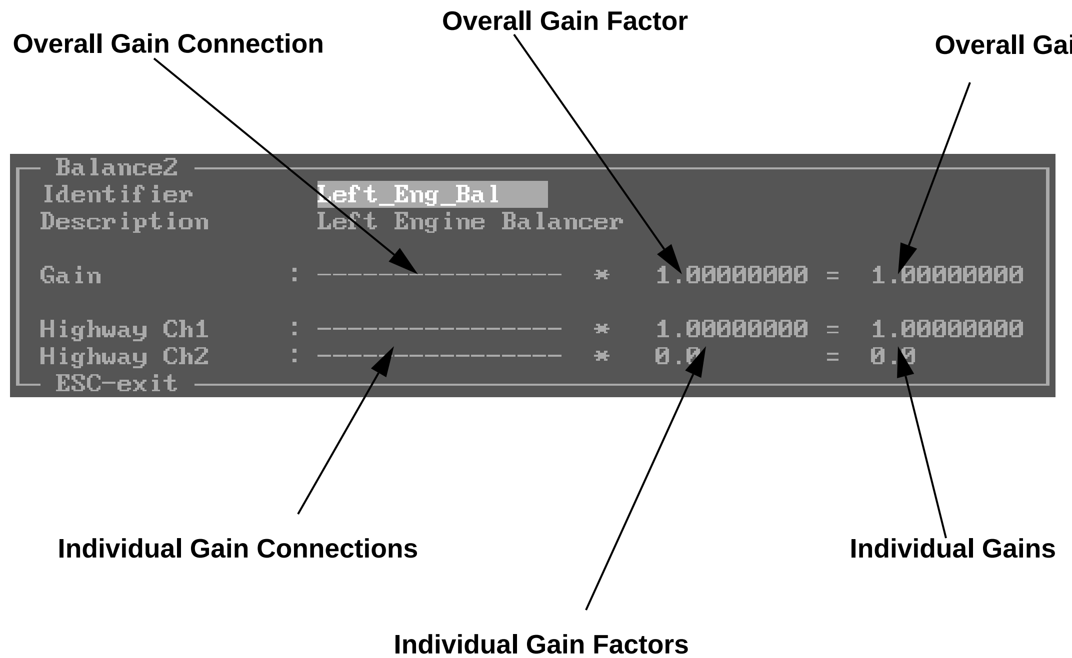
Balancer



Balancer

The balancer is a feeder which allows a single signal source to be mixed onto a number of channels on the mixing highway. This allows you to 'balance' the signals between highway channels in a similar way to which the balance control works on a hi-fi system

Balancer



Overall Gain Connection

Connection to a control object which provides the overall amplitude gain for all highway feeds.

Overall Gain Factor

Value for gain, or scaling factor for overall gain connection value.

Overall Gain

Overall amplitude gain of waveform generated by synthesizer.

Individual Gain Connection

Connection to a control object which provides the amplitude gain for each of the individual highway feeds.

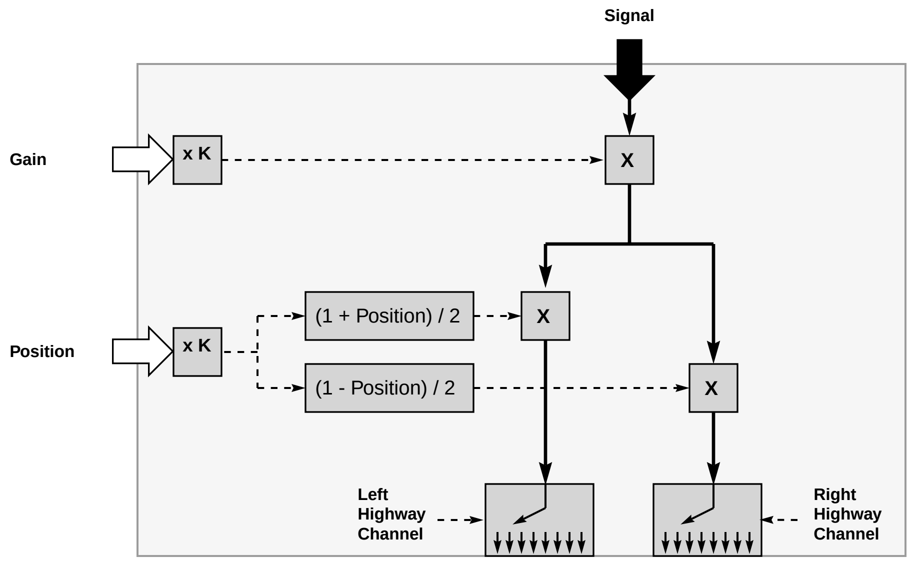
Individual Gain Factor

Value for gain, or scaling factor for each individual gain connection value.

Individual Gain

Amplitude gain of waveform generated by synthesizer for each of the highway feeds.

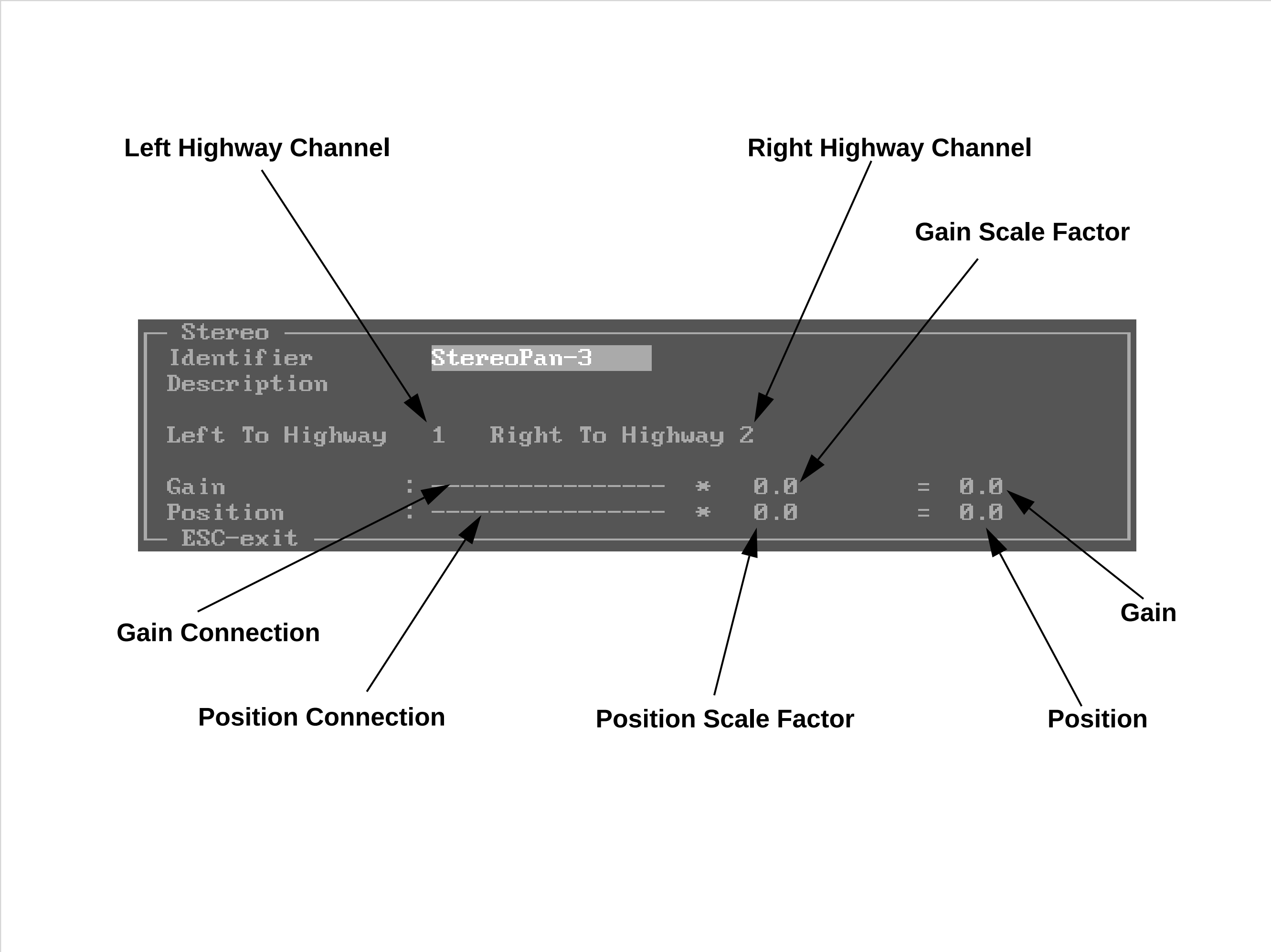
Stereo Pan



Stereo Pan

The Stereo Pan allows a single source to be mixed between two channels, such that it can be panned from one side to the other with a single control.

Stereo Pan



Left Highway Channel

Predominant Highway channel to which signal source will be added when Position > 0.0 (1 thru 8).

Right Highway Channel

Predominant Highway channel to which signal source will be added when Position< 0.0 (1 thru 8).

Gain Connection

Connection to a control object which provides the overall amplitude gain.

Gain Scale Factor

Value for gain, or scaling factor for gain connection value.

Gain

Amplitude gain of waveform generated by synthesizer.

Position Connection

Connection to a control object which provides the position.

Position Scale Factor

Value for position, or scaling factor for position connection value.

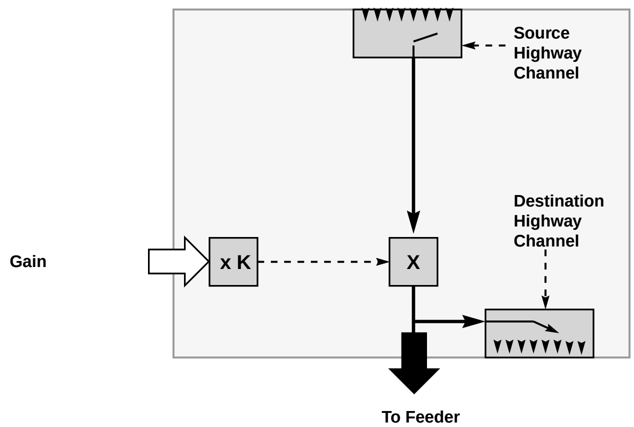
Position

Value of position, 0.0 allows signal to be equally mixed between left and right channels. If position >= 1.0 then signal is only on left channel. If position <= -1.0 then only on right. Values between 1.0 and -1.0 provide sound in each channel, with:

$$LeftGain = (1 + Position) / 2$$

$$RightGain = (1 - Position) / 2$$

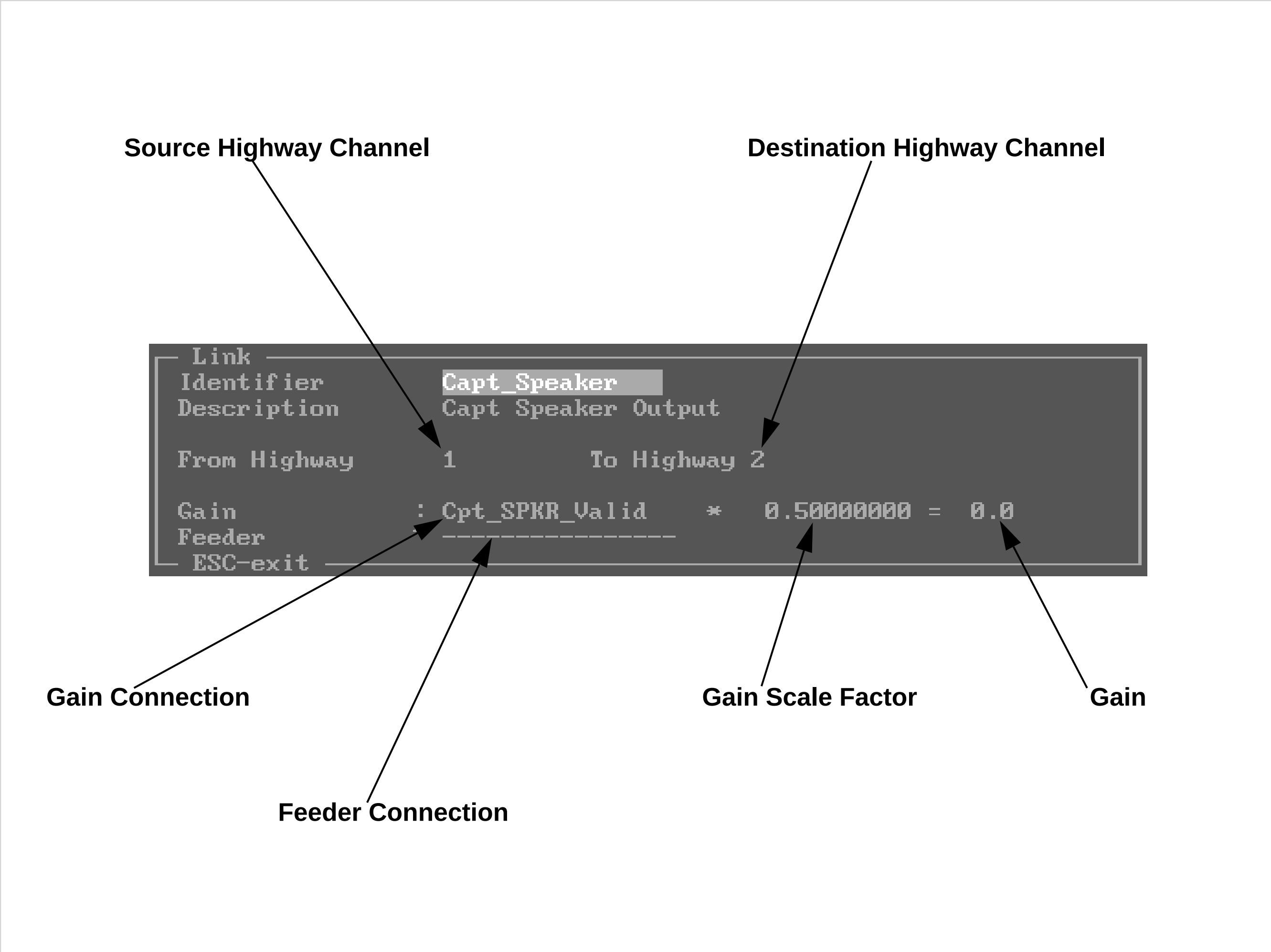
Link



Link

This feeder provides a way of linking the output of one highway channel to another. The source channel can be gain controlled and mixed into one or more other highway channels. Note: care should be taken to ensure that there is no feedback from the destination channels to the source channel, else there can be either a feedback howl effect, or a d.c. level build up resulting in the signal output being driven to one or other rail.

Link



Source Highway Channel

The highway channel used as the source for the signal to be linked to one or more destination highway channels. In a master/slave synthesizer configuration, a negative value is used to access the highway channel on the other synthesizer (e.g. -3 selects highway channel 3 on the other synthesizer as the source highway channel).

Destination Highway Channel

The highway channel into which the source channel is to be mixed. Note: The feeder connection allows more than one channel to be used as the destination. In that case the destination channel should be set to zero.

Gain Connection

Connection to a control object which provides the overall amplitude gain.

Local Gain Value

Value for gain, or scaling factor for gain connection value.

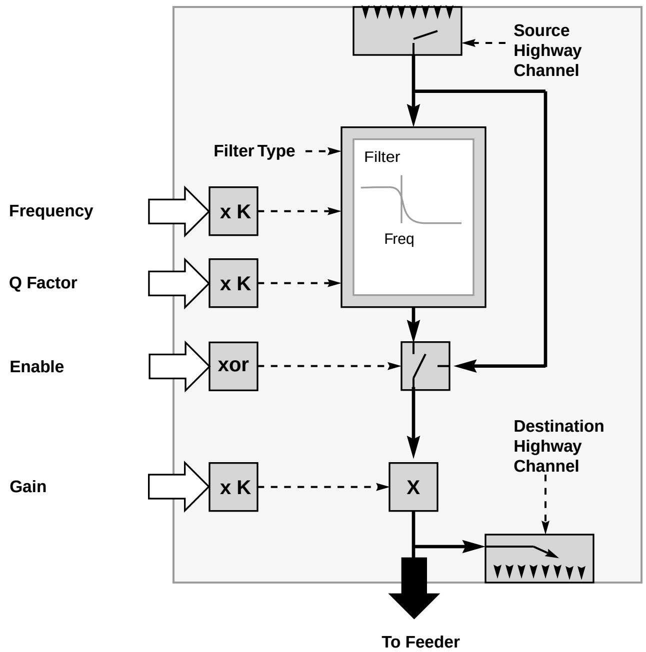
Gain

Amplitude gain of signal linked from source to destination highway.

Feeder Connection

Connection to a feeder, which adds the source highway channel into the signal highway.

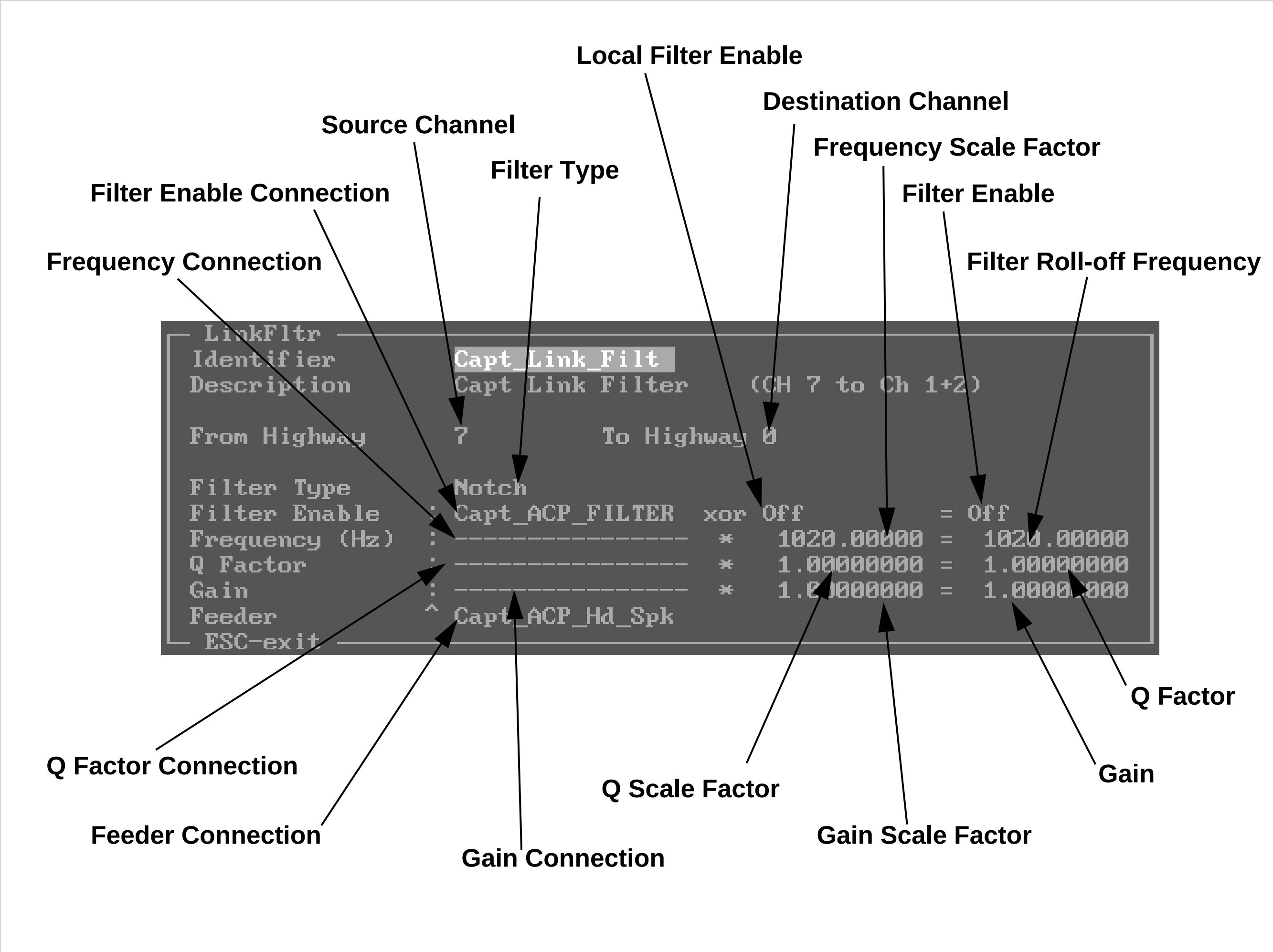
Link Filter



Link Filter

This feeder provides a way of linking the output of one highway channel to another via a filter. The source channel can be gain and frequency controlled and mixed into one or more other highway channels. The type of filtering can be selected from low-pass, band-pass or high-pass. The filter quality factor, roll-off frequency, and gain can be controlled by input variables from elsewhere in the model, or from the host interface. Note: care should be taken to ensure that there is no feedback from the destination channels to the source channel, else there can be either a feedback howl effect, or a d.c. level build up resulting in the signal output being driven to one or other rail.

Link Filter



Source Highway Channel

The highway channel used as the source for the signal to be linked to one or more destination highway channels. In a master/slave synthesizer configuration a negative value is used to access the highway channel on the other synthesizer (e.g. -3 selects highway channel 3 on the other synthesizer as the source highway channel).

Destination Highway Channel

The highway channel into which the source channel is to be mixed. Note: The feeder connection allows more than one channel to be used as the destination. In that case the destination channel should be set to zero.

Filter Type

Selects a two pole filter type from; Low-pass, Band-pass, High-pass, Low-passQ, Band-passQ, High-passQ, Notch or AllPass. The three Q filters are amplitude adjusted such that the filter has unity gain at the roll-off frequency, and maintains this gain as the quality factor is increased. The band-pass filters have the low-pass and high-pass poles at the same roll-off frequency.

Filter Enable Connection

Control object connection to provide filter enable control from elsewhere in model.

Link Filter

Local Filter Enable

Local value for the filter enable. If an enable connection is made then this provides an exclusive or function for inverting the enable control state

Filter Enable

The current filter enable state, when on the filter will be active, when off the signal will be passed through with gain control only.

Frequency Connection

Control object connection to provide overall frequency control from elsewhere in model.

Frequency Scale Factor

Scaling factor for frequency control value.

Frequency

Roll-off frequency (in Hertz) of filter. If the frequency connection is blank then the frequency scale factor is used as the frequency value, else the frequency is the scale factor times the output result of the control object.

Q Factor Connection

Control object connection to provide quality factor control from elsewhere in model.

Q Scale Factor

Scaling factor for Q factor control value.

Q Factor

Quality factor for filter. If the Q factor connection is blank then the Q scale factor is used as the Q value, else the Q is the scale factor times the output result of the control object.

Gain Connection

Connection to a control object which provides the overall amplitude gain.

Local Gain Value

Value for gain, or scaling factor for gain connection value.

Gain

Amplitude gain of signal linked from source to destination highway.

Feeder Connection

Connection to a feeder, which adds the source highway channel into the signal highway.

RIU Feeders (RIU Package)

A model can have up to 24 audio channels coming into it (the *input highways*) and 24 audio channels being sent out (the *output highways*). In a system with RIU’s, all of the audio comes from and goes to the RIU’s.

The model takes sound from the input highways through the Audio In and Vox objects. It puts sound out to the output highways with a variety of feeders, including buffers, balancers, and others.

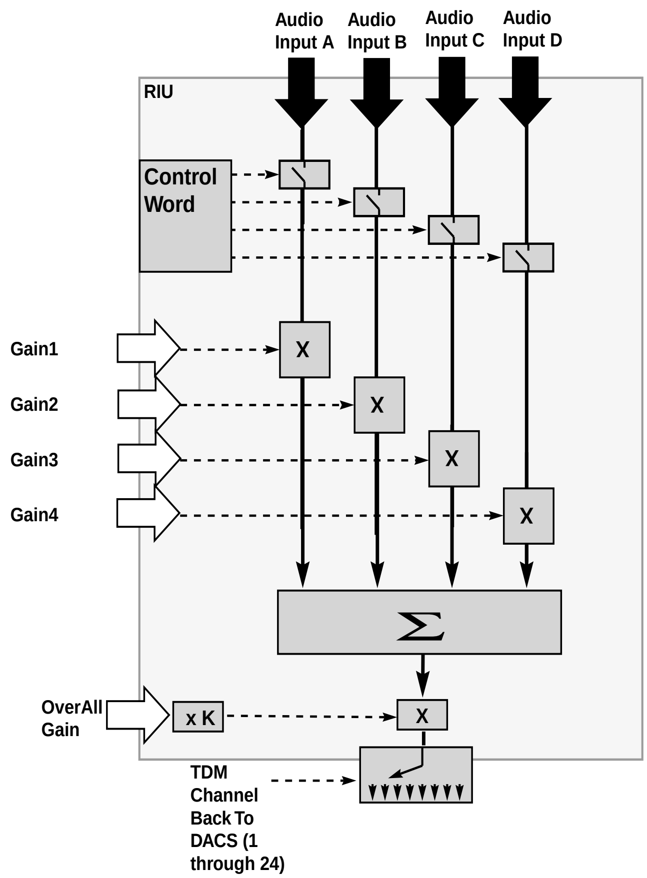
Each RIU has four analog inputs, four analog outputs, CODECs, and a DSP processor. Because it has an on board DSP processor, it can mix sounds locally.

The RIU feeders map the input and output highways to the RIU inputs and outputs. Each TDM card can be attached to up to 16 RIUs, giving a total of 64 audio inputs and 64 audio outputs. Since a model can only handle 24 inputs and 24 outputs, not all of these audio inputs and outputs can be used independently.

The mappings do not need to be one to one - a single output highway can be sent to many RIU outputs, and inputs on a single RIU can be mixed together on the RIU and assigned to a single input highway. The feeders also allow sound to be routed from an RIU input to an output on the same RIU, without using any input or output highways.

If two output highways are assigned to the same RIU output, they will get mixed together on the RIU.

RIU Audio Input



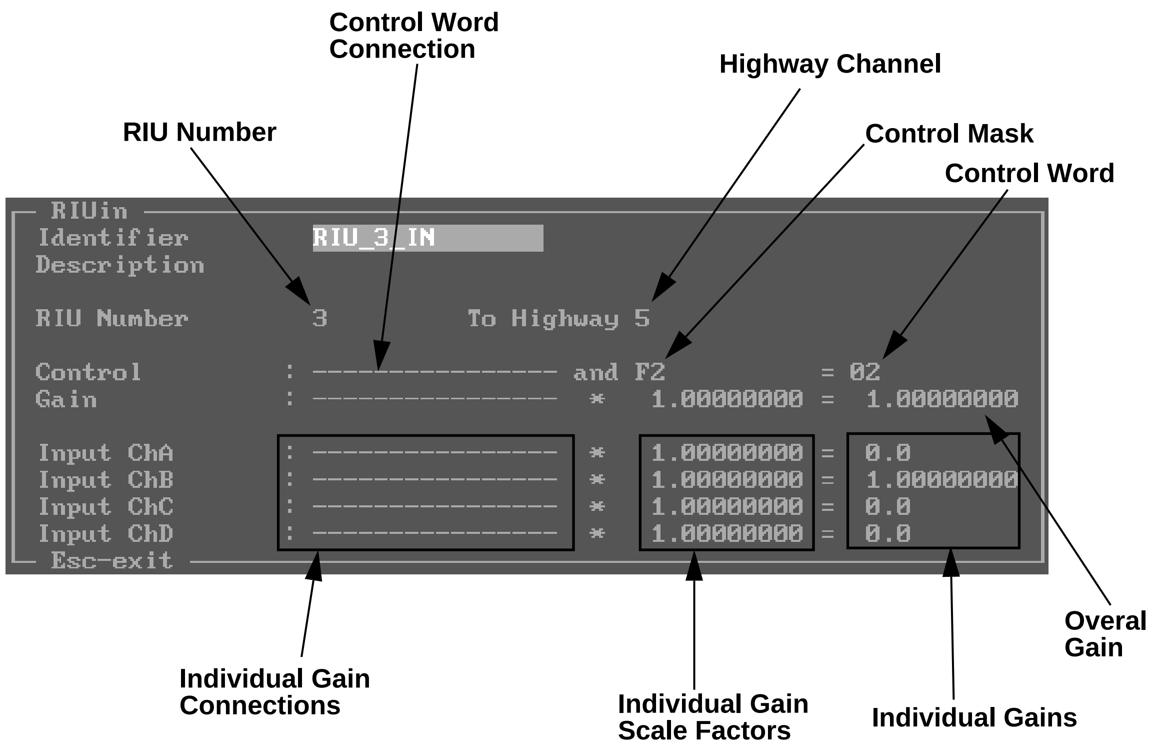
RIU Audio Input

The RIU audio input feeder assigns the audio inputs from a single RIU to one of the input highways.

This feeder specifies a particular RIU. The four audio inputs from the RIU are mixed together and assigned to an input highway. Once the input highway is assigned, the sound is brought into the model by using the Audio In object or the Vox object.

The four physical inputs on the RIU can be mixed in any proportion and assigned to one input highway. The mixing ratio can be dynamically changed in the model. If you want more than one independent audio input from an RIU, just use more than one RIU Audio Input objects.

RIU Audio Input



Highway Channel

Input highway channel to which the audio will be routed (1 thru 24). To get the audio from the input highway, the Audio In object or Vox object is used.

RIU Number

Specifies the RIU from which the input audio will be taken. This can be set from 1 to 16, and corresponds to the setting of a rotary switch on the RIU. A value of zero will render the object inoperative, as it does not specify a valid RIU.

Control Word Connection

A connection to an 8 bit control word which determines the control word. Only the 4 least significant bits are used.

Control Word Mask

Provides a mask for the control word connection. In the absence of a connection, this provides a control word.

Control Word

The control word is determined by the Control Word Connection and the Control Word Mask.

It determines which of the four input channels are to be mixed together. The highest order four bits are automatically set to zero. The least significant bit corresponds to input channel A on the RIU.

Individual Gain Connections

Provides connections for the individual gains.

RIU Audio Input

Individual Gain Scale Factors

Provide scale factors for the individual gain connections. In the absence of gain connections, these provide the gains themselves.

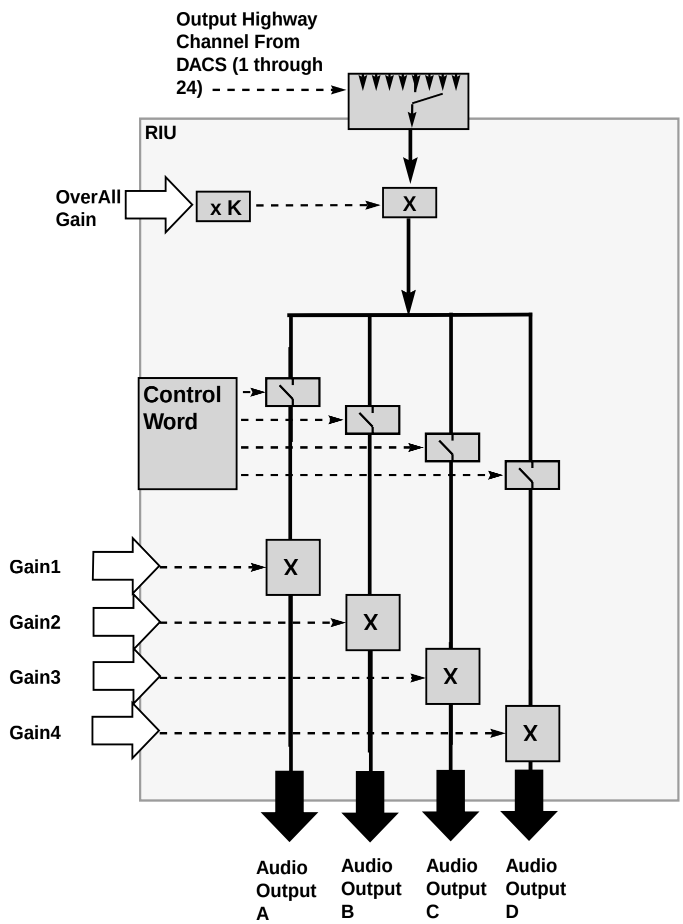
Individual Gains

Indicates the multiplicative factor for each input signal. After the signals are multiplied by the individual gains, they are mixed together.

Overall Gain

After the inputs are multiplied by the individual gains and mixed together, they are multiplied by the overall gain and sent to the input highway.

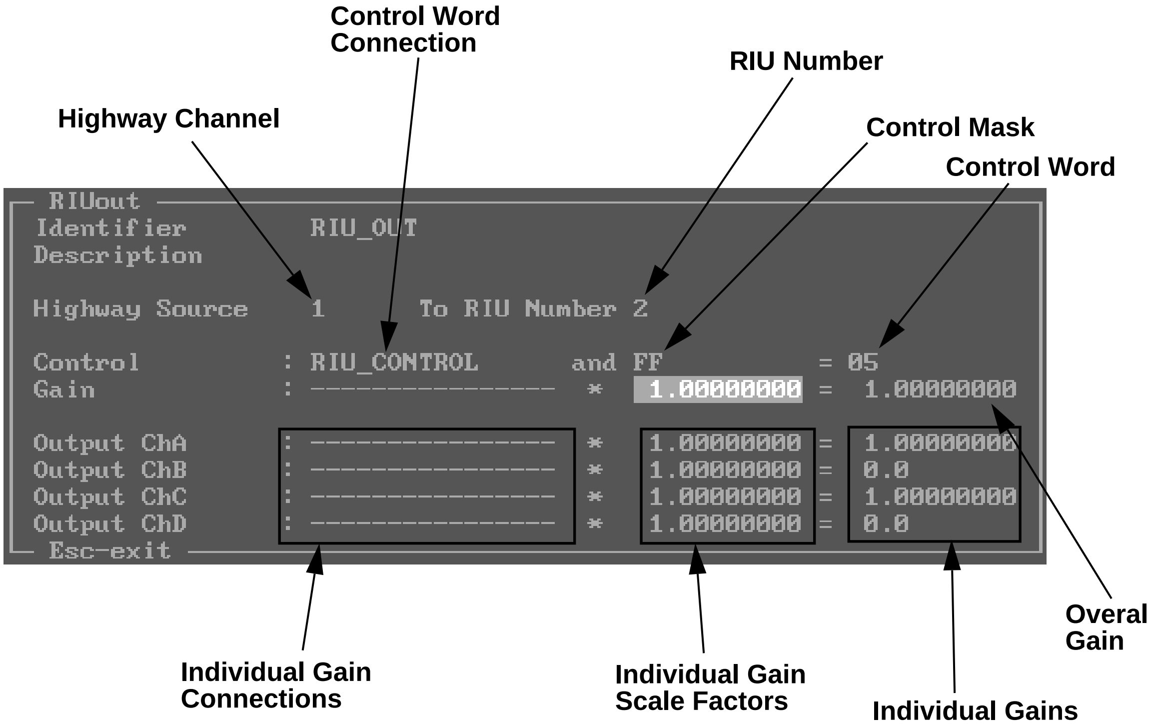
RIU Audio Output



RIU Audio Output

The RIU audio output feeder assigns an output highway channel to RIU outputs. The feeder acts as a local balancer for the RIU, taking one highway channel and splitting the sound out to the four audio outputs on the RIU in any proportion specified.

RIU Audio Output



Highway Channel

Output highway channel which will be the source of the sound for the RIU. The output highway channel can be from 1 to 24. The total number of output highway channels is assigned in the model configuration window. To put audio onto the output highway, use another feeder (such as the buffer or the balancer.)

RIU Number

Specifies the RIU to which the output audio will be sent.

Control Word Connection

A connection to an 8 bit control word which determines the control word. Only the 4 least significant bits are used.

Control Word Mask

Provides a mask for the control word connection. In the absence of a connection, this provides a control word.

Control Word

Determines which of the four output channels are to receive audio from the output highway. The highest order four bits are automatically set to zero. The lease significant bit corresponds to output channel A on the RIU.

Individual Gain Connections

Provides connections for the individual gains.

RIU Audio Output

Individual Gain Scale Factors

Provide scale factors for the individual gain connections. In the absence of gain connections, these provide the gains themselves.

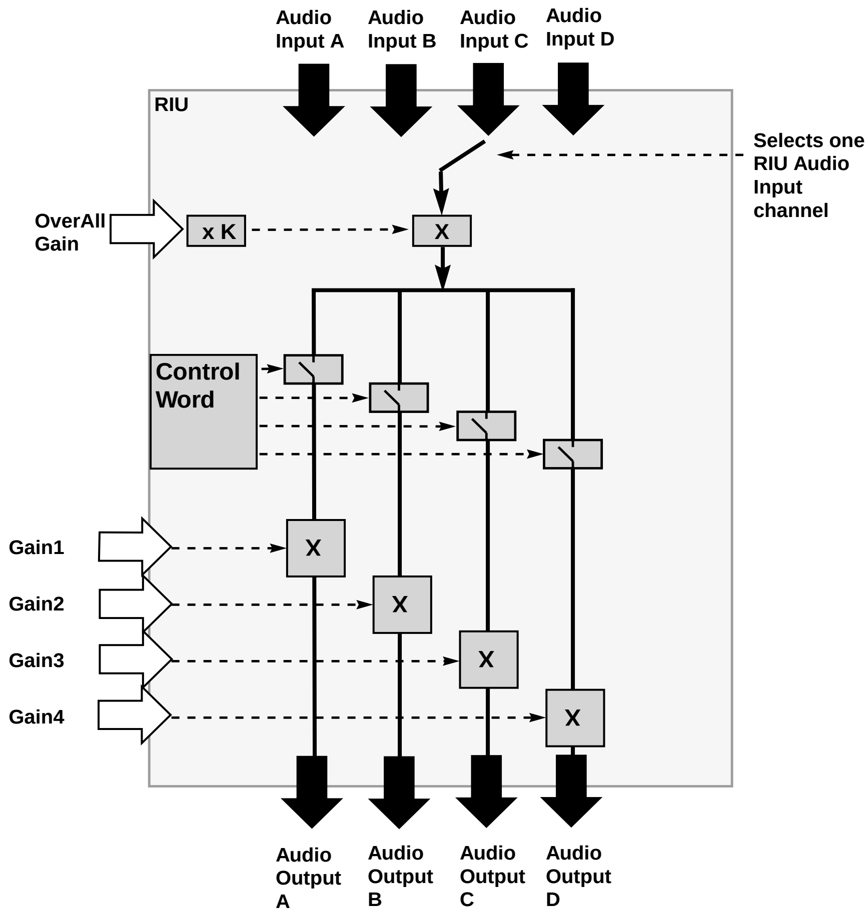
Individual Gains

Indicates the multiplicative factor for the output audio for each audio channel that it is routed to.

Overall Gain

Multiplier for the output signal before it is multiplied by the individual gains.

Local RIU

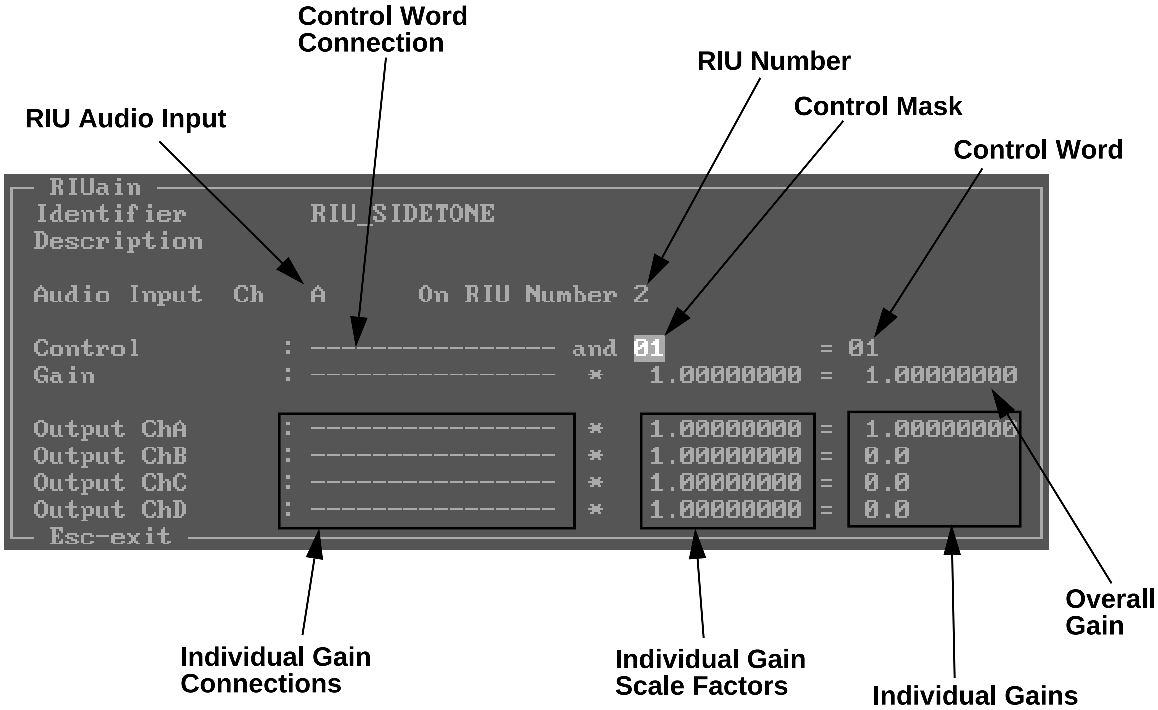


Local RIU

The local RIU feeder controls local mixing of inputs to outputs on an RIU. It is used mainly for sidetone control.

By mixing the sidetone locally, only one output highway needs to be used if two channels at the RIU are going to have different sidetones but the same output otherwise.

Local RIU



RIU Number

Specifies which RIU is to have the local sound mixing.

RIU Audio Input

Determines which audio input is to be routed to the audio outputs.

Control Word Connection

A connection to an 8 bit control word which determines the control word. Only the 4 least significant bits are used.

Control Word Mask

Provides a mask for the control word connection. In the absence of a connection, this provides a control word.

Control Word

Determines which of the four output channels are to receive audio from the audio input. The highest order four bits are automatically set to zero. The lease significant bit corresponds to output channel A on the RIU.

Individual Gain Connections

Provides connections for the individual gains.

Individual Gain Scale Factors

Provide scale factors for the individual gain connections. In the absence of gain connections, these provide the gains themselves.

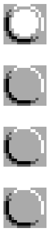
Local RIU

Individual Gains

Indicates the multiplicative factor for the output audio for each audio channel.

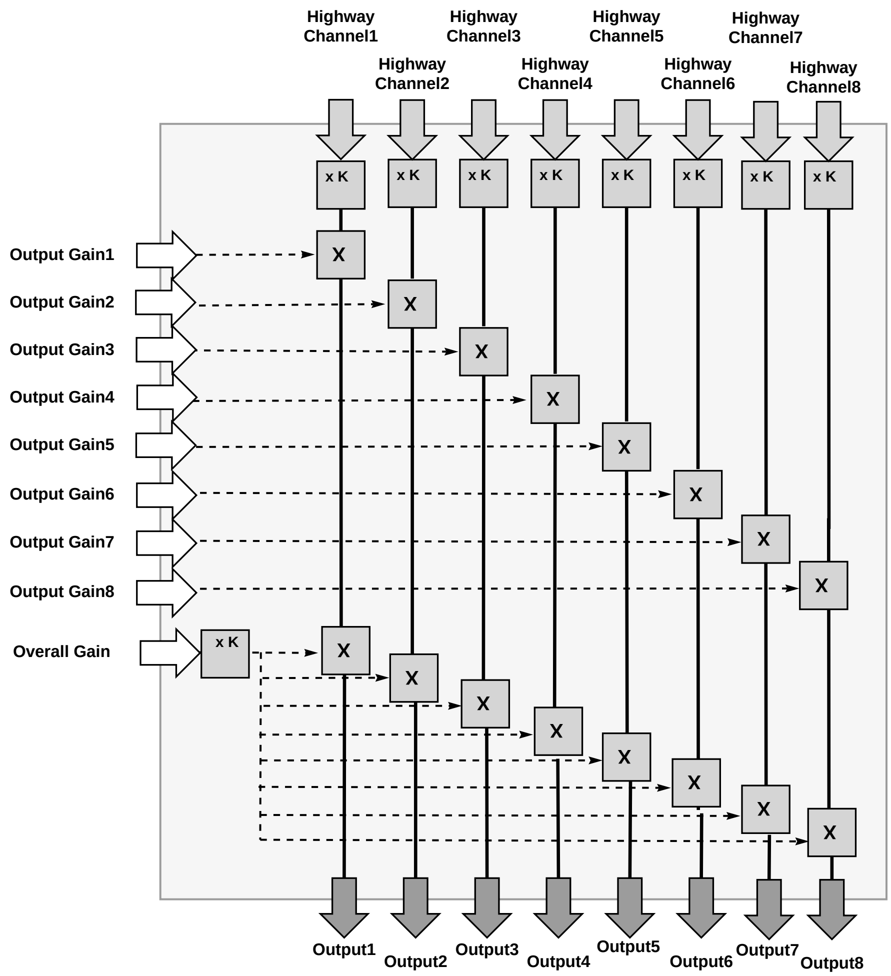
Overall Gain

Multiplier for the audio input signal before it is multiplied by the individual gains.



Analog Out Feeder

Output

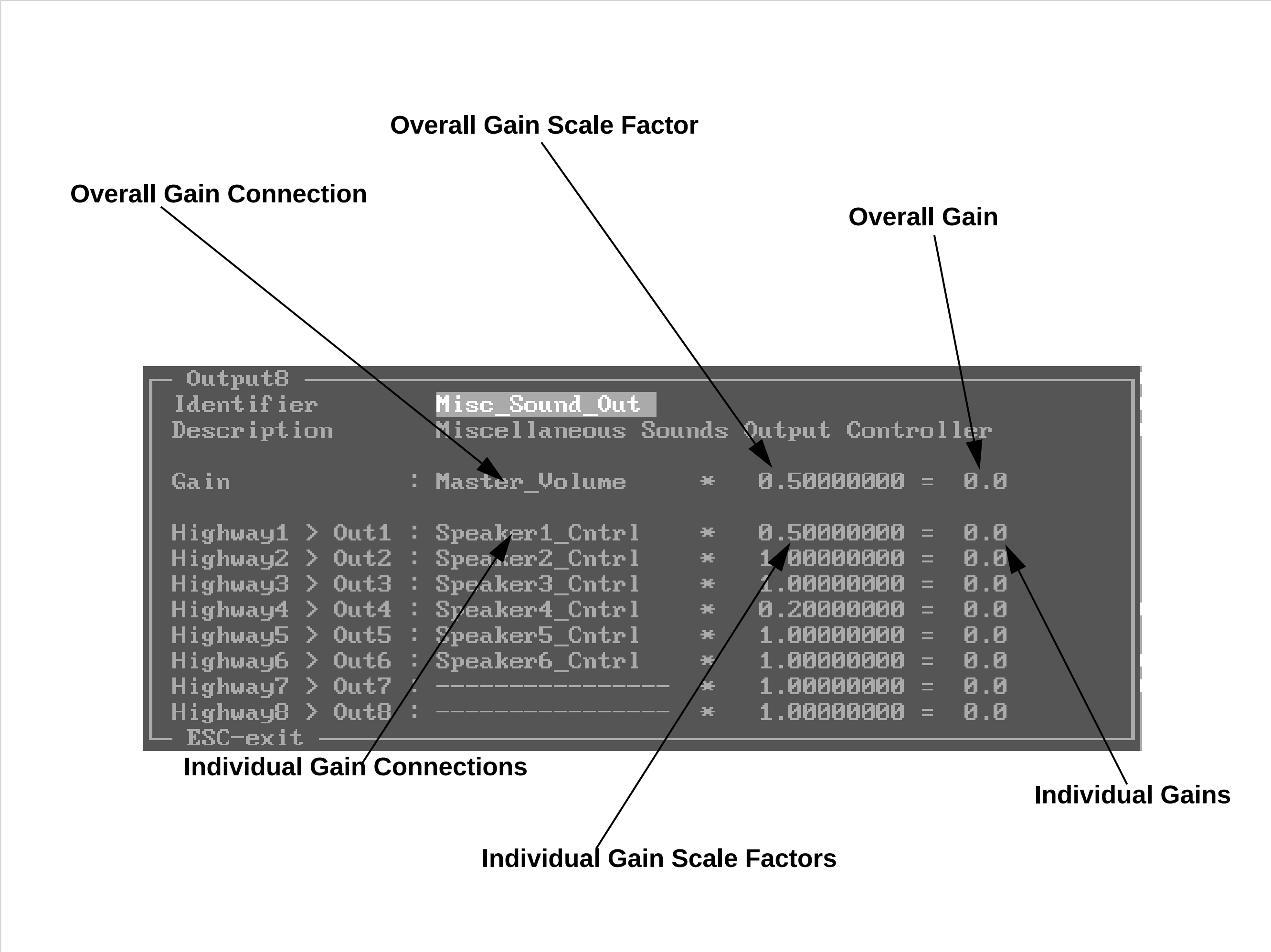


Output

The analog output object connects all of the highway channels to the outputs of the waveform synthesizer.

Only one analog output object can be present in each waveform synthesizer. Its channel width is determined automatically from the hardware configuration of the synthesizer.

Output



Overall Gain Connection

Connection to a control object which provides the overall amplitude gain.

Overall Gain Scale Factor

Value for gain, or scaling factor for overall gain connection value.

Overall Gain

Overall amplitude gain for all outputs.

Individual Gain Connections

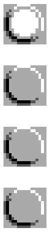
Connection to a control object which provides the amplitude gain for each output.

Individual Gain Scale Factors

Value for gain, or scaling factor for each output gain connection value.

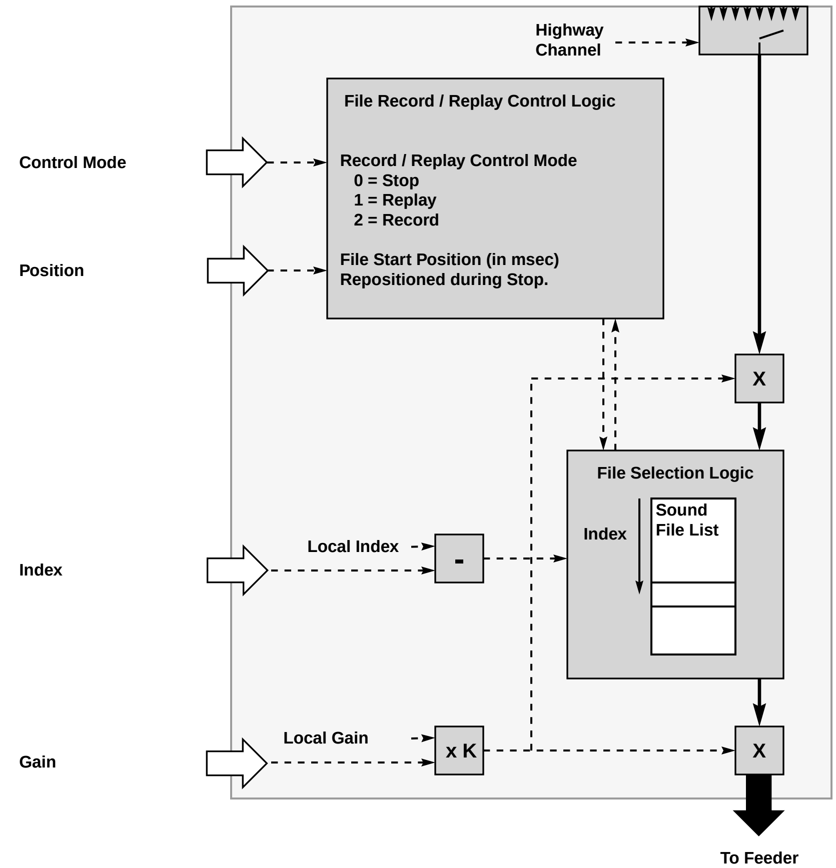
Individual Gains

Amplitude gain for each output.



Record Replay Feeders (Record/Replay Package)

Record / Replay



Record / Replay Feeder

The record replay object provides the ability to record any output highway channel to or from a selected file on the hard disk, under host computer control.

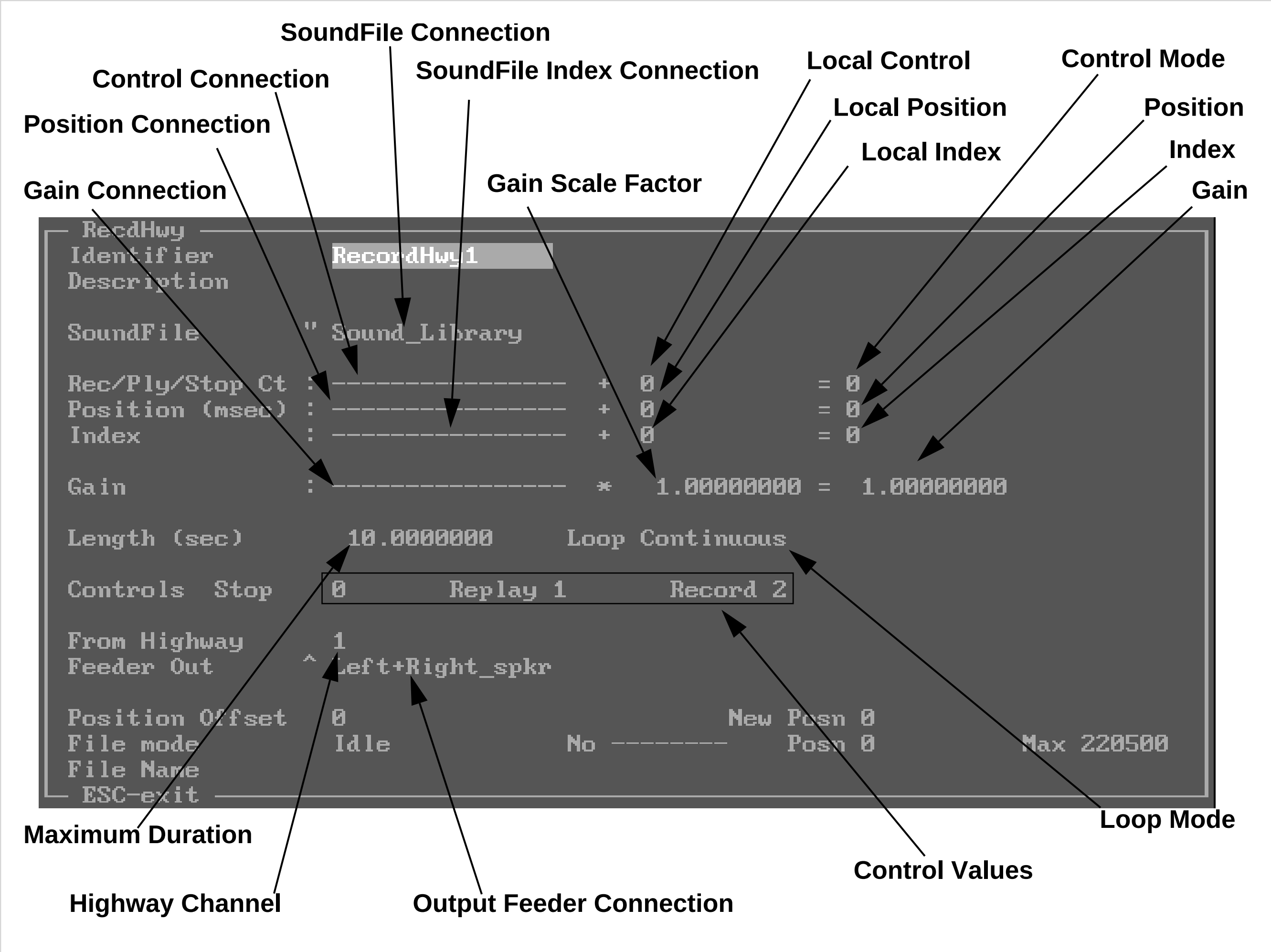
For replay, control is provided over which combination of the highway channels receive the composite sound on replay. This allows different combinations to be used for recording and replaying, e.g. a single channel record such as pilot's headset output to be replayed to all of the headsets.

A position input provides the host with a repositioning capability for both record and replay.

The record object can be linked to a file group such that a different series of files can be used for the recording process. This is useful for demonstration or debrief purposes.

The Record/Replay Feeder is identical to the Record/Replay signal, except that it takes a Highway channel as its input instead of a signal object.

Record / Replay



SoundFile Connection

Sound file selection from model's soundfile directory list

Control Connection

Record / replay mode connection to elsewhere in model.

Local Control

Provides local offset for control mode value (useful for testing only).

Control Mode

The record/replay control variable, with defaults of 0 = Stop, 1 = Replay, 2 = Record.

NOTE: this field should not be changed directly from replay to record, or vice versa. Instead, the field should be made to go to stop after either record or replay.

Position Connection

File position connection to elsewhere in model.

Local Position

Provides local offset for position value (useful for testing only).

Position

File start position (32 bit integer in milliseconds) from start or end of file. This value is used to reposition the file while in the stop mode. When in continuous loop mode, a positive value provides an offset forward in the file, while a negative offset repositions the file back from the last stop position. When not in loop mode, a positive position provides an offset from the start of the file, while a

Record / Replay

negative number is an offset from the end of the file. A value of zero leaves file pointer at its current position. (See the figure on the next page.)

Soundfile Index Connection

Control object connection to provide file selection control from elsewhere in model.

Local Index

Local value for the file index. If a connection is made then this value is added to the incoming index value, else it provides a default value for the file index.

Index

The current value of the file index. This index is used to select one of a group of files from the sound files list. It is only active when the soundfile connection is made to a file group, and the index matches that of one file in the list. If no matches are found then no file is recorded or replayed.

Gain Connection

Control object connection to provide input/output amplitude gain control from elsewhere in model.

Gain Scale Factor

Scaling factor for input/output gain control value.

Gain

Input/output amplitude gain of file record source. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Loop Mode

The continuous loop mode sets the recording and replay such that the effective end of recording marker moves in a continuous loop. This allows the file to act as a recorder for the last few minutes of sound which is continuously overwritten until a stop command is issued. If this mode is Off the file is recorded once only, completing when the maximum duration is reached.

Control Values

The record/replay control variable, uses default values of 0 = Stop, 1 = Replay, 2 = Record. These fields allow the user to define other more convenient mappings of these three commands.

Maximum Duration

The maximum record/replay time allocated to the soundfile (in seconds).

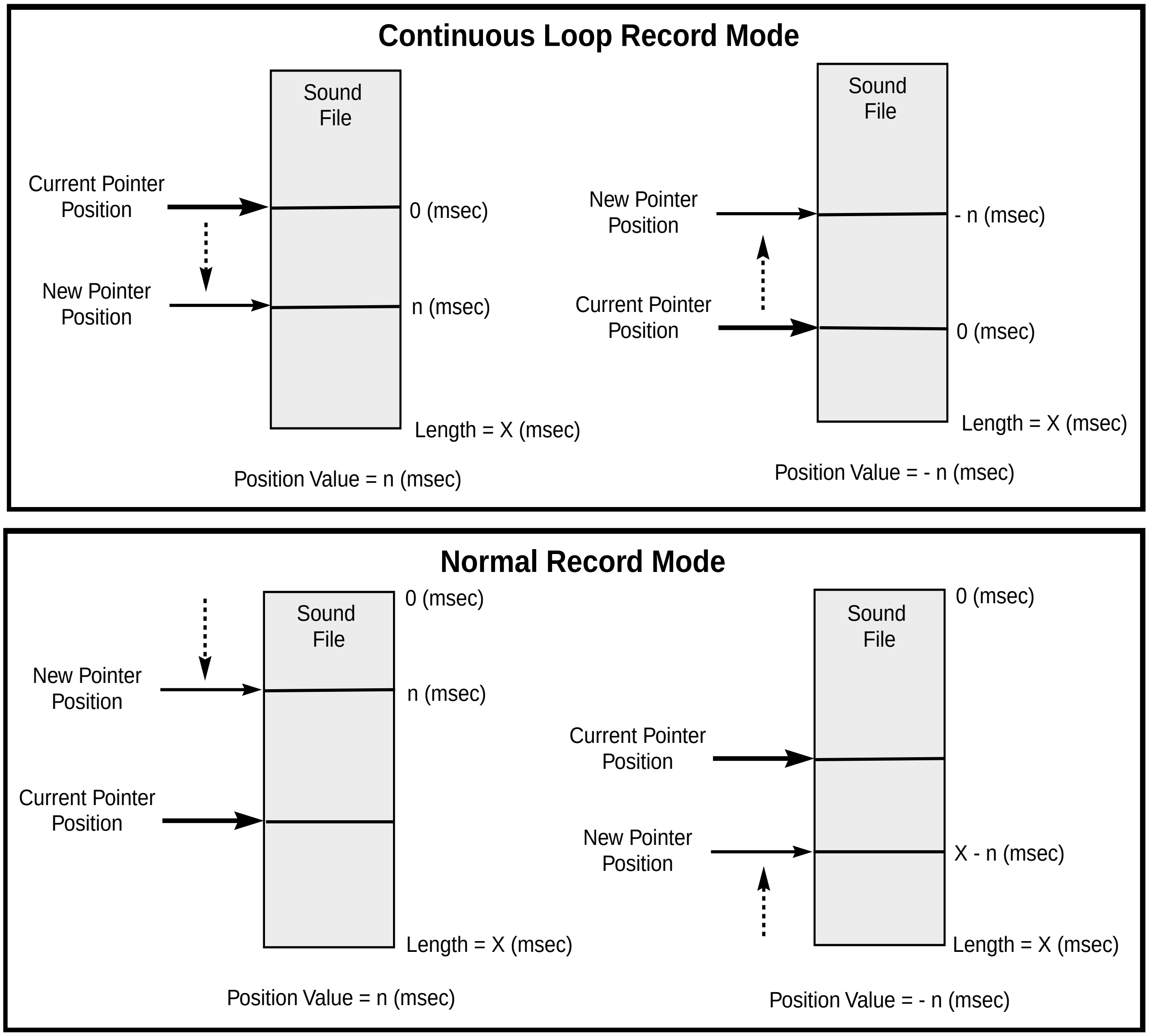
Highway Channel

Selects which output highway channel provides the source for the recorder.

Output Feeder Connection

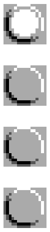
Connection to a feeder, which adds the replayed sound source into the signal highway.

Record / Replay



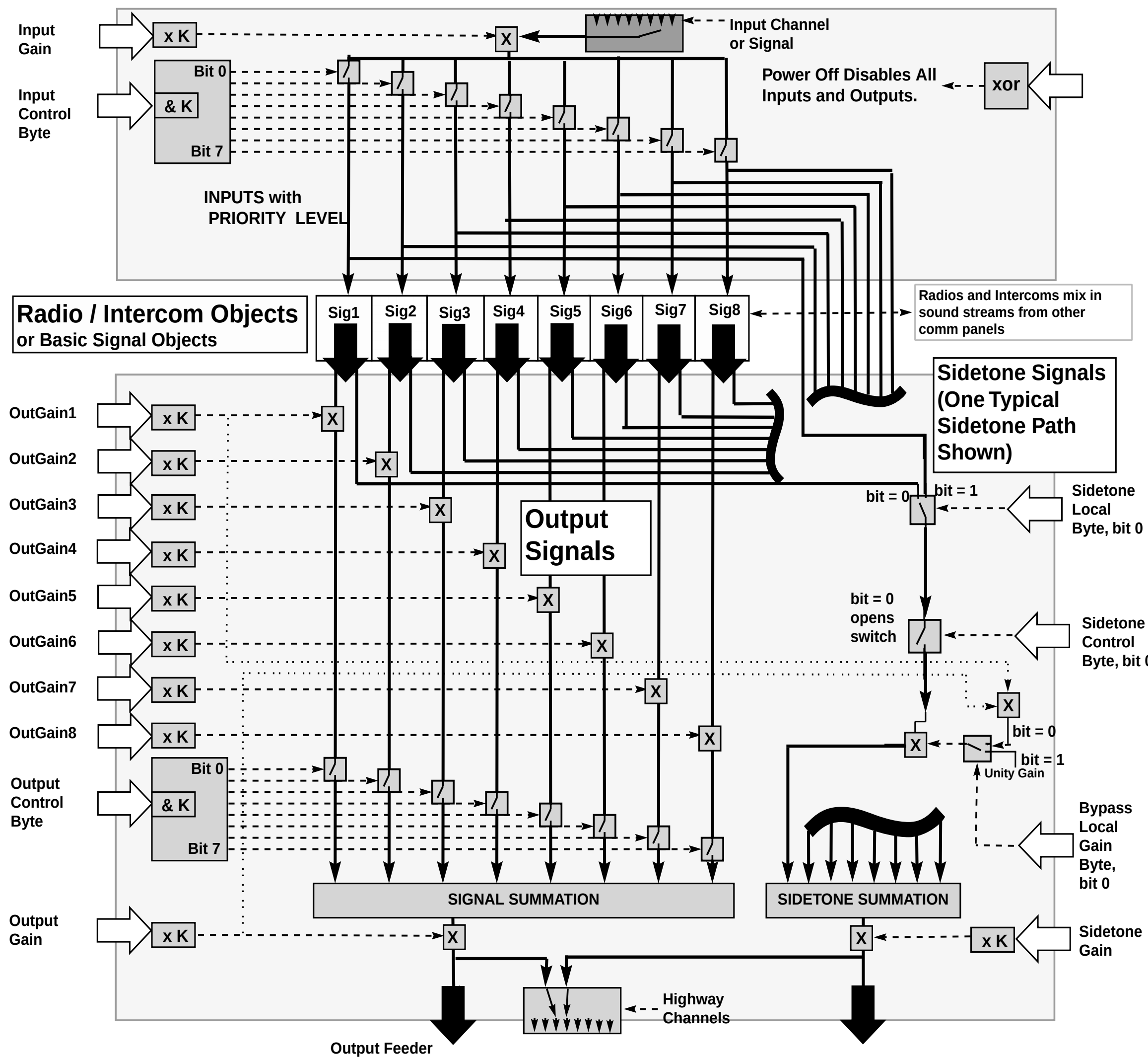
Figure

Pointer Positions in Normal and Continuous Loop Mode



Communication Panels (Radio & Intercom Package)

Comm Selector Panel



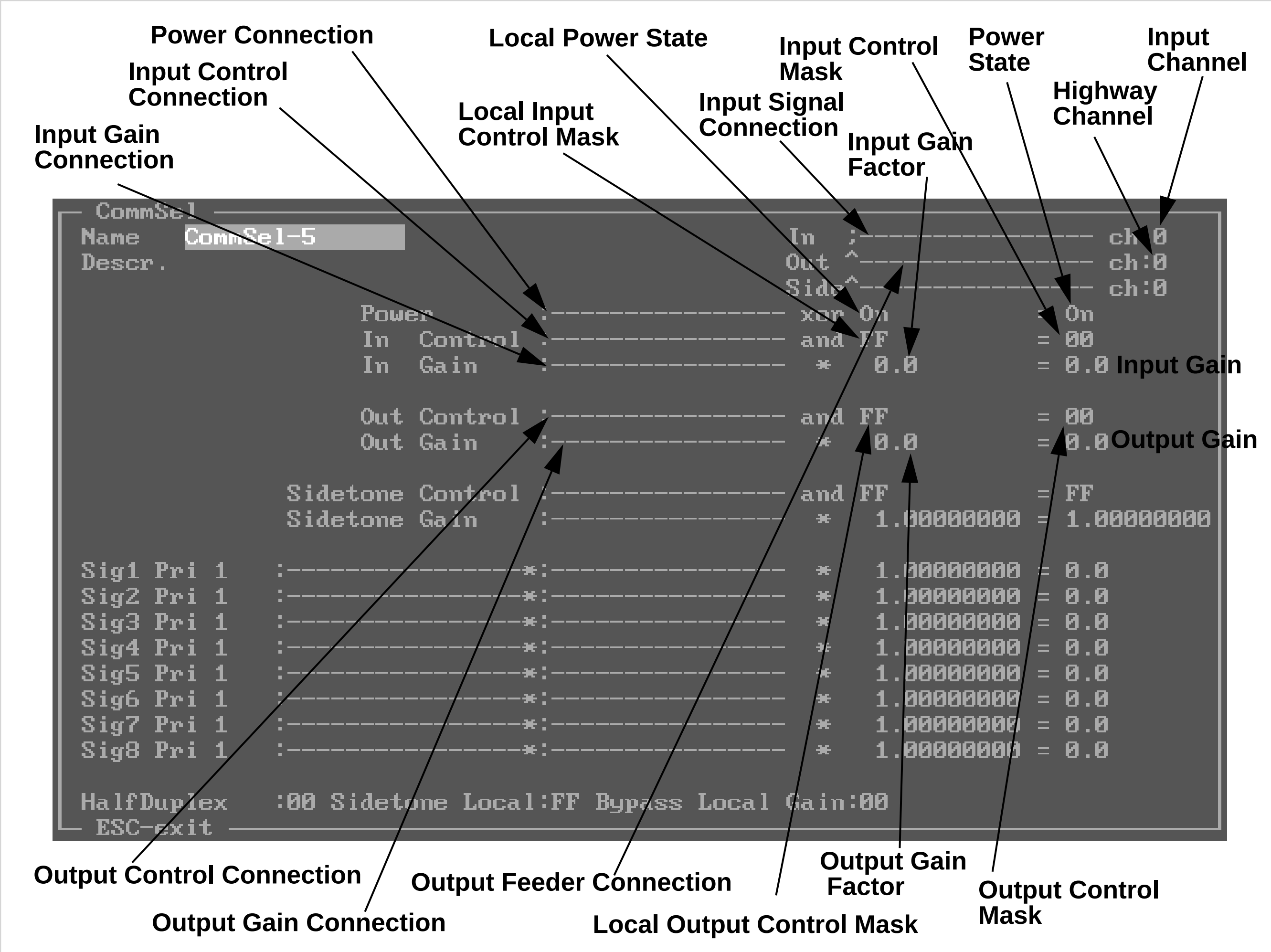
Communications Selector Panel

The Communications Selector Panel provides a simulation of a radio/ communications control panel. However, it provides for separate signal and sidetone paths to allow more flexible control over sidetones. The outputs signals can be mixed into multiple highway channels via the main output and sidetone feeders.

Local on/off control is provided over the eight output signals and eight sidetone signals, as well as individual and overall signal gains.

Input microphone gain and control provide for microphone keying when used in conjunction with a radio object.

Comm Selector Panel



Input Channel

Input channel used for input (1 thru 8, with 0 input disabled). Set to 0 to take the input from the Input Signal Connection.

Input Signal Connection

Connection to input signal to comm panel. This connection is used instead of the input channel when, for example, a VOX is needed.

Highway Channel

Output Highway channel used for total signal output. (1 thru 8, with 0 output disabled). Set this field to zero if the output or sidetone feeders are in use.

Power Connection

Control object connection to provide power control from elsewhere in model.

Local Power State

Local value for the power state. If an enable connection is made then this provides an exclusive-or function for inverting the power control state.

Power State

The comm panel power state; when On the panel is active, when Off it is inactive, simulating a power off condition.

Comm Selector Panel

Input Control Connection
Connection to a control object which provides control of which signal connections have the input microphone routed to them.

Local Input Control Mask
Provides either a local input signal enable, or a bit mask of input control connection value.

Input Control Mask
Bit control value for microphone input channels. A zero bit disables the microphone input thru to the appropriate signal. The microphone input is then passed thru to the selected object. Bit 0 (lsb) controls signal 1, bit 7 (msb) signal 8, with a 1 being on and a 0 being off.

Input Gain Connection
Connection to a control object which provides the input amplitude gain for the signal feed from the selected input channel to the enabled signals.

Input Gain Factor
Value for gain, or scaling factor for input gain connection value.

Input Gain
Input (microphone) gain.

Output Control Connection
Connection to a control object which provides a bit mask enable for each of the eight signals in the comm. panel.

Local Output Control Mask
Provides either a local output signal enable mask, or a bit mask of output control connection value.

Output Control Mask
Bit control value for channel selection. A zero bit disables the output of the appropriate signal. Bit 0 (lsb) controls signal 1, bit 7 (msb) signal 8, with a 1 being on and a 0 being off.

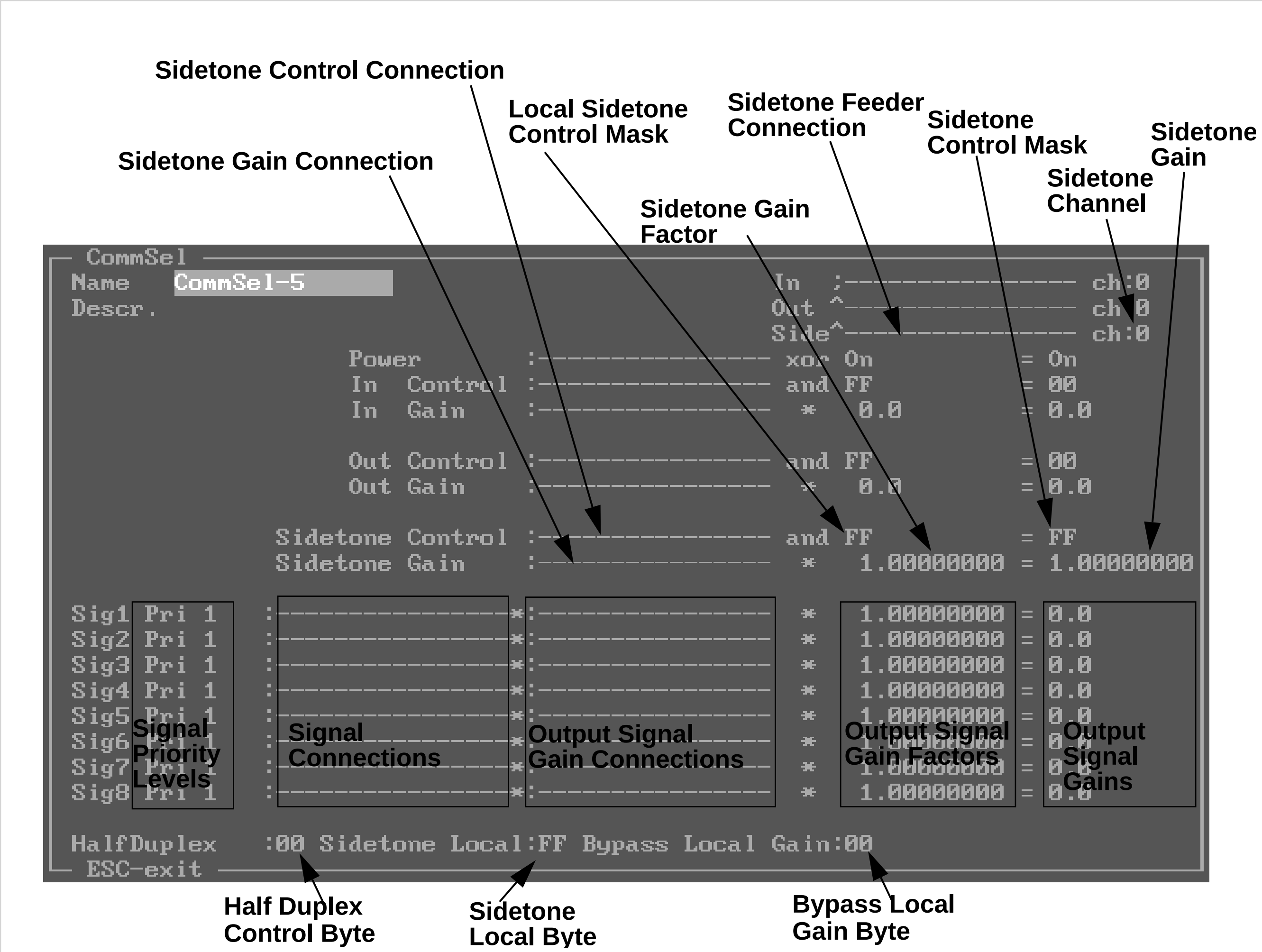
Output Gain Connection
Connection to a control object which provides the overall output amplitude gain for the signal feed onto the selected highway channel.

Output Gain Factor
Value for gain, or scaling factor for overall output gain connection value.

Output Gain
Overall comm. panel output gain

Output Feeder Connection
Connection to a feeder, which provides a mechanism for outputting the main composite output onto the signal highway. This field can be left blank if the highway channel value is non-zero.

Comm Selector Panel



Sidetone Control Connection

Connection to a control object which provides a bit mask enable for the sidetone for each of the eight signals in the comm. panel.

Local Sidetone Control Mask

Provides either a local sidetone signal enable mask, or a bit mask of sidetone control connection value.

Sidetone Control Mask

Bit control value for sidetone channel selection. A zero bit disables the output of the appropriate signal. Bit 0 (lsb) controls signal 1, bit 7 (msb) signal 8, with a 1 being on and a 0 being off.

Sidetone Gain Connection

Connection to a control object which provides the overall sidetone amplitude gain for the signal feed onto the selected highway channel.

Sidetone Gain Factor

Value for gain, or scaling factor for overall sidetone gain connection value.

Sidetone Gain

Overall comm. panel sidetone gain

Sidetone Channel

Output channel for the Sidetone sounds. This can be set to the same channel as the Output Channel. Set to 0 to use the Sidetone Feeder Connection.

Comm Selector Panel

Sidetone Feeder Connection

Connection to a feeder, which provides a mechanism for outputting the composite sidetone output onto the signal highway. This field can be left blank if the highway channel value is non-zero.

Signal Connections

Connections to signal objects (radios, intercoms or basic signals) to be used for channels 1 thru 8 of the Comm Panel. For radio and intercom signals, the Comm Panel input is fed to the radio or intercom and the output and sidetones are fed back to the Comm Panel. The signal priority is sent to intercom objects only.

For basic signals, the signal output is fed to the Comm Panel, but the Comm Panel input is not used.

Signal Priority Levels

Priority level for each of the eight signals. This is used by the Intercom Object to determine which sounds are heard on the Intercom Bus. 1 is the highest priority, and 255 is the lowest. A priority of 0 is a wildcard - signals with a priority of 0 will always get mixed on the bus, regardless of the current priority level.

Output Signal Gain Connections

Connection to a control objects which provide the individual gain for each output signal.

Output Signal Gain Factors

Gain scale factors for each output signal.

Output Signal Gains

Final output signal gains for each signal (1 thru 8), including overall gain factor, control bit (on/off) and individual gain connection values.

Sidetone Local Byte

An eight bit mask which determines, for each Comm Panel channel, whether the sidetone is taken from the connected object or is taken from the Comm Panel input. A 1 value of the bit means the sidetone is generated locally, while a 0 bit means it is taken from the object connected via the Signal Connection. Bit 0 (lsb) controls signal 1, bit 7 (msb) controls signal 8.

Bypass Local Gain Byte

An eight bit mask which controls whether the sidetone gain is multiplied by the output signal gain. If the bit for a signal connection is set to 0, the sidetone gain for that channel is multiplied by the Output Signal Gain for the signal connection. Bit 0 (lsb) controls signal 1, bit 7 (msb) controls signal 8. This provides simultaneous control for the sidetone and output volumes.

If the bit is set to one, the sidetone gain is unaffected by the Output Signal Gain.

HalfDuplex Byte

An eight bit mask which determines, for each channel, whether the communication with the connected object is half duplex.

For a bit set to 1, the channel is half duplex (i.e. sound cannot be simultaneously sent to and received from the connected object.) A bit value of 0 means the comm panel can send and receive at the same time to the connected object.

Half Duplex would be used, for example, in an intercom setup where you can only hear if you are not talking.

Global Channels

Global Channels

Global Channels are a method of providing communication between models running on different DSP boards within the same DACS.

Normally, when you have models running on two or more DSP boards in a DACS, they act independently. You can load and save models to them independently, and all the sounds they process and generate will be independent of each other.

Sometimes, however, it’s necessary to pass sound from one model to another. This would be the case, for example, when you want to have more than eight people on an intercom bus, or if you want them to all hear the same signal.

Global Channels are used to accomplish this passing of sound back and forth. To use them, your DACS must be properly configured.

Setting up the DACS for DSP to DSP communication

In order for the separate DSPs within the DACS to communicate, they must be set up as follows:

1) The hardware should be properly configured. This is done by ASTi before a system is shipped to the customer.

2) In the Model Timing Window, designate the master and slaves. Model Builder 3.10 allows one master and two slaves, which allows up to three cards to pass sound. The master and slave(s) can all intercommunicate. Any DSP that is not designated as a master or a slave will not be able to communicate with other DSPs.

-OR-

designate the Master and Slave(s) in the .CFG file. (See SYSTEM CONFIGURATION, Master_DSP and Slave_DSP).

3) Set the sample rate of the DSP you’ve designated as the master in the Model Configuration Window. The slaves will all automatically be set to this sample rate. Other than determining the sample rate, there is no difference between master and slave DSPs. All models that are communicating must be running at the same sample rate.

NOTE: Not all DSP boards need to be communicating. It is perfectly OK to have two boards hooked up as a Master/Slave and communicating via global channels, and to have a third board running independently at a different sample rate. This might be done, for instance, when simulating communications (several boards linked) and Aural Cues (One board, possibly running at a higher sample rate for higher sound fidelity) on the same DACS.

Communicating between DSPs after the DACS is properly set up

Once the DACS settings are properly set up, there are three ways of sending signals across global channels.

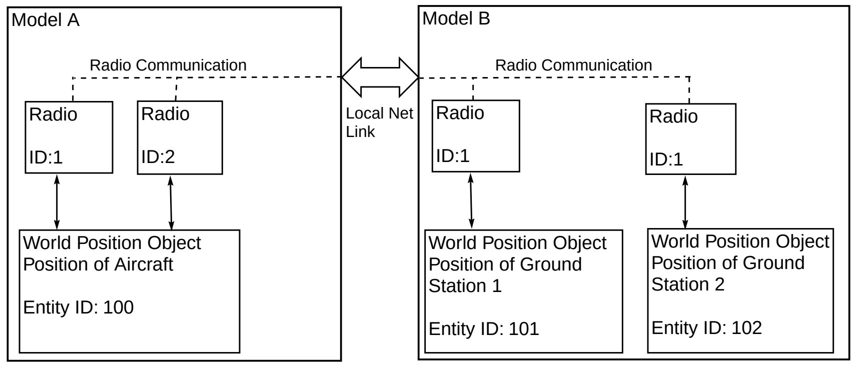
The first way is to designate a global channel in the GLB field of a signal object. Allowed values of global channels are 1 through 64 (Earlier versions of Model Builder only allow 31 global channels.). A value of zero means no global channel is attached. When a global channel is designated, the sound stream from the signal will get sent out on the designated global channel. To receive this signal in another model, use the GlobalIn signal. This will pull the sound stream off of the global channel for use in the normal way within the model. Only one sound in a given model should be directed to a given global channel. If two sounds from the same model are designated to the same global channel, one of them won't be put on. However, if two different cards put sound on the same global channel, the sounds will get mixed together.

The second way to use the global channels is through the Intercom object. To set up an intercom bus that operates across DSP boards, create an intercom object in each model and designate them all with the same global channel in the Glb field of the intercom object. They will all form one intercom bus, and any voice or signal sent into the bus in one model can be heard in the other models as well.

The third way to use global channels is indirectly through the Radio objects. In this case, global channels are not specifically designated, but used internally by the software as needed. To have radios communicating across DSP boards to different models, attach the radio to a world position. In that world position, specify the radio as a "LocalNet" in the Entity Id field, and assign an entity ID in the Entity Number field. Then designate the radio ID in the RadioID field of the radio object. All radios communicating over the local net must have a unique combination of entity ID and radio ID for them to be modeled properly. All radios designated in this way will be able to communicate between models in separate DSP boards.

As an example for the third case, suppose you have two models, A and B, running on two DSP cards. Suppose also that each model has two radios, and all the radios should be able to communicate. Suppose that the radios in model A are both on the same aircraft, while the radios on model B are separate ground stations.

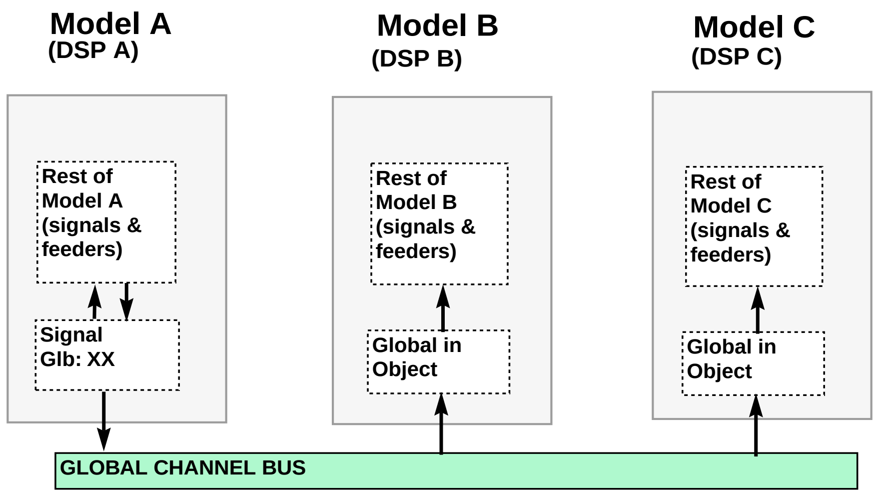
The diagram below demonstrates how these radios could be assigned entity and radio ID's:



The two radios in model A would have ID's 100:1 and 100:2, and the radios in model B would B 101:1 and 102:1 (The ID is given as entity ID:radio ID). Each radio has a unique combination of entity and radio ID's, so all the radios would be able to communicate (provided they are on the same frequency, in range, etc.)

Note: If you are set up to run DIS or VoiceNet Radios, you do not need to set up the Local Net. Radios on other boards will automatically communicate with each other in this case.

Global In Signal



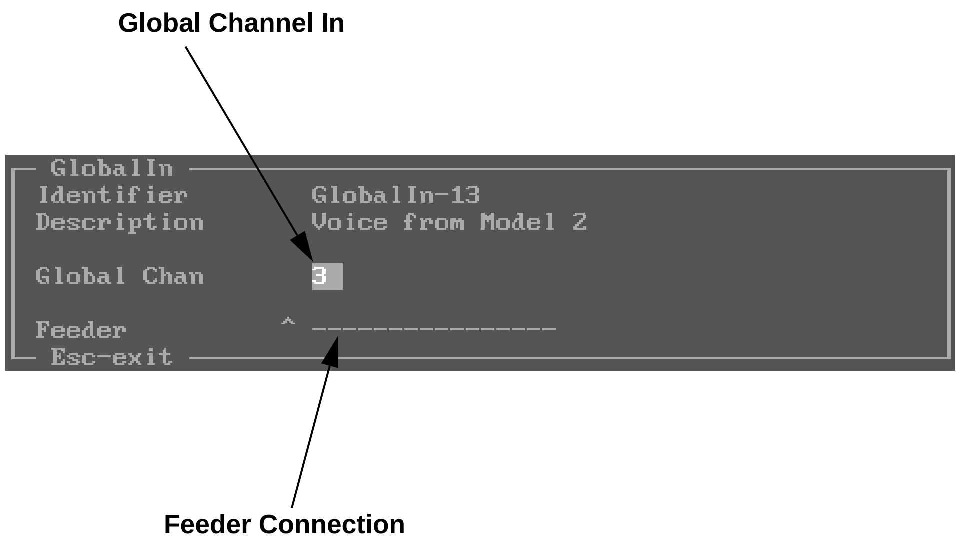
Sending a signal from Model A to Models B and C.

Global In Signal

The Global In signal is used to pull in a sound stream off of a global channel that was sent out from another model on another DSP card. To send a signal out on a global channel, designate the Glb field in the signal to a non zero number. The signal will go out on that global channel, and will be picked up by all Global In signals in other models with the same global channel number.

If more than one signal is sent to the same global channel, the signals will be mixed together.

Global In Signal



Feeder Connection

Connection to a feeder, which adds the GlobalIn signal into the signal highway.

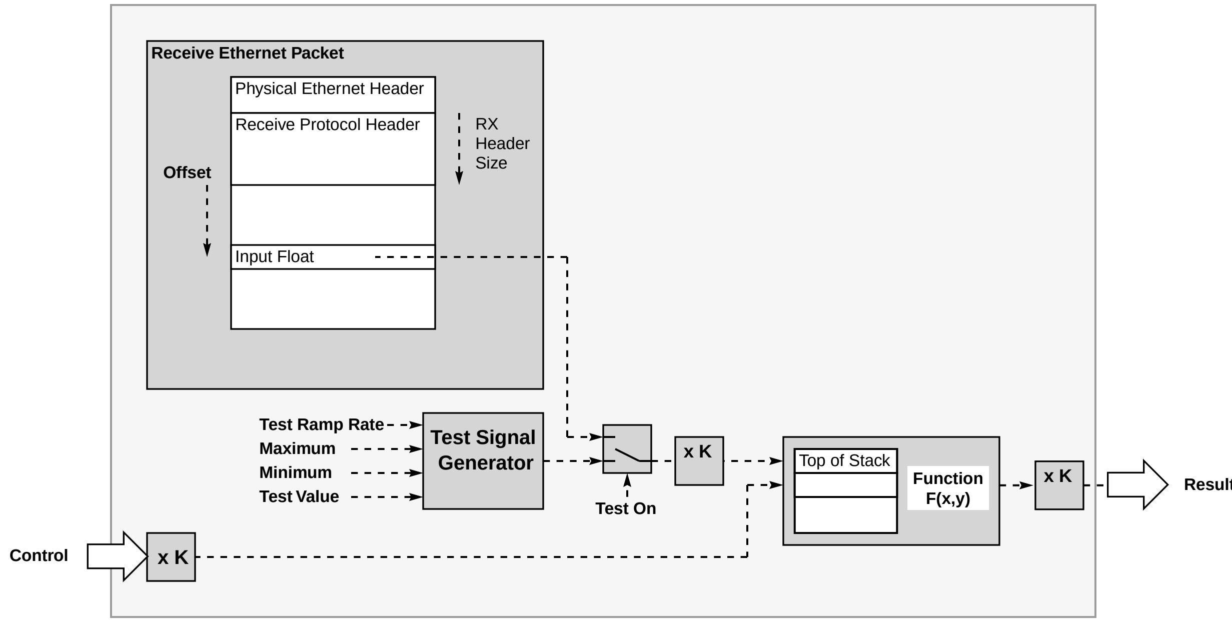
Global Channel In

Global channel number for the sound stream that the Global In object receives.

Controls

Host Inputs

Float Input

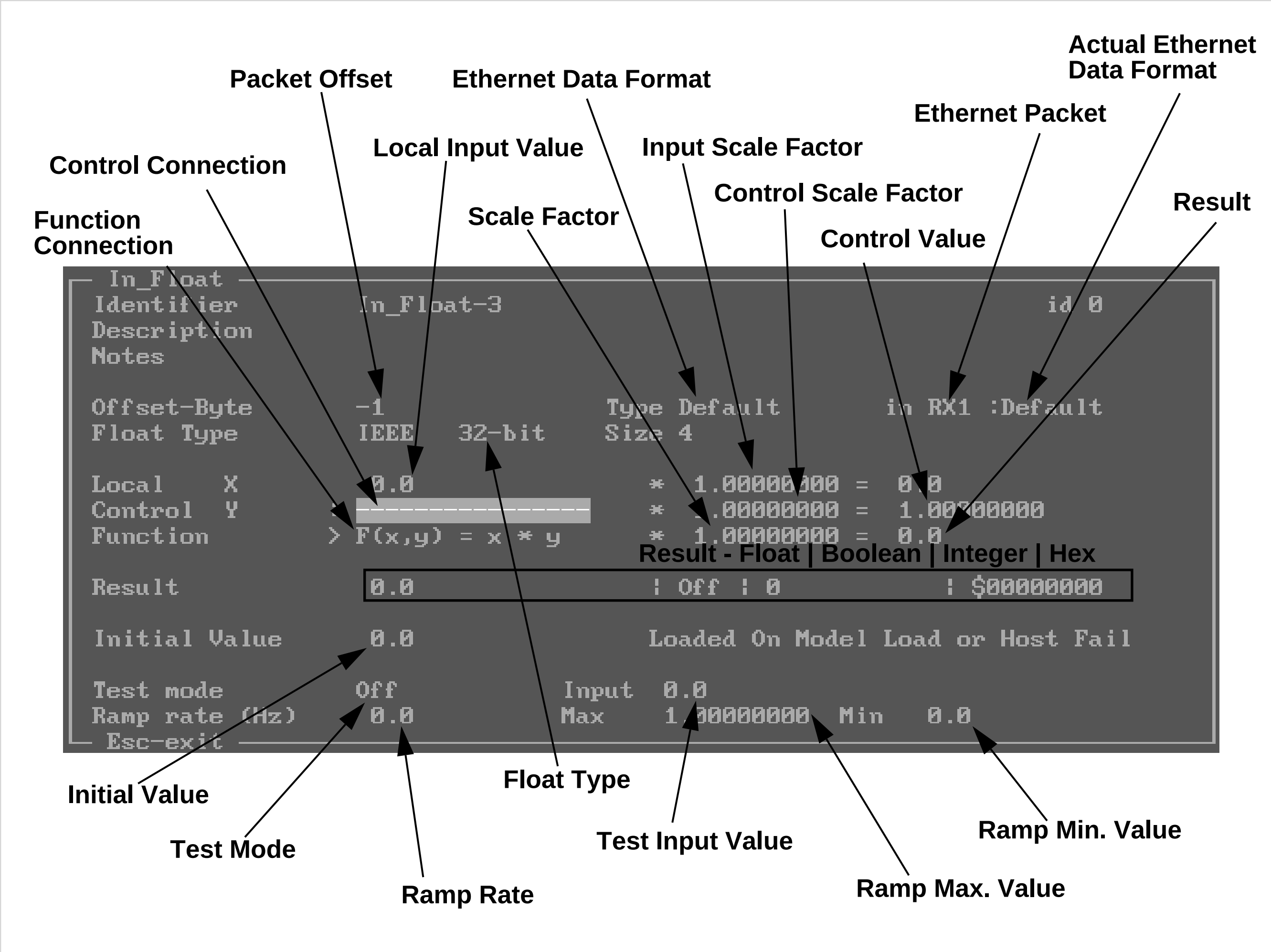


Float Input

This provides an interface between the host's Ethernet control packet and the object orientated model for any single precision real variable.

It also provides a local test mechanism for simulating host inputs.

Float Input



Packet Offset

Offset to input variable from beginning of ethernet packet, in bytes.

Ethernet Data Format

Format of the ethernet variable, includes Default, Big Endian, Little Endian, and Encore formats.

Ethernet Packet

Ethernet packet used for variable access.

Actual Ethernet Data Format

Format of the ethernet variable used to read data, usually same as above, except when default type, then the format used is that specified on the ethernet status page.

Float Type

Format of floating point number.

Local Input Value

The current value of the Float being read from the ethernet packet.

Input Scale Factor

Local scale factor for the input variable prior to control function manipulation.

Control Connection

Connection to another control object for overall control of the floating point result.

Control Scale Factor

Local scale factor for the control variable.

Float Input

Control Value

Control value used as the second variable for the function lookup, allowing scaling or offsetting by another model control.

Function Connection

Selection of input scaling function. With the input value on the top of the stack and the control value on the bottom. The default (no selection) function multiplies the input value by the control value.

Scale Factor

Scale factor for function result.

Initial Value

Initial value loaded either at model load, or on each failure of the host communication link

NOTE: the initial value mode is selected for the entire model on the model configuration page.

Test Mode

Test mode enable flag allows user to override ethernet input value. Useful for debug purposes only.

Test Value

Value for result when in test mode, and ramping disabled.

Ramp Rate

Rate (in Hertz) at which to change result value between maximum and minimum values. Active only when Test mode is on.

Ramp Min. Value

Minimum value for result when in test ramp mode.

Ramp Max. Value

Maximum value for result when in test ramp mode.

Result - Float

Output result of input, floating point value.

Result - Boolean

Output result of input, as a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

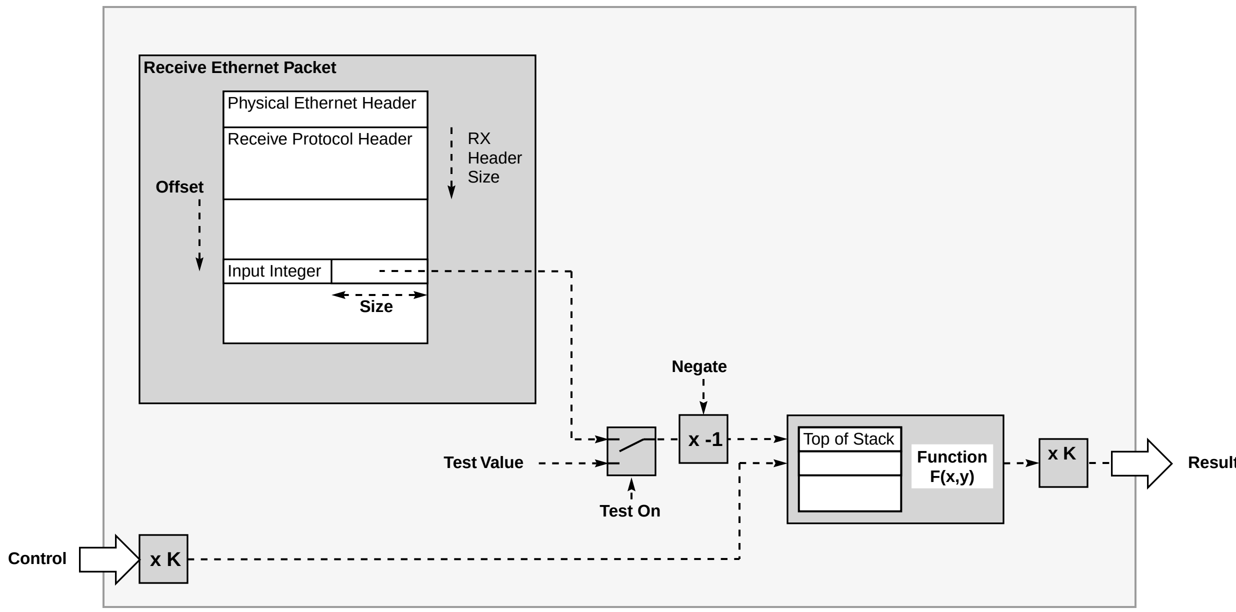
Result - Integer

Output result of input, integer value, a rounding of the floating point result.

Result - Hex

Output result of input, integer value in hexadecimal, a rounding of the floating point result.

Integer Input

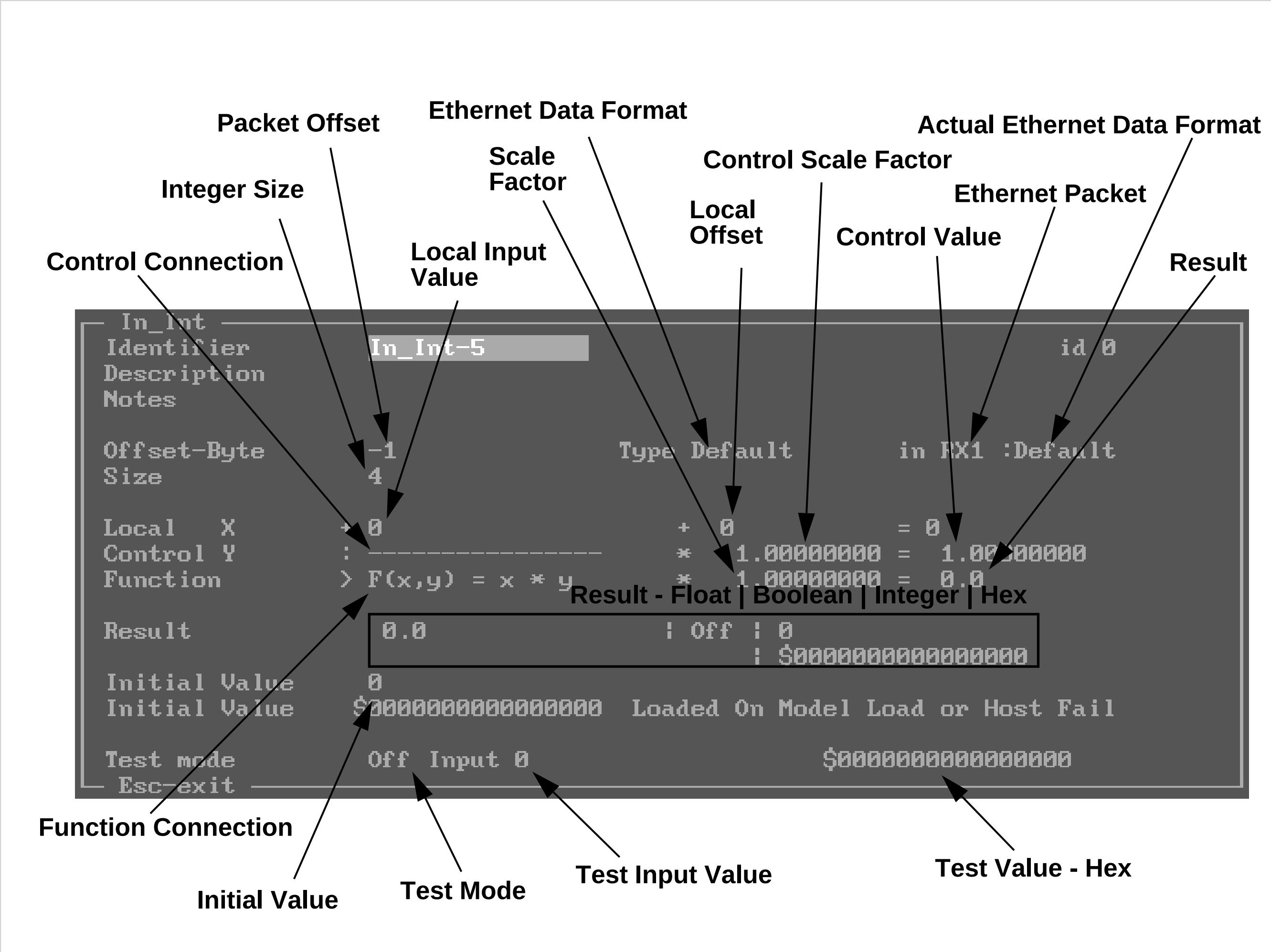


Integer Input

This provides an interface between the host's Ethernet control packet and the object orientated model for any integer variable. The integer can be 1, 2 or 4 bytes long.

A local test mode is provided to override the incoming host integer.

Integer Input



Packet Offset

Offset to input variable from beginning of ethernet packet, in bytes.

Ethernet Data Format

Format of the ethernet variable, includes Default, Big Endian, Little Endian, and Encore formats.

Ethernet Packet

Ethernet packet used for variable access.

Actual Ethernet Data Format

Format of the ethernet variable used to read data, usually same as above, except when default type, then the format used is that specified on the ethernet status page.

Integer Size

Size of integer variable in ethernet packet. Valid sizes, 1,2, and 4 byte integers.

Local Value

The current value of the Integer being read from the ethernet packet.

Local Offset

Offset added to the input integer.

Control Connection

Connection to another control object for overall control of the floating point result.

Control Scale Factor

Local scale factor for the control variable.

Integer Input

Control Value

Control value used as the second variable for the function lookup, allowing scaling or offsetting by another model control.

Function Connection

Selection of input scaling function. With the input value on the top of the stack and the control value on the bottom. The default (no selection) function multiplies the input value by the control value.

Scale Factor

Scale factor for function result.

Result - Float

Output result of input, floating point value, and gain times input integer.

Result - Boolean

Output result of input, as a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

Result - Integer

Output result of input, integer value.

Result - Hex

Output result of input, integer value, hexadecimal format.

Initial Value

Initial value loaded either at model load, or on each failure of the host communication link

NOTE: the initial value mode is selected for the entire model on the model configuration page.

Test Mode

Test mode enable flag allows user to override ethernet input value. Useful for debug purposes only.

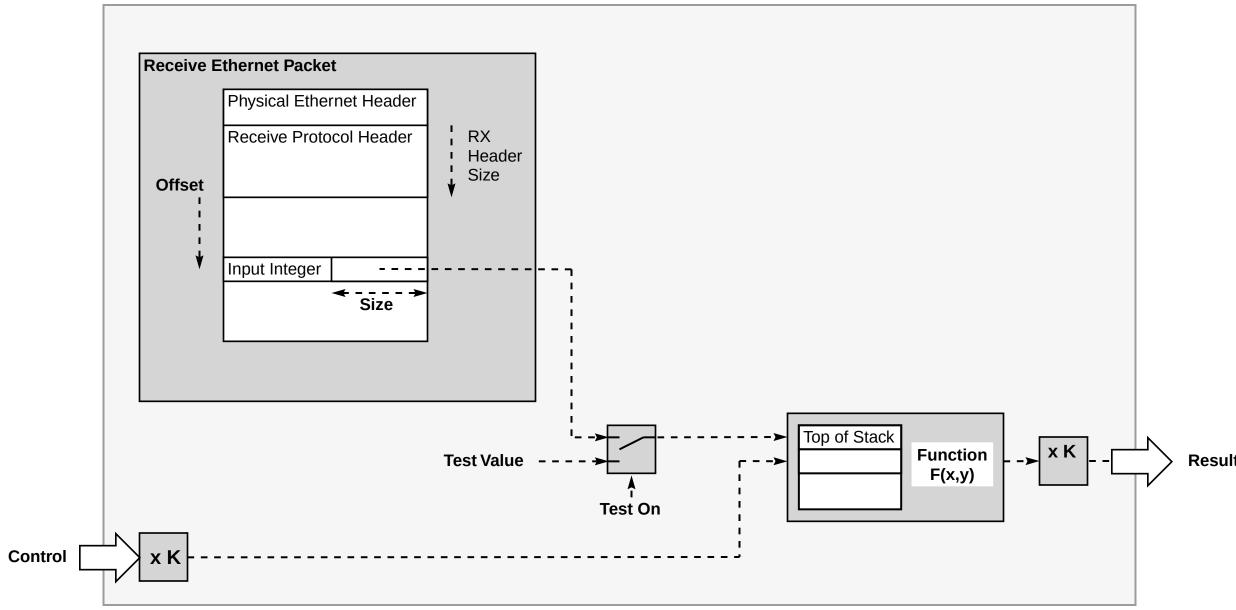
Test Value

Value for result when in test mode - Decimal

Test Value - Hex

Value for result when in test mode - Hexadecimal

Unsigned Integer Input

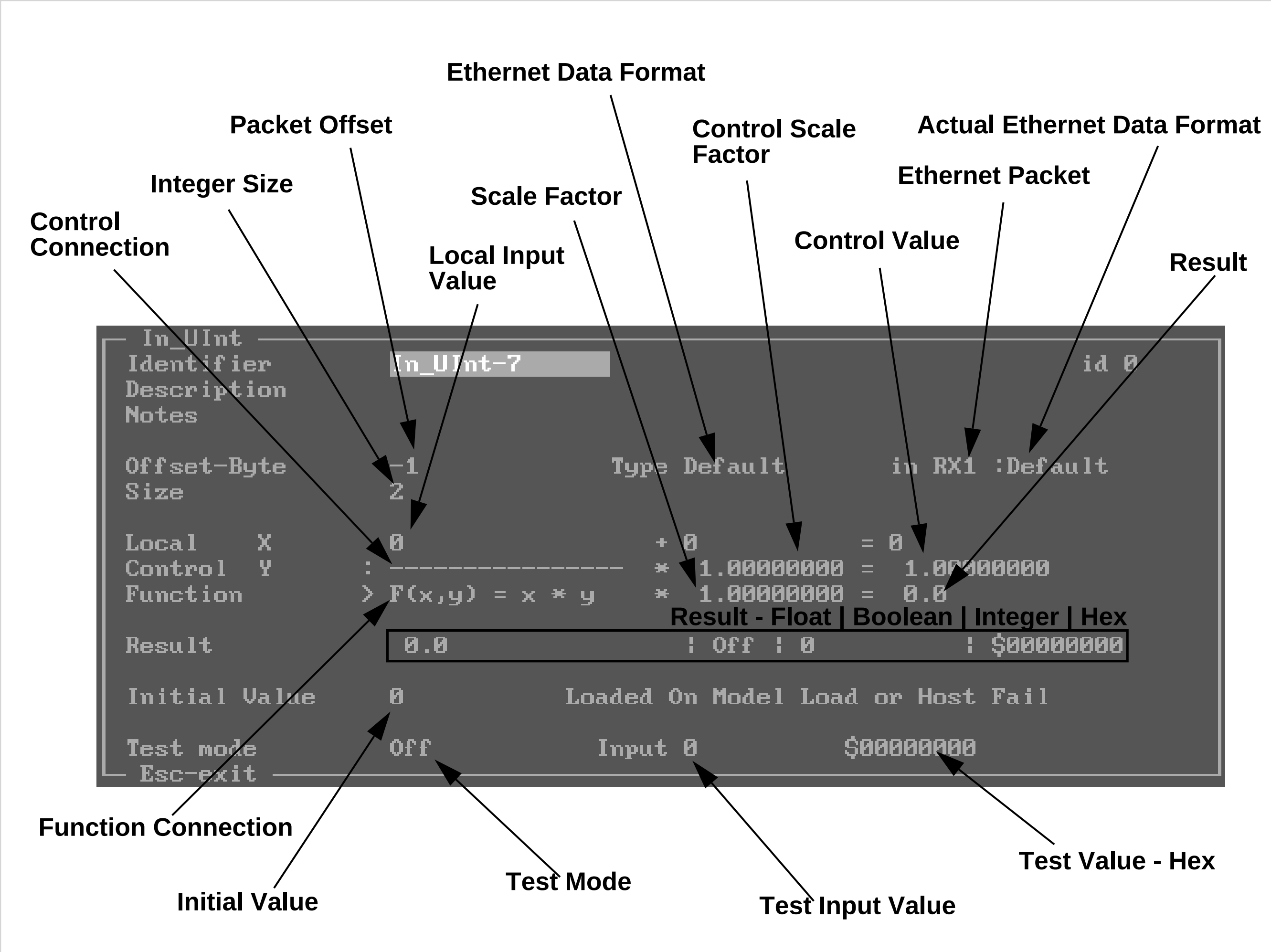


Unsigned Integer Input

This provides an interface between the host's Ethernet control packet and the object orientated model for any unsigned integer variable. The integer can be 1 or 2 bytes long.

A local test mode is provided to override the incoming host integer.

Unsigned Integer Input



Packet Offset

Offset to input variable from beginning of ethernet packet, in bytes.

Ethernet Data Format

Format of the ethernet variable, includes Default, Big Endian, Little Endian, and Encore formats.

Ethernet Packet

Ethernet packet used for variable access.

Actual Ethernet Data Format

Format of the ethernet variable used to read data, usually same as above, except when default type, then the format used is that specified on the ethernet status page.

Integer Size

Size of integer variable in ethernet packet. Valid sizes, 1,2, and 4 byte integers.

Local Value

The current value of the Integer being read from the ethernet packet.

Control Connection

Connection to another control object for overall control of the floating point result.

Control Scale Factor

Local scale factor for the control variable.

Unsigned Integer Input

Control Value

Control value used as the second variable for the function lookup, allowing scaling or offsetting by another model control.

Function Connection

Selection of input scaling function. With the input value on the top of the stack and the control value on the bottom. The default (no selection) function multiplies the input value by the control value.

Scale Factor

Scale factor for function result.

Result - Float

Output result of input, floating point value, gain times input integer.

Result - Boolean

Output result of input, as a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

Result - Integer

Output result of input, integer value.

Result - Hex

Output result of input, integer value, hexadecimal format.

Initial Value

Initial value loaded either at model load, or on each failure of the host communication link.

NOTE: the initial value mode is selected for the entire model on the model configuration page.

Test Mode

Test mode enable flag allows user to override ethernet input value. Useful for debug purposes only.

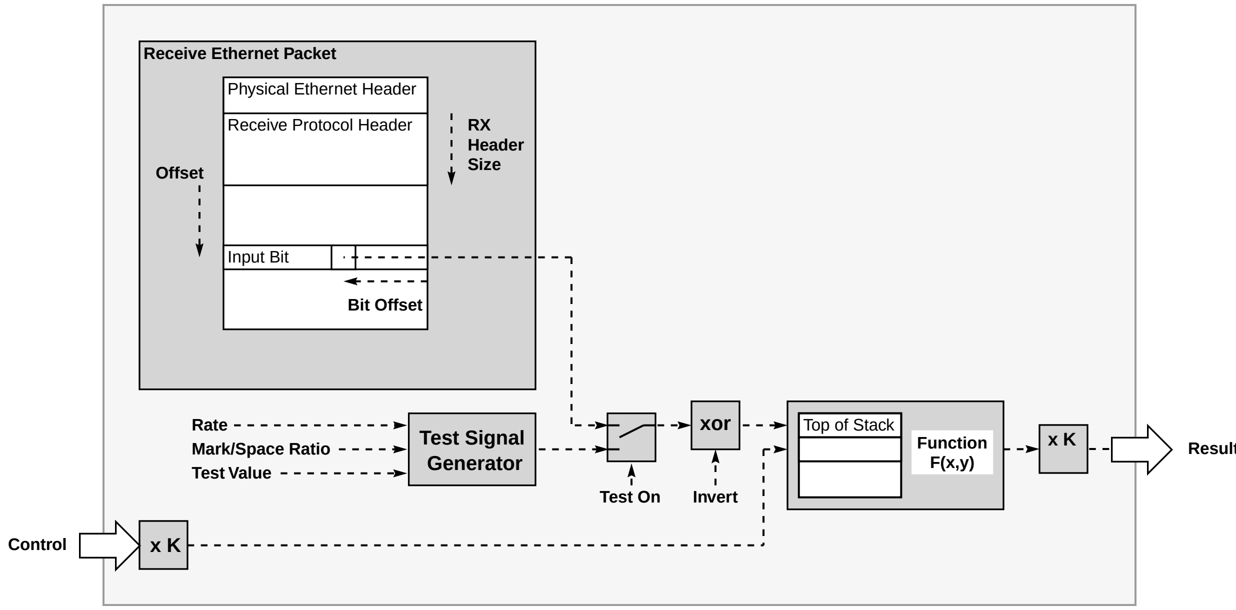
Test Value

Value for result when in test mode - Decimal

Test Value - Hex

Value for result when in test mode - Hexadecimal.

Boolean Input

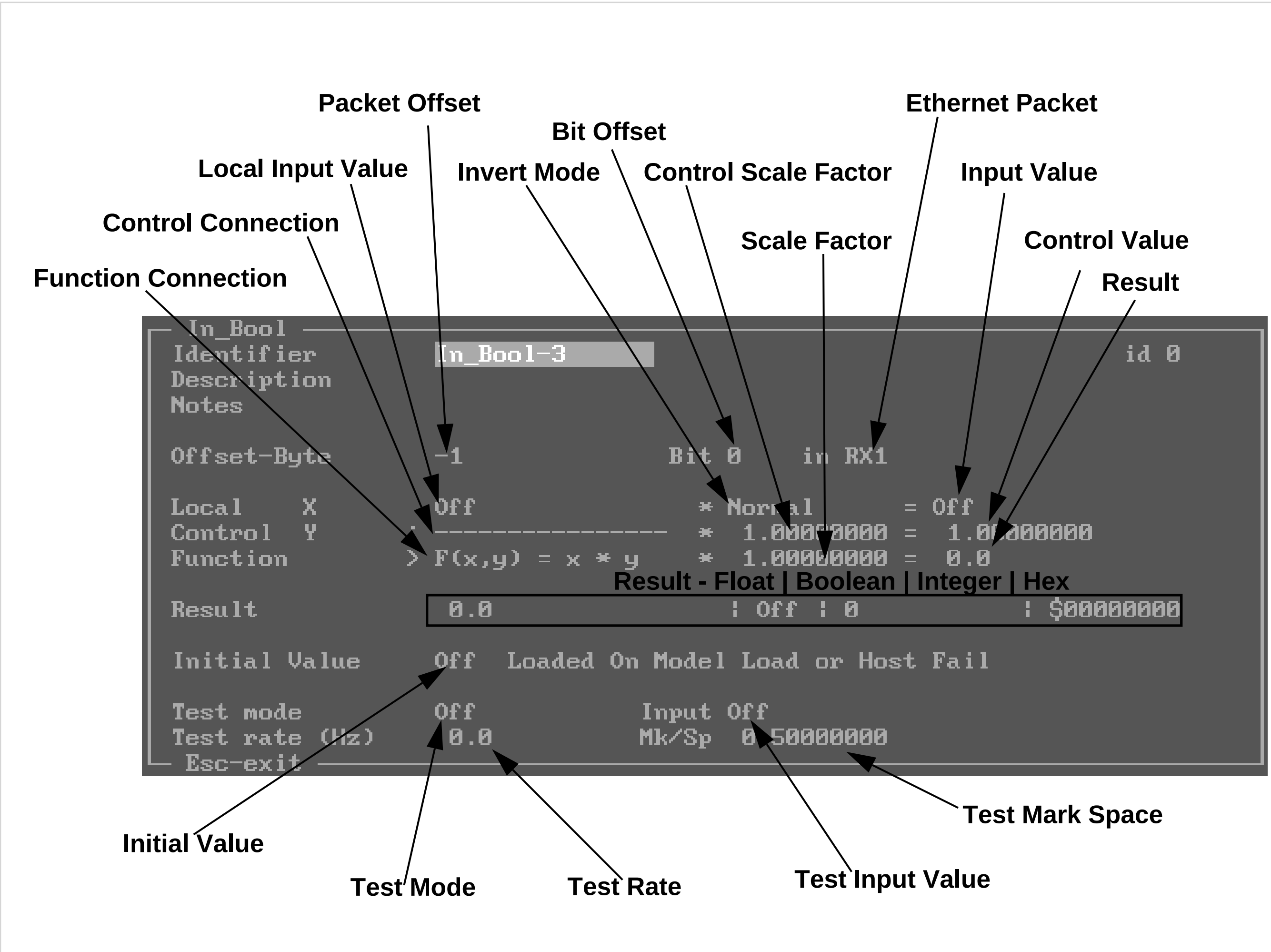


Boolean Input

This provides an interface between the host's Ethernet control packet and the object orientated model for any digital boolean variable (either byte or bit wide).

A local test mode is provided to override the incoming host boolean.

Boolean Input



Packet Offset

Offset to input variable from beginning of ethernet packet, in bytes.

Bit Offset

Bit in ethernet byte to be used as boolean, 0 is least significant bit, 7 is most significant.

Ethernet Packet

Ethernet packet used for variable access.

Local Value

The current value of the Boolean being read from the ethernet packet.

Invert Mode

Provides local logic inversion.

Input Value

The current value of the Boolean being read from the ethernet packet after inversion.

Control Connection

Connection to another control object for overall control of the floating point result.

Control Scale Factor

Local scale factor for the control variable.

Control Value

Control value used as the second variable for the function lookup, allowing scaling or offsetting by another model control.

Boolean Input

Function Connection

Selection of input scaling function. With the input value on the top of the stack and the control value on the bottom. The default (no selection) function multiplies the input value by the control value.

Scale Factor

Scale factor for function result.

Result - Float

Output result of input, floating point value. 1.0 if input bit true, 0.0 if input bit false.

Result - Boolean

Output result of input boolean.

Result - Integer

Output result of input, integer value, a rounding of the floating point result.

Result - Hex

Output result of input, integer value in hexadecimal format, a rounding of the floating point result.

Initial Value

Initial value loaded either at model load, or on each failure of the host communication link

NOTE: the initial value mode is selected for the entire model on the model configuration page.

Test Mode

Test mode enable flag allows user to override ethernet input value. Useful for debug purposes only.

Test Mark Space

Mark space ratio for test toggling of output result.

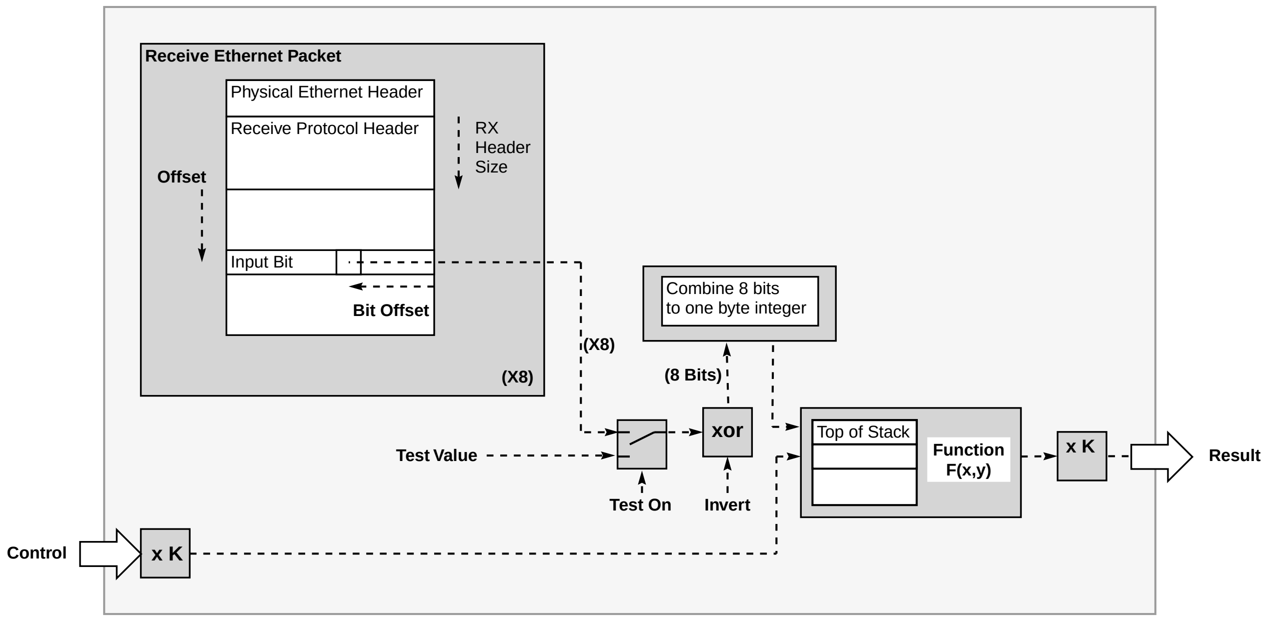
Test Value

Value for result when in test mode.

Test Rate

Rate (in Hertz) at which result value is toggled. Active only when Test mode is on.

8 Bit Input

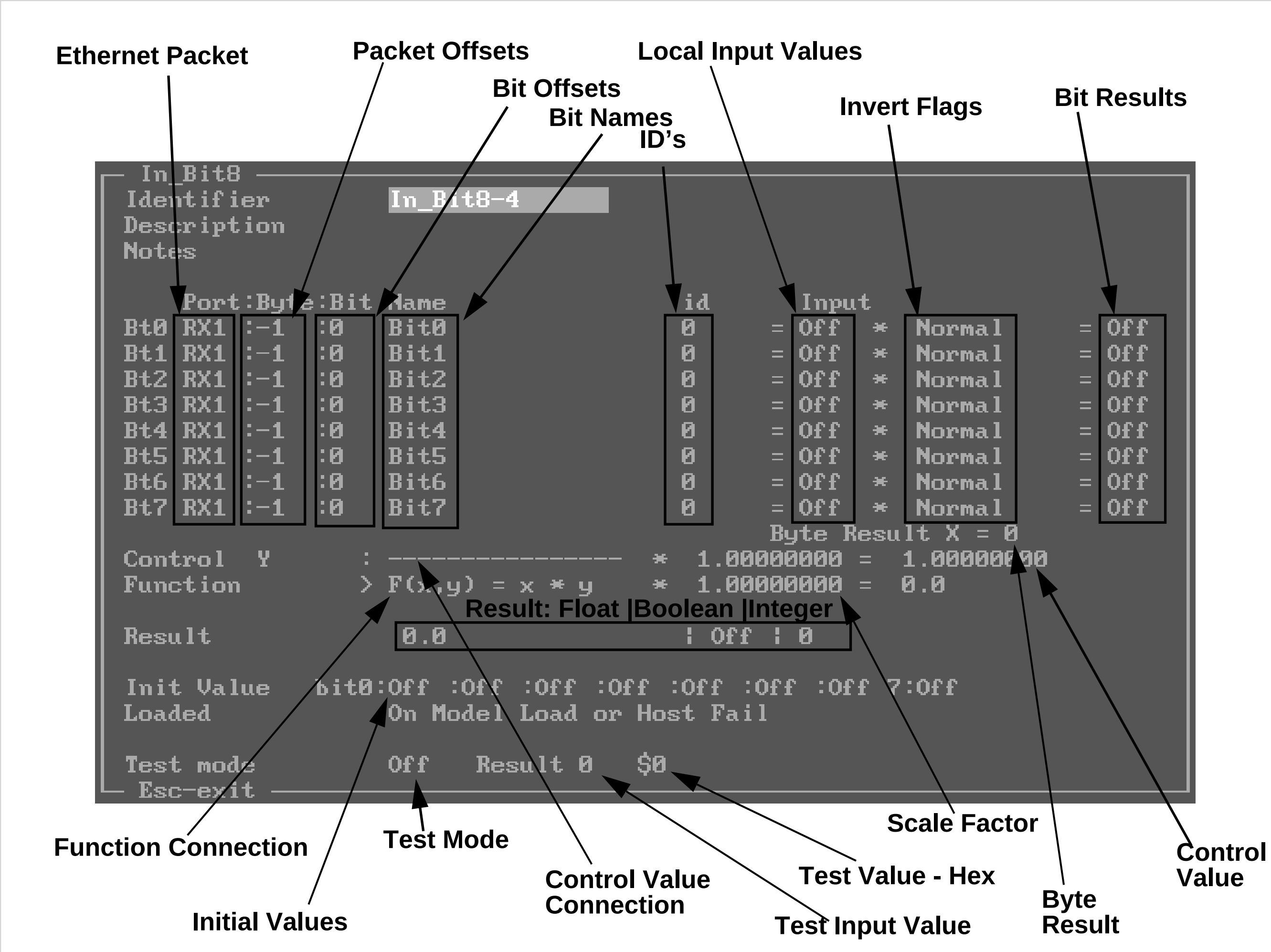


8 Bit Input

This provides an interface between the host's Ethernet control packet and the object orientated model. It pulls 8 bits from anywhere in the receive packet and combines them into a single byte.

A local test mode is provided to override the incoming host values.

8 Bit Input



Packet Offsets

Offsets to input variables from beginning of ethernet packet, in bytes.

Bit Offsets

Bit in ethernet bytes to be used as input, 0 is least significant bit, 7 is most significant.

Ethernet Packet

Ethernet packet used for variable access.

Local Input Values

The current values of the bits being read from the ethernet packet.

Bit Names

Names assigned to each bit. These names have no effect on the model operation, and are solely for the convenience of the user.

ID's

These numbers are not currently used.

Invert Flags

Provide local bit inversion.

Input Values

The current values of the bits being read from the ethernet packet after inversion.

Bit Results

Result, after input and inversion, for each bit.

Byte Result

Byte value of the combined 8 result bits.

8 Bit Input

Control Value Connection

Connection to another control object for overall control of the floating point result.

Control Scale Factor

Local scale factor for the control variable.

Control Value

Control value used as the second variable for the function lookup, allowing scaling or offsetting by another model control.

Function Connection

Selection of input scaling function. With the input value on the top of the stack and the control value on the bottom. The default (no selection) function multiplies the input value by the control value.

Scale Factor

Scale factor for function result.

Result - Float

Output result of input, floating point value. The eight bits are converted to a number from 0-255. This integer is acted on by the function and multiplied by the scale value to give the Float result.

Result - Boolean

Off if the float result is negative or less than 0.25, On if the float result is greater than 0.75. Between 0.25 and 0.75 the result is hysteretic - if off, it will stay off until the result float reaches 0.75. If on, it will stay on until the float falls below 0.25.

Result - Integer

Output result, a rounding of the floating point result.

Initial Values

Initial values loaded either at model load, or on each failure of the host communication link

NOTE: the initial value mode is selected for the entire model on the model configuration page.

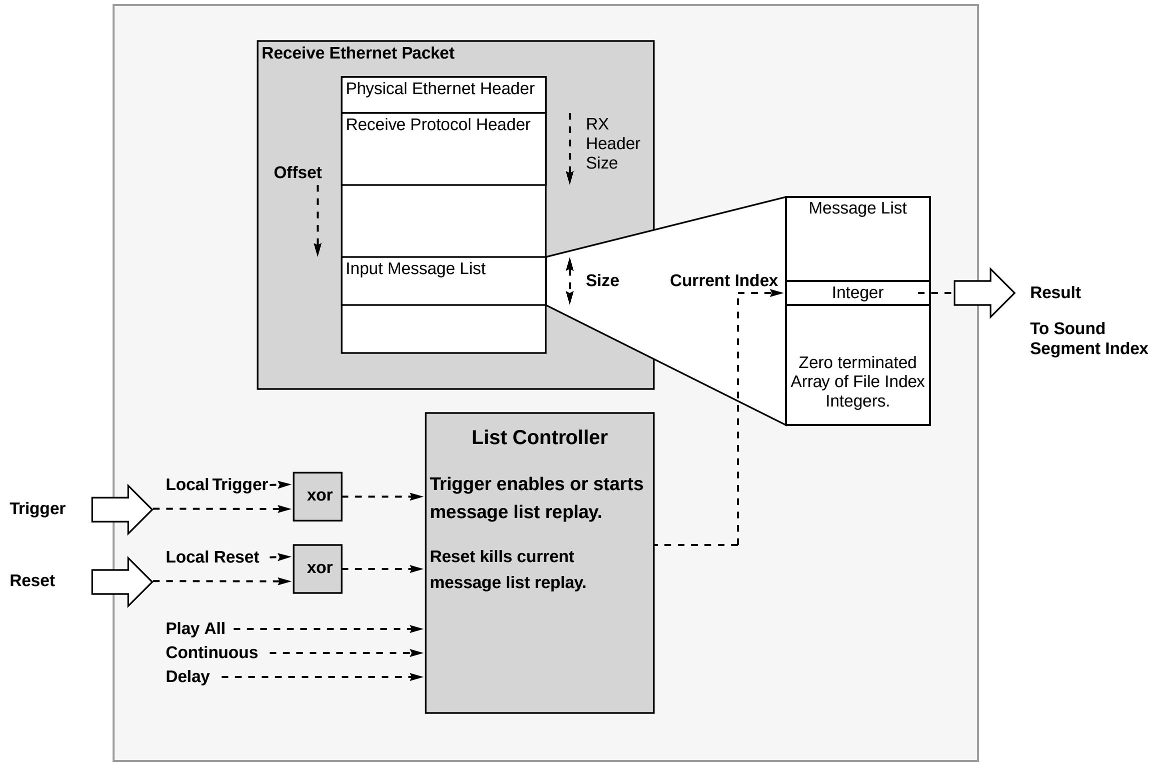
Test Mode

Test mode enable flag allows user to override ethernet input value. Useful for debug purposes only.

Test Value

Value for input bits in test mode, expressed as an integer from 0 to 255.

Message List



Message List Input

This provides an interface between the host's Ethernet control packet and the play sound object, allowing a sequence of separate sound files to be played.

The list controller maintains an internal pointer into the message list which increments each time the sound replay object completes a file replay. This off-loads the host computer from timing tasks associated with replaying voice messages consisting of multiple adjacent sound files.

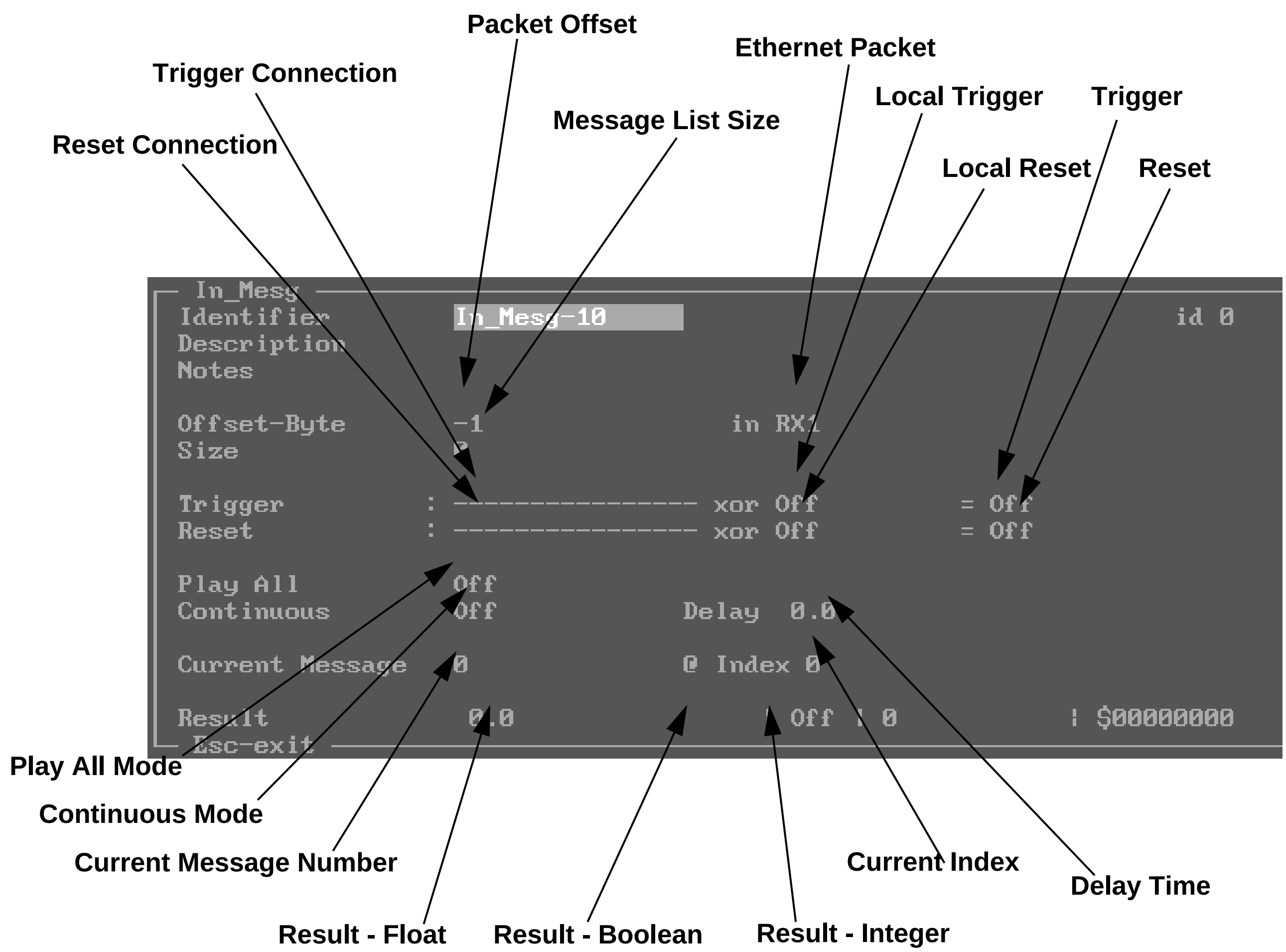
The output from this object can also be transmitted back to the host to allow a list completion value to be monitored.

External control of the message is provided by trigger and reset inputs. The trigger provides start stop control, while the reset allows for the forced truncation of a currently playing message.

The message list is defined either by the size of the memory area, or by a zero termination index. This allows messages shorter than the predefined size to be sent. As a result zero is unavailable as a file index code.

Currently the message list object only supports byte wide indexes, i.e. 1 through 255.

Message List



Packet Offset

Offset to start of message list from beginning of ethernet packet, in bytes.

Ethernet Packet

Ethernet packet used for variable access.

Message List Size

Maximum size of list allocated in ethernet packet (in bytes). The list can be zero terminated for variable length message lists.

Trigger Connection

Control object connection to provide start/stop control from elsewhere in model.

Local Trigger

Local value for the trigger. If no connection is made this allows the trigger to be left permanently in the on position. If a trigger connection is made then this provides an exclusive or function for inverting the trigger control state.

Trigger

The current trigger state, a value of on starts the list playing. If in continuous mode the list replays while this trigger is on, else if in one-shot mode the list replays once for each off to on transition of the trigger.

Message List

Reset Connection

Control object connection to provide start/stop control from elsewhere in model.

Local Reset

Local value for the reset.

Reset

The current reset state, a value of on kills the currently playing list and inhibits the trigger.

Play All Mode

Mode control flag, if set on forces the list to be played in its entirety, if off the list will stop playing when the trigger switches off.

Continuous Mode

If this flag is on the message list will be repeated continuously while the trigger is in the on state. When the trigger changes to off the list will cease playing, note if the play all flag is set then the list will play to completion.

Delay Time

When in continuous mode the delay time provides a space (in seconds) between restarting the playing of the list.

Current Message Number

The current value in the message list, i.e. the value to be sent to a play sound file index.

Current Index

The current index into the message list. With 0 being inactive, and 1 being first byte in list.

Result - Integer

Current list object value as a 32bit integer.

Result - Float

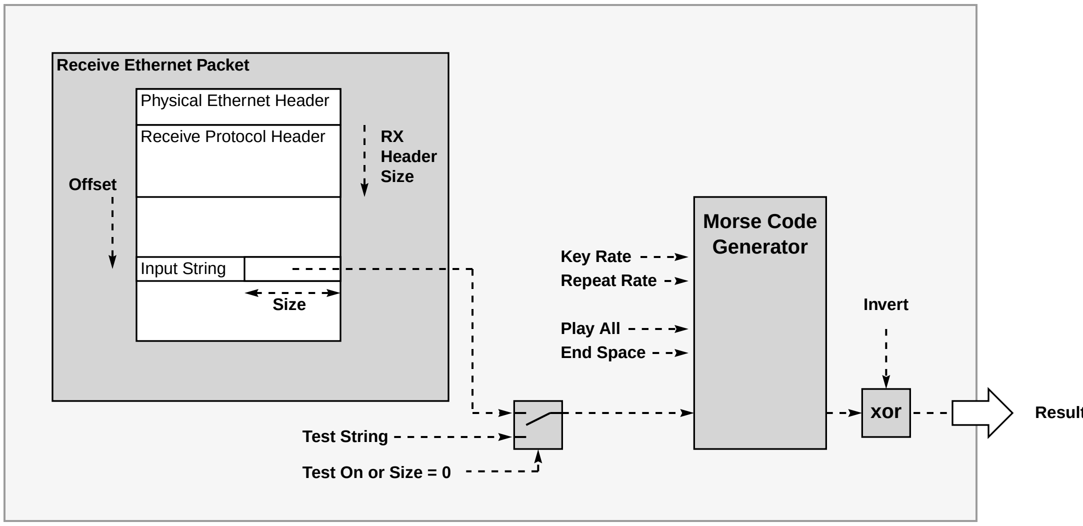
Output result of current message list object converted to a floating point value.

Result - Boolean

Output result of current message list object converted to a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

Host Morse Inputs (Nav-Aids Package)

Morse Keyer

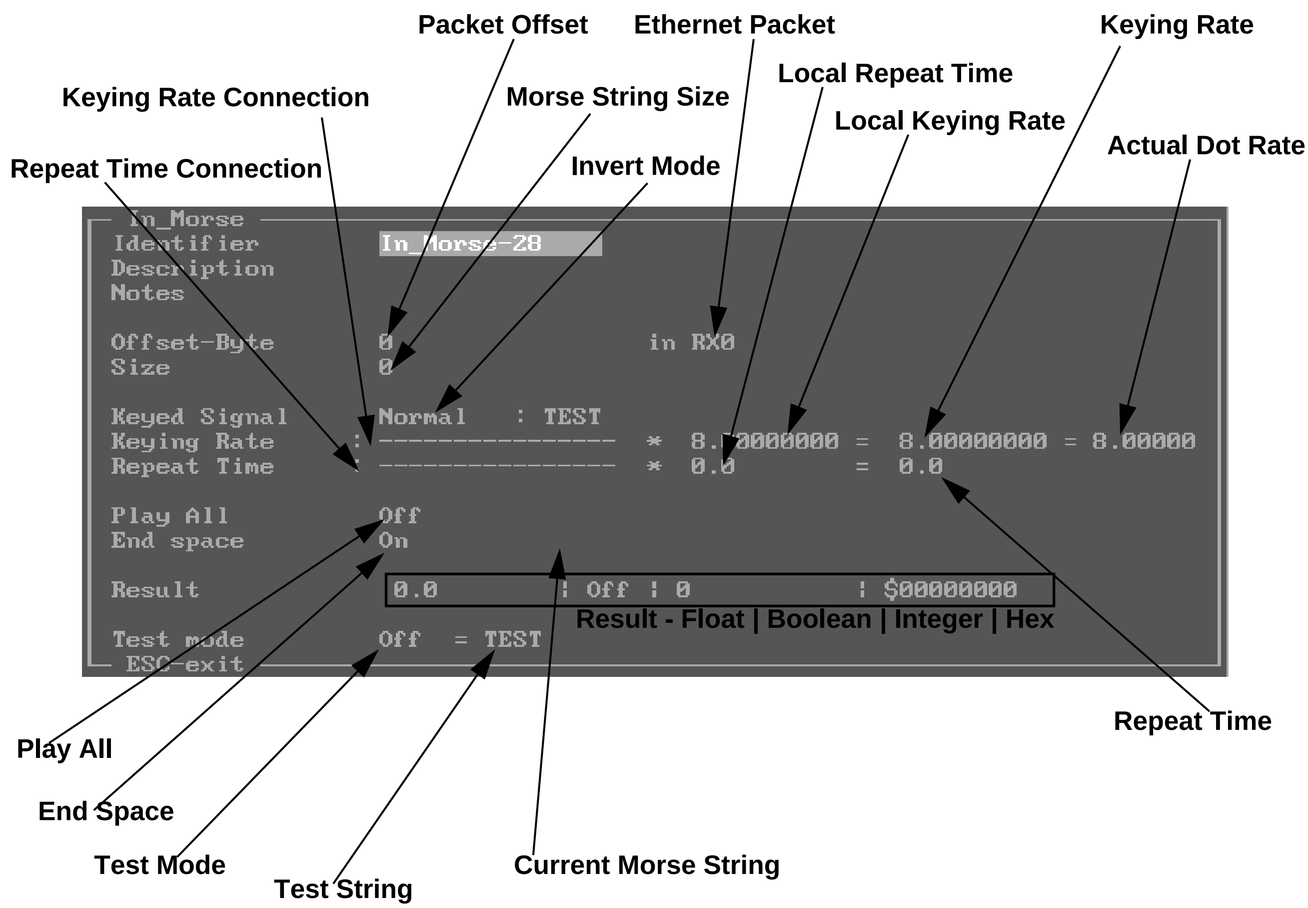


Morse Keyer Input

This provides an interface between the host's Ethernet control packet and the object orientated model for any ASCII string driven morse code sequence.

It decodes the incoming zero terminated ASCII string into the correct sequence of on/off pulses required for morse code communication. In addition to the usual letters and numbers defined in the morse code, it also includes the characters (* and -) to represent individual dot and dash combinations. The morse keyer has configuration control for the keying rate, whether an end of word space should be placed at the end of the string, and whether the current string should always be keyed to completion.

Morse Keyer



Packet Offset

Offset to input variable from beginning of Ethernet packet, in bytes.

Ethernet Packet

Ethernet packet used for variable access.

Morse String Size

Maximum size of morse string allocated in Ethernet packet. The string can be zero terminated for variable length strings. A Size value of zero forces the use of the test string.

Keying Rate Connection

Connection to another control object for overall morse key rate.

Local Keying Rate

Local key rate or scale factor for key rate connection.

Keying Rate

Morse code key rate in dots per second. (default 8 dots per second = 125 ms per dot).

Repeat Time Connection

Connection to another control object for morse string repeat time.

Local Repeat Time

Local repeat time or scale factor for repeat time connection.

Repeat Time

Repeat period (in seconds) for retransmission of morse code string.

Morse Keyer

Actual Dot Rate

Morse code dot rate. Note: this is the rate in use by the keyer selected as the nearest rate which can be derived by integer subdivision from the model execution rate.

Play All

Mode flag for forcing entire ident string to be completed before starting a new string.

End Space

Mode flag to control whether the morse string is terminated with an inter-word gap (6 spaces) before re-transmission.

Invert Mode

Provides local logic inversion of keying.

Current Morse String

Morse code string currently being used.

Result - Float

Output result of input, floating point value.

Result - Boolean

Output result of input, as a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

Result - Integer

Output result of input, integer value, a rounding of the floating point result.

Result - Hex

Output result of input, integer value in hexadecimal, a rounding of the floating point result.

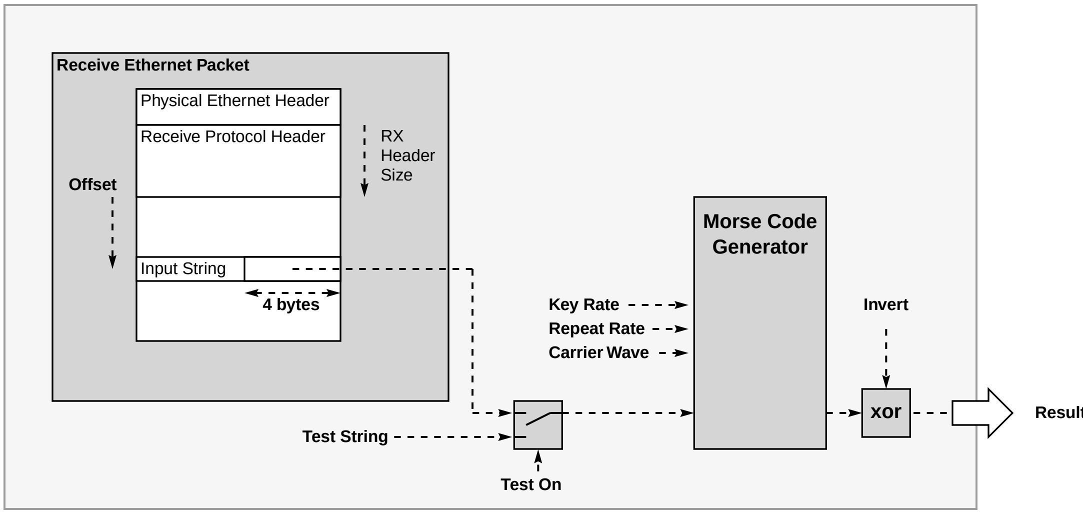
Test Mode

Test mode enable flag allows user to override Ethernet input value. Useful for debug purposes only.

Test String

Morse code string used when in test mode.

Morse Identifier

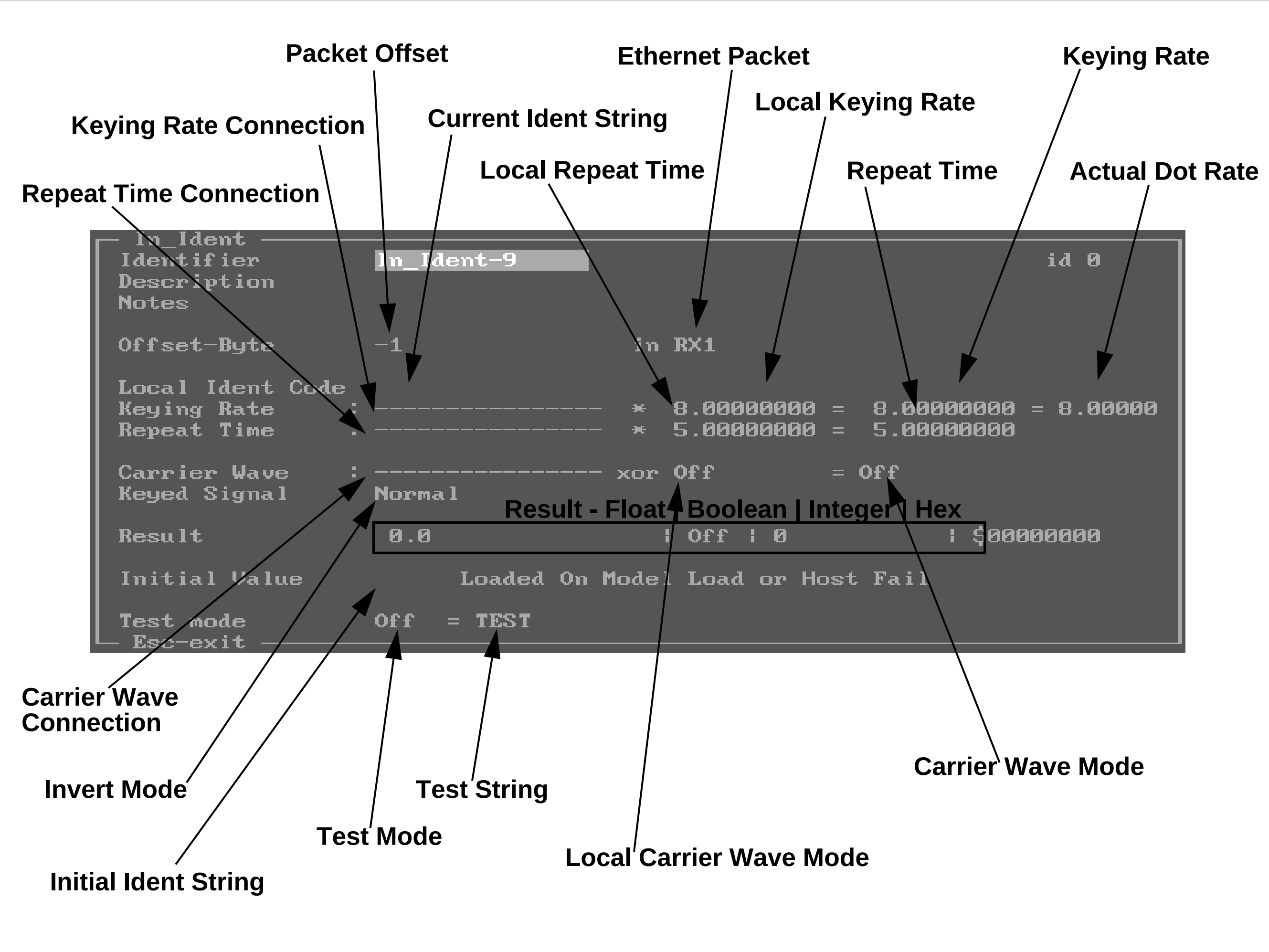


Morse Identifier Input

This provides an interface between the host's Ethernet control packet and the object orientated model for any 4 character ASCII string identifier sequence.

It decodes the incoming zero terminated ASCII string into the correct sequence of on/off pulses required for ident code communication. In addition to the usual letters and numbers defined in the morse code, it also includes the characters (* and -) to represent individual dot and dash combinations. The morse keyer has configuration control for the keying rate, and the repeat rate for the identifier.

Morse Identifier



Packet Offset
Offset to input variable from beginning of Ethernet packet, in bytes.

Ethernet Packet
Ethernet packet used for variable access.

Current Identifier String
Morse code string currently being used.

Keying Rate Connection
Connection to another control object for overall morse key rate.

Local Keying Rate
Local key rate or scale factor for key rate connection.

Keying Rate
Morse code key rate in dots per second. (default 8 dots per second = 125 ms per dot).

Repeat Time Connection
Connection to another control object for morse string repeat time.

Local Repeat Time
Local repeat time or scale factor for repeat time connection.

Repeat Time
Repeat period (in seconds) for retransmission of morse code string.

Morse Identifier

Actual Dot Rate

Morse code dot rate. Note: this is the rate in use by the keyer selected as the nearest rate which can be derived by integer subdivision from the model execution rate.

Carrier Wave Mode Connection

Connection to another control object for carrier wave control.

Local Carrier Wave Mode

Local carrier mode flag or exclusive-or function for carrier wave connection.

Carrier Wave Mode

Provides control of carrier wave state during gaps in identifier. When On the identifier has spaces appended to the front and back of the string, and the carrier wave is on when not keying morse string. Intended for use with the continuous wave non-directional beacon.

Invert Mode

Provides local logic inversion of keying.

Result - Float

Output result of input, floating point value.

Result - Boolean

Output result of input, as a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

Result - Integer

Output result of input, integer value, a rounding of the floating point result.

Result - Hex

Output result of input, integer value in hexadecimal, a rounding of the floating point result.

Initial Identifier String

Morse code string loaded at start-up or on host communication fail.

Test Mode

Test mode enable flag allows user to override Ethernet input value. Useful for debug purposes only.

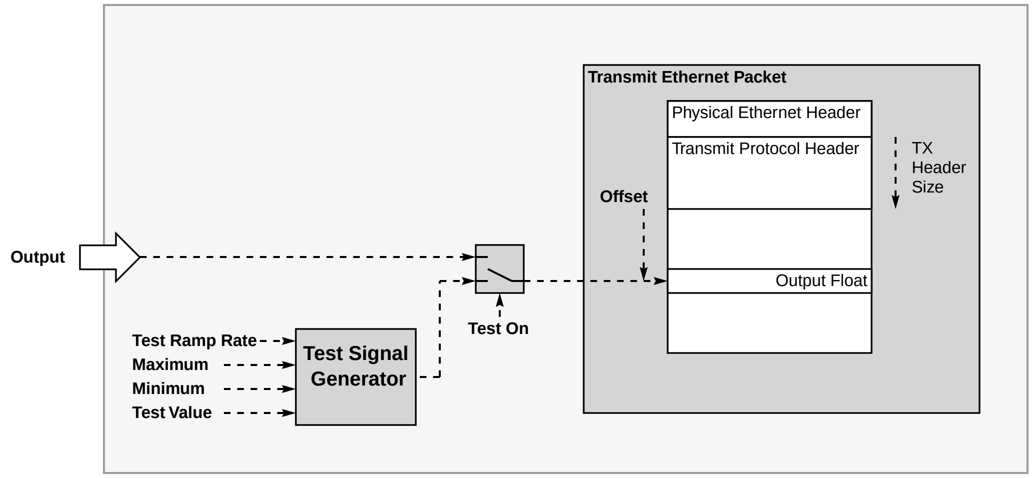
Test Identifier String

Morse code string used when in test mode.

Host Outputs

The host outputs provide objects for connecting any control variable back to the transmission Ethernet packet.

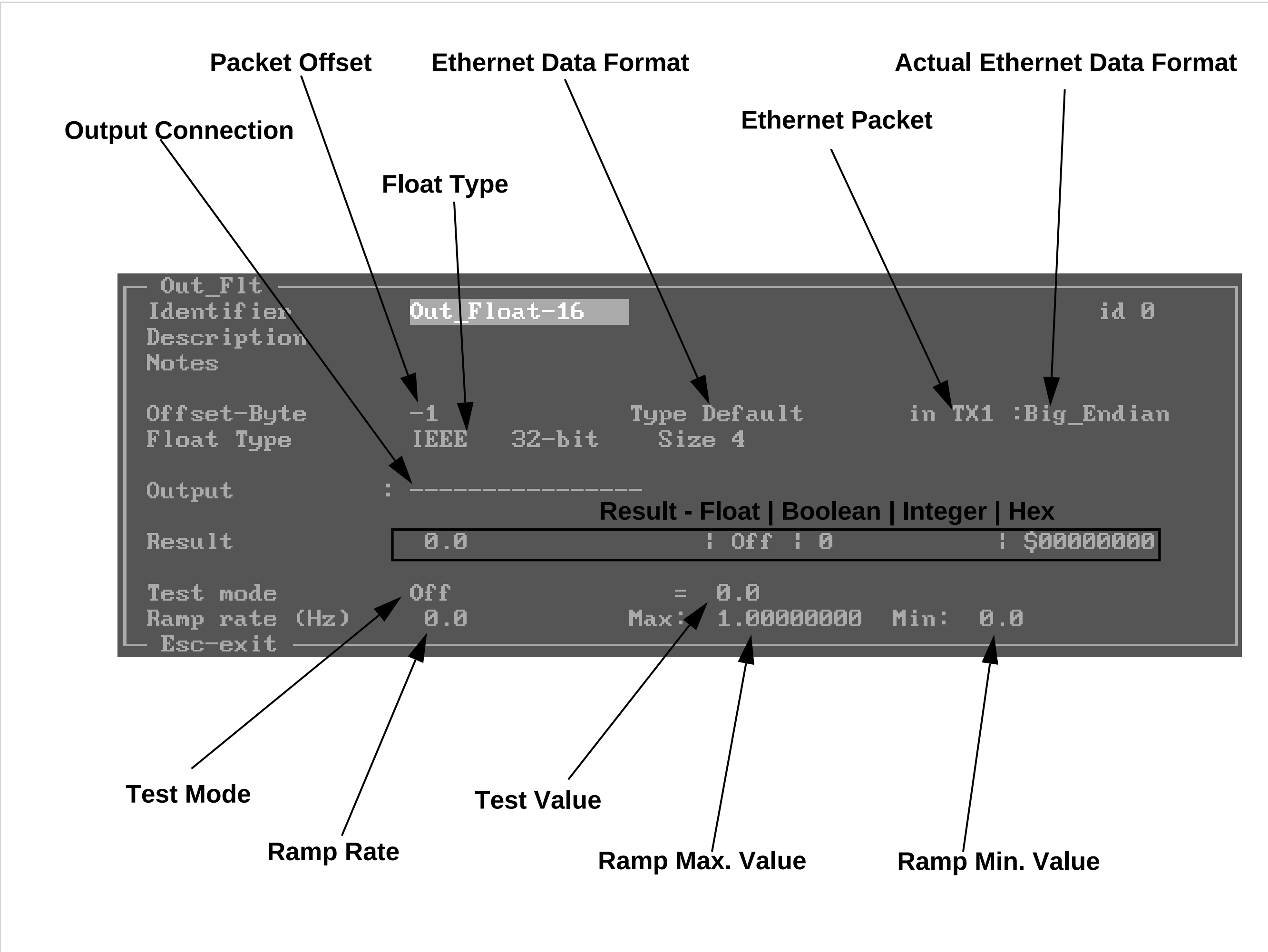
Float Output



Float Output

This provides an interface between a control variable (floating value) in the model and the Ethernet transmission packet.

Float Output



Packet Offset

Offset to output variable from beginning of Ethernet packet, in bytes.

Ethernet Data Format

Format of the Ethernet variable, includes Default, Big Endian, Little Endian, and Encore formats.

Ethernet Packet

Ethernet packet used for variable access.

Float Type

Format for floating point number.

Actual Ethernet Data Format

Format of the Ethernet variable used to write data, usually same as above, except when default type, then the format used is that specified on the Ethernet status page.

Output Connection

Connection to control object result to be sent to Ethernet packet.

Test Mode

Test mode enable flag allows user to override output value.

Test Value

Value for result when in test mode, and ramping disabled.

Ramp Rate

Rate (in Hertz) at which to change result value between maximum and minimum values. Active only when Test mode is on.

Ramp Min. Value

Minimum value for result when in test ramp mode.

Float Output

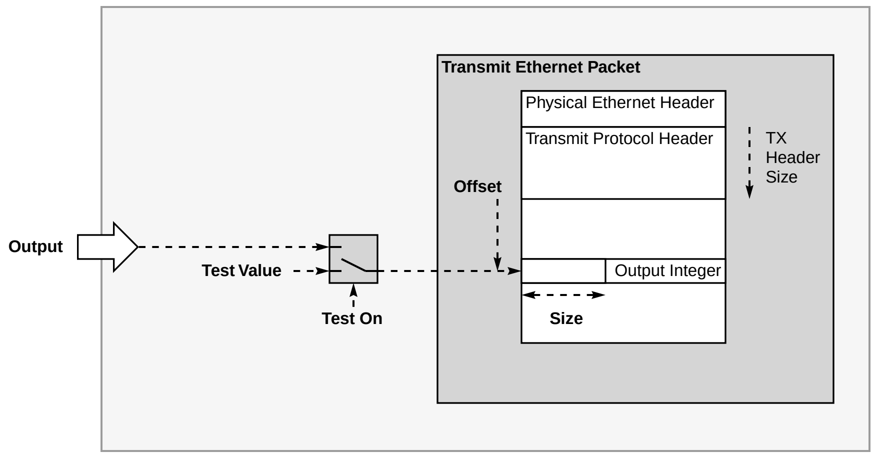
Ramp Max. Value

Maximum value for result when in test ramp mode.

Result - Float

Floating point value to be sent to Ethernet packet.

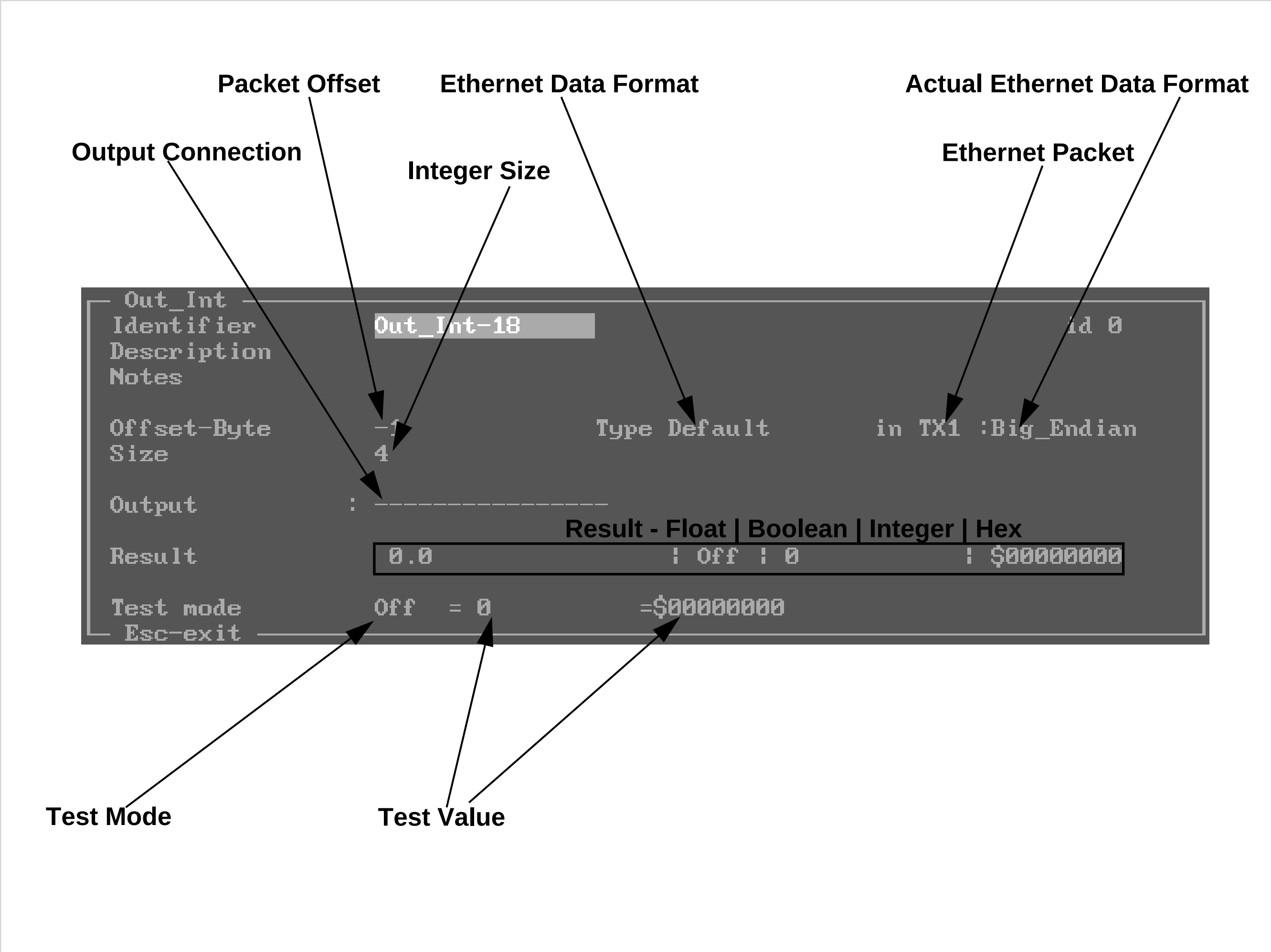
Integer Output



Integer Output

This provides an interface between a control variable (integer value) in the model and the Ethernet transmission packet.

Integer Output



Packet Offset

Offset to output variable from beginning of Ethernet packet, in bytes.

Ethernet Data Format

Format of the Ethernet variable, includes Default, Big Endian, Little Endian, and Encore formats.

Ethernet Packet

Ethernet packet used for variable access.

Actual Ethernet Data Format

Format of the Ethernet variable used to write data, usually same as above, except when default type, then the format used is that specified on the Ethernet status page.

Integer Size

Size of integer variable in Ethernet packet. Valid sizes, 1,2, and 4 byte integers.

Output Connection

Connection to control object result to be sent to Ethernet packet.

Test Mode

Test mode enable flag allows user to override output value.

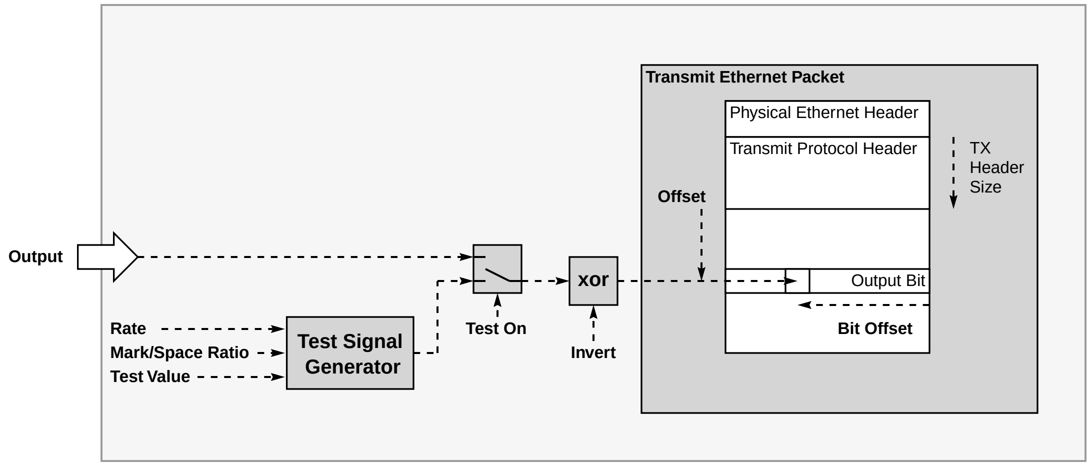
Test Value

Value for result when in test mode - Decimal or Hexadecimal.

Result - Integer

Output result, integer value.

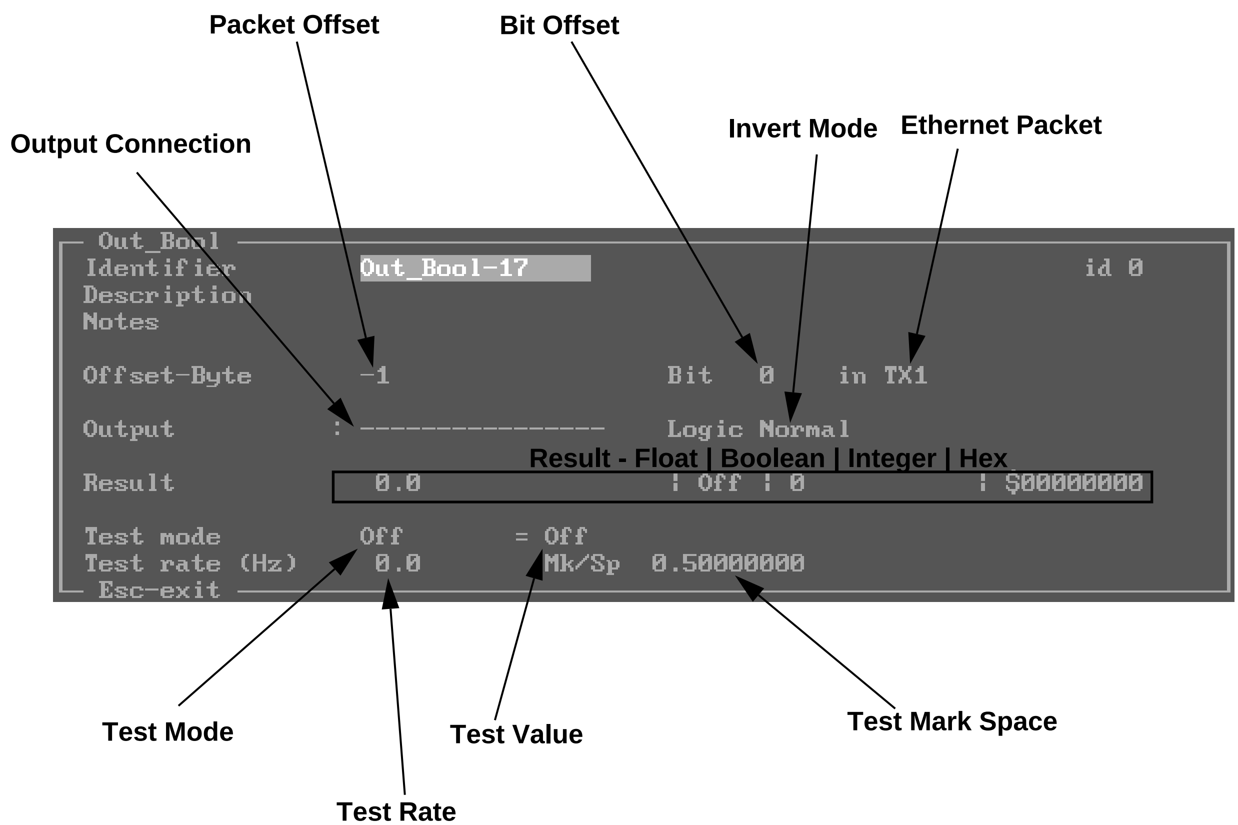
Boolean Output



Boolean Output

This provides an interface between a control variable (boolean value) in the model and the Ethernet transmission packet.

Boolean Output



Packet Offset

Offset to output variable from beginning of Ethernet packet, in bytes.

Bit Offset

Bit in Ethernet byte to be used as boolean, 0 is least significant bit, 7 is most significant.

Ethernet Packet

Ethernet packet used for variable access.

Output Connection

Connection to control object result to be sent to Ethernet packet.

Invert Mode

Provides local logic inversion.

Test Mode

Test mode enable flag allows user to override output value.

Test Mark Space

Mark space ratio for test toggling of output result.

Test Value

Value for result when in test mode.

Test Rate

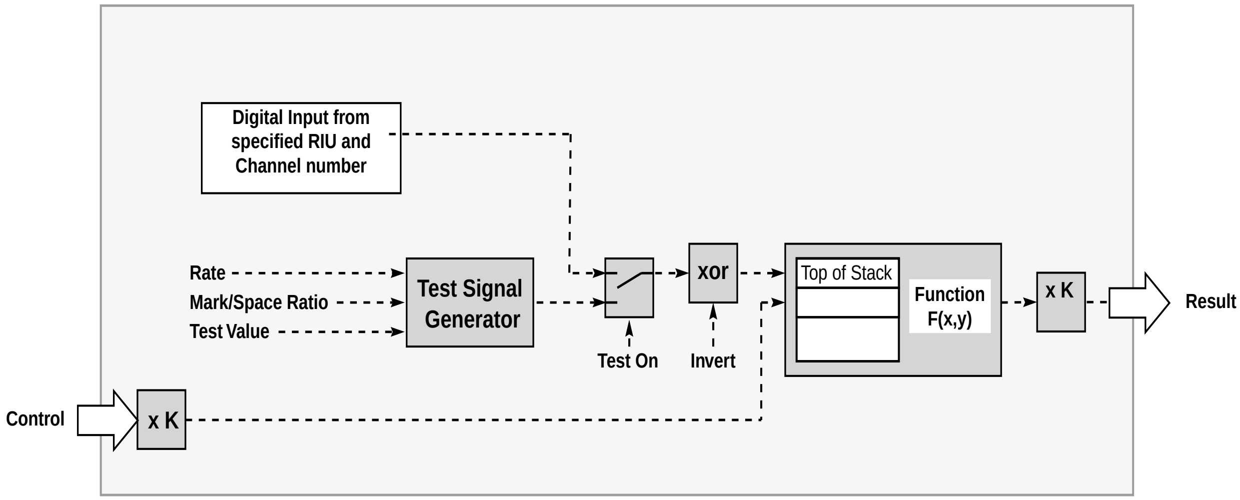
Rate (in Hertz) at which result value is toggled. Active only when Test mode is on.

Result - Boolean

Output result of input boolean.

RIU Digital I/O (RIU Package)

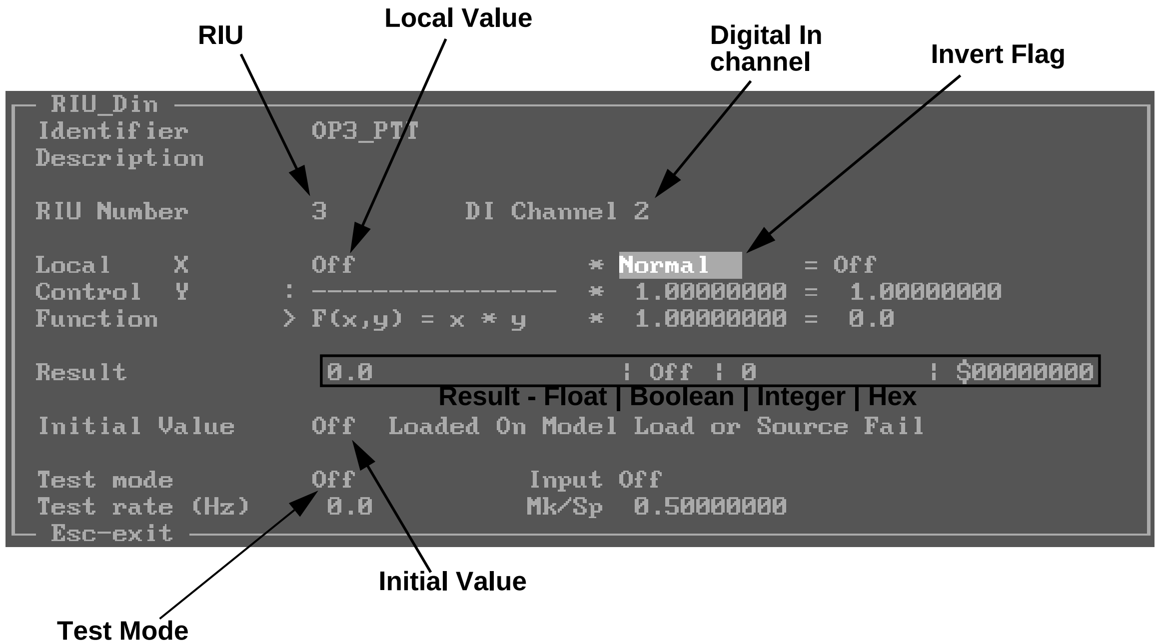
RIU Digital Input



RIU Digital Input

The RIU digital input object provides access to the digital inputs on the RIU. Each RIU has 4 single bit digital inputs, numbered 0 to 3. In order for an RIU digital input to be read, an RIU audio input object for the RIU must be put in the feeders list.

RIU Digital Input



RIU

Determines which RIU the digital input is being read from. The RIU number is set by a rotary switch on the RIU, and can range from 1 to 16. A value of zero tells the object not to take input from any RIU, so the Local Value will just remain at the Initial Value.

Digital In Channel

Determines which digital input channel is being read. Di channel 0 is the digital input on channel A of the RIU. Di channel 3 is the digital input on channel D of the RIU.

Di channels 4 through 10 are only accessible on custom packaged RIU's.

Local Value

The local value being read from the RIU. If the RIU is not on the TDM ring, or if an RIU analog input feeder for the RIU has not been put in the feeders list, the local value will be taken from the initial value. The value will be either ON or OFF.

Invert Flag

Logically inverts the local value.

Result

The final result for the object. Four values are given - float, boolean, integer, and Hex. The appropriate format will be used depending on where the object is attached. (e.g., if the object is plugged into a gain, the float value will be used.)

RIU Digital Input

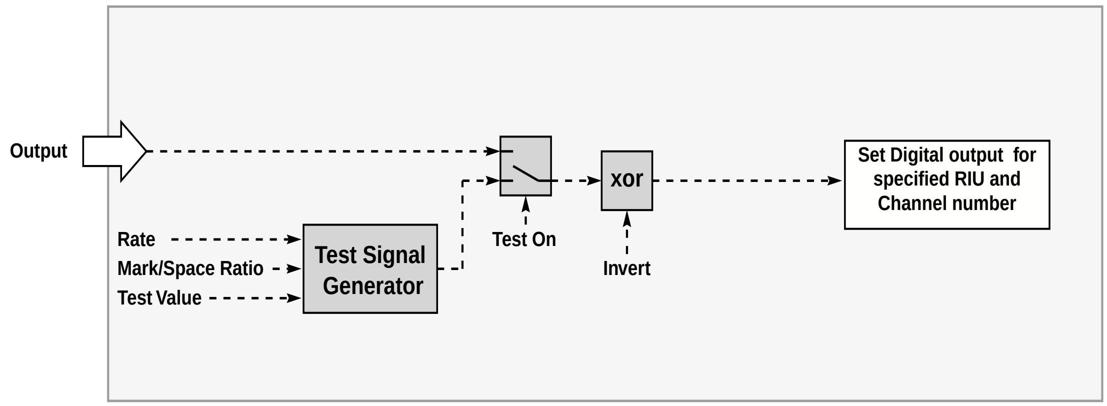
Test Mode

Overrides the local value from the RIU. The test mode is set to OFF when the model is loaded.

Initial Value

The initial value is loaded into the local value at start-up. It may also be loaded when the RIU fails, based on the initialize inputs flag set in the model configuration window.

RIU Digital Output

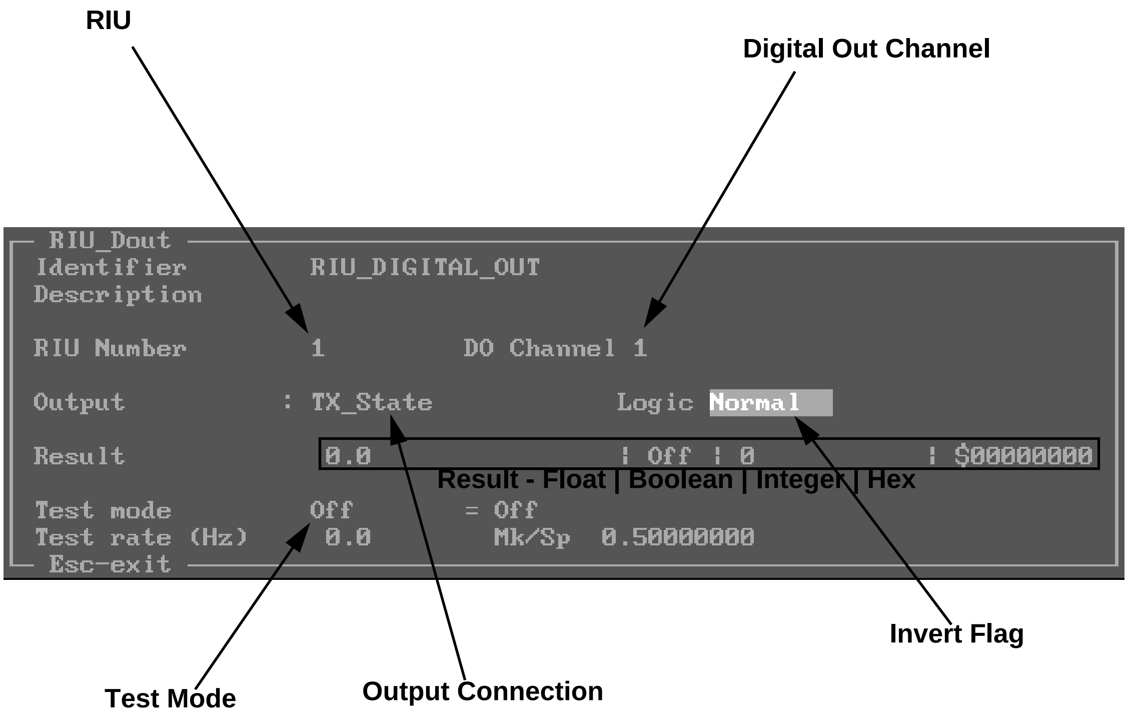


RIU Digital Output

The RIU digital output allows the software to drive a digital output on the RIU. Each RIU has four single bit digital outputs.

For information on the wiring of the digital output, see the Operations and Maintenance manual.

RIU Digital Output



RIU

Determines which RIU the digital output is being sent to. The RIU number is set by a rotary switch on the RIU, and can range from 1 to 16. If this value is zero, no RIU will receive the Digital Output.

Digital Out Channel

Determines which digital output channel is being driven. Digital Out channel 0 is the digital output on channel A of the RIU. Channel 3 is the digital output on channel D of the RIU.

Output Connection

Connection to a control object which determines the value to be driven.

Test Mode

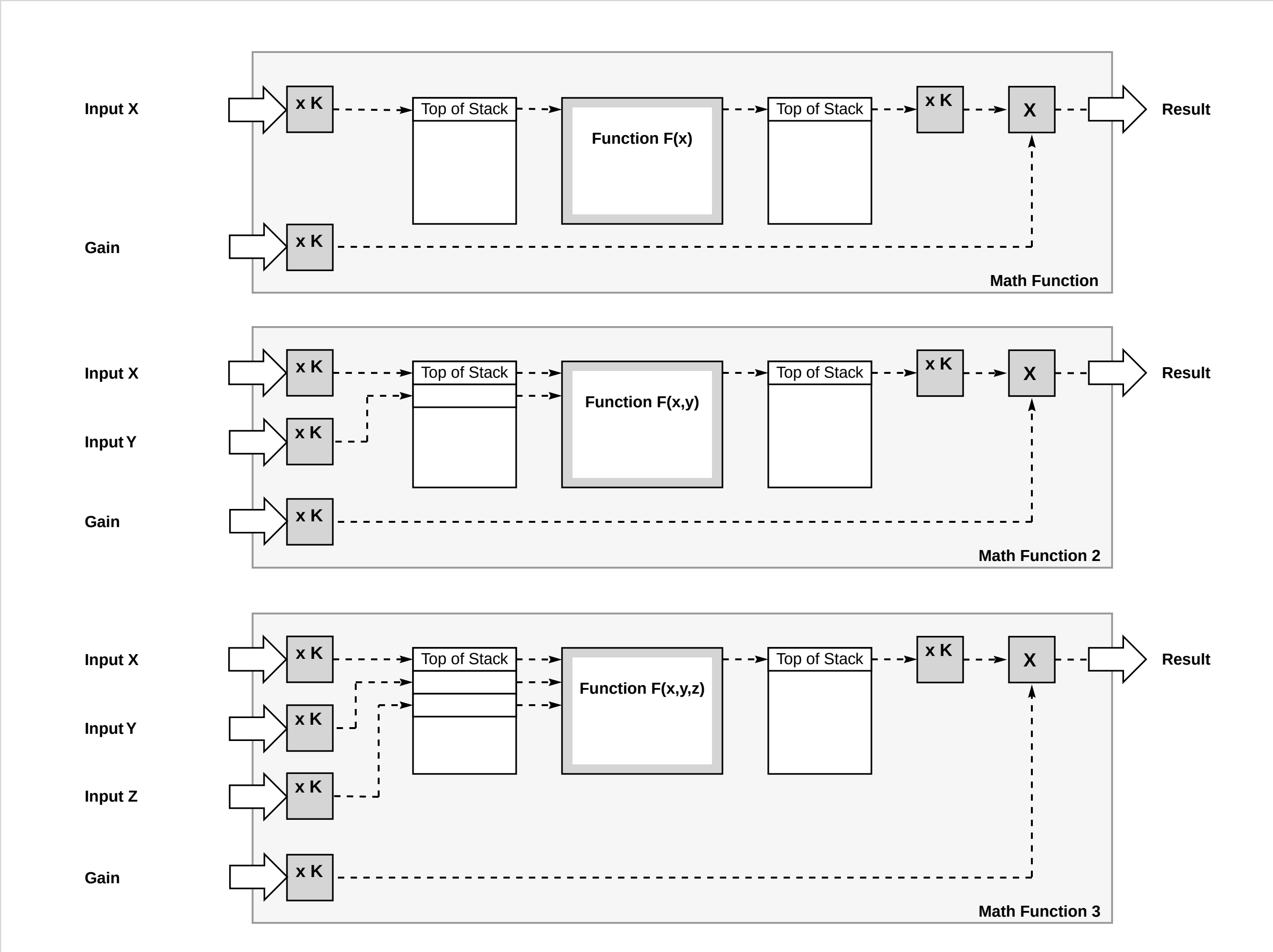
If the test mode is on, the digital output is driven from the test mode and the output connection is ignored.

Invert Flag

If on, the invert flag logically inverts the value of the Output connection before sending it to the digital output.

Internal Controls

Math Function

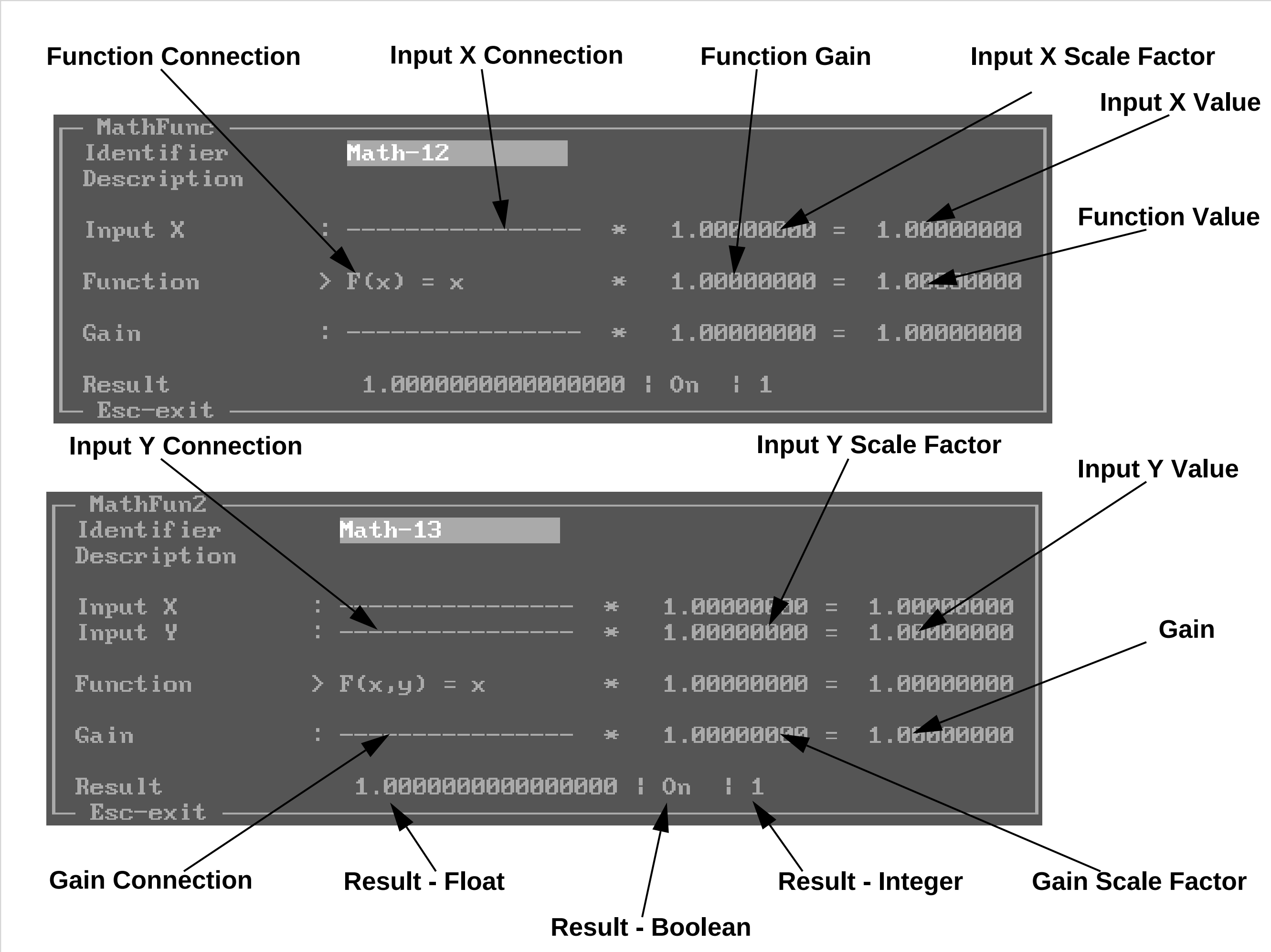


Math Function

The math function objects provide table lookups or mathematical functions within the model. Three types of math function are supported, single, double and triple variable, i.e. function of input X, function of input X and Y or function of input X,Y and Z. This object is connected to a list of different math functions described in the “Functions” section of this manual.

Each of these functions is a stack orientated calculation, with the first variable on the top of the stack, and the last at the bottom. The discrete functions can be grouped together to form more complicated function, all of which place their results on the stack, thus allowing the result of one function to be the input for the next.

Math Function



Input X Connection
Connection to another control object which provides the first variable to be used by the math function.

Input Y Connection
Connection to another control object which provides the second variable to be used by the math function.

Input Z Connection
Connection to another control object which provides the third variable to be used by the math function.

Input X,Y,Z Scale Factor
Scale factor for variables before application of selected function.

X Value.
First variable used in function, appears a top stack element.

Y Value
Second variable used in function, appears as second stack element.

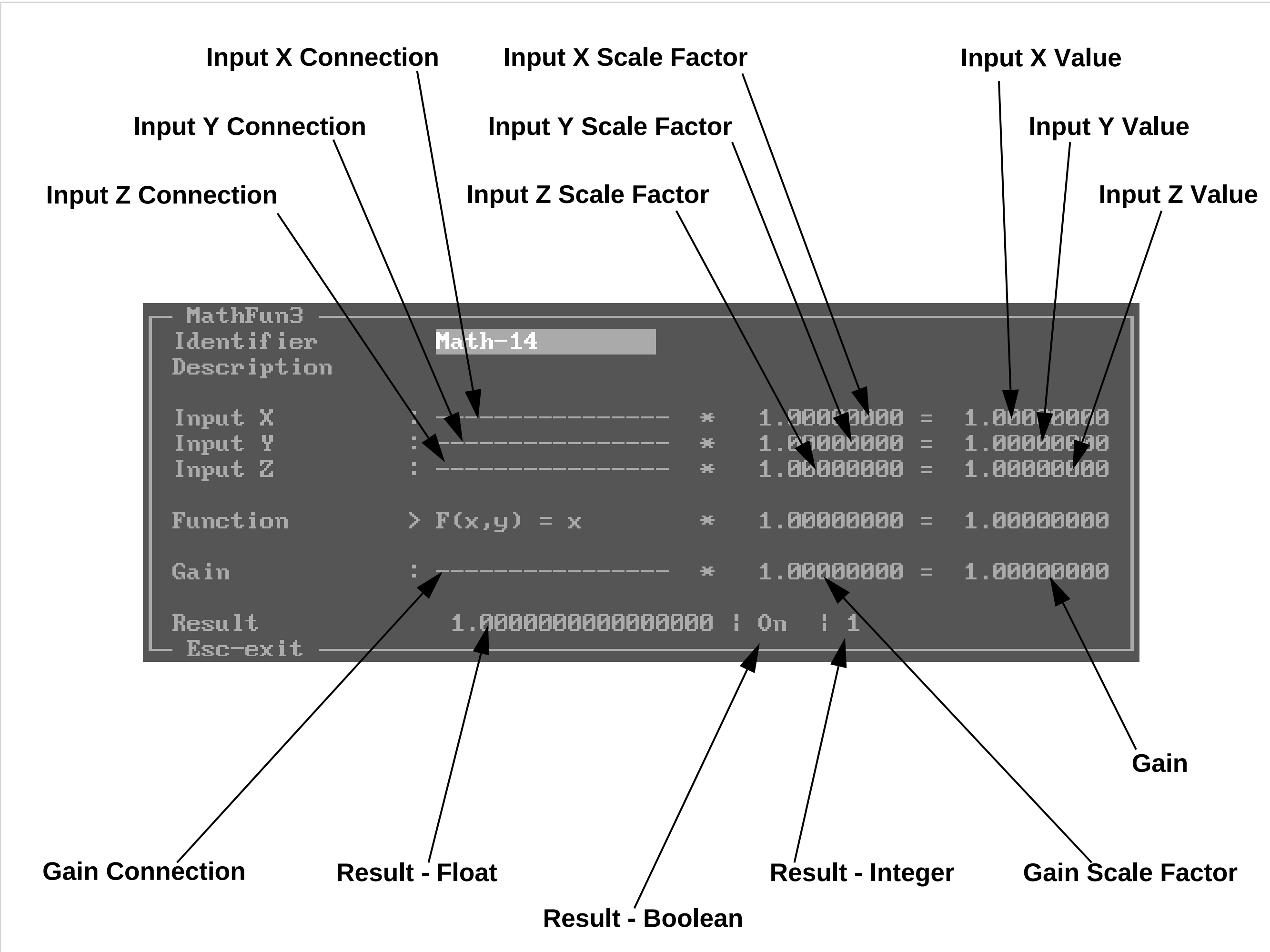
Z Value
Third variable used in function, appears as third stack element.

Function value
Value after function evaluation.

Function Connection
Connection to the function that will act on the input value(s).

Function gain
Scaling factor for function output.

Math Function



Gain Connection

Control object connection to provide overall gain control of the math function result from elsewhere in the model.

Gain Scale Factor

Scaling factor for overall gain control value.

Gain

Overall gain of the math function output result. If the gain connection is blank then the gain scale factor is used as the gain value, otherwise the gain is the scale factor times the output result of the control object.

Result - Float

Output result of math function, floating point value.

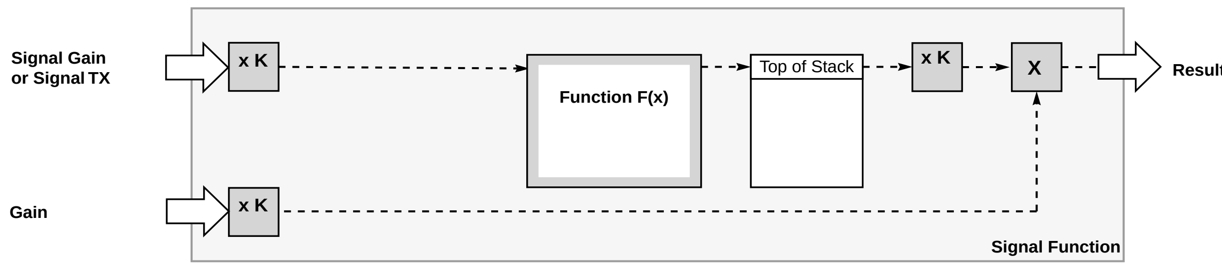
Result - Boolean

Output result of math function, as a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

Result - Integer

Output result of math function, integer value, a rounding of the floating point result.

Signal Function

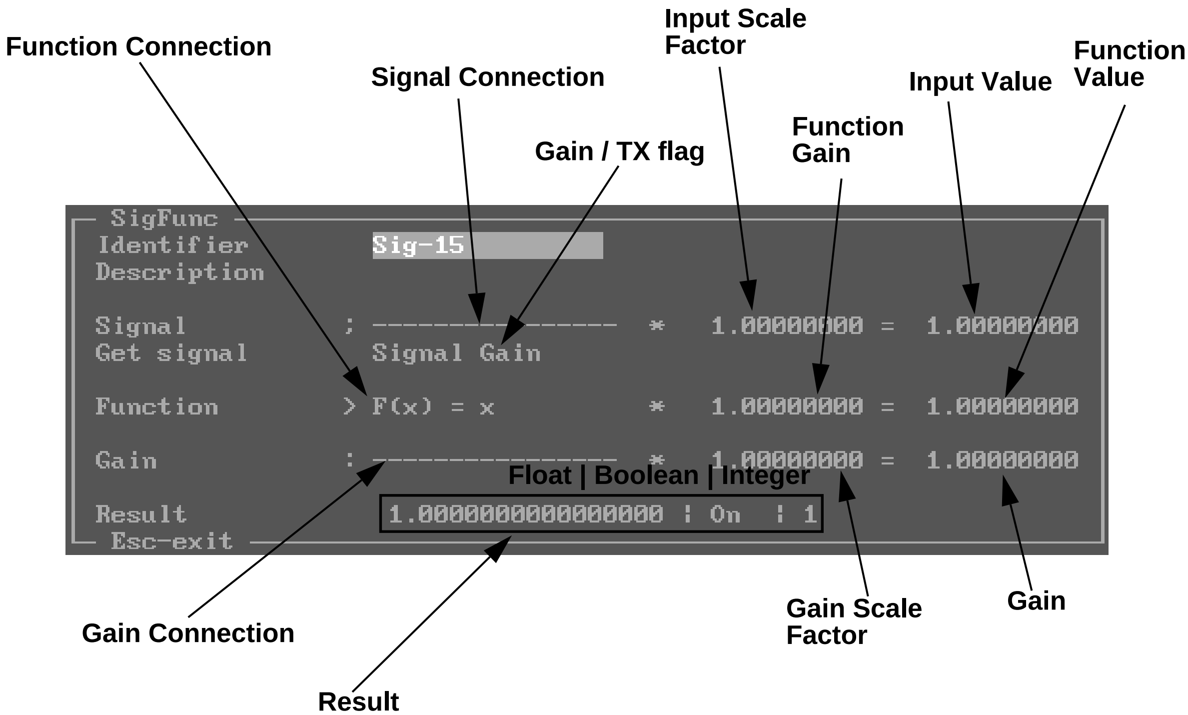


Signal Function

The signal function is similar to the single variable math function, except that it takes as input either the gain of a signal or the signal Tx value (1 if Tx is on, 0 otherwise).

This object is useful for obtaining radio Tx states, RX signal/noise, marker tone on/off state, and other signal object results for use as a control value or for transmission of the state back to the host computer via one of the output objects.

Signal Function



Gain / TX Flag

This flag determines if the input value is the signal gain, or the TX value. If it is set to TX, the input value is one when the signal is transmitting (has a non zero amplitude) and zero otherwise.

Signal Connection

Connection to the signal object that provides the input value.

Input Scale Factor

A scale factor for the input value taken from the Signal Connection.

Input Value

The value of the input to the function in the function connection.

Function Connection

Connection to the function that acts on the input value.

Function Value

Value after function look-up.

Function Gain

Scaling factor for function output.

Gain Connection

Control object connection to provide overall gain control of the math function result from elsewhere in the model.

Gain Scale Factor

Scaling factor for overall gain control value.

Signal Function

Gain

Overall gain of the math function output result. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Result - Float

Output result of math function, floating point value.

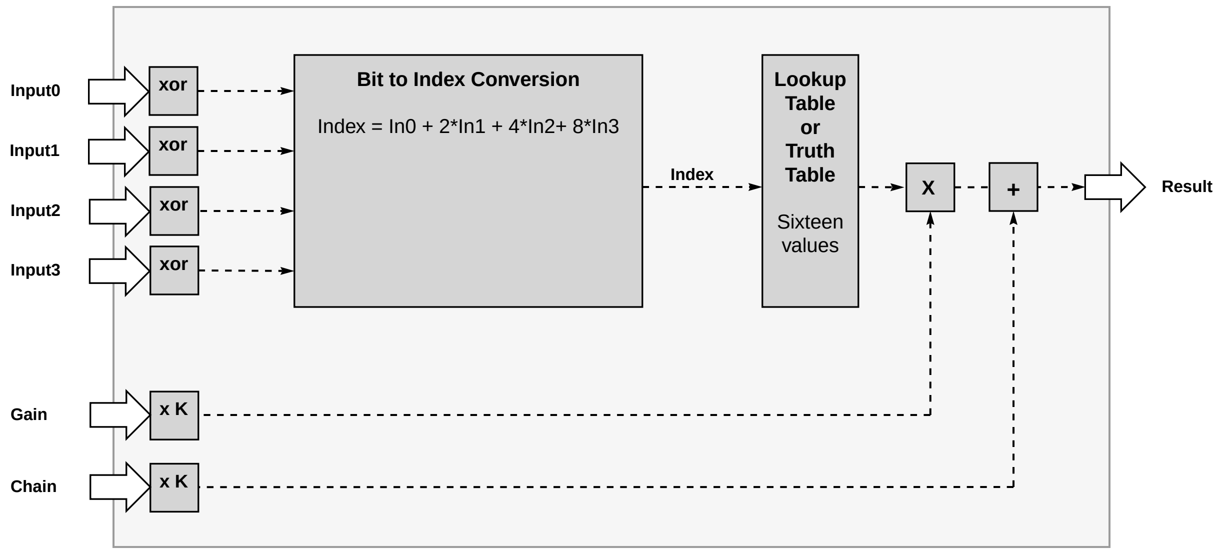
Result - Boolean

Output result of math function, as a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

Result - Integer

Output result of math function, integer value, a rounding of the floating point result.

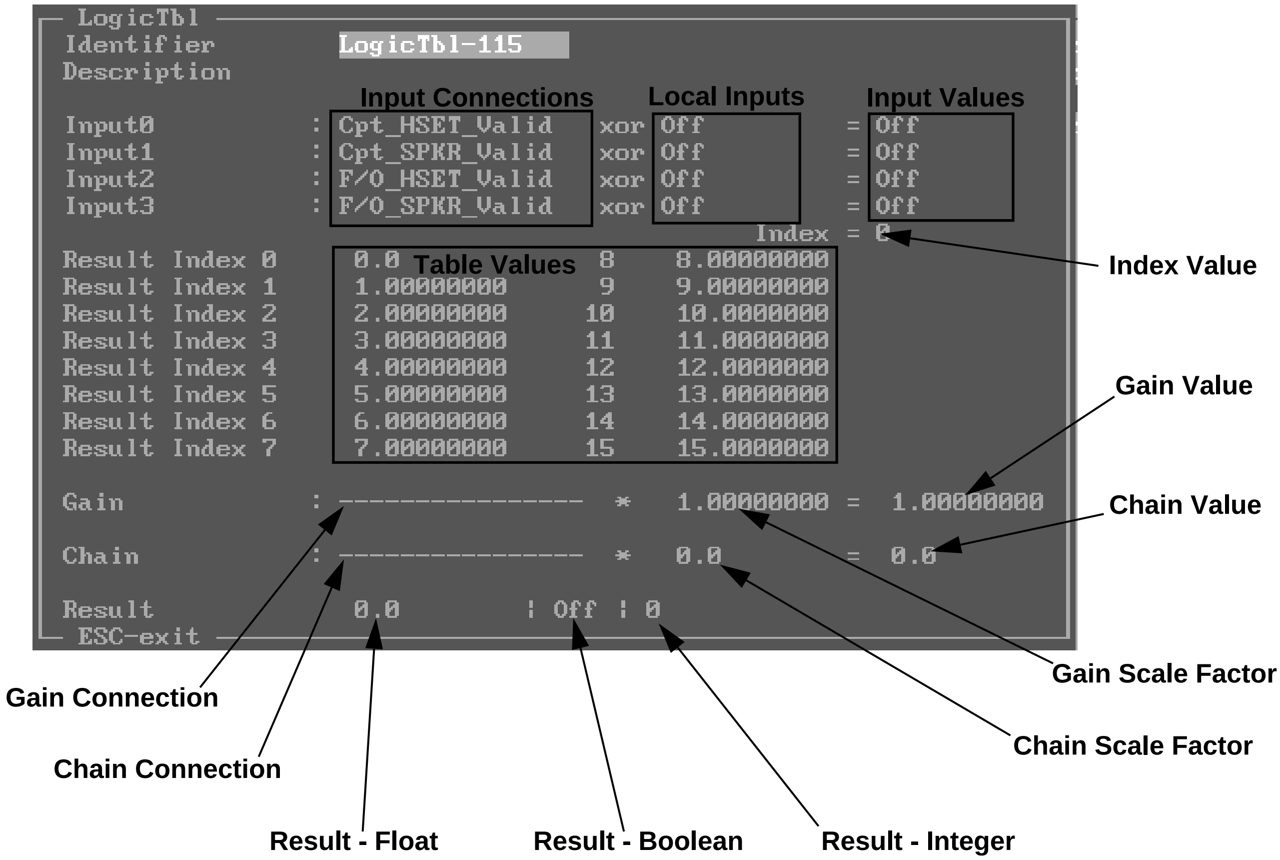
Logic Table



Logic Table

The Logic table object provides a mechanism for combining up to four boolean controls into a single function. The four inputs are combined to form a 4 bit number which is used as an index into a 16 value array. This array contains floating point values, so that a combination of control functions can be achieved in a simple fashion.

Logic Table



Input Connections
Connection to control object which provide the boolean values to be assembled into a 4 bit index.

Local Inputs
Local input value, or if connection present, logic inversion flag.

Input Values
Boolean inputs to table lookup.

Index Value
Current index value used for table lookup.

Table Values
16 values selected by index value.

Gain Connection
Control object connection to provide overall gain control of the result from elsewhere in the model.

Gain Scale Factor
Scaling factor for overall gain control value.

Gain Value
Overall gain of the output result. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Chain Connection
Control object connection to provide a chaining input value from a logic table elsewhere in the model.

Logic Table

Chain Scale Factor

Scaling factor for chain value.

Chain Value

Chain value to be added to logic table value. This provides a simple way of chaining two logic tables together to form an eight bit table lookup.

Result - Float

Output result of logic function, floating point value.

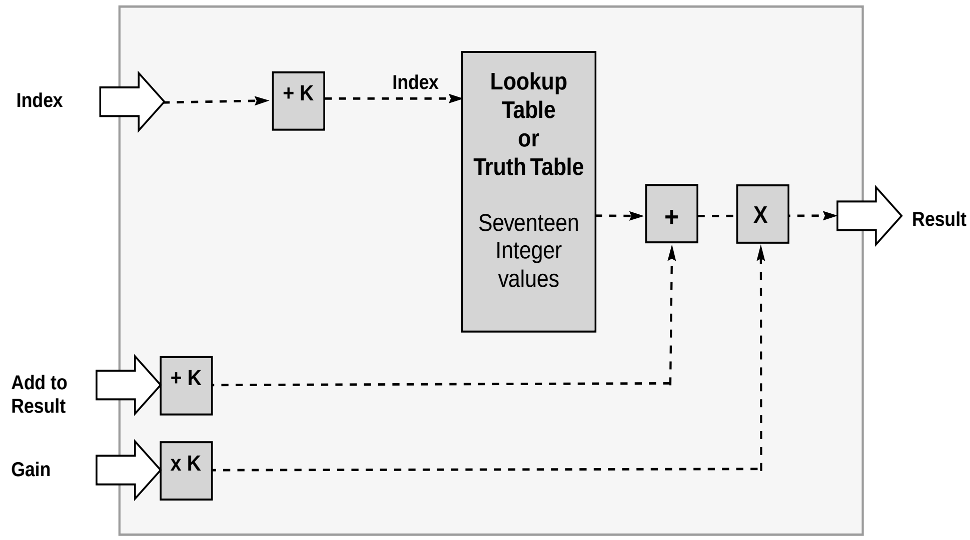
Result - Boolean

Output result of logic function, as a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

Result - Integer

Output result of logic function, integer value, a rounding of the floating point result.

Integer Table



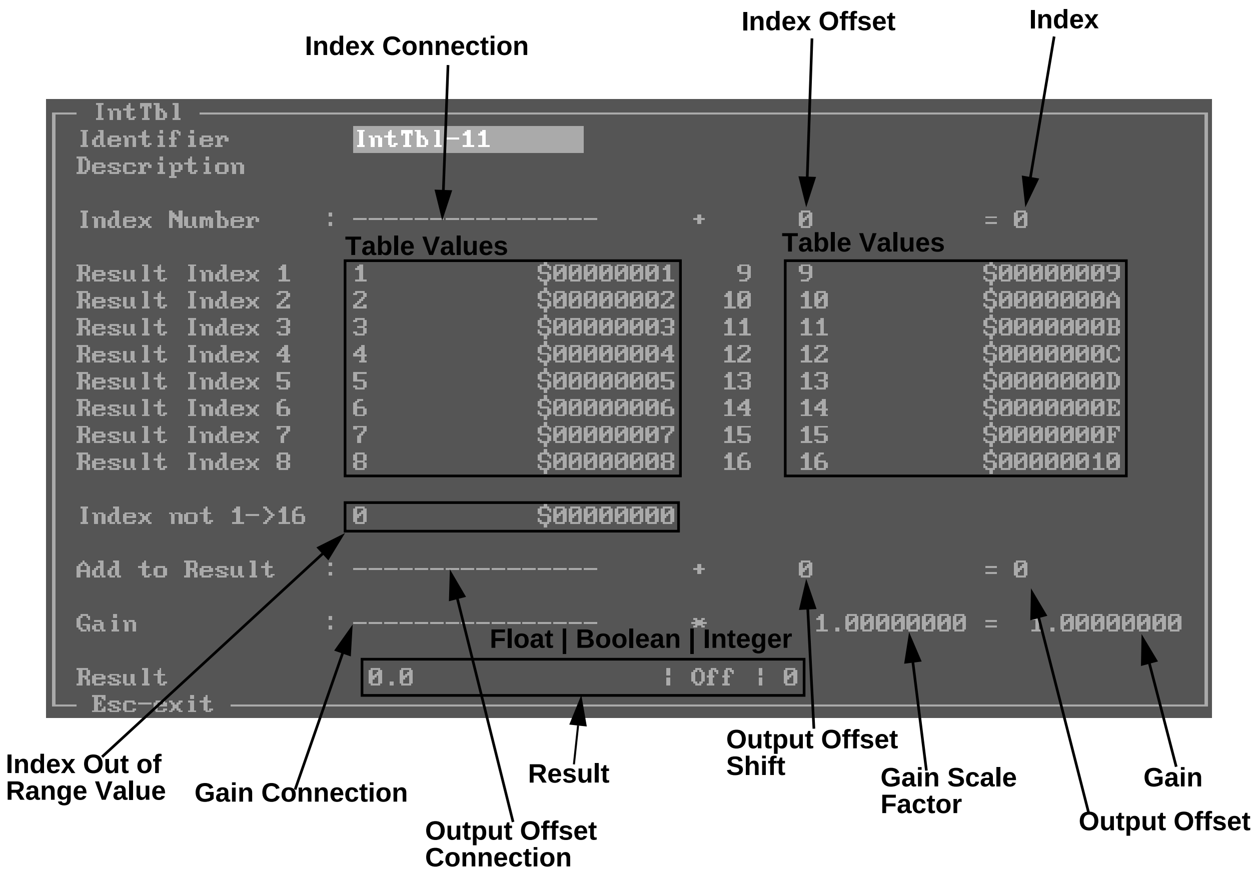
Integer Table

The integer table provides a simple look up function for integer values.

It takes an integer input from an integer connection to which it adds an offset to get an index. If the index is between one and sixteen, the result is determined by the table. If the index is outside this range, the result is given by the index out of range value.

The table values are integers.

Integer Table



Index Connection

Connection to a control object that provides an index value.

Index Offset

Offset value for the index connection.

Index

Current index value used for table lookup.

Table Values

16 values selected by index value.

Index Out of Range Value

If the index is not between 1 and 16, the result is the value given in this field.

Output Offset Connection

Connection to a control object that provides an offset to the output.

Output Offset Shift

Offset value for the output offset connection.

Output Offset

An offset value that is added to the value looked up in the table.

Gain Connection

Control object connection to provide overall gain control of the result from elsewhere in the model.

Gain Scale Factor

Scaling factor for overall gain control value.

Integer Table

Gain Value

Overall gain of the output result. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Result - Float

Output result of logic function, floating point value.

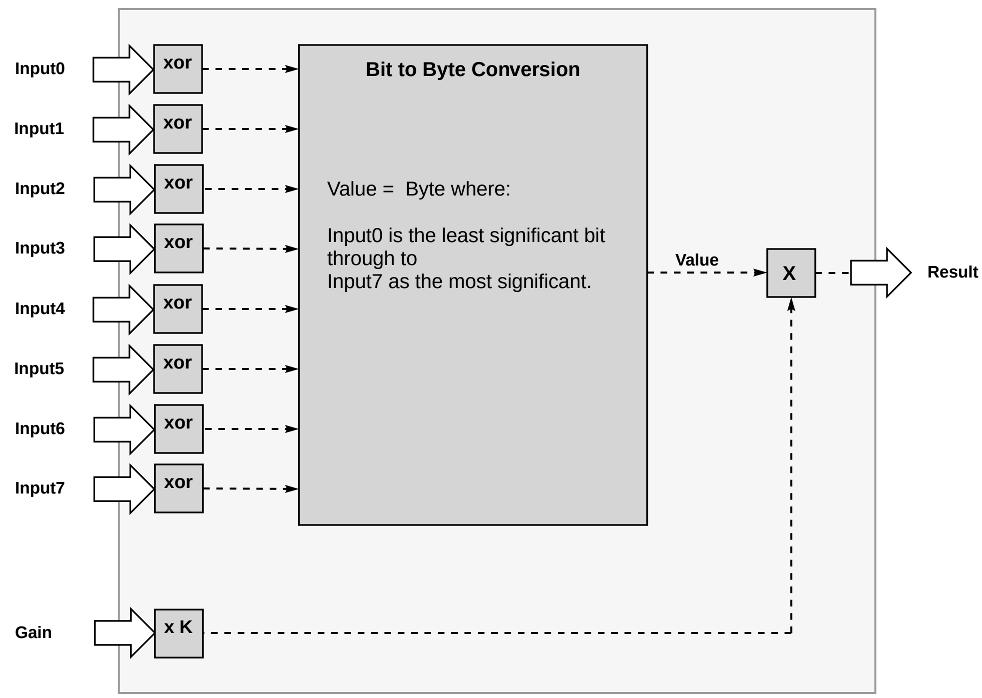
Result - Boolean

Output result of logic function, as a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

Result - Integer

Output result of logic function, integer value, a rounding of the floating point result.

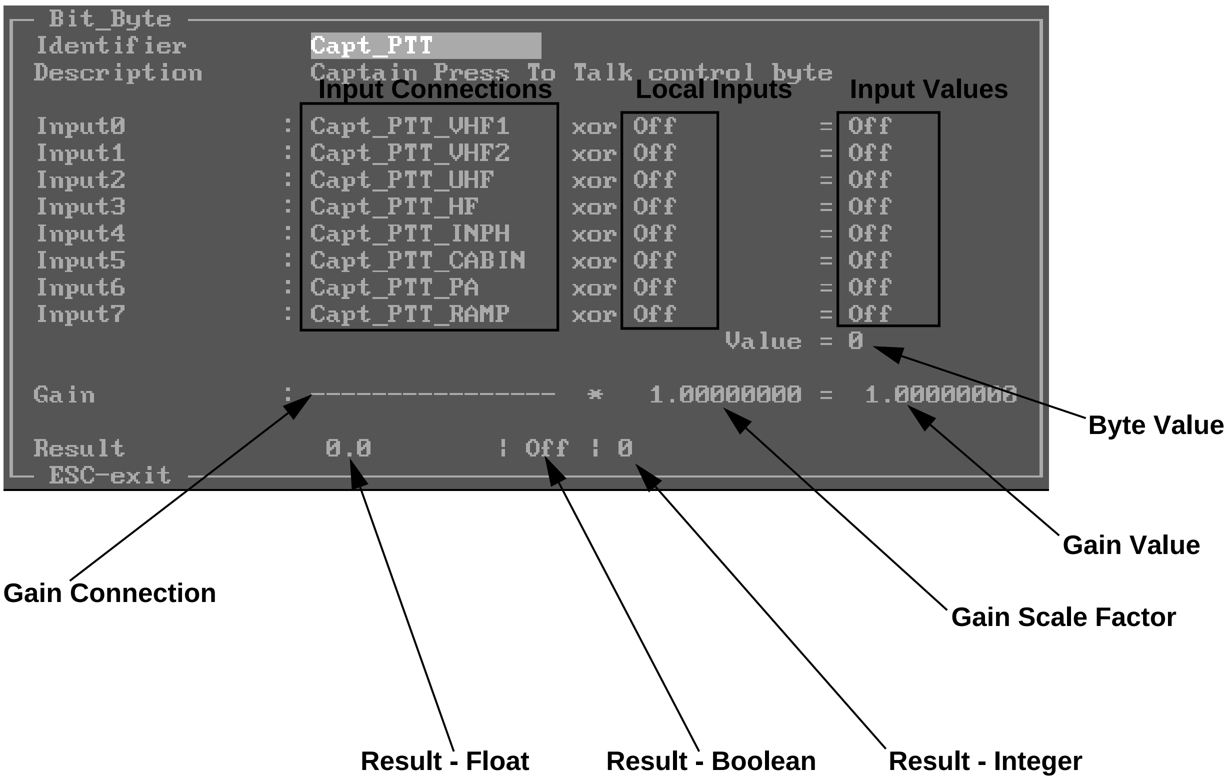
Bit to Byte



Bit to Byte

The Bit to Byte object provides a mechanism for combining up to eight boolean controls into a single byte wide value.

Bit to Byte



Input Connections

Connection to control object which provide the boolean values to be assembled into an 8 bit byte.

Local Inputs

Local input value, or if connection present, logic inversion flag.

Input Values

Boolean inputs to table lookup.

Byte Value

Current byte value from eight bits.

Gain Connection

Control object connection to provide overall gain control of the result from elsewhere in the model.

Gain Scale Factor

Scaling factor for overall gain control value.

Gain Value

Overall gain of the output result. If the gain connection is blank then the gain scale factor is used as the gain value, else the gain is the scale factor times the output result of the control object.

Result - Float

Output result of bit to byte function, floating point value.

Bit to Byte

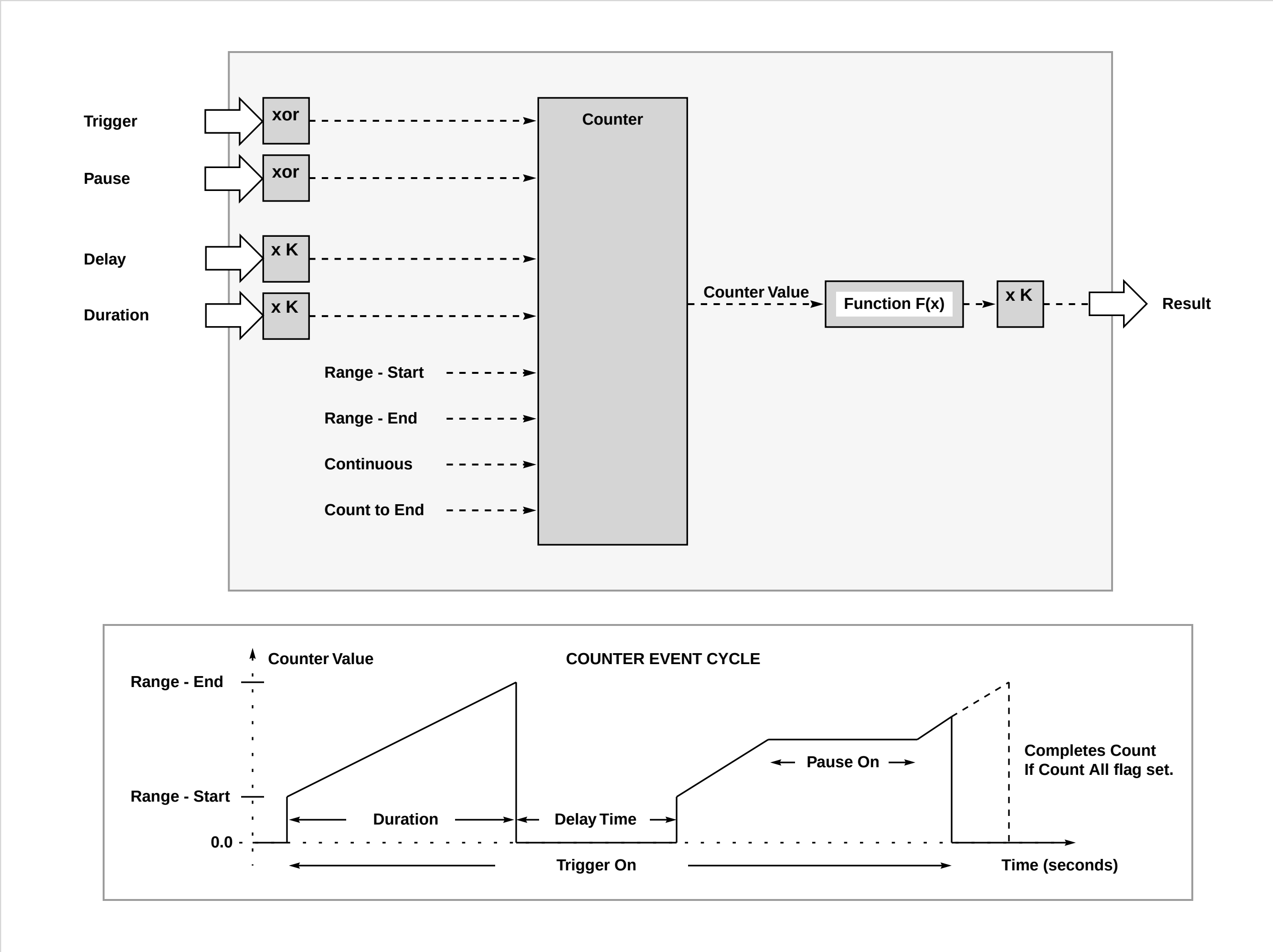
Result - Boolean

Output result of bit to byte function, as a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

Result - Integer

Output result of bit to byte function, integer value, a rounding of the floating point result.

Counter

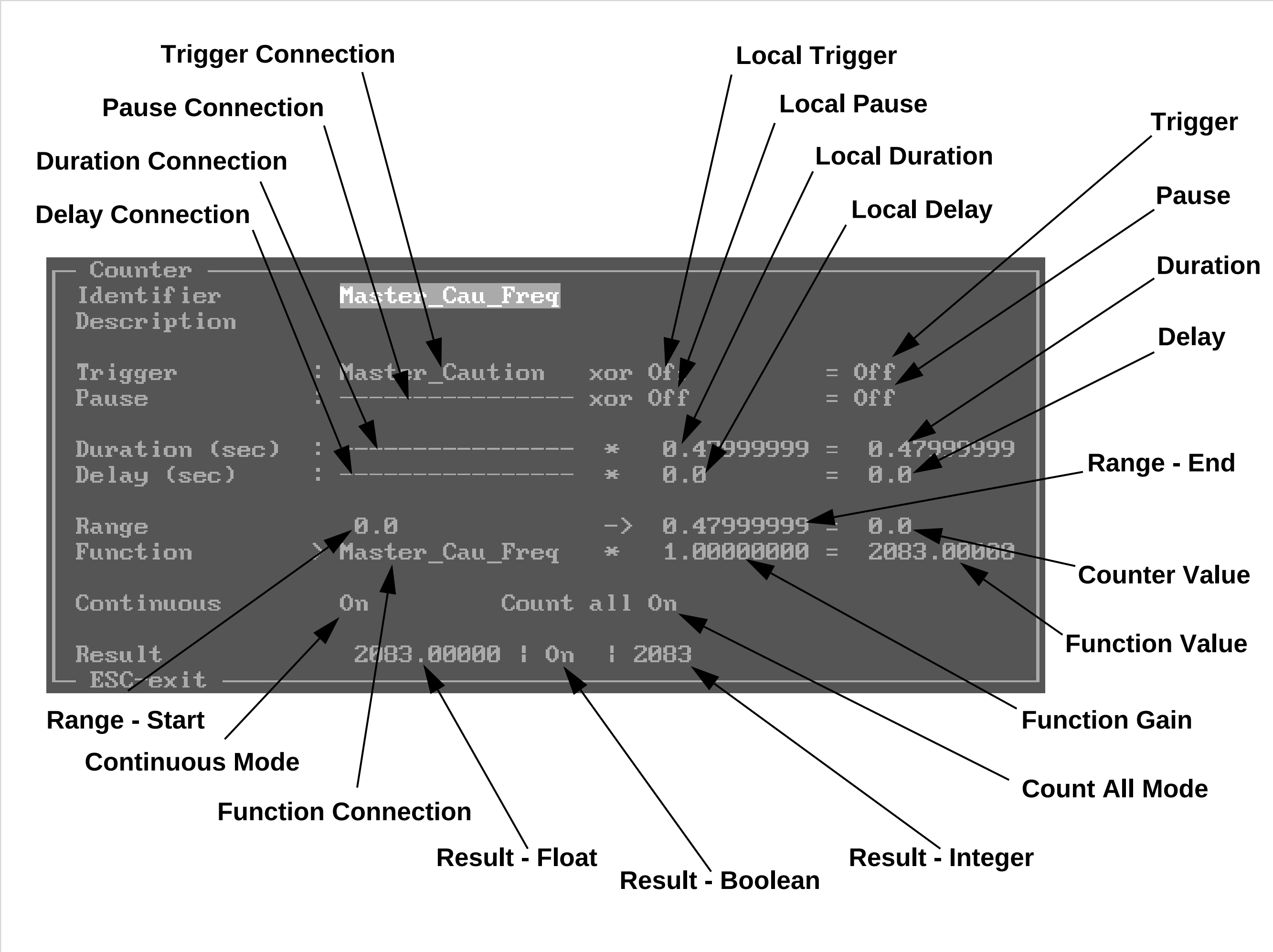


Counter

The Counter provides a general purpose event or continuous ramping function.

In single shot mode it provides an externally triggered function lookup suitable for amplitude or frequency control of signal sources for explosions, touch-down thumps, etc., where a pre-recorded sound file is not available or applicable. When set to continuous it provides a table driven modulation of waveforms, where the modulation rate is slower than the overall model execution rate (i.e. 0 to 10 hertz).

Counter



Trigger Connection

Connection to another control object (usually a boolean input) which causes counter to run when trigger is on.

Local Trigger

Local value for the trigger. If no connection is made this allows the trigger to be left permanently in the on position. If a trigger connection is made then this provides an exclusive or function for inverting the trigger control state

Trigger

Counter trigger state, either the local trigger or the exclusive or function of the local trigger with the boolean value of the trigger connection.

If in continuous mode the counter runs while this trigger is on, else if in one-shot mode the counter runs once for each off to on transition of the trigger.

Pause Connection

Connection to another control object (usually a boolean input) which causes counter to pause when pause is on.

Local Pause

Local value for the pause. If a pause connection is made then this provides an exclusive or function for inverting the pause control state

Counter

Pause
Counter pause state, either the local pause or the exclusive or function of the local pause with the boolean value of the pause connection. The counter will pause at its current value when pause is On.

Duration Connection
Connection to another control object which provides host control over the duration period.

Local Duration Value
Value for duration, or scaling factor for duration connection value.

Duration
Length of time for which counter runs in seconds.

Delay Connection
Connection to another control object which provides host control over the delay period.

Local Delay Value
Value for re-start delay, or scaling factor for delay connection value.

Delay
Delay time (in seconds) when in continuous mode this is the time between the counter ending and restarting again.

Range - Start
Starting value for counter.

Range - End
Ending value for counter.

Counter Value
Counter value used for function look-up.

Function Connection
Function used to convert from counter range to result, used to step through a table or polynomial.

Function Gain
Scaling factor for function output.

Function Value
Counter value after function look-up.

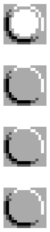
Continuous Mode
If On counter runs continuously while trigger is on, else runs once per off to on transition of trigger.

Count All Mode
If set counter will run through full range after off to on transition of trigger, else counter runs only while trigger is on, resetting to zero when trigger is off.

Result - Float
Output result of counter, floating point value.

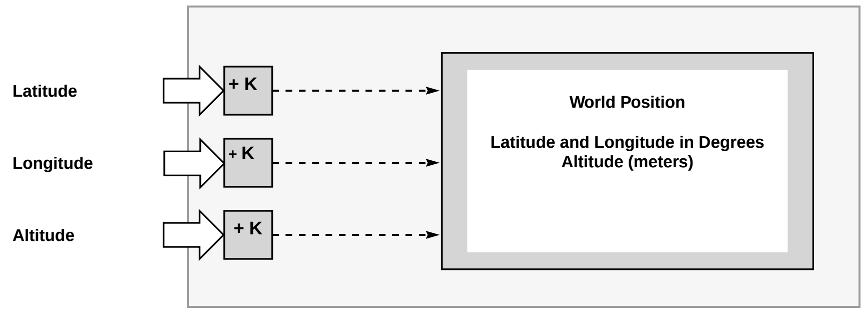
Result - Boolean
Output result of counter, as a boolean. The boolean (On / Off) is a digital comparison of the float value based on a 0.3 and 0.7 low and high threshold value. Below 0.3 is Off, above 0.7 is On, the 0.4 difference provides a hysteresis value.

Result - Integer
Output result of counter, integer value, a rounding of the floating point result.



Navigation Controls (Radios & Nav-Aids Packages)

Geodetic World Position

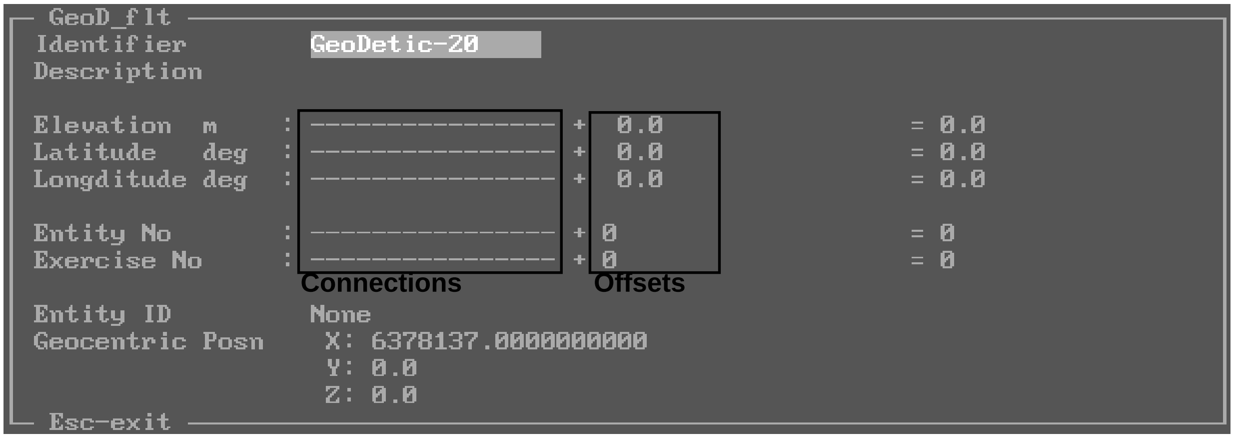


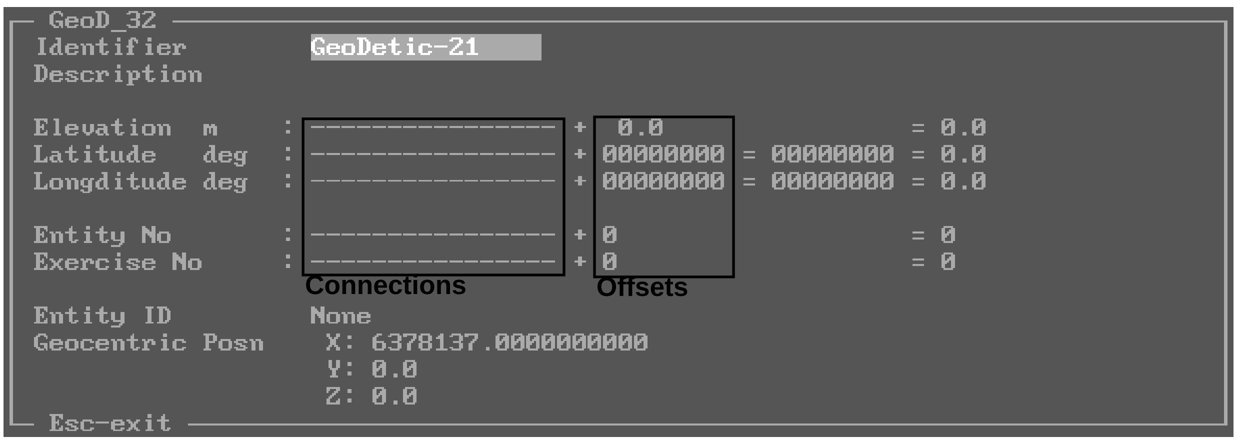
Geodetic World Position

The Geodetic World Position object provides a simple location feature for radio and transmitter positioning. The World positions of the transmitter and receiver are used to compute diminishing power with an inverse square law, as well as occulting by the earth for line of sight transmissions. The model of the earth is a smooth ellipsoid (model WGS84).

The Geodetic World position comes in two flavors - Geodetic float and geodetic 32 bit. The Geodetic float specifies the latitude and longitude as float variables, while the 32 bit express them as 32 bit integers. The 32 bit integer world position provides smoother connection when a world position goes from 360 to 0 degrees (i.e. crosses the international date line).

Geodetic World Position





Latitude Connection

Connection to another control object which provides the Latitude in degrees.

Longitude Connection

Connection to another control object which provides the Longitude in degrees.

Altitude Connection

Connection to another control object which provides the altitude in meters.

Latitude, Longitude, Altitude Offsets

Offsets for position variables. If the connection fields are empty, the offsets become the position values.

Latitude Value

Latitude in degrees.

Longitude Value

Longitude in degrees.

Altitude Value

Altitude in meters.

Geocentric Posn (X,Y,Z)

The X,Y,Z values, in meters from the center of the Earth, corresponding to the given Latitude, Longitude, and Altitude.

Entity Number

Specifies the DIS entity ID of the simulated entity which the attached radio object is associated with.

Geodetic World Position

Exercise Number

Specifies the exercise number which the attached radio object is associated with. Radios can only communicate with other radios with the same exercise ID. The exercise ID is required for all world positions, whether or not DIS is being used.

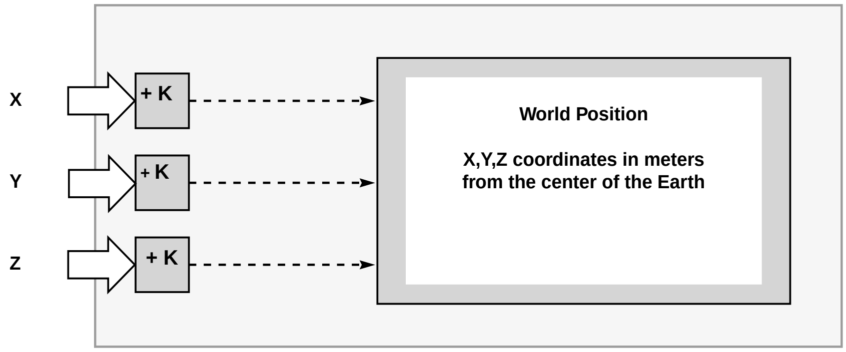
NOTE: An exercise number of zero is INVALID.

Entity ID

Specifies whether the radio or transmitter connected to the world position is communicating only within the model, to other models in the same DACS (Local Net), to other DACS via VoiceNet (VoiceNet), or over DIS.

If one of these nets is specified, the entity ID will consist of three numbers - the Host and site ID’s, given in the Ethernet Control Network Menu, and the entity ID given in the World Position object. In addition, the connected object will also be given an exercise ID.

Geocentric World Position



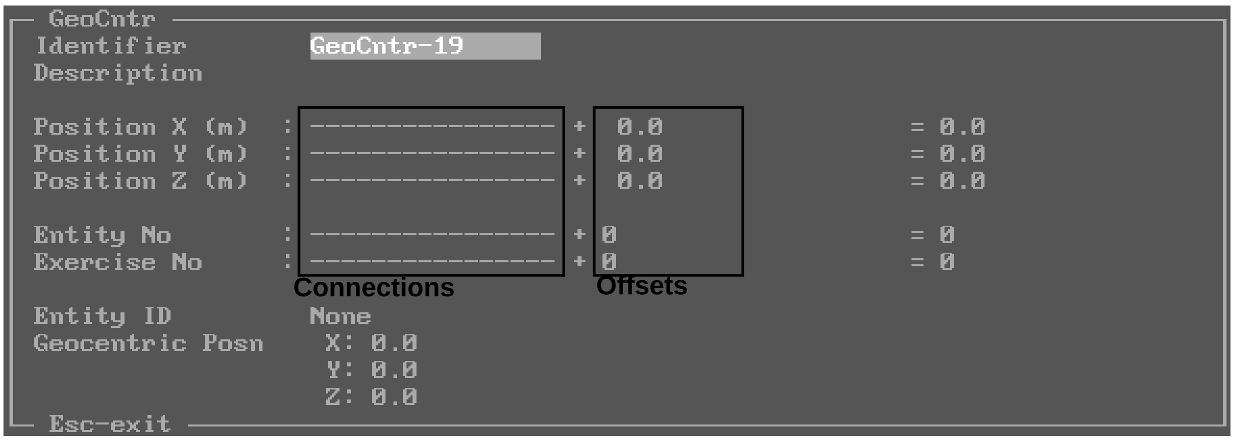
Geodetic World Position

The Geocentric World Position object provides a simple location feature for radio and transmitter positioning. The World positions of the transmitter and receiver are used to compute diminishing power with an inverse square law, as well as occulting by the earth for line of sight transmissions. The model of the earth is a smooth ellipsoid.

The Geocentric World position is identical to the Geodetic World Position, except that the input values for the position are given in terms of X, Y and Z meters from the center of the Earth.

If the world position is 0,0,0 (i.e. the center of the Earth), then the ranging effects of any attached radio will be turned off, and the radio will clearly receive all transmissions on its frequency.

Geocentric World Position



X,Y, and Z Connections

Connections to other control objects which provides the X,Y, and Z coordinates.

X,Y, and Z Offsets

Offsets for position variables. If the connection fields are empty, the offsets become the position values.

X,Y and Z Values

The position, given in meters, relative to the center of the earth. If a radio is placed at (0,0,0) (i.e. the center of the earth), the ranging will be turned off and the radio will receive any signal broadcast on its frequency, regardless of its power or position.

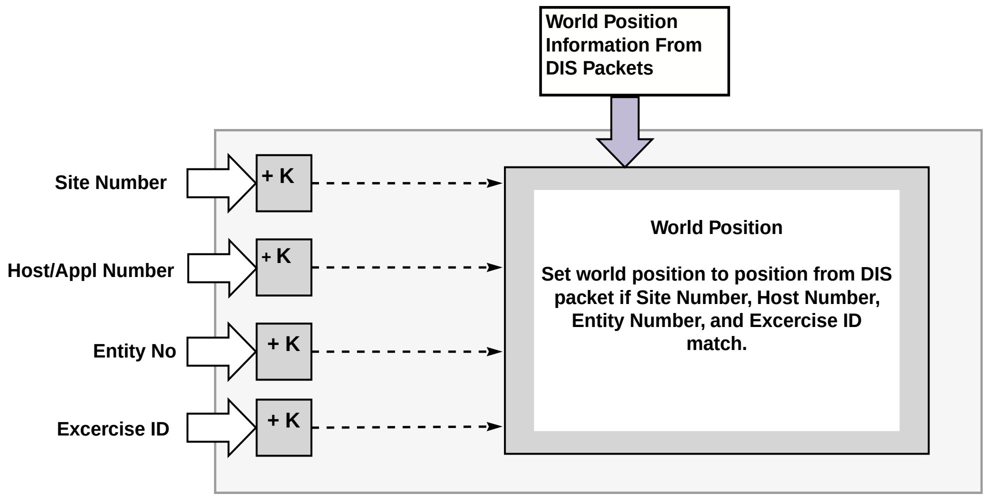
Geocentric Posn (X,Y,Z)

The X,Y,Z values, in meters from the center of the Earth. These are the same as the X,Y, and Z values.

Other Fields

The remaining fields are identical to the ones in the geodetic world position objects.

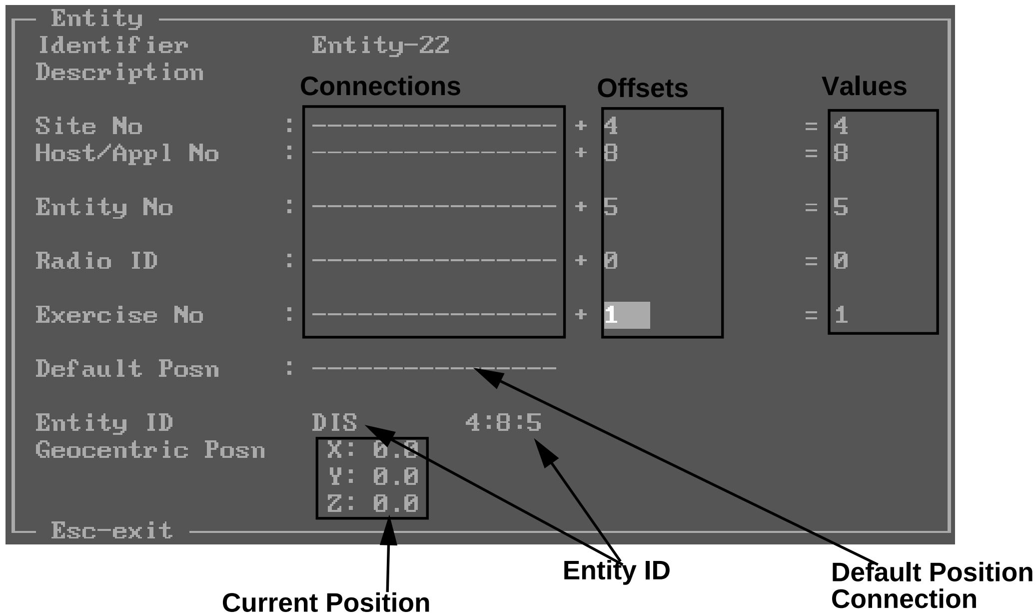
Entity



Entity

The Entity object is used in DIS to attach a radio transmitter or receiver to an external DIS entity. The world position of the radio is taken from the world position of the DIS entity as the new position information comes in over DIS. This allows the radio simulation of an object to be run on the DACS while the position and other information is being simulated elsewhere on DIS.

Entity



Connections

Connections to other control object that provides the site number, Host/ Application number, Entity Number, Radio ID, and Exercise Number.

Offsets

Provide offsets for the Site number, Host/Application number, Entity number, Radio ID and Exercise Number. If any of the connection fields are empty, the offset provides the value.

Site Number

This provides the Site number of the DIS entity that the entity object is attaching to.

Host /Application Number

This provides the Host/Application number of the DIS entity that the entity object is attaching to.

Entity Number

This provides the Entity number of the DIS entity that the entity object is attaching to. The Site, Host, and Entity Number uniquely specify one DIS entity.

Radio ID

This field is included for compatibility with older models. It is not currently used. The Radio ID for any radio object attached to an entity is given in the radio object itself.

Entity

Exercise Number

This provides the Exercise number of the DIS entity to which the entity object is attached.

NOTE: An exercise number of zero is INVALID.

Default Posn

This provides a connection to a world position object, which specifies the position of the entity before any DIS packets have arrived given its location.

The entity ID and exercise numbers specified in the attached world position object are ignored.

If a radio is attached to an entity object with no default position, and no entity PDUs are being received over the network, then the radio will not transmit or receive any signals.

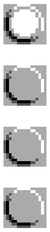
Entity ID

The word (none, Local, Voicenet, DIS) determines whether the entity ID is a Local, Voicenet, or DIS. The number specifies the entity ID.

This object will normally only be used for DIS. The specifying of ID numbers for Local Net and Voice net radios should be done through the world position object.

Current Position

Current position of the specified entity. This value is from the last DIS packet received with the entity's ID and position information, or from the Default Position if no packet has yet been received, or if the last packet was times out.



Functions

Table Function

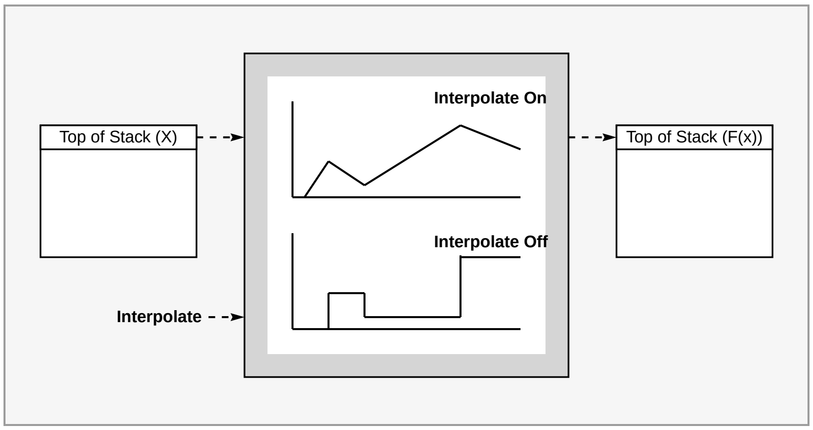


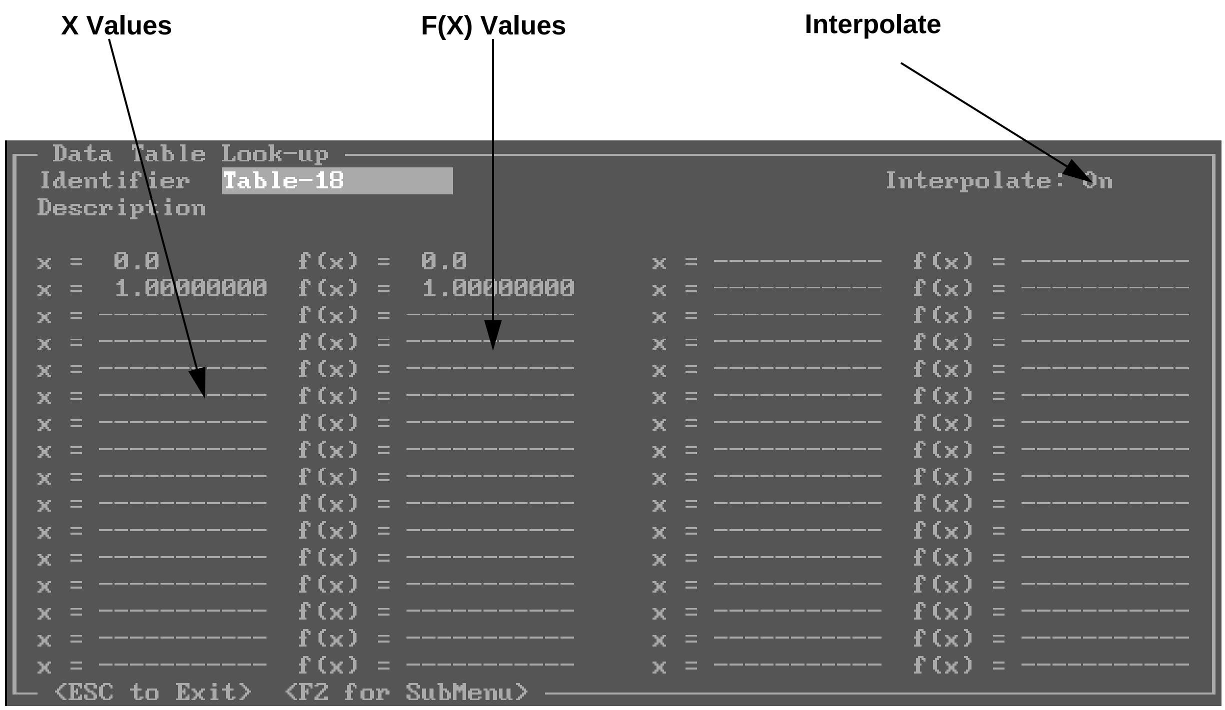
Table Function

The table function provides a single function look-up in a table with up to 32 breakpoints (floating point). The breakpoint table is editable on-line, and is stored as part of the model. A single table is reusable in several parts of the same model.

An interpolate control allows the table to provide single variable straight line interpolation of data, or non-interpolated, stepwise changes in data.

The top of stack value is replaced by its corresponding value from the table.

Table Function



Interpolate

Interpolate On enables straight line interpolation of data.

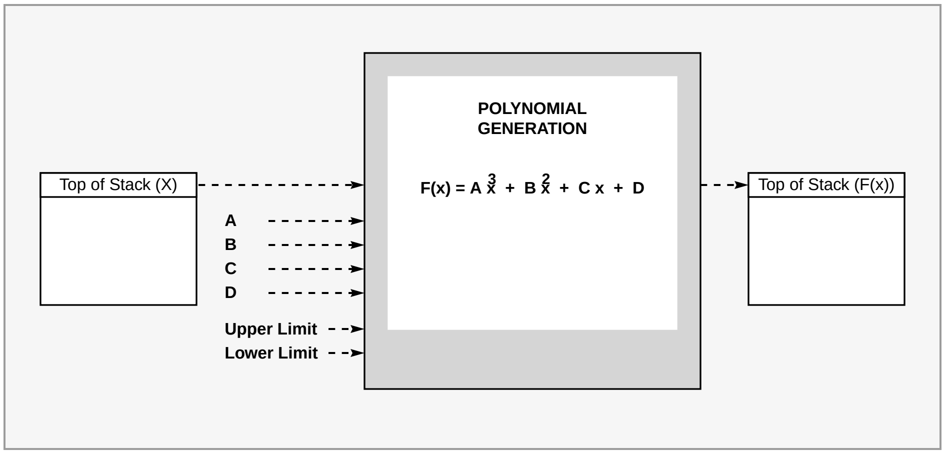
X Values

X axis values for table lookup.

F(X) Values

Function values for each x value breakpoint in table.

Polynomial



Polynomial

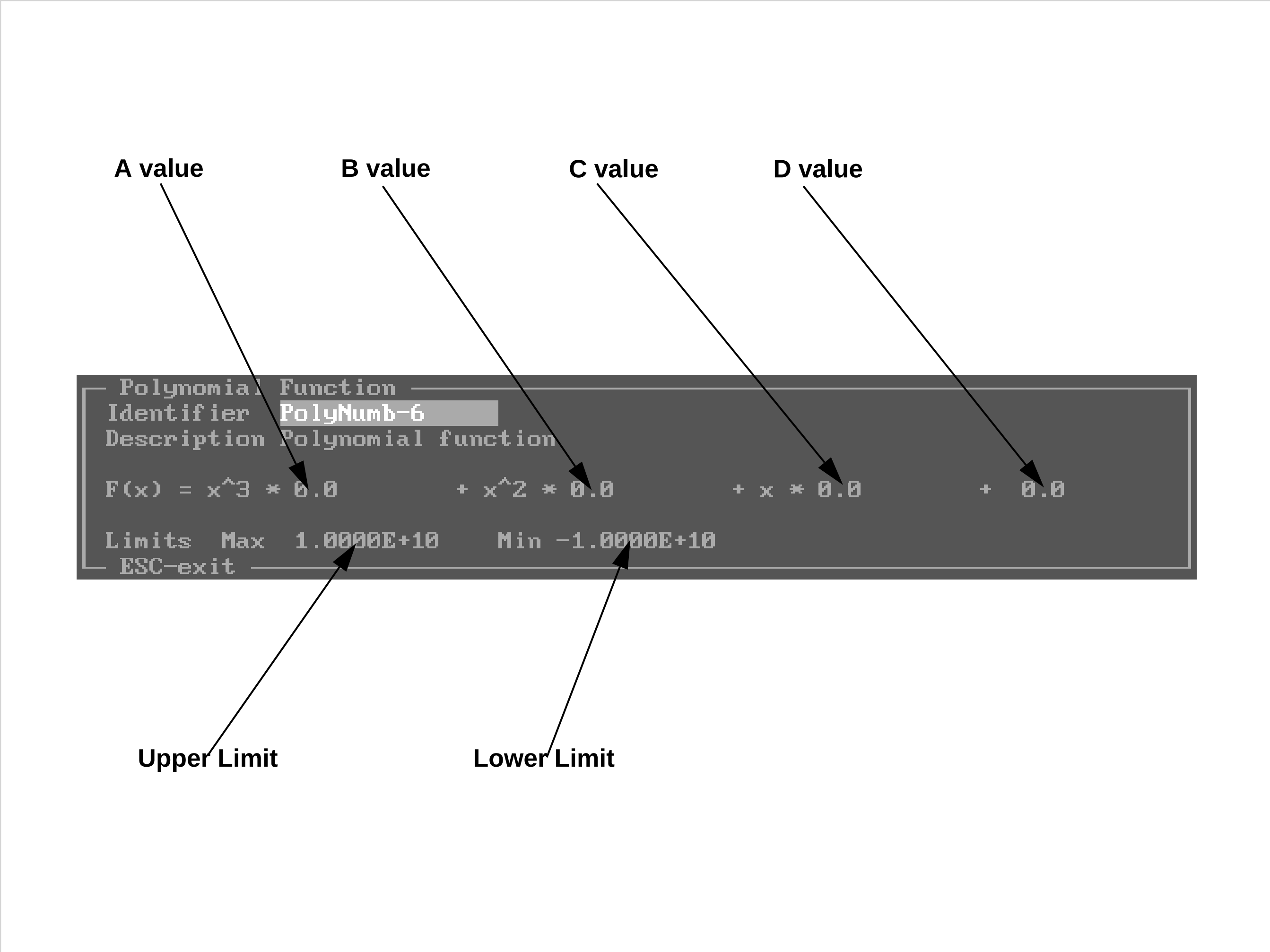
The polynomial function provides a simple third order expansion from the input value (X) with user definable factors.

$F(X)= A.X^3 + B.X^2 + C.X + D$

The result is held between the upper and lower limits.

The top stack value is replaced by its polynomial value.

Polynomial



A value

X cubed scale factor.

B value

X squared scale factor.

C value

X scale factor.

D value

Constant offset value.

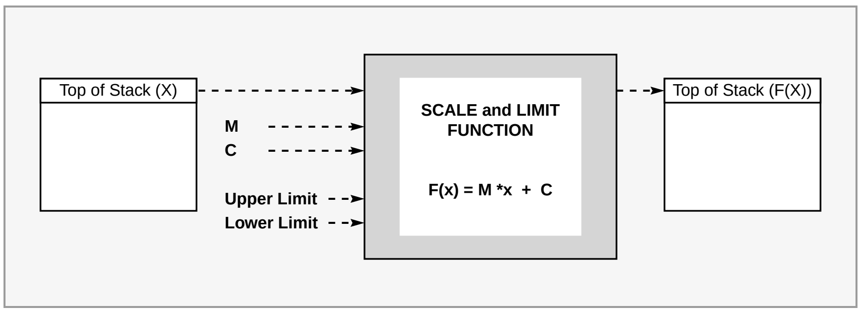
Upper Limit

Maximum value for F(X).

Lower Limit

Minimum value for F(X).

Scale / Limit



Scale and Limit

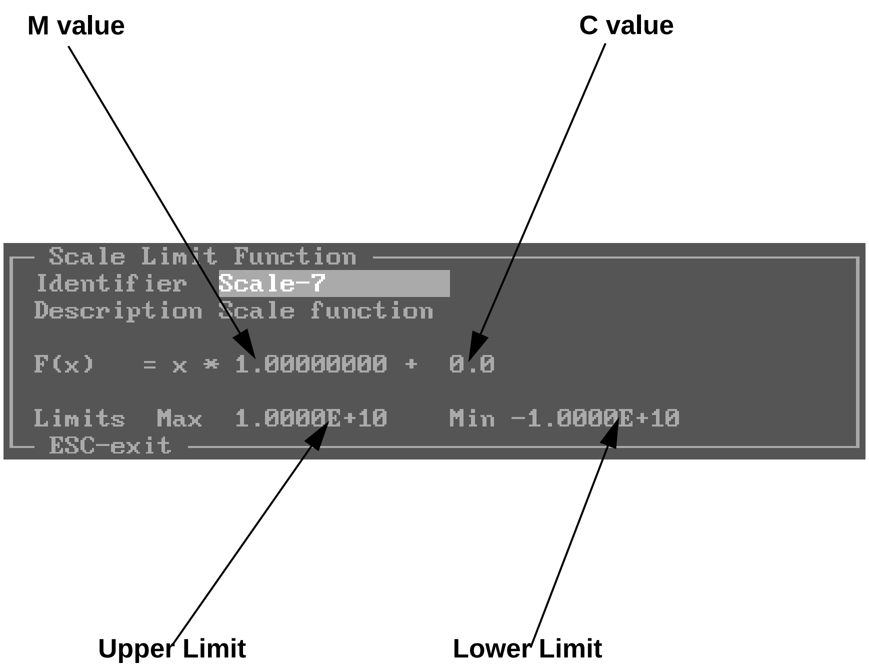
The scale and limit function provides a simple first order expansion from the input value (X) with user definable factors, limited between an upper and lower limit.

$F(x) = M*X + C$

The result is held between the upper and lower limits.

The top stack value is replaced by its scaled value.

Scale / Limit



M value

X scale factor.

C value

Constant offset value.

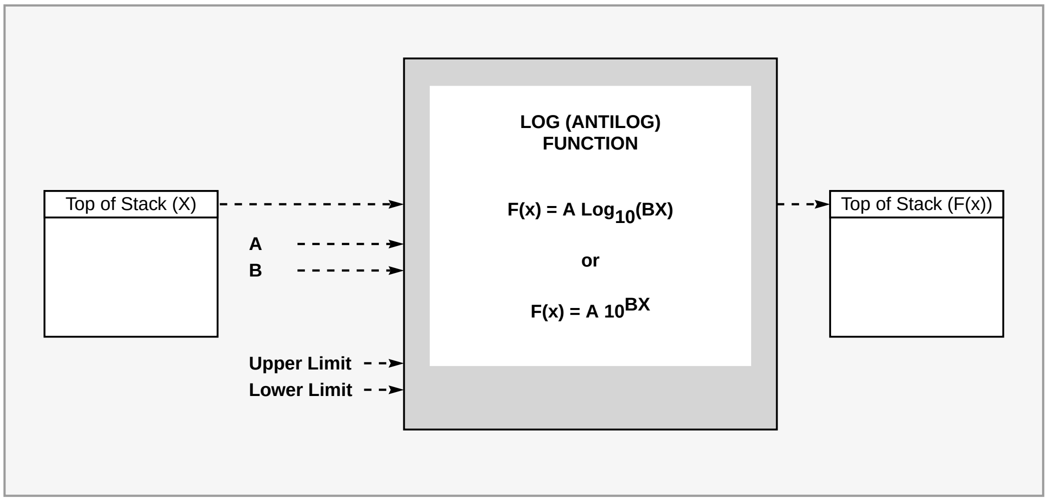
Upper Limit

Maximum value for F(X).

Lower Limit

Minimum value for F(X).

Log / Antilog

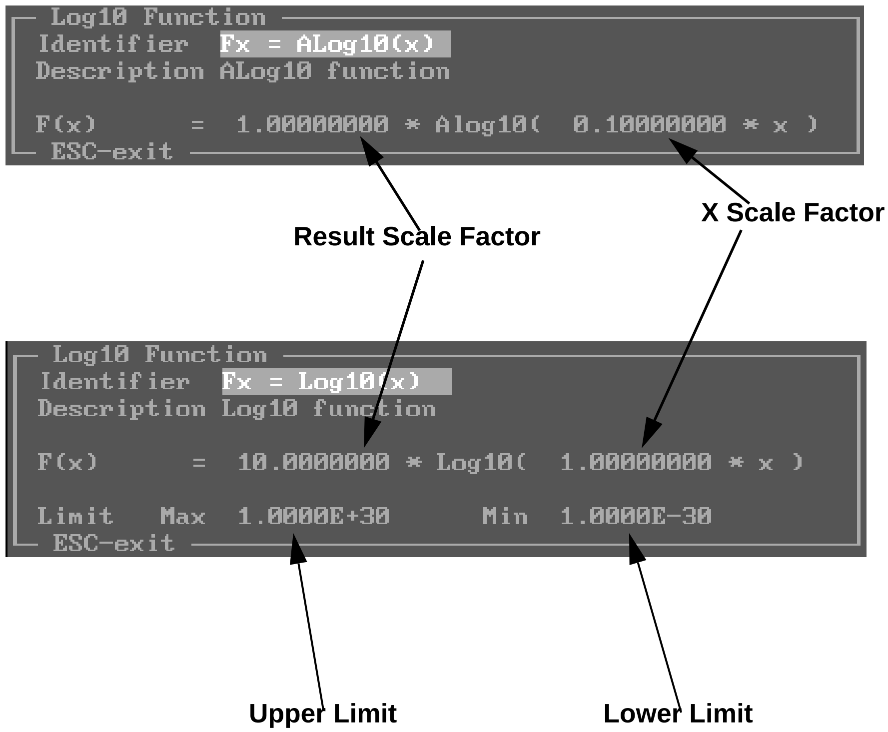


Log / Antilog

The log and antilog functions provide the log and antilog (base 10) of the input value on top of the stack.

The top stack value is replaced by the function result.

Log / Antilog



X Scale Factor

Scale Factor for the input (X) value.

Result Scale Factor

Scale Factor for the result.

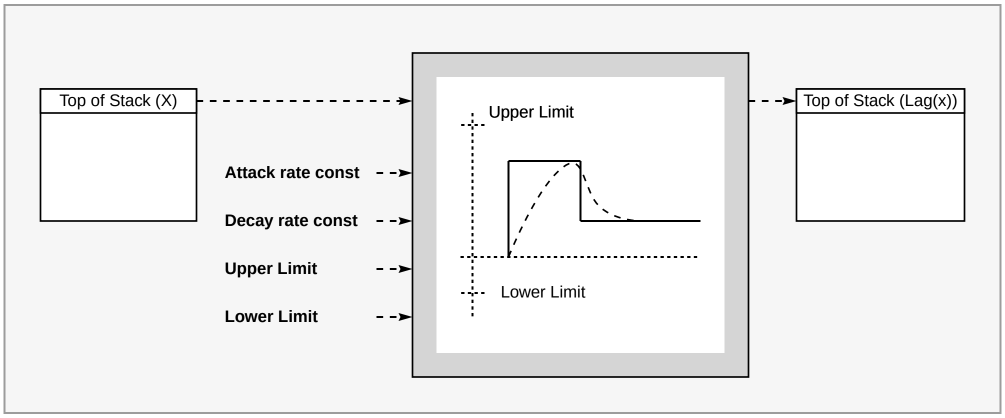
Upper Limit

Maximum value for F(X).

Lower Limit

Minimum value for F(X).

Lag Filter Function



Lag Filter Function

The lag filter provides a simple slew-rate limiting filter which is useful for fade-in and fade-out effects.

The filter function is defined as:

$Y_N = Y_{N-1} + K(X_N - Y_{N-1})$

Where

X_N = new input value

Y_N = new output value

Y_{N-1} = last frame's output value

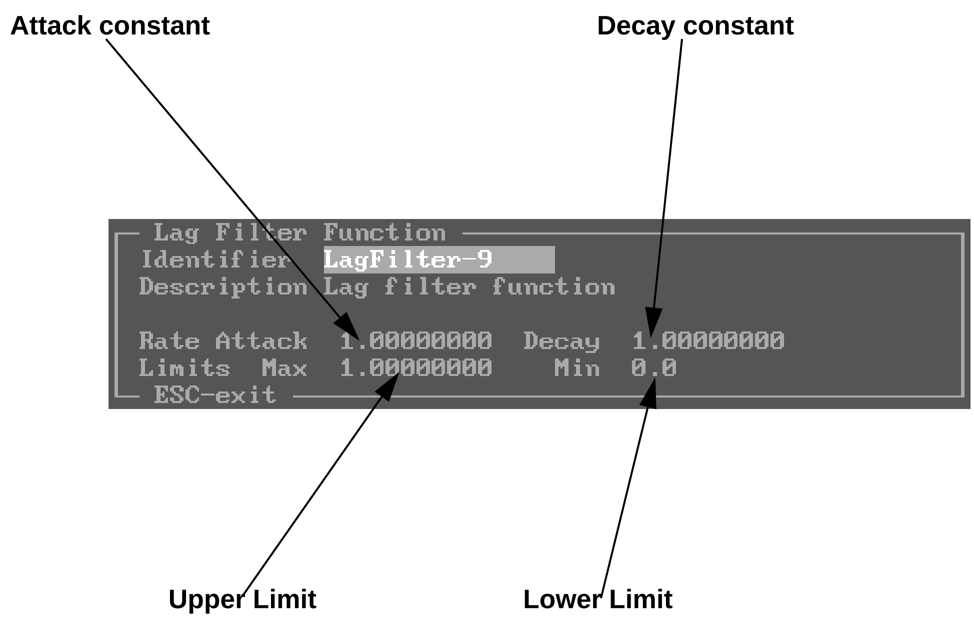
K = Attack const If $X_N > Y_{N-1}$

K = Decay const If $X_N < Y_{N-1}$

The result is held between the upper and lower limits.

The top stack value is replaced by its filtered value.

Lag Filter Function



Attack constant

Filter attack rate constant.

Decay constant

Filter decay rate constant.

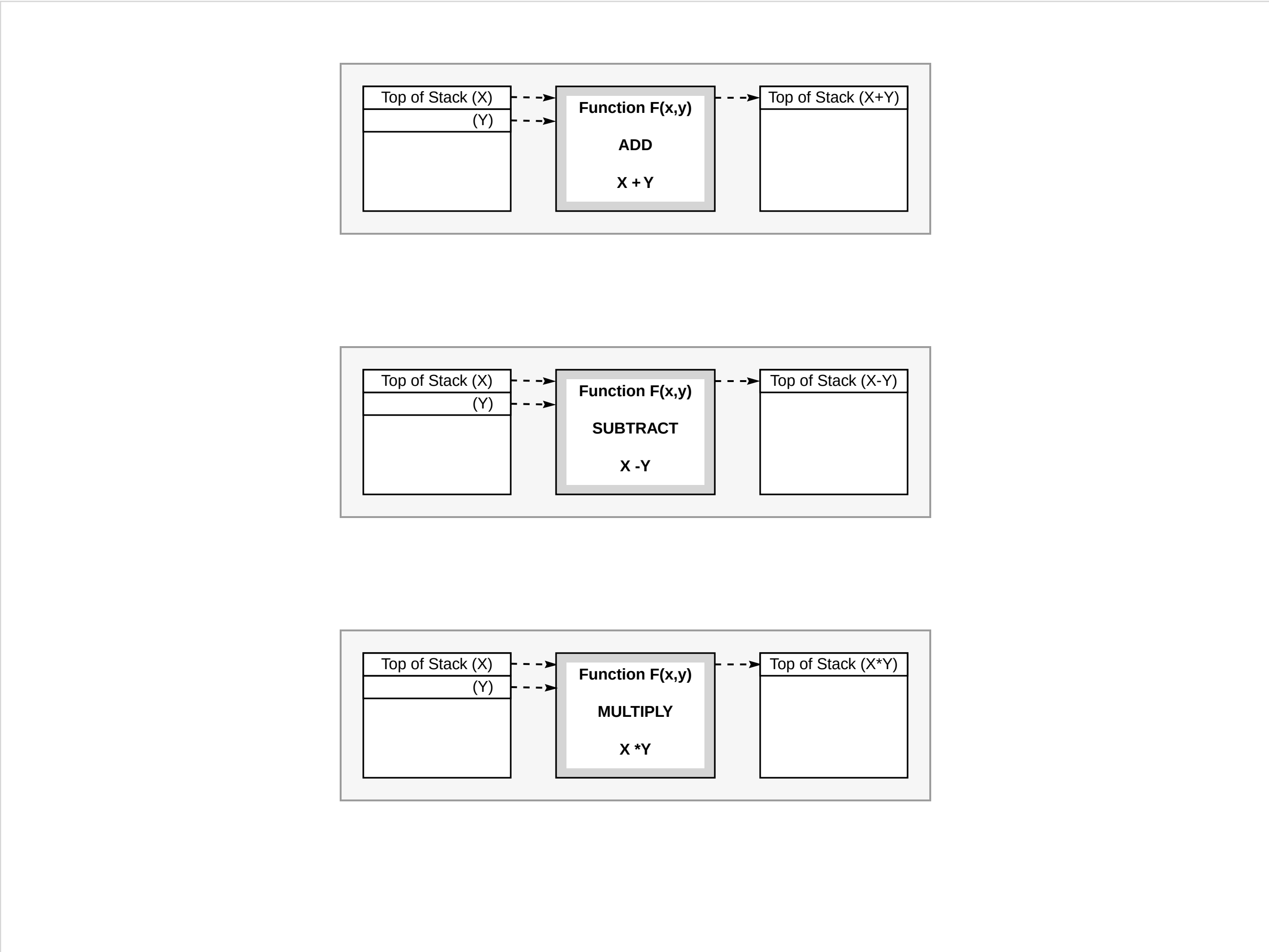
Upper Limit

Maximum limit value for filter accumulation to prevent saturation of integrators.

Lower Limit

Minimum limit value for filter accumulation to prevent saturation of integrators.

Adder/Subtractor/Multiplier



Adder/ Subtracter/ Multiplier

The basic math functions provide simple addition, subtraction, or multiplication of two input variables.

Adder/Subtractor/Multiplier

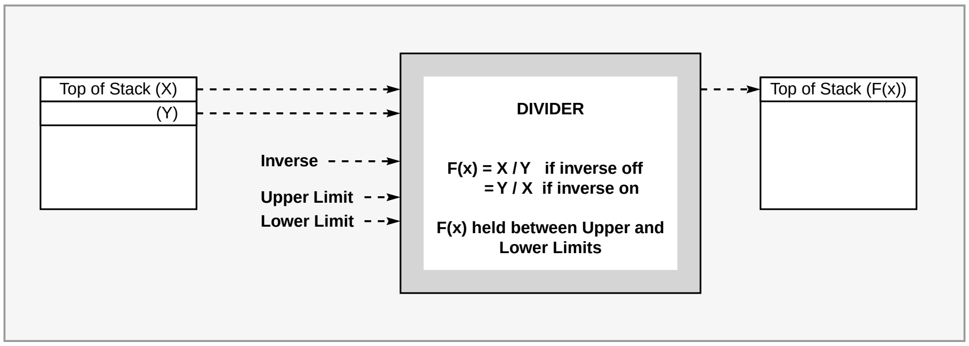
```
┌─ Add Function ──────────────────────────────────────────┐
│ Identifier  F(x,y) = x + y                              │
│ Description Add function                                │
└─ ESC-exit ──────────────────────────────────────────────┘
```

```
┌─ Subtract Function ─────────────────────────────────────┐
│ Identifier  F(x,y) = x - y                              │
│ Description Subtract function                           │
└─ ESC-exit ──────────────────────────────────────────────┘
```

```
┌─ Multiply Function ─────────────────────────────────────┐
│ Identifier  F(x,y) = x * y                              │
│ Description Multiply function                           │
└─ ESC-exit ──────────────────────────────────────────────┘
```

These functions have no configurable parameters.

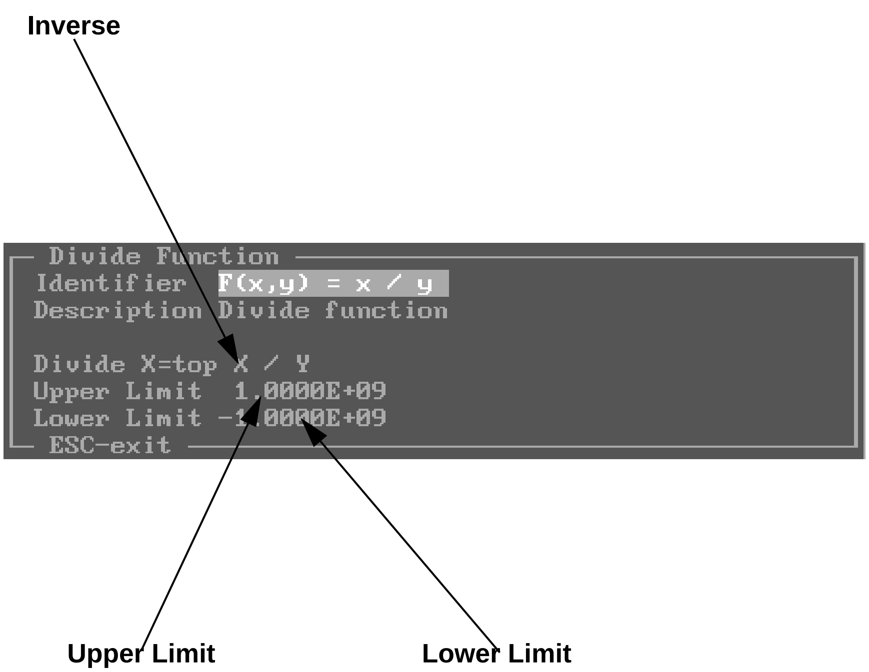
Divider



Divider

This basic math function provides a simple division of two input variables.

Divider



Inverse

Inverts normal division of X by Y, i.e. F(X,Y) = X/Y else F(X,Y) = Y/X, where X is top of stack, and Y is second on stack.

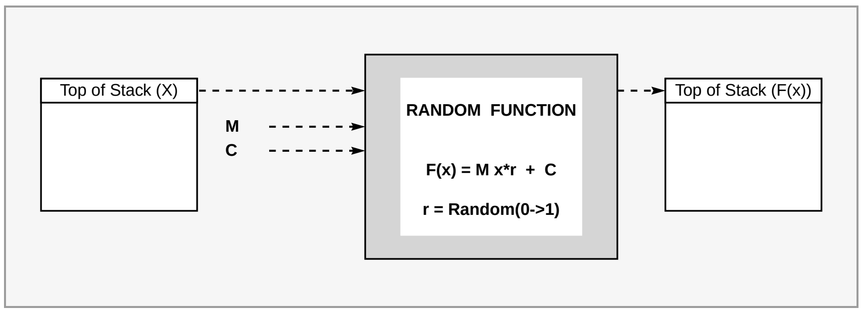
Upper Limit

Maximum value for F(X).

Lower Limit

Minimum value for F(X).

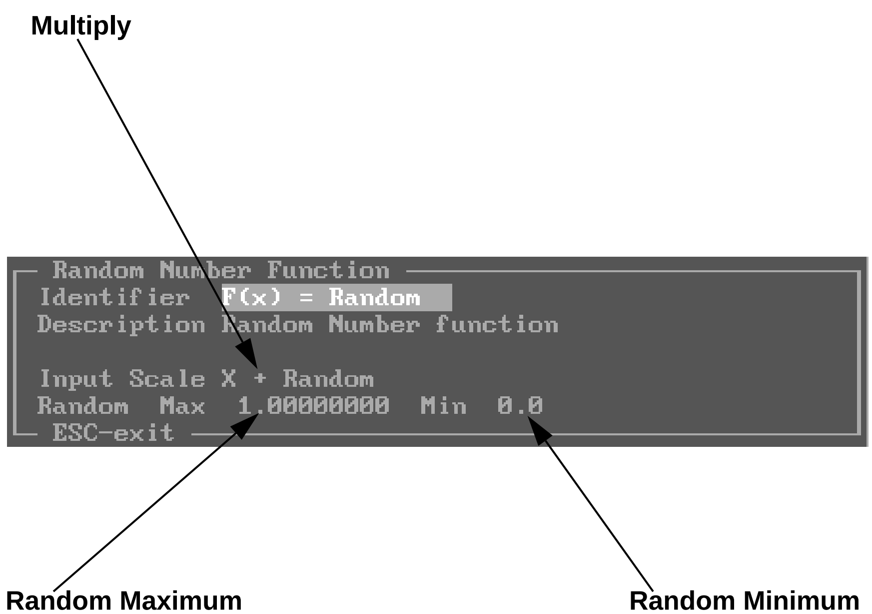
Random Number



Random Number

Provides a simple random number addition or multiply factor for the top stack value. The random number is limited between upper and lower limits, and is added to the input value. A multiply flag allows the random number to be used as a scale factor for the input value.

Random Number



Multiply

Controls whether the random number is added to the input value or multiplied by the input value. If set to "*" then F(X) = Random Number * X, otherwise F(X) = Random Number + X.

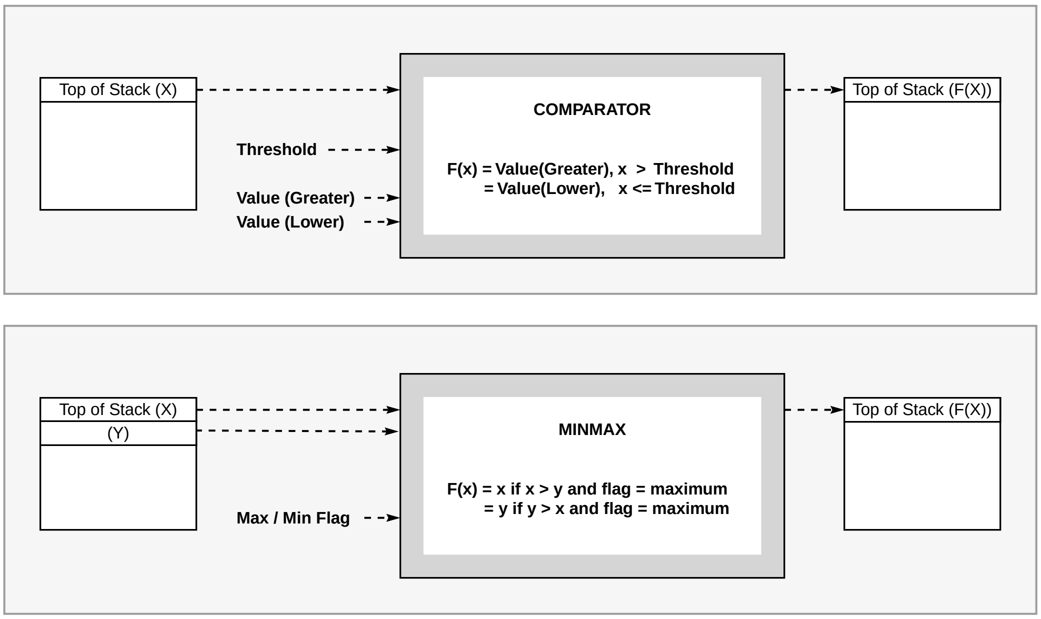
Random Maximum

Maximum random number value.

Random Minimum

Minimum random number value.

Comparator / MaxMin

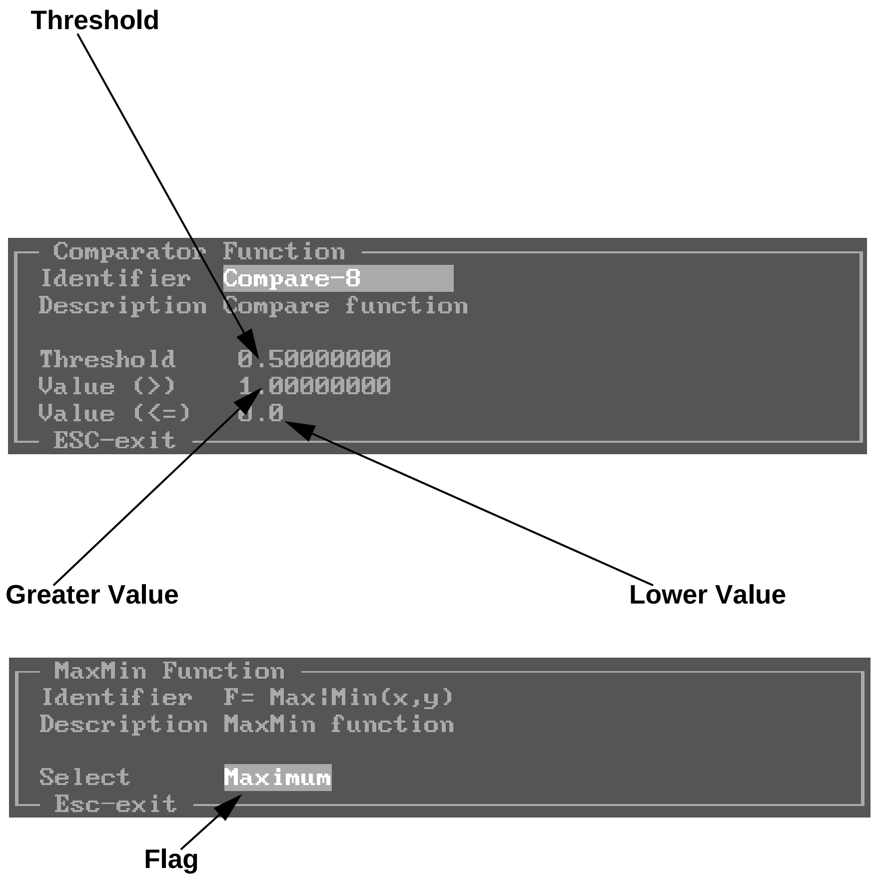


Comparator and MaxMin

The comparator function provides a simple comparison of the top stack input with a set threshold. The top stack element is then replaced by a value dependent on the threshold comparison.

The maxmin function compares the top two values of the stack and returns either the maximum or the minimum, depending on the flag value.

Comparator / MaxMin



Threshold

Threshold for input value comparison.

Greater Value

Output value when input greater than threshold.

Lower value

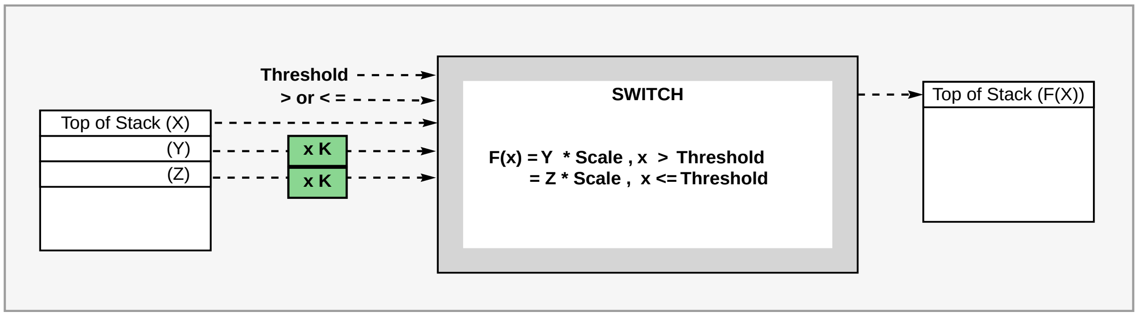
Output value when input less than or equal to threshold.

Flag

If the flag is set to Maximum, the MaxMin function returns the greater of X and Y.

If the flag is set to Minimum, the MaxMin function returns the lesser of X and Y.

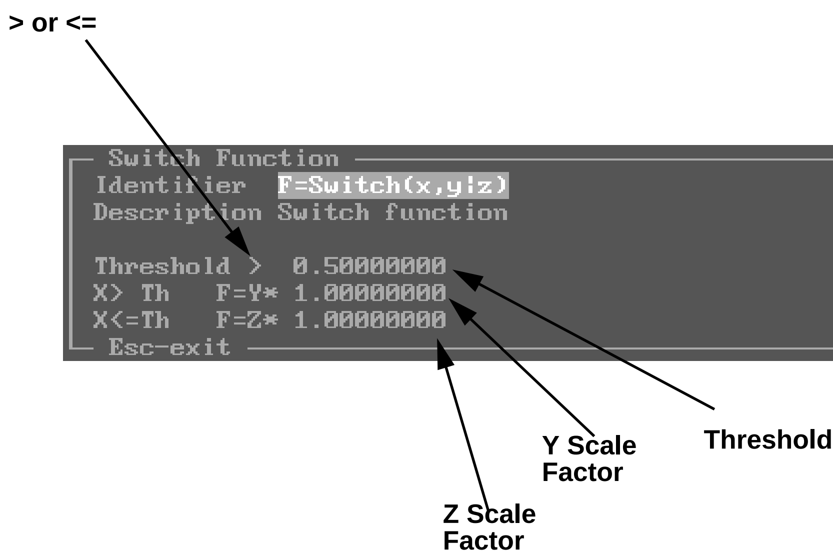
Switch



Switch

The switch is similar to the comparator, except that it takes its outputs from the Y and Z values (the second and third values on the stack) instead from fixed fields.

Switch



Threshold

Threshold for input value comparison.

Y Scale Factor

Scale factor for the Y value.

Z Scale Factor

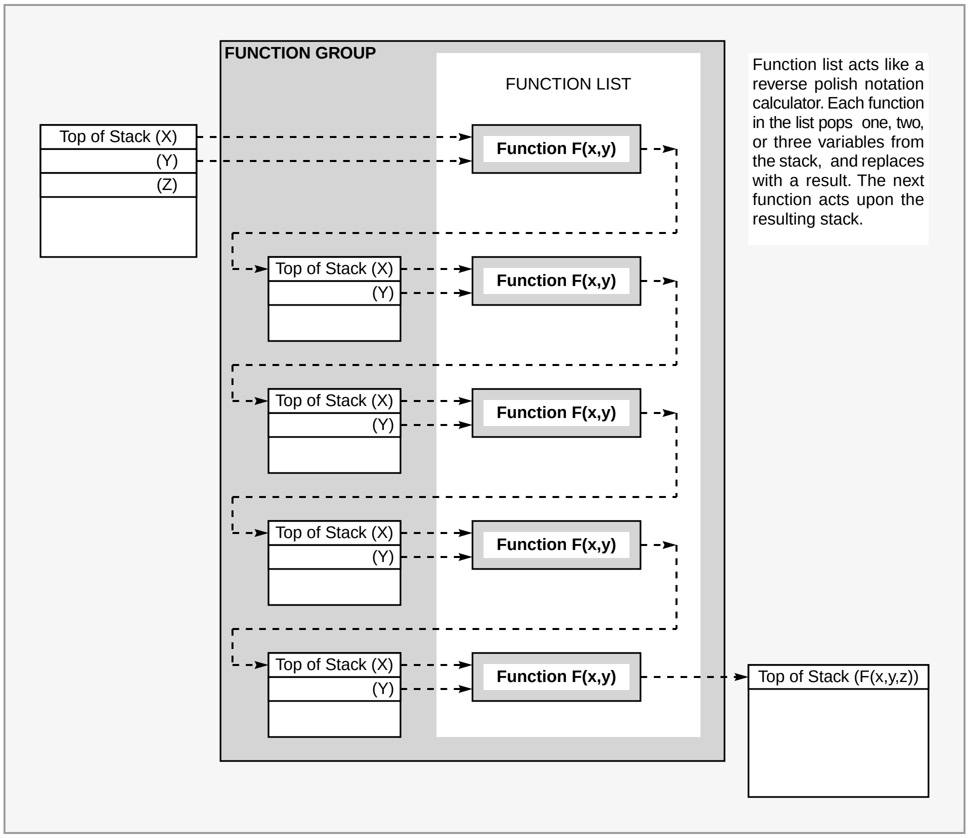
Scale factor for the Z value.

> or <=

If this field is >, then the function value will be Y times the Y scale factor if X is greater than the threshold, otherwise it will be Z times the Z scale factor.

If this field is <=, then the function value will be Y times the Y scale factor if X is less than or equal to the threshold, otherwise it will be Z times the Z scale factor.

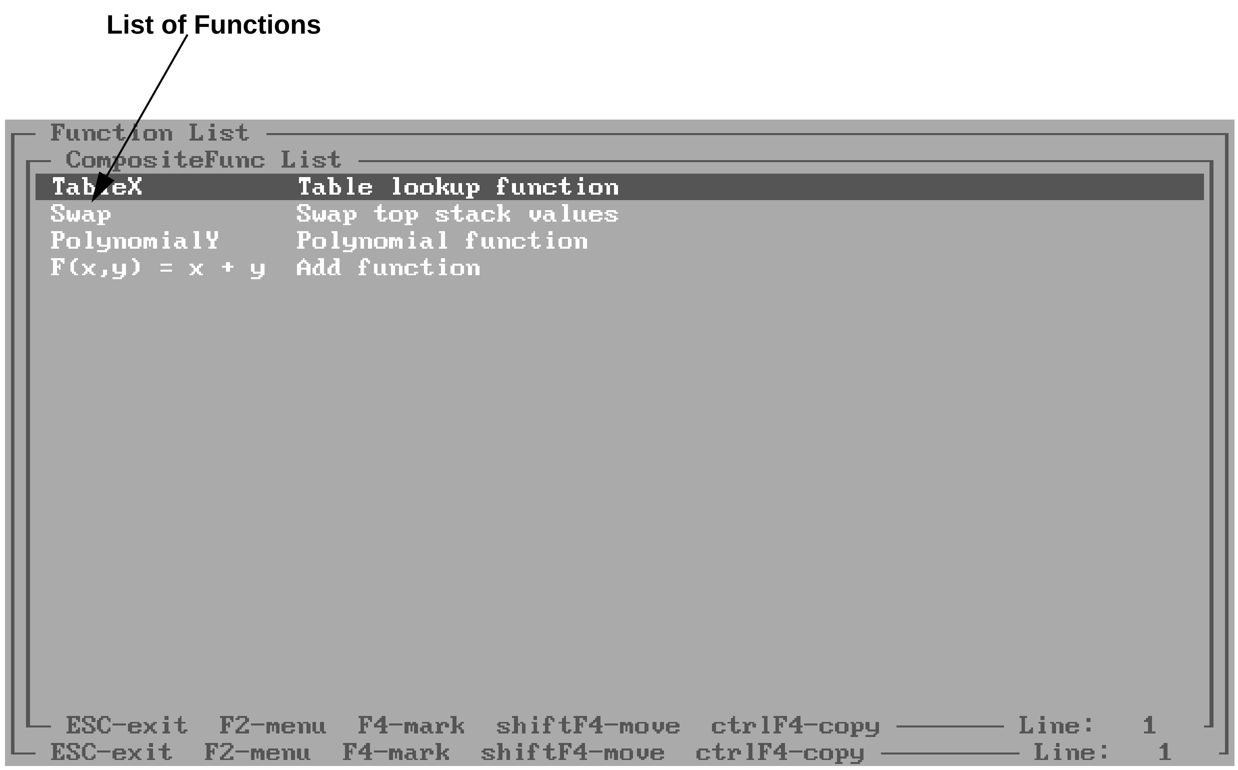
Function Group



Function Group

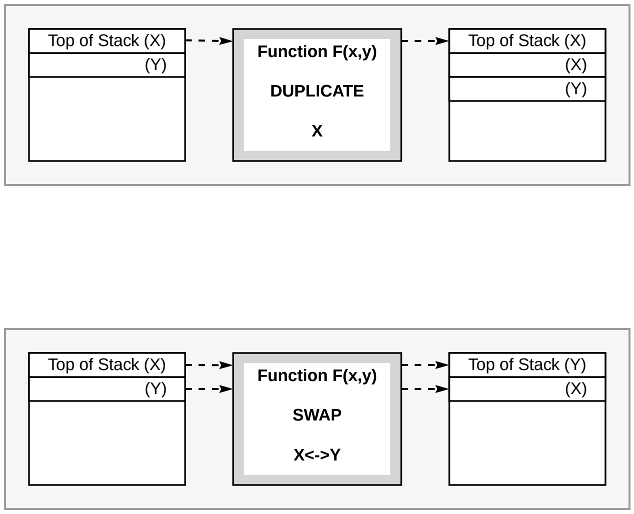
The function group provides a mechanism for producing more complicated functions based on the existing simple functions. The group is a list of functions that act upon the eight element deep stack. Each function operates on the top stack elements, then passes its resulting stack on to the next function. The order of operation is that of the function list with the top function acting on the stack first.

Function Group



The function group list has no configurable parameters.

Swap / Duplicate



Swap / Duplicate

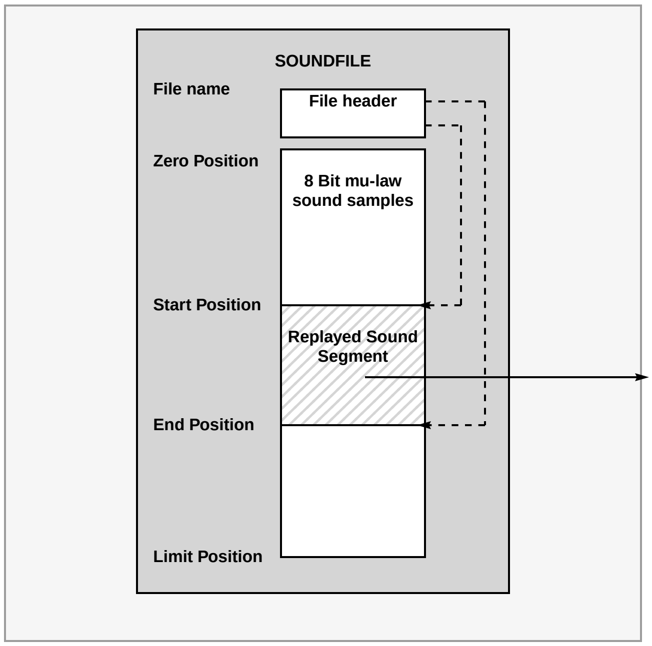
The swap and duplicate functions are only available inside of a function group. They provide simple stack manipulation, for duplicating the top stack element, or for swapping the top two elements.

Swap / Duplicate

These functions have no configurable parameters.

Sound Files

Sound File

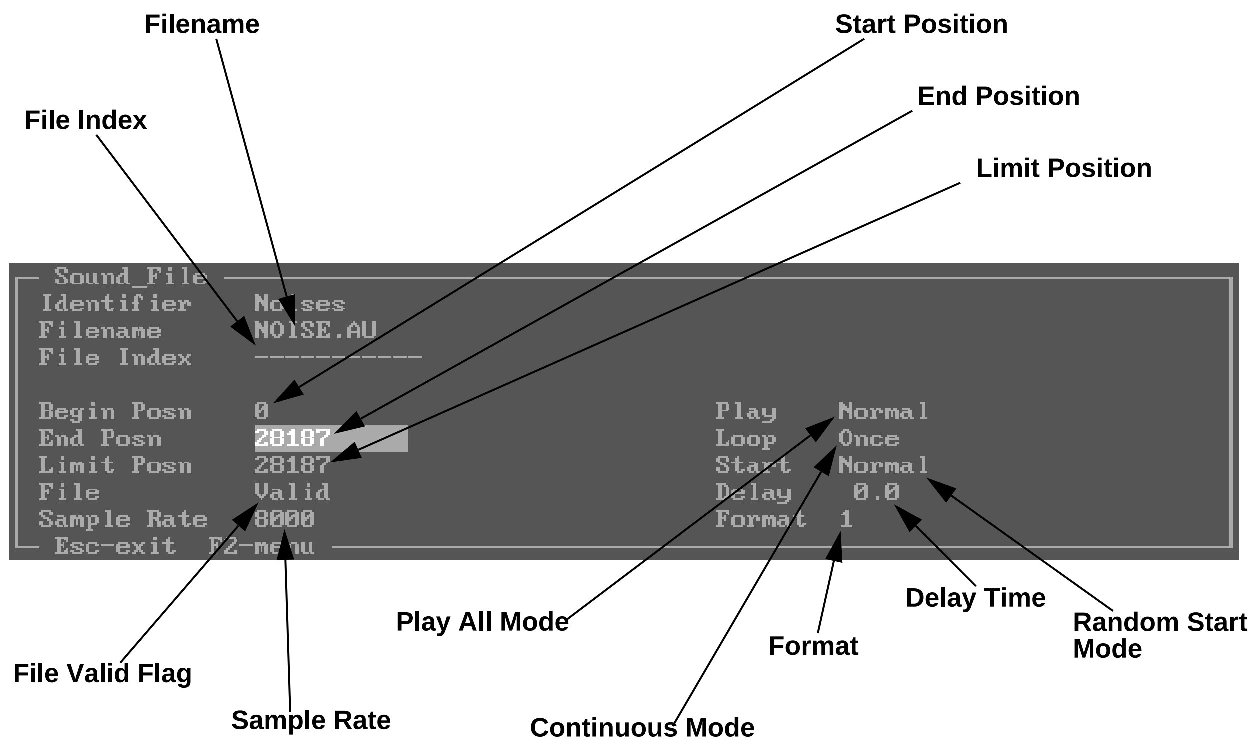


Sound File

A sound file contains a recorded sound which can be played by the Replay Sound signal. The sound can be recorded by the Record Sound signal or the Record Highway feeder.

Sounds which have no dynamically varying elements (except for overall volume level), are best handled as fixed off-line recorded sound segments stored in sound files. (E.g. Missile launch)

Sound File



Filename

Sound file name.

File Index

File index number, used for indexing into a group of files (only valid when the file has been installed in a file group).

Play All Mode

Mode control flag, if set On forces the sound file to be played in its entirety, if Off the file will stop playing when the trigger switches off, if this occurs before the file has completed.

Continuous Mode

If this flag is On, the sound file will be repeated continuously while the trigger is in the On state. When the trigger changes to Off, the sound will cease playing. If the play all flag is set then the sound file will play to completion.

Delay Time

When in continuous mode the delay time provides a space (in seconds) between restarting the play of the sound file.

Start Position

Start position for replay, allows the sound to be trimmed at the front.

Sound File

End Position

End position for replay, allows the trailing end of a recording to be trimmed.

Limit Position

Last sample in sound file, defines maximum value of start or end position values.

Random Start Mode

This field can either be set to "Random" or "Normal" If set to "Normal", the Play Sound signal will begin playing at the beginning of the file, or at the begin offset. If set to Random, the Play Sound signal will have a counter running, and will start the sound at the counter position when the sound is triggered.

Sample Rate

Indicates the number of sound samples per second in the sound file.

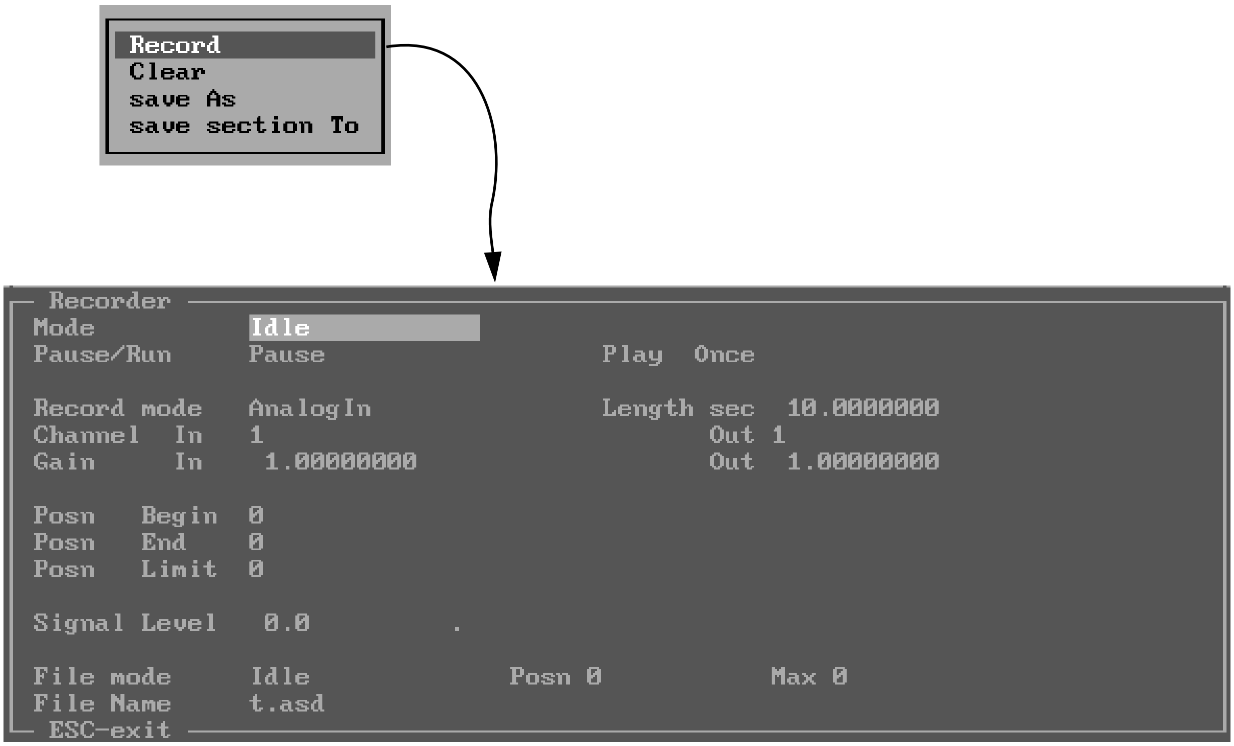
Format

Indicates the format of the sound file. 1 is an AU file and 256 is an ASD file.

File Valid Flag

Indicates whether the sound file exists and is a valid format.

Recorder Panel



The recorder panel is used to record sounds into sounds files “off-line” for playback during model execution. Sounds can be recorded either directly from an analog input or from a channel highway (meaning the input sound can be filtered or modified by a model.) It also allows a sound file to be “cropped”, eliminating the need for precise timing control over the starting and stopping the recording.

Sounds cannot be recorded during normal model execution with this panel. That can be done with the Record Objects which come with the record/playback software.

In order to play back sounds from the Recorder Panel, the “Output” object must be in the feeders list. (See the “Analog Out” feeder object.).

To get to the recorder panel, hit F2 from the Soundfile object. In addition to the record panel, you can clear the sound file with the “Clear” option, or save the section of the soundfile between “Begin Posn” and “End Posn” to a new soundfile, thus cropping the ends off of the file.

Mode

This can be set to playing, recording, or idle. Hitting the ‘+’ key toggles between playing and idle, while hitting the ‘-’ key toggles between recording and idling.

Pause/Run

Pauses the playback or recording.

Play

Recorder Panel

This can be set to Once or Continuous. If set to Once, the sound playing back will stop at the end of the file during playback from this panel. If set to continuous, the sound will repeat from the beginning when the end of the file is reached.

Recording always stops at the end of the file.

Record Mode

If this is set to AnalogIn, the recorded sound comes from the channel specified in channel in. If it is set to Highway, the recorded sound is taken from the sound highway specified by channel in. For recording from the highway, some model must be set up so that sound is being played on the highway.

Length sec

Gives the maximum length of the sound file, in seconds. The mode will automatically switch from record to idle when the recorded sound reaches this length.

Channel In

Specifies the analog input or highway input that is being recorded.

Channel Out

Specifies the Analog output that the sound is played back to.

Gain In

Specifies the input gain for the sound being recorded. Note that if you are recording from a microphone or low level audio signal, it's itsnecessary to set the preamplifier gain in the model configuration window as well.

Gain Out

Specifies the output gain during sound playback.

Posn Begin, Posn End

Specifies the starting and ending position for the sound playback, in sound samples. Right after recording, the begin position will be zero and the end position will be proportional to the length of the sound.

After the sound is recorded, these numbers can be changed to cut off some portion of the beginning and the end of the sound file. Once the desired cropping is achieved, you can use the "save section to" option in the sound file sub menu to save the cropped sound to its own file.

Posn Limit

The maximum position that the Posn End can be adjusted to. This is the number of sound samples in the sound file.

Signal Level

During recording, this field indicates the strength of the signal being recorded.

File Mode

Indicates whether the file is playing, recording, or idle.

File Posn

Indicated the current position in the file.

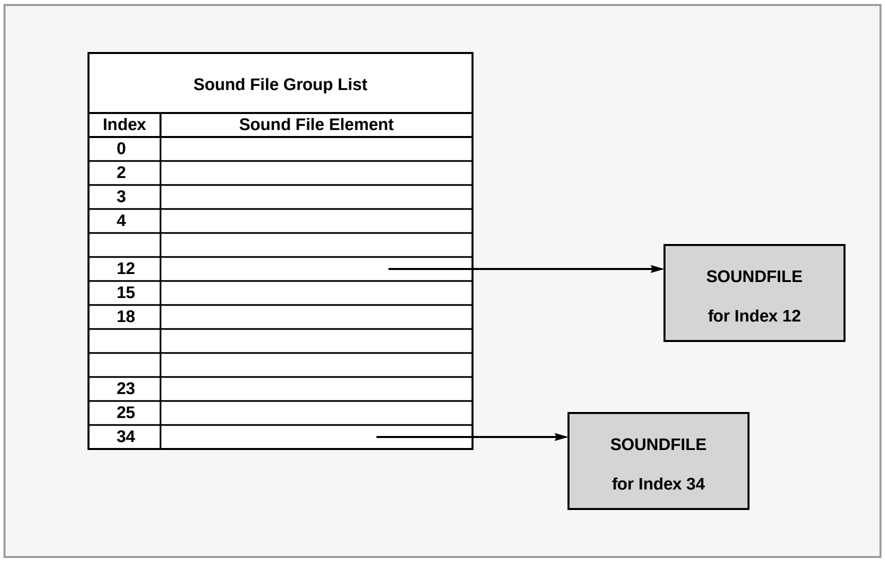
Max

Indicates the maximum position in the file. This number is equal to the length, in seconds, times the sample rate.

File Name

Indicates the name of the sound file being used.

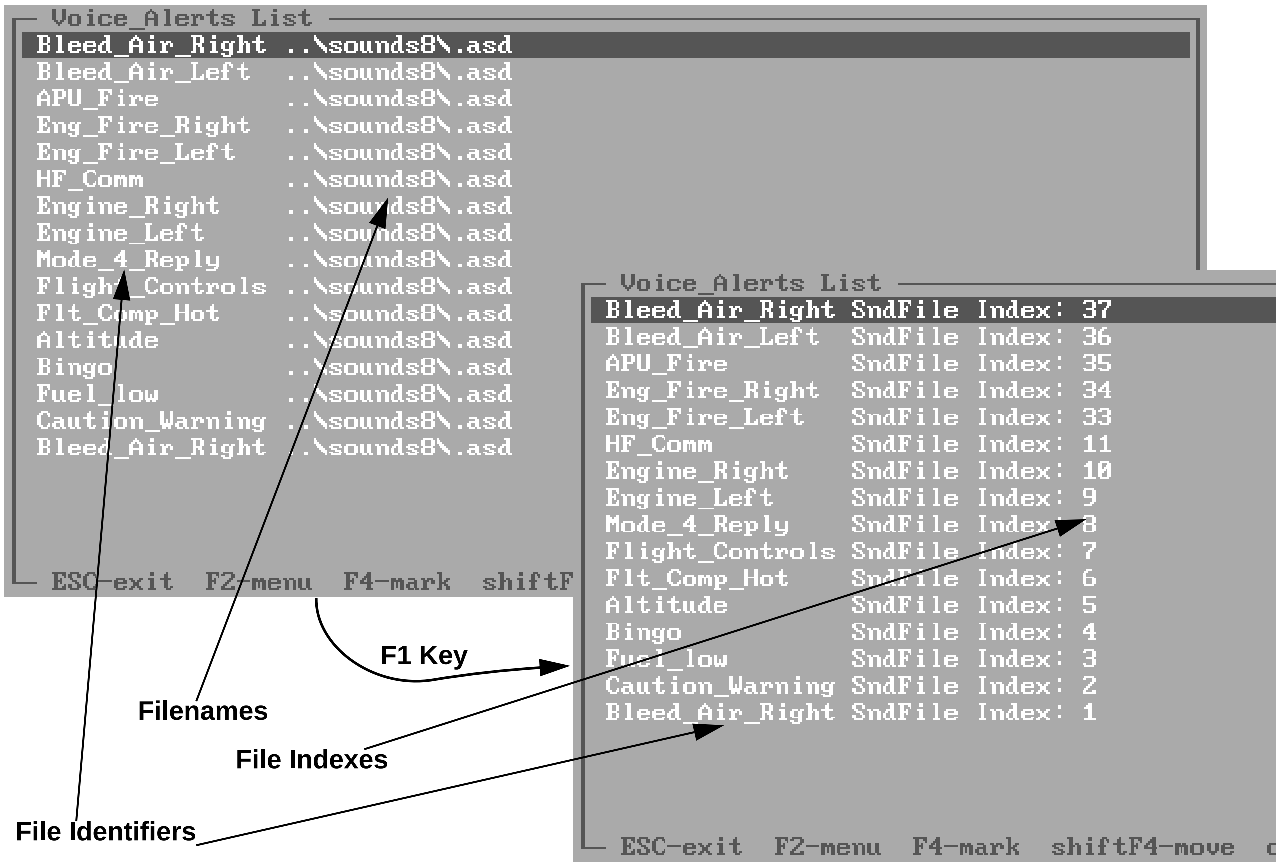
Sound File Group



Sound File Group

The sound file group provides an indexed directory of sound files. The index is a 16 bit integer which is user definable and unique to the individual sound group. When used with any sound replay object it allows the host computer to dynamically select files for replay from a predefined set. (e.g. selecting high speed or low speed gunfire, or different airport facility ATIS messages.)

Sound File Group



File Identifiers

Object name identifiers for each sound file.

Filenames

Pathname and filename for each sound file in the group.

File Indexes

List of file indexes used to fetch correct sound file from group. The indexes need to be unique to the group.

NOTE: The file indexes page is displayed by toggling the page format with the F1 key.

System Configuration

Configuration Files

What is a Configuration file?

There are two conceptual levels at which the user defines the application-specific features of the ASTi DACS.

One of these pertains to the actual cues generated, and their routing to the various user microphones and speakers. This level, in principle, is a function of the platform being simulated, the specific tank, aircraft, or submarine etc. This is referred to as the overall "Custom Model". It is comprised of several individual "models" running on each DSP.

The other level of customization, being addressed in this chapter, is related to the system installation, such as the Host Ethernet interface, or the model iteration rate. This area of user-specific configuration is termed the "System Configuration" and is modified using special "System Configuration Commands". These configuration details are, in general, not a direct function of the platform being simulated.

The System Configuration File is an ASCII text file containing system wide configuration commands which are used to set ModelBuilder's internal variables

Saving System Configuration Files

Within Model Builder the user is able to make changes to these System Configuration files on the relevant interactive pages, as described in the following sections of this chapter, and observe the results immediately, as with changes to the custom model. However, in order to save changes to System Configuration files, the user cannot just save the relevant page changes, but must edit the System Configuration file using System Configuration Commands. The list of System Configuration Commands and their associated syntax, is listed at the end of this chapter. The various pages that present System Configuration data are discussed in the next sections

Using the Configuration file

The ModelBuilder can be configured to your system requirements using a configuration file. This is an ASCII text file containing system wide configuration commands which are used to set ModelBuilder's internal variables.

When ModelBuilder is initiated using the MB command, an additional parameter may be appended to the command line to define the configuration file to be used during initialization.

If no parameter is specified, ModelBuilder uses the configuration file DEFAULT.CFG.

An example configuration file is shown below

Configuration file commands

Each command consists of a single line of ASCII text of the form

[Command] = [parameter]

The command line parser is case insensitive, and the square brackets [_] are not part of the command line.

The semi-colon can be used on a line to place comments against commands. All characters after the **;** are ignored by the command parser.

Sample Configuration File

```
Model1 = engines.mdl
Model2 = comms.mdl
Number_DSPs = 2
Sound_directory = ..\sounds22\
Ethernet:Local_Raw = 00-00-00-00-00-01
Ethernet:Raw_Source = 00-00-00-00-00-02
```

It should be noted that all configuration files have a.CFG suffix

The configuration file can be used to specify:

- The Ethernet address and mode parameters.
- The DIS address and mode parameters.
- The Voicenet address and mode parameters.
- The sound models to be loaded automatically.
- The default directory where the soundfiles are stored.
- The number of DSPs in the system.
- The user registration key code.

System Configuration Pages

The pages listed below are those in which the user is given access to the system configuration parameters.

Model Timing Window

Ethernet D.I.S. Protocol Window

Ethernet Control Window

Ethernet Status Window

Model Configuration Window *

The commands and readouts for these pages are dealt with individually, page by page in the following sections.

* Note: This is the only instance where system configuration parameters are presented on a custom model. The model configuration window is described in detail in the menus section at the beginning of this manual.

Sound Model Configuration file commands

Model

Command String; **MODEL[N] =** <path><filename>

Where [**N**]=Model number. Maximum value is equal to the number of DSP cards in the DACS

And where <path> = Standard DOS convention

And <filename> = Standard 8 character DOS convention

The Model line allows you to define the directory in which all sound models are stored.

The default: values are:-

Model1 = untitled.mdl

Model2 = untitled.mdl

Model3 = untitled.mdl

Model4 = untitled.mdl

Notes: Note that a path must be specified for each model

DSPGain

This command sets the pre-amp gain for individual DSP Input Channels.

Command String; **DSP<integer#>GAIN<channel #>=** <Decimal Number>

Where < integer#> = index number of the DSP card being set (Card 1 through 8) and,

Where < channel#> = channel number (1 through 8) of DSP for which the gain is being set. (i.e. each DSP card contains eight input/output audio channels.) and,

where <Decimal Number> = **YYY** Decimal number in range 1 thru 1000 that represents the selected gain.

When no DSP card number is specified then the gain applies to all DSP cards in the DACS. Similarly, when no channel number is specified the gain value applies to all eight channels.

E.g.1. **DSPGAIN=10** sets all pre-amp gains to 10 for all channels on all DSPs

E.g.2. **DSP2GAIN=10** sets all pre-amp gains on DSP card # 2 to 10 for all channels on all DSPs

E.g. DSP1GAIN3=50

DSP1GAIN4=50

sets pre-amp gains to 50 for channels 3 and 4 on DSP1

Default value is **DSPGAIN=1**

Model_Rate

Command String; **MODEL_RATE =** <Decimal Number>

Where <Decimal Number> = **YYY** Decimal number that represents the selected iteration rate for the model in Hz ranging from 1 to 100 Hz

Default value is **MODEL_RATE =24.**

Sound_Directory

Command String; **SOUND_DIRECTORY =** <Path>

Where <Path> = path to directory in which all soundfile are stored.

Default value is no path. i.e. current directory.

Where different models need different values then the model number should be appended to the command string. e.g.

Format: Sound_Directory1 = [path]
Sound_Directory2 = [path]
Sound_Directory3 = [path]
Sound_Directory4 = [path]

Number_DSPs

Command string; **NUMBER_DSP =** <Decimal Number>

Where <Decimal Number> = **Y** Decimal number. Allowable values 0 through 6.

The Number _DSPs command defines the number of DSPs present in the hardware configuration. This configuration command is used to disable error messages from non-existent DSPs, and to prevent the system error code sent back to the host from detecting a non-existent DSP.

Default value; Number_DSP = 4

Notes: To run ModelBuilder stand-alone on a PC rather than the target hardware, use Number_DSP = 0. Model Builder generates DSP interrupts from the PC timers instead of the DSPs.

Master_DSP

Command String; **MASTER_DSP** =<Decimal Number>

Where <Decimal Number> = **Y** Decimal number. Allowable values 0 through 6.

Sets the master DSP for multiple DSPs chained together via their communication ports.

Default value : **MASTER_DSP =0**

Note: Value of zero disables Master DSP communication.

Slave_DSP

Command String; SLAVE_DSP =<Decimal Number> [, <Decimal Number>]

Where <Decimal Number> = **Y** Decimal number. Allowable values 0 through 6.

Sets the slave DSP or DSPs for multiple DSPs chained together via their communication ports. In most cases, only three DSP cards can be linked, with one master and two slaves.

Default: **SLAVE_DSP =0**

PreLoad

Command String; **PRELOAD=** <Decimal Number>

Where <Decimal Number> = **YYYYY** Decimal number. Allowable values 0 through 32000.

Sets the number of 512 byte sectors of a sound file that are preloaded into the disk cache. One second of a 16 kHz sampled sound file occupies 32 sectors of disk space.

Default: value is: **PRELOAD=1000**

Sound_Extension

Command String; **SOUND_EXTENSION**= <File Extension>

Where <File Extension> = **.AAA** DOS file extension type.

Allowable values :

.asd = ASTi sound files

.au = SUN sound files

.snd = NEXT sound files

Default setting is : **SOUND_EXTENSION**= **.asd**

Linked_Highway

Command String; **LINKED_HIGHWAY=** <Mode>

Where <Mode> = **ON** or **OFF**

In the Linked Highway mode the models in the separate DSPs are all connected for the purpose of communicating data streams. Since this mode adds a significant processing burden it is only used when necessary.

Default setting: **LINKED_HIGHWAY=OFF**

AI_Address

Command String; **AI_ADDRESS=**<Hexadecimal Number> [,<Hexadecimal Number>]

Where <Hexadecimal Number> = **YYY** Decimal number 100 through 3FF

The word *default* may also be substituted instead of the hexidecimal number. It is recommended that this form be used in case the addresses of the card(s) change.

This command enables operation of up to two AI cards in the DACS. Hence, two fields are provided in the command line, one for each card in the system.

Example: **AI_ADDRESS = Default**

will set card one to the address $208.

Example: **AI_ADDRESS = Default, Default**

will set card one to the address $208 and card two to the address $108.

If the command is not included, Model Builder will not look for an AI card. If the line is included and there is a problem with the board or the board is not installed an error message will appear in the Model Builder application software.

DI_Address

Command String; **DI_ADDRESS=**<Hexadecimal Number> [,<Hexadecimal Number>]

Where <Hexadecimal Number> = **YYY** Decimal number 100 through 3FF. The word *default* may be substituted instead of the hexidecimal number. It is recommended that this form be used in case the addresses of the card(s) change.

This command enables operation of up to two DI cards or the DI section of up to two DI/DO cards in the DACS. Hence, two fields are provided in the command line, one for each card in the system.

Example: **DI_ADDRESS = Default**

will set card one to address to $200.

Example: **DI_ADDRESS = Default, Default**

will set card one to address $200 and card two to address $100.

Example: **DI_ADDRESS = Default**

DO_ADDRESS = Default

will enable the DI and DO section on one DI/DO card in the DACS.

Example: **DI_ADDRESS = Default, Default**

DO_ADDRESS = Default, Default

will enable the DI and DO section on two DI/DO cards in the DACS.

If the command is not included, Model Builder will not look for a DI card. If the line is included and there is a problem with the board or the board is not installed an error message will appear in the Model Builder application software.

DO_Address

Command String; **DO_ADDRESS =** <Hexadecimal Number> [,<Hexadecimal Number>]

Where <Hexadecimal Number> = **YYY** Decimal number 100 through 3FF. The word *default* may be substituted instead of the hexidecimal number. It is recommended that this form be used in case the addresses of the card(s) change.

This command enables operation of the DO section of up to two DI/DO cards in the DACS. Hence, two fields are provided in the command line, one for each card in the system. This command should be used in conjunction with the command **DI_ADDRESS =** to enable both the DI and DO sections on the DI/DO cards installed in the DACS.

Example: **DI_ADDRESS = Default**

DO_ADDRESS = Default

will enable the DI and DO section on one DI/DO card in the DACS and set the card address to $200.

Example: **DI_ADDRESS = Default, Default**

DO_ADDRESS = Default, Default

will enable the DI and DO section on two DI/DO cards in the DACS and set card one to address $200, card two to address $100.

If the command is not included, Model Builder will not look for a DI/DO card. If the line is included and there is a problem with the board or the board is not installed an error message will appear in the Model Builder application software.

NONSTANDARD Commands

The following commands are not for general customer use. They should only be used with the oversight of an ASTi engineer. The effects of these commands are not guaranteed, and may change without notice.

Byte_Order

This command is not for general customer use.

Command String; **BYTE_ORDER =** <Type>

Where <Type> = **LITTLE_ENDIAN** or **BIG_ENDIAN**

This provides the start-up default for the input objects, which defines the byte order of the incoming host data on all of the available interfaces. Little_Endian is used by Intel based equipment which has the least significant byte of a word at the lower address, Big_Endian is the Motorola form with the most significant byte at the lower address.

The default setting is : **BYTE_ORDER = LITTLE_ENDIAN**

Global_Dsp

This command is not for general customer use.

Command String; **Global_Dsp =** <number>

Where <number> is an integer from 1 to 5 indicating the number of cards on the global DSP bus.

CODEC_Sync

This command is not for general customer use.

Command String; **CODEC_Sync =** <state>

Where <state> is **ON** or **OFF**.

Forces the CODECs of the cards on the Global DSP bus to take their CODEC clock from the global bus. This command would only be set to **OFF** under the direction of an ASTi engineer, otherwise this command should not be used.

Default setting: **CODEC_Sync = ON**

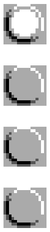
Maintenance_Page

This command is not for general customer use.

Command String; **Maintenance_Page =** <state>

Where <state> is **ON** or **OFF**.

Causes an extra “maintenance page” to appear in the main menu, for help with trouble shooting. No information appears on the maintenance page which does not appear elsewhere - it is merely gathered in one place.



Ethernet Configuration Commands

Overview of Ethernet Header Conventions and Message Passing

Introduction

Before proceeding to the detailed explanations of the System Configuration pages which relate to the use of Ethernet, a brief overview of some basic Ethernet concepts is in order to give the less familiar reader some kind of context for the various configuration options. This overview only touches upon those protocols of relevance to the user of the DACS in a training environment and ignores most of the complexity and detail that underlies the use of Ethernet for networking. The actual data transmitted in an Ethernet packet is preceded by the following (possible) sequential header data.

First Header

Ethernet Header - 14 bytes of data. Source and Destination Physical Addresses (6 bytes each), Type, or Length (2 bytes).

Second Header

I.P. Header - 20 bytes. Contains Source and Destination I/P Addresses.

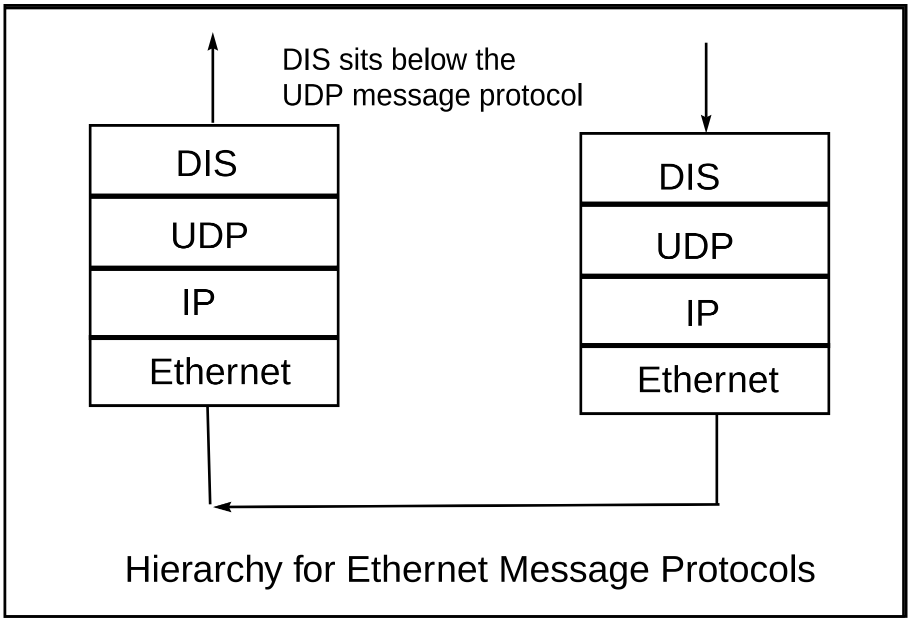
Third Header

UDP Header - 8 bytes. Contains Receiver and Transmitter port Addresses.

Fourth Header

DIS PDU See current DIS standards.

The hierarchy of Ethernet protocols can be regarded as the layered series of communication options as shown in the diagram on the next page.



The Ethernet Header

This is primarily of significance in raw mode when there is no alternative to use of the basic Physical Addresses.

The Physical Addresses are unique to every Ethernet card sold and are not user reprogrammable.

The Ethernet Header contains the following 14 bytes of data:

Destination field =6 bytes hexadecimal XX-XX-XX-XX-XX-XX. This is the Physical Address of the device to which the packet is to be routed. Broadcast mode is designated by FF-FF-FF-FF-FF-FF.

Multicast mode is designated by 01-XX-XX-XX-XX-XX

Source field = 6 bytes hexadecimal XX-XX-XX-XX-XX-XX. This is the Physical Address of the device originating the packet. This address will change if a packet is routed via other intermediate devices.

Length/Type Field = 2 bytes hexadecimal= XX-XX. Type of packet protocol being used. E.g. Raw or (08-00) for I/P mode.

On 802.3 networks this is a length field containing the number of bytes in the packet.

If Raw mode is used, then everything in the packet after the first 14 bytes is data, including the areas discussed below as I/P.and UDP headers. Even in the Raw mode the device can filter the incoming Ethernet packet by the transmitter source physical address.

The I/P Header

The next 20 bytes are the I/P header information.

The I/P Source Address is usually used to supercede the Physical Address of the workstation on the net. This allows the station to retain an address that is independent of the Ethernet card that is being used to connect to the network.

The fourth and fifth word of the header, respectively are the Source and Destination I/P Addresses These are 32 bit addresses with the decimal field configuration: YYY.YYY.YYY.YYY. where each field must be in the range 0 through 255

The Destination Address is a standard 32 bit IP address that contains the information to uniquely identify the network and a specific host on that network.

Although an IP address contains a network part and a host part, the format of these parts is not the same for every IP address. The three main classes of address are A, B, and C. In a class A network the first 8 bits of the 32 bit word identify the network and the remaining 24 bits identify the specific host within that network. For class B it is 16 bits and 16 bits respectively, and for class C, the first 24 bits are used to designate the network, with the remaining 8 bits used for the host address.

UDP Header

The User Datagram Protocol gives application programs direct access to a datagram. This allows applications to exchange messages over the network with a minimum of protocol overhead.

The UDP header contains two consecutive 16 bit fields within the first 32 bit word that define the Source (transmitter) port and Destination (receiver) port respectively. The format of these is decimal YYYY.

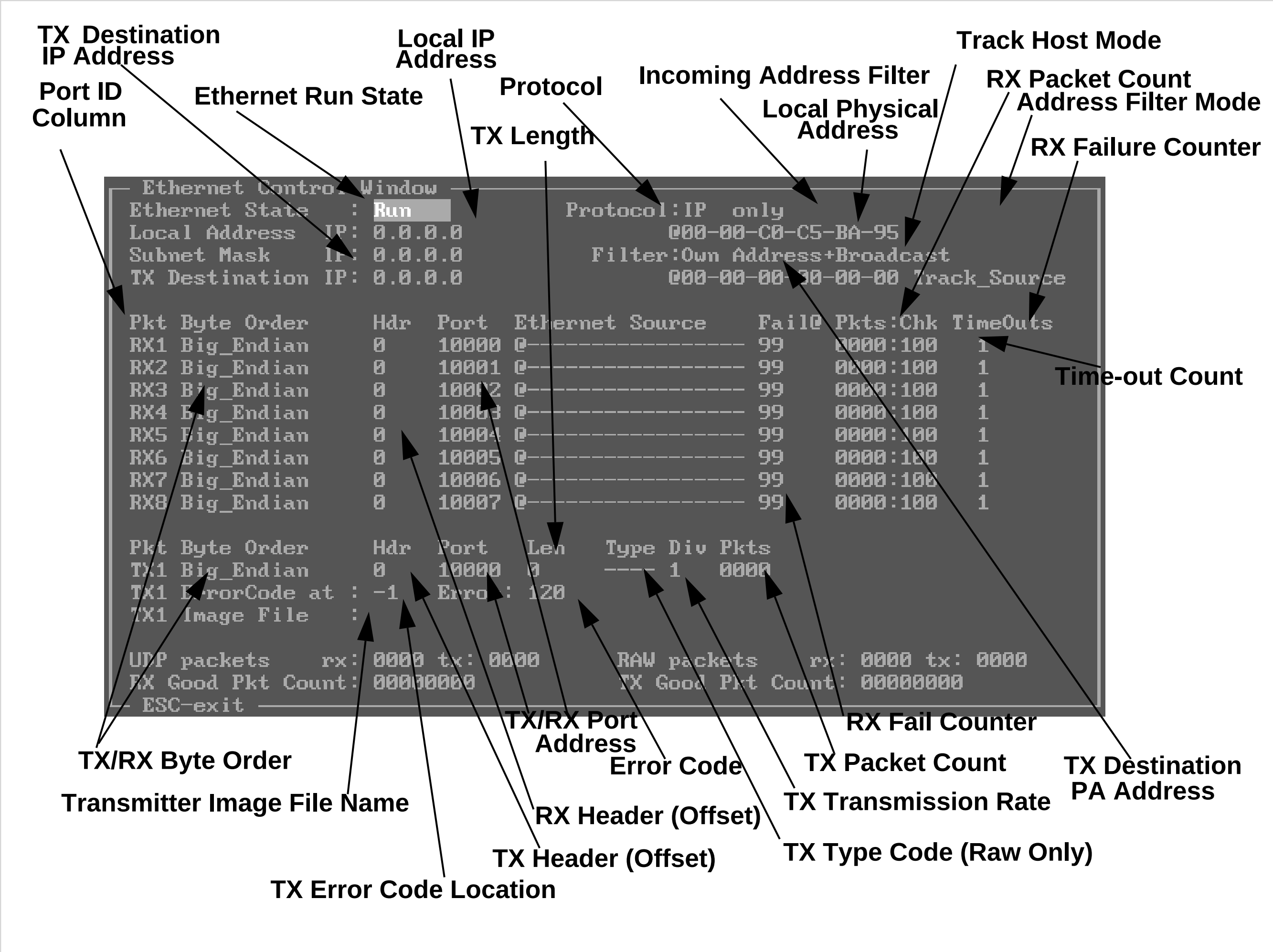
These port addresses denote the software application or “service” within the device which is the source of, or the destination for, the data packet.

For instance, the user may wish to assign multiple receiver port (RX) identities to various “Models” within the same DACS in order to provide different aural cue or communications models for multiple simulation entities. When Model Builder attaches model parameters to fields within the Ethernet packet, the data is configured to a specific receiver port address from 1 to 8. (RX1-RX8) each with a unique port address.

Similarly, within a given simulation exercise the simulators will usually all share a common IP. RX port address or “service”. For example the I/ITSEC '95 DIS UDP Port address was 6995. This means that any device participating in the DIS demonstrations will assign the Destination Address of 6995 to any data that is being broadcast to other participants in the demonstration. Similarly, the simulators will filter all external data upon that port address when importing Ethernet packets into their own application.

Model Builder allows the DACS user to specify up to eight receiver (Rx) port addresses; RX1 thru RX8 with addresses of 10,000 through 10,007 respectively.

Ethernet Control Window



Ethernet Run State

This field can be toggled between **Run** and **Freeze**.

Protocol

This indicates which Ethernet protocol is in use. It toggles between **IP only**, **Raw,** and **IP+RAW**. Raw mode is a legacy for backwards compatibility to older versions of model builder, and should not be used in new systems.

Local IP Address

This is the IP address of the local device.

Local Physical Address

This is the PA address of the DACS.

Incoming Address Filter

This field can be toggles between different options: **Own_Address/ Own_Address+Broadcast/ Multicast/All Multicast/ Promiscuous**.

Transmitter Destination -IP Address

This is the address of the device to which the transmitter packet will be addressed.

Transmitter Destination- Physical Address

This is the physical address of the device to which the transmitter packet will be addressed.

Ethernet Control Window

Transmitter ID-Track Host Mode

When in Track Host Mode the Model Builder software automatically enters the IP address of the source device as the Transmitter Destination Address.

Port ID

This column shows the port addresses from 1 to 8 within the own-device.

RX(N) Byte Order

This column indicates the mode of the header information

RX(N) Header Offset

The offset position (in bytes) in the received header, from which the header/data commences (in excess of the standard 14 bytes).

RX(N) Port Address

The IP address assigned to this port within the own-device.

RX(N) Ethernet Source

The Physical Address assigned to this port.

RX(N) Fail Counter

The number of frame cycles that must elapse without a packet being received to trigger an error message.

RX Packet Count

The number of packets received.

Time-out Count Check

The number of frames elapsed on the time-out counter since the last packet.

RX Failure Counter

The number of frames elapsed on the time-out counter since the last packet.

TX Byte Order

This column indicates the mode of the header information (Little_Endian/ Big_Endian) for the packet transmitted to the host.

TX Header Offset

The offset position (in bytes) in the transmitted header, from which the header/data commences (in excess of the standard 14 bytes).

TX Port Address

The IP address assigned to the destination address for the transmission packet.

TX Length

The number of bytes in the transmitted data packet (including header).

Tx Type

This field contains user defined type data.

TX Transmitter Rate

The number of frame cycles that elapse between packets transmitted by the host.

TX Packet Count

The count of Transmitter packets sent to host.

TX Error Code Location

The offset position in the transmitted header from which the data commences.

Error Code

The code of any errors that have occurred. This code is written to the TX error code location in the TX buffer.

Once an error has occurred and this field has a non zero value, new error codes are locked out and the same error code is sent repeatedly. To allow new error codes, set this field back to zero. This field can also be manually set to check the transmission of the error code to the host computer.

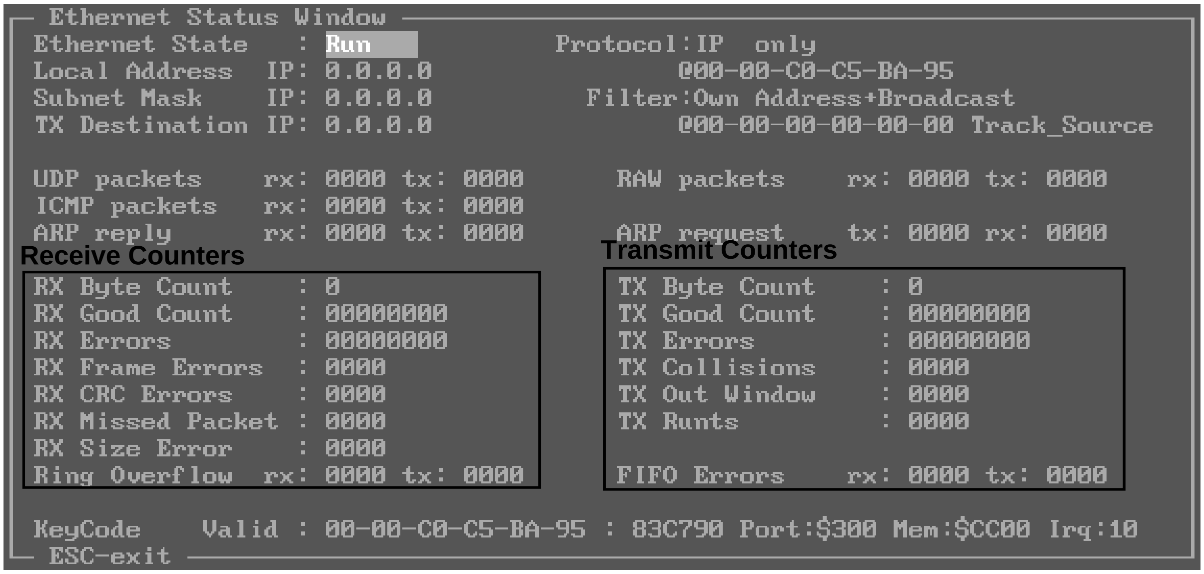
Transmitter Image File

Sets the name of the Ethernet image file that will be loaded upon initialization.

UDP and RAW RX/TX/good packet counts

These fields at the bottom of the page are self explanatory. Their main purpose is to indicate general activity levels.

Ethernet Status Window



The Ethernet Status Window indicates the state of the Ethernet link to the host computer in more detail than the control window. Much of the information in the status window is repeated from the control window.

ICMP Packets rx/tx

Internet Control Monitoring Packets received and sent. When a “ping” command is received by the DACS over the internet, the receive counter is incremented. When it replies to the ping, the tx counter is incremented.

ARP reply rx/tx

In the IP protocol, the host computer will send out an ARP request to get the Ethernet address of the DACS. When the DACS receives this request and replies to it, the host can send packets to the DACS.

ARP request rx/tx

When the DACS sends an ARP request to get the Ethernet address of the host computer, the tx counter is incremented. When it receives a reply, it updates the TX destination Ethernet address and increments the rx counter. This is used when the DACS is sending control data back to the host computer.

Ethernet Status Window

Transmit Counters

Indicates the activity of packets being sent from the DACS to the host computer. Used for trouble shooting problems with the transmission of packets over the Ethernet network.

TX Collisions

Indicates the number of times the DACS attempted to send out an Ethernet packet at the same time another computer was attempting to use the network. A large number of collisions indicates a heavily loaded network.

Receive Counters

Various counters used for troubleshooting problems with receiving packets over the Ethernet link.

Ethernet Configuration Commands

Ethernet

Command string; **Ethernet=**<Mode>

Where <Mode>= **ON** / **OFF**

The Ethernet host interface can be disabled by setting Ethernet = Off.

The default value is Ethernet = ON

Ethernet:Local

Command string; **Ethernet:LOCAL=**<Physical Address>

Where <Physical Address> is entered in the format **XX-XX-XX-XX-XX-XX** where each field is a pair of hexadecimal numbers separated by a dash.

OR

Command string; **Ethernet:LOCAL=**<IP Address>

Where <IP Address> = **YYY.YYY.YYY.YYY** Where each field is a decimal number in the range 0 through 255 separated by periods.

The Local command sets either the IP (Internet Protocol) address or the physical Ethernet address. If the address is a valid IP address then the system will operate in IP mode, else the system will work in a raw physical mode with the address string being decoded as a six byte hexadecimal physical address.

Default: **Ethernet:LOCAL= 0.0.0.0**

or: **Ethernet:LOCAL=** ROM address of Ethernet hardware.

When used as a physical address the local command overrides the physical address of the Ethernet card in the DACS. This allows the user to configure the physical address of the sound system to any value and remove the need for any network address changes if the Ethernet board ever needs to be replaced.

Warning: To prevent problems associated with broadcast and multicast logic it is recommended that the first two digits in physical Ethernet address be zero i.e. Local = 00-XX-XX-XX-XX-XX.

Note: If this command is not used then the ROM address in the Ethernet card will be used as a local address. However since currently there is no support for automatic address resolution, this address will need to be added to the host computer network database. Should the Ethernet module fail and need to be replaced the new modules address will now need to be overridden by the old address or the host database will need to be updated. We recommend that this local address be overridden as part of the system design to ensure that maintenance personnel need to undertake no software changes when resolving hardware problems.

Note: A Physical Address of all zero should be avoided as some networks treat this as a broadcast address.

Ethernet:Local_IP

Command string; **Ethernet:LOCAL_IP=** <IP Address>

Where <IP Address> = **YYY.YYY.YYY.YYY** Decimal number where each field can be 0 thru 255.

This command sets the local IP address of the own device. This address supersedes the PA address to make the own device address independent of the physical Ethernet card.

The default value is**: Ethernet:LOCAL_IP= 0.0.0.0**

Ethernet:Subnet_Mask

Command string;**Ethernet:SUBNET_MASK=**<IP Address>

Where <IP Address> = **YYY.YYY.YYY.YYY** (Decimal number where each field can be 0 thru 255.)

This command sets the subnet mask bits for determining IP broadcast addresses.

The default is the correct subnet mask for the local IP address based upon device class.

Ethernet:Local_Raw

Command string; **Ethernet:LOCAL_RAW=**<Physical Address>

Where <Physical Address>= **XX-XX-XX-XX-XX-XX** Hexadecimal numbers where each field can be 0 thru FF.

This command sets the local Physical Address of the own device.

The default value is the PA of the installed Ethernet card

Ethernet:IP_Mode

Command string; **Ethernet:IP_MODE**=<Mode>

Where <Mode>= **IP_ONLY** or

<Mode>= **RAW_ONLY** or

<Mode>= **IP+RAW_**

This command forces the Ethernet protocol into IP mode (Internet Protocol), including UDP, ARP, and ICMP. The IP+RAW mode allows both IP and raw packets to be received into RX1 throughRX8

Default: **Ethernet:IP_MODE = IP_ONLY**

Ethernet:RX_Mode

Command string; **Ethernet:RX_MODE=**<Mode>

Where <Mode> = **OWN_ADDRESS** or

<Mode> =**BROADCAST** or

<Mode> = **MULTICAST** or

<Mode> =**ALL_MULTICAST** or

<Mode> =**PROMISCUOUS**

OWN_ADDRESS will read only those PDUs with the own-device Ethernet Physical Address in the header of the incoming transmission packet.

BROADCAST will read those PDUs with the own-device Ethernet Physical Address and Broadcast Address in the header.

MULTICAST will read those PDUs with the own-device Ethernet Physical Address, and Broadcast Address, and Multicast addresses in use.

ALL_MULTICAST will read all of the above packets plus all Multicast addresses.

PROMISCUOUS will read <u>every</u> packet and should only be used for debug purposes.

Note. On the Ethernet Control Page this field is toggled between these settings.

The default value is **Ethernet:RX_MODE =BROADCAST**

Ethernet:RX[N] _RAW_Source

Command string; **Ethernet:RX**[N]**_RAW_SOURCE**=<Physical Address>

Where [N] is an optional suffix to RX,

and where <Physical Address> = **XX-XX-XX-XX-XX-XX** Hexadecimal number.

This sets the Ethernet Physical Address upon which incoming packets will be filtered.

If n is 1 thru 8 then this filter is set only for the receiver port with the specified index. If no index number is used then the filter address is valid for all receiver ports

The default value is **Ethernet:RX**[N]**_RAW_SOURCE**=**FF-FF-FF-FF-FF-FF**. i.e. all incoming sources.

Ethernet:RX_Header

Command string; **Ethernet:RX_HEADER=** <Decimal Number>

Where <Decimal Number>= **YY**

This gives the DACS a value for the byte offset that will be used for all incoming packet headers to account for longer headers than the standard 14 byte physical layer. Example: A value of 4 would move the zero input offset down to the 18th byte in the packet.

The default value is Ethernet:RX_Header = 0

Ethernet:RX[N]_UDP_Port

Command string; **Ethernet:RX[N]_UDP_PORT=**<Address>

Where [N] is an optional suffix to RX,

and where <Address>= **YYYYY** Decimal Number

This is the port address of the “service” within the DACS that will be used to filter incoming data packets by their UDP destination port address

This sets the UDP Address upon which incoming packets will be filtered.

If n is 1 thru 8 then this filter is set only for the receiver port with the specified index. If no index number is used then the UDP filter address is valid for all receiver ports

The default values are 10,000 through10,007

Ethernet:RX[N]_Byte_Order

Command string; **Ethernet:RX[N]_BYTE_ORDER=**<type>

Where [N] is an optional suffix to RX,

and where <Type> = **Little_Endian** or **Big_Endian**

This provides the start-up default for the input objects, which defines the byte order of the incoming Ethernet packet. Little_endian is used by Intel based equipment which has the least significant byte of a word at the lower address, Big_Endian is the Motorola form with the most significant byte at the lower address.

The default setting is**: Ethernet:RX[N]_BYTE_ORDER=Little_Endian**

Ethernet:RX[N]_Fail_Count

Command string; **Ethernet:RX[N]_FAIL_COUNT=**<Number>

Where <Number> = **YYYYY**, a number from 0 to 65535 that represents the number of iteration periods necessary to trigger the counter, and where [N] is an optional suffix to RX that specifies the port to which the count applies. If no suffix is used, the count is set for all ports.

The count sets the maximum delay that can occur before an error message is returned to the host computer. This period is defined by the number of iteration frames that must elapse. A value of 0 means that there is no error message generated. The missed frame count is used for re-initializing the model control parameters. When the missed packet count, which increments once per model frame, reaches this value, the initial values are reloaded into the control inputs. The control inputs remain frozen in their initial state until a valid packet is received from the host computer, at which time the host fail counter is reset to start another time-out cycle.

The default value is 99.

Ethernet:TX_Length

Command string; **Ethernet:TX_LENGTH=**<Decimal Number>

Where <Decimal Number > =YYYY

This sets the length of the transmit packet back to the host computer (in bytes). This includes the fourteen bytes for source, destination and length fields that make up the physical link level Ethernet header. A value of zero disables the transmission of a data packet from the DACS back to the host computer. The minimum value for N is 60 and the maximum is 1500. Note: The packet transmission is synchronous with the model execution rate (e.g. 30 Hz) and asynchronous with the Ethernet reception.

The default value is **Ethernet:TX_LENGTH**=0

Ethernet:TX_UDP_Port

Command string; **Ethernet:TX_UDP_PORT=**<Decimal Number>

Where <Decimal Number > =YYYY in the range 0 to 65535

This command sets the outgoing Ethernet UDP Destination Address (Transmission port number) to which the Transmission data packet will be routed. The primary application for the transmitter packet is to return error information back to the simulation host computer.

The default value is **Ethernet:TX_UDP_PORT=10000**.

Ethernet:TX_Header

Command string; **Ethernet:TX_HEADER=** <Decimal Number>

Where <Decimal number> = **YY**

This allows a byte offset to be added to all transmit packet offsets to account for longer headers than the standard 14 byte physical layer. Example: A value of 4 would move the zero input offset down to the 18th byte in the packet.

The default value is **Ethernet:TX_HEADER = 0**

Ethernet:TX_Destination_IP

Command string; **Ethernet:TX_DESTINATION_IP=**<IP Address>

Where <IP address>= **YYY.YYY.YYY.YYY**. Decimal number

This command sets the destination address for the transmit Ethernet packet as an IP address.

The default value is:

Ethernet:TX_DESTINATION_IP=0.0.0.0 (i.e. no valid IP address)

Ethernet:TX_Byte_Order

Command string; **Ethernet:TX_BYTE_ORDER=**<Mode>

Where <Mode> = **LITTLE_ENDIAN** or **BIG_ENDIAN**

This provides the start-up default for the output objects, which defines the byte order of the outgoing Ethernet transmission packet. Little_endian is used by Intel based equipment which has the least significant byte of a word at the lower address, Big_Endian is the Motorola form with the most significant byte at the lower address.

The default value is:

Ethernet:TX_BYTE_ORDER=LITTLE_ENDIAN

Ethernet:TX_Error

Command string; **Ethernet:TX_ERROR**= <Decimal Number>

Where <Decimal Number>= **YYYY**

The DACS provides a system status/error code which can be inserted into the transmit packet. The TX1_Error Code command specifies where the error code byte is located in the packet. A value of -1 disables the error code, a value between 0 and 1500 specifies which packet byte will be overwritten by the error code. Note: this offset is the same as the offsets used for the input objects, i.e. offset 0 is the first byte after the 14 byte link level header.

The default value is: **Ethernet:TX_ERROR**=-1

Ethernet:TX_Destination_Raw

Command string; **Ethernet:DESTINATION_RAW=**<Physical Address>

Where <Physical Address>= **XX-XX-XX-XX-XX-XX** The hexadecimal six byte address

This command sets the destination address for the transmit Ethernet packet as a Physical Address.

The default value is :

Ethernet:DESTINATION_RAW=FF-FF-FF-FF-FF-FF (broadcast address)

Ethernet:TX_Type

Command string; Ethernet:TX_TYPE= <Hexadecimal Number>

Where <Hexadecimal Number> = **XX**

This command sets a user defined value into the transmit packet's protocol type field. A value of zero enables the use of the type field in the Ethernet header, as an IEEE 802.3 length field.

The default value is Ethernet:TX_Type = 0

Ethernet:Track_Source

Command string; **Ethernet:TRACK_SOURCE=**<Mode>

Where <Mode> = **ON** or **OFF**

This command sets the outgoing Ethernet IP or Raw Destination Address (Transmission port number) to that of the incoming source. This mode is referred to as "Track_Source" on the Ethernet Control Window page.

The default value is**: Ethernet:TRACK_SOURCE=ON**

Ethernet:TX_Image

Command string; **Ethernet:TX_IMAGE=** <File Name>

Where <File Name> = <**PATH**> <**FILE_NAME**>

Sets the name of an Ethernet image file that is loaded into the TX buffer at initialization.

The default value is no filename.

Ethernet:TX_Rate

Command string; **Ethernet:TX_RATE**= <Decimal Number>

Where <Decimal Number>= **YYYY**. Decimal number between 1 and 1000.

This command sets the interval at which the transmission packet is sent back to the host. The number entered into the variable field represents the number of frame iterations between transmission packets. This is shown as the Division rate on the Ethernet Control page.

The default value is **Ethernet:TX_RATE**= **1**

Ethernet:TX_on_RX

Command string; **Ethernet:TX_ON_RX** = <Mode>

Where <Mode> = **ON** or **OFF**

This sets the Ethernet driver into a special mode whereby the latest transmit packet is sent back to the host computer in response to receiving a valid packet from the host. This is useful where the host Ethernet driver wants to effectively control when the sound system transmits on the network.

The default setting is **Ethernet:TX_ON_RX =OFF**

Note: This command is only supported on the earlier versions of Model Builder.

Bit3 Configuration Commands

NOTE - Bit3 is only supported up to Model Builder 3.05. There is no Bit 3 support in later versions of Model Builder.

Bit3

Format: Bit3 = [on] [off]

The Bit3 host interface can be disabled by setting Bit3 = Off.

Default: Bit3 = ON

Bit3:RX_Offset

Format: Bit3:RX_Offset = [N]

This sets the memory offset for the receive area in the 32k byte Bit3 memory buffer. (Note: N is hexadecimal address)

Default: Bit3:RX_Offset = 0

Bit3:TX_Offset

Format: Bit3:TX_Offset = [N]

This sets the memory offset for the transmit area in the 32k byte Bit3 memory buffer. (Note: N is hexadecimal address)

Default: Bit3:TX_Offset = 4000

Bit3:RX_Frame_Count

Format: Bit3:RX_Frame_Count = [N]

The RX_Frame_Count sets the location in receive memory for the receive watchdog counter from the host. This byte counter is provided by the host simulation software to indicate that the simulation variables in dual port memory are valid. If the counter fails to change an internal host fail counter is incremented once per model frame. Once this counter reaches it’s limit the input controls are re-initialized and the Bit3 memory is disabled until this frame counter starts running consistently again. Note: Setting this location to -1 disables the host fail feature and keeps Bit3 memory permanently valid.

Default: Bit3:RX_Frame_Count = 0

Bit3:TX_Frame_Count

Format: Bit3:TX_Frame_Count = [N]

The TX_Frame_Count sets the location in receive memory for the transmit watchdog counter to the host. This byte counter is incremented one per model frame, providing a method for the host to determine whether the DACS is running. Note: Setting this location to -1 disables the counter.

Default: Bit3:TX_Frame_Count = 0

Bit3:Host_Fail_Count

Format: Bit3:Host_Fail_Count = [N]

The missed frame count used for re-initializing the model control parameters. When the missed packet count, which increments once per model frame, reaches this value the initial values are reloaded into the control inputs. When the receive counter starts running again the new data in the Bit3 memory is used for input controls.

Default: Bit3:Host_Fail_Count = 100

Bit3:Byte_Order

Format: Bit3:Byte_Order = [type]

where [type] = Little_Endian or Big_Endian or Encore

This provides the start-up default for the input objects, which defines the byte order of the Bit3 memory. Little_endian is used by Intel based equipment which has the least significant byte of a word at the lower address, Big_Endian is the Motorola form with the most significant byte at the lower address. The Encore format uses Big_Endian with support for the non-IEEE floating point used on SEL/Gould/Encore computers.

Default: Bit3:Byte_Order = Little_Endian

Bit3:TX_Error

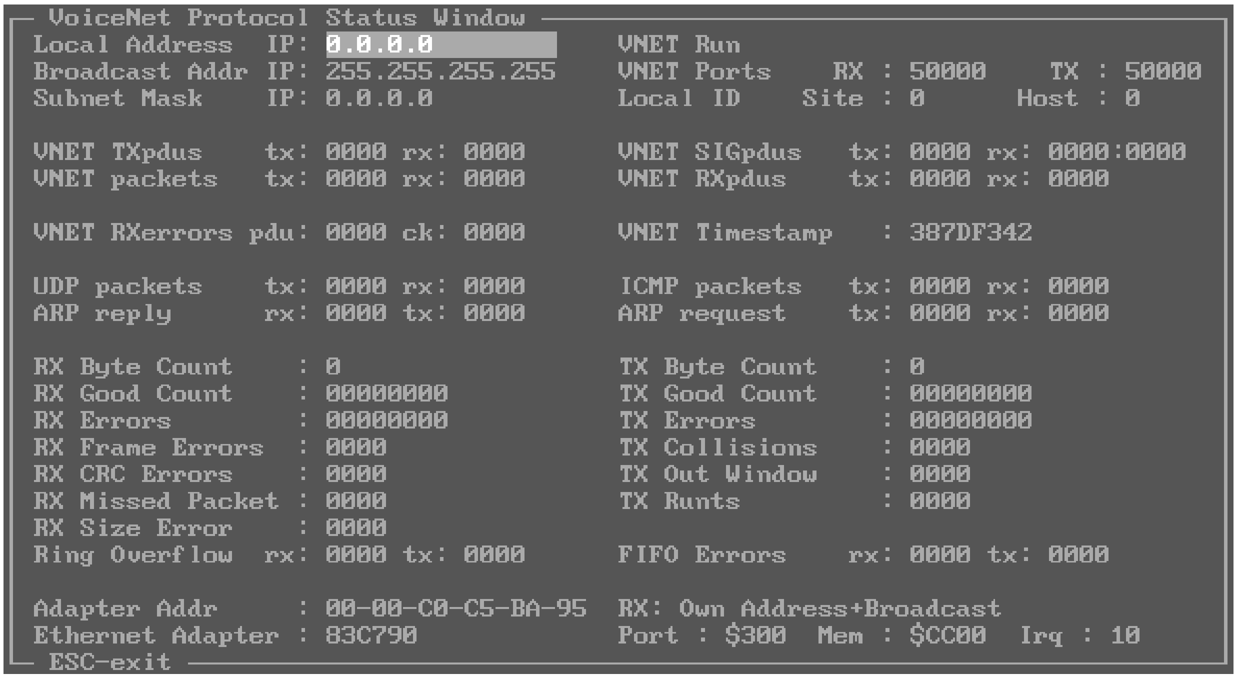
Format: Bit3:TX_Error = [N]

The DSS provides a system status/error code which can be inserted into the transmit memory. This command specifies where the error code byte should be placed. A value of -1 disables the error code, a value between 0 and 1500 specifies which packet byte should be overwritten by the error code.

Default: Bit3:TX_Error = -1

Voice Net Configuration Commands

VoiceNet Status Window



The Voice Net Protocol Status Window indicates the state of the DACS connection to the voice network. In normal use, the user definable fields will be defined in the configuration file. The configuration file commands are given below.

Local Address

Gives the IP address of the DACS for Voice Net purposes. This address does not need to be the same as the IP address used in the host interface.

VNET Ports RX/TX

IP ports for sending and receiving Voice Net packets. See the UDP_Port, TX_UDP_Port, and RX_UDP_Port configuration commands.

Local ID Site and Host

Determines the Voice Net site and host numbers for the DACS. This numbers are usually the last two numbers of the IP address (e.g. for IP address 192.42.172.186, the site would normally be 172 and the host 186). The site and host set here determine the site and host numbers for the Voice Net information from the DACS. An object in a model can use a different site and host ID through a connection to the Entity control object.

Subnet Mask

See the configuration command Subnet_Mask.

VoiceNet Status Window

VNET TXPDUs rx/tx

Indicates the number of transmitter PDUs that have been sent and received. A transmitter PDU contains the transmitter information of the radio broadcasting, with links to the signal PDUs.

VNET SIGPDUs tx/rx

Indicates the number of signal PDUs being sent and received. The signal PDUs contain the actual voice information.

VNET Packets tx/rx

Total number of Voice Net packets sent and received.

VNET RXPDUs

Indicates the number of receiver PDUs being sent and received. Voice Net receivers send out PDUs with their world position to aid in monitoring the Voice Net exercise.

VNET RX Errors PDU/ck

Indicates the number of bad Voice Net packets received. ck indicates the number of packets received with bad checksums.

UDP packets tx/rx

Total number of packets sent and received.

ARP Reply rx/tx

Indicates the number of ARP requests for an Ethernet address received and replied to.

ARP Request rx/tx

Indicates the number of ARP requests for an Ethernet address that have been sent out, and the number of replies received.

Receive Counters

Various counters for monitoring Voice Net network activity and problems. RX byte count is the number of bytes of data received over Voice Net. The RX good count is the number of good packets received over Voice Net. The rest of the counters indicate errors.

Transmit Counters

Various Counters for monitoring Voice Net network activity and problems. The TX byte count is the number of bytes of Voice Net data sent out over the Voice Net network by the DACS. The good frame count indicates the number of good packets sent out over Voice Net.

The TX collisions counts the number of times that the DACS and another computer attempted to start sending a packet at the same time. If this is incrementing a lot it may indicate the network is heavily loaded. Occasional collisions are normal and to be expected.

Voice Net Configuration Commands

Voice_Net

Command string; **VOICE_NET**=<Mode>

Where <Mode> = [**ON**] / [**OFF**]

The Ethernet Voice Net driver can be disabled by setting **VOICE_NET = OFF**

The default value is **VOICE_NET= ON**

Voice_Net:Local_IP

Command string; **VOICE_NET:LOCAL_IP**=<IP Address>

Where <IP Address> = **YYY.YYY.YYY.YYY** . (Decimal number where each field can be 0 thru 255)

The Voice Net Local command sets the local IP address.

The default value is **VOICE_NET:LOCAL_IP=255.255.255.255**

Voice_Net:Broadcast_IP

Command string; **VOICE_NET:BROADCAST_IP=**<IP Address>

Where <IP Address> =**YYY.YYY.YYY.YYY**. (Decimal number where each field can be 1 thru 255.)

The Broadcast command sets the outgoing PDU destination IP address. This is usually a broadcast IP address, but can be a unique IP address if desired

The default value is **VOICE_NET:BROADCAST_IP**=**255.255.255.255**

Voice_Net:UDP_Port

Command string; **VOICE_NET:UDP_PORT=**< Decimal Number>

Where < Decimal Number> =**YYYY**Y. (Decimal number in the range 0 through 65535.)

This command sets both the source and destination UDP port addresses.

The default value is**VOICE_NET:UDP_PORT=50000**

Voice_Net:TX_UDP_Port

Command string; **VOICE_NET_TX_UDP_PORT=**< Decimal Number>

Where < Decimal Number> = **YYYYY**. (Decimal number in the range 0 through 65535.

This command sets the destination UDP port addresses This is the port address of the voice network “service” within the destination devices that is used to filter incoming data packets by their UDP destination port address.

The default value is **VOICE_NET:TX_UDP_PORT=50000**

Voice_Net:RX_UDP_Port

Command string; **VOICE_NET_RX_UDP_PORT=**< Decimal Number>

Where < Decimal Number> = **YYYYY**. (Decimal number in the range 0 through 65535.

This command sets the receive UDP Port address. This is the port address of the voice network “service” within the DACS that will be used to filter incoming data packets by their UDP destination port address.

The default value is **VOICE_NET:RX_UDP_PORT=50000**

Voice_Net:RX_Mode

Command string; **VOICE_NET:RX_MODE=**<Mode>

Where <Mode> = **OWN_ADDRESS** or,

<Mode> =**BROADCAST** or,

<Mode> = **MULTICAST** or,

<Mode> =**ALL_MULTICAST** or,

<Mode> =**PROMISCUOUS**

OWN_ADDRESS will read only those PDUs with the own-device Ethernet Physical Address in the header of the incoming transmission packet.

BROADCAST will read those PDUs with the own-device Ethernet Physical Address and Broadcast Address in the header.

MULTICAST will read those PDUs with the own-device Ethernet Physical Address, and Broadcast Address, and Multicast addresses in use.

ALL_MULTICAST will read all of the above packets plus all Multicast addresses.

PROMISCUOUS will read <u>every</u> packet and should only be used for debug purposes.

Note. On the Voice Net Control Page this field is toggled between these settings.

The default value is **VOICE_NET:RX_MODE =BROADCAST**

Voice_Net:Site

Command string; **VOICE_NET:SITE=**<Decimal Number>

Where <Decimal Number> = **YYYY**.

This sets the Site Address for the own device. The Site Address is the same for all PDUs emanating from the DACS

The default value is **VOICE_NET:SITE=1**

Voice_Net:Application
or Voice_Net:Host

Command string; **VOICE_NET:APPLICATION (**or **Host) =**<Decimal Number>

Where <Decimal Number> = **YYYY**.

This sets the Host or Application Address for all PDUs emanating from the DACS

The default value is **VOICE_NET:APPLICATION=1**

Voice_Net:Exercise

Command string; **VOICE_NET:EXERCISE=**<Decimal Number>

Where <Decimal Number> =**YYYY**

This sets the network filter such that only Voice Net packets for this exercise are received through the network driver. A value of zero allows all exercises through this filter.

Note : Set this to zero if the DACS is to support multiple exercises simultaneously.

The default value is **VOICE_NET:EXERCISE=0**

Voice_Net:Exercise_Offset

Command string; **VOICE_NET:EXERCISE_OFFSET**=<Decimal Number>

Where <Decimal Number> = **YYYY**

The host computer for the simulation entity to which the DACS platform is connected has the ability to remotely set values for Exercise filter. This command identifies the number of bytes by which the Exercise number is offset from location zero in the Ethernet packet transmitted to the DACS from the simulator host.

A value of -1 disables the host control of the exercise filter

The default value is**VOICE_NET:EXERCISE_OFFSET**= **-1**.

Voice_Net:Subnet_Mask

Command string; **VOICE_NET:SUBNET_MASK=**<IP Address>

Where <IP Address> = **YYY.YYY.YYY.YYY** (Decimal number where each field can be 0 thru 255.)

This command sets the subnet mask bits for determining IP broadcast addresses.

The default is the correct subnet mask for the local IP address based upon device class.

UNPUBLISHED COMMANDS

Voice_Net:Protocol

Command string; <VOICE_NET:PROTOCOL=[type variable]>

]

Variable = [3] [4] or [5]

3 = DIS 2.0.3

4 = DIS 2.0.4

5= DIS IEEE

This command sets the revision level of DIS standard for inter-operability between DACS units.

The default value is DIS 2.0.4

DIS Configuration Commands

The DIS Protocol Status Window indicates the state of the DACS connection to the DIS network. In normal use, the user definable fields will be defined in the configuration file. The configuration file commands are given below.

Local Address

Gives the IP address of the DACS for DIS purposes. This address does not need to be the same as the IP address used in the host interface.

Protocol Version

DIS protocol being used. See the Protocol configuration command.

DIS Ports RX/TX

IP ports for sending and receiving DIS packets. See the UDP_Port, TX_UDP_Port, and RX_UDP_Port configuration commands.

Local ID Site and Host

Determines the DIS site and host numbers for the DACS. These numbers are usually the last two numbers of the IP address (e.g. for IP address 192.42.172.186, the site would normally be 172 and the host 186). The site and host set here determine the site and host numbers for the DIS information from the DACS. An object in a model can use a different site and host ID through a connection to the Entity control object.

Subnet Mask

See the configuration command Subnet_mask.

DIS TXPDUs rx/tx

Indicates the number of transmitter PDUs that have been sent and received. A transmitter PDU contains the transmitter information of the radio broadcasting, with links to the signal PDUs.

DIS SIGPDUs tx/rx

Indicates the number of signal PDUs being sent and received. The signal PDUs contain the actual voice information.

DIS Packets tx/rx

Total number of DIS packets sent and received.

DIS RXPDUs

Indicates the number of receiver PDUs being sent and received. DIS receivers send out PDUs with their world position to aid in monitoring the DIS exercise.

DIS RX Errors PDU/ck

Indicates the number of bad DIS packets received. ck indicates the number of packets received with bad checksums.

UDP packets tx/rx

Total number of packets sent and received.

ARP Reply rx/tx

Indicates the number of ARP requests for an Ethernet address received and replied to.

ARP Request rx/tx

Indicates the number of ARP requests for an EEthernetthernet address that have been sent out, and the number of replies received.

Receive Counters

Various counters for monitoring DIS network activity and problems. RX byte count is the number of bytes of data received over DIS. The RX good count is the number of good packets received over DIS. The rest of the counters indicate errors.

Transmit Counters

Various Counters for monitoring DIS network activity and problems. The TX byte count is the number of bytes of DIS data sent out over the DIS network by the DACS. The good frame count indicates the number of good packets sent out over DIS.

The TX collisions counts the number of times that the DACS and another computer attempted to start sending a packet at the same time. If this is incrementing a lot it may indicate the network is heavily loaded. Occasional collisions are normal and to be expected.

DIS Options Window

```
┌─ D.I.S. Protocol Options Window ──────────────────────────────────────────────┐
  Protocol Version : 4    PDUs/Pkt: Multiple
  PDU TimeOut (sec): 5    Moving  : 2
  PDU Posn Delta(m):  500.000000
  RX Sample Delay  : 160

  MuLaw Samples Max: 480       Min: 200
  CVSD  Samples Max: 1600      Min: 800

  DIS Exercise RX1@: -1   Exercise: 0
  CVSD Encode Cecom: 2        CCTT: 7
  UDP Check Sum  tx: Generate   rx: Verify
  Non Zero Pad     : Invalid
  Zero Entity ID   : Invalid
  Track TX PDUs    : All


  DIS TXpdus     tx: 0000 rx: 0000      DIS SIGpdus    tx: 0000 rx: 0000.0000
  DIS packets    tx: 0000 rx: 0000      DIS RXpdus     tx: 0000 rx: 0000
└─ Esc-exit  PgUp/PgDn-page 1of2────────────────────────────────────────────────┘
```

```
┌─ D.I.S. Protocol Options Window ──────────────────────────────────────────────┐
  Protocol Version : 4    PDUs/Pkt: Multiple
  PDU TimeOut (sec): 5    Moving  : 2
  PDU Posn Delta(m):  500.000000
  RX Sample Delay  : 160

  MuLaw Samples Max: 480       Min: 200
  CVSD  Samples Max: 1600      Min: 800

  Freq  txPDU Count: 0     Free:64    of 64    Miss:0     Err:0
  DIS   txPDU Count: 0     Free:64    of 64    Miss:0     Err:0    Signals: 0
  Local txPDU Count: 0     Free:64    of 64    Miss:0     Err:0
  DIS   rxPDU Count: 0     Free:64    of 64    Miss:0     Err:0
  Local rxPDU Count: 0     Free:64    of 64    Miss:0     Err:0
  RXTXpathPDU Count: 0     Free:64    of 64    Miss:0     Err:0
  DIS   enPDU Count: 0     Free:64    of 64    Miss:0     Err:0    Search: 0

  DIS TXpdus     tx: 0000 rx: 0000      DIS SIGpdus    tx: 0000 rx: 0000.0000
  DIS packets    tx: 0000 rx: 0000      DIS RXpdus     tx: 0000 rx: 0000
└─ Esc-exit  PgUp/PgDn-page 2of2────────────────────────────────────────────────┘
```

PDUs/Pkt

This can be set to single or multiple. See the configuration command Single_PDUS.

PDU Timeout

Determines the number of seconds a PDU packet is considered valid after it is sent. See the configuration command Time_Out.

Moving

When a transmitter or receiver moves more than a certain amount, it sends out a new transmitter (or receiver) PDU to indicate its new position. If it doesn’t move in the amount of time (in seconds) specified in this field, a transmitter (or receiver) PDU is sent out anyway.

DIS Options Window

The amount of movement required to trigger the sending of a new transmitter or receiver PDU is given by the PDU Position Delta.

PDU Posn Delta

When a transmitter or receiver moves by more than the amount given in this field in one frame, then a new tranmitter PDU will be generated and sent out. If the position change from frame to frame does not exceed this field in the time specified in the Moving field, a transmitter PDU will be sent out anyway.

RX Sample Delay

This determines how long a received signal PDU is held before sound playback is commenced. The time is given in number of samples. This is done to prevent gaps in the sound from occurring due to differences in the amount of time different packets take to travel over the network.

DIS Exercise

Filters incoming DIS PDUs based on the exercise ID. If set to zero, all incoming exercise ID’s are accepted.

This feature is a legacy to earlier versions of model builder. Currently, the exercise ID number is determined for each transmitter and receiver individually in the World Position Object or Entity Object.

DIS Exercise RX1@

If this is a positive number, the exercise ID is taken from the RX1 ethernet packet sent from the host computer.

This feature is a legacy to earlier versions of model builder. Currently, the exercise ID number is determined for each transmitter and receiver individually in the World Position Object or Entity Object.

Zero Entity ID

Determines whether an entity ID of zero is considered valid or invalid.

UDP Check Sum tx/rx

The tx field determines whether a checksum for the DIS packets should be generated or set to zero. The rx field determines whether the checksum of incoming packets should be checked for validity or ignored.

MuLaw/CVSD samples Max/Min

Determine the maximum and minimum number of sound samples each DIS signal PDU generated by the DACS will contain.

See the configuration commands CVSD_Samples and MuLaw_Samples.

CVSD Encode Cecom:

Sets the Encoding type field in the signal PDU for signals sent with Cecom compatible CVSD. The default is 2. This should be set for compatability with other systems using Cecom CVSD.

CVSD Encode CCTT:

Sets the Encoding type field in the signal PDU for signals sent with CCTT compatible CVSD. The default is 7. The default standard is for radios to be CCTT compatible.

DIS Configuration Commands

DIS

Command string; **DIS=**<Mode>

Where <Mode>= **ON** or **OFF**

The Ethernet DIS driver can be disabled by setting DIS = Off.

The default value is **DIS= ON**

DIS:Single_PDUS

Command string; **DIS:SINGLE_PDUS=**<Mode>

Where <Mode>= **ON** or **OFF**

In single PDU mode the DIS driver will only allow a single PDU per Ethernet packet. This mode exists to force compatibility with other manufacturers products that will only accept single PDU/Ethernet packets.With this mode disabled (Off) the DACS will transmit multiple PDUs in a single Ethernet packet when this is appropriate.i.e. when there is more than one active transmitter in the DACS. This mode makes marginally better use of the available bandwidth.

The default value is : **DIS:SINGLE_PDUS=OFF**

DIS:TX_UDP_Checksum

Command string; **DIS:TX_UDP_CHECKSUM=**<Mode>

Where <Mode>= **ON** or **OFF**

If <Mode> = off, no checksum is computed for outgoing packets and the checksum field of outgoing packets is set to zero.

The default value is **DIS:TX_UDP_CHECKSUM = OFF**

DIS:RX_UDP_Checksum

Command string; **DIS:RX_UDP_CHECKSUM=**<Mode>

Where <Mode>= **ON** or **OFF**

If <Mode> = off, the checksum is not computed for incoming PDUs. If <Mode> = on, the checksums of incoming packets are computed and compared to the checksum in the header. If they are different, the packet is rejected.

The default value is **DIS:TX_UDP_CHECKSUM = OFF**

DIS:UDP_Checksum

Command string; **DIS:UDP_CHECKSUM=**<Mode>

Where <Mode>= **ON** or **OFF**

Sets TX_UDP_Checksum and RX_UDP_Checksum to <Mode>.

The default value is **DIS:TX_UDP_CHECKSUM = OFF**

DIS:RX_Delay

Command string; **DIS:RX_DELAY=**<Samples>

Where <Samples>= 160 to 800.

Sets the RX sample delay, in samples. This is the amount of time between when a signal PDU is first received and when playback of the signal begins. The time is given in sound samples.

The purpose of the delay is to prevent gaps in the signal stream due to variations in the packet propagation time across the network.

The default value is **DIS:RX_DELAY** = 160.

DIS:Zero_Entity

Command string; **DIS:ZERO_ENTITY=**<Mode>

Where <Mode>= VALID or INVALID.

Determines whether a zero entity ID is considered valid or not. This only applies to received packets. Any transmitter modeled in Model Builder with an entity ID of zero will not transmit over DIS.

The default value is **DIS:ZERO_ENTITY** = INVALID

DIS:Broadcast_IP

Command string; **DIS_BROADCAST_IP=**<IP Address>

Where <IP Address>= **YYY.YYY.YYY.YYY** Decimal number where each field can be 0 thru 255.

The Broadcast command sets the outgoing PDU destination address.

The default value is: **DIS_BROADCAST_IP=255.255.255.255**

DIS:Local_IP

Command string; **DIS:LOCAL_IP=**<IP Address>

Where <IP Address>= **YYY.YYY.YYY.YYY** Decimal number where each field can be 0 through 255.

This command sets the local IP address of the DACS on the DIS port. This is used as the source IP address for all IP packets on the DIS port.

The default value is : **DIS:LOCAL_IP=0.0.0.0**

Note : 0.0.0.0 is an invalid IP address. For DIS to work this IP address must be set to a specific value.

DIS:UDP_Port

Command string; **DIS:UDP_PORT=**<Decimal Number>

Where <Decimal Number>= **YYYYY** Decimal number in the range 0 thru 655535.

This command sets both the source and destination UDP port addresses.

The default value is **: DIS:UDP_PORT= 6994**

DIS:TX_UDP_Port

Command string; **DIS:TX_UDP_PORT=**<Decimal Number>

Where <Decimal Number> = [YYYY] Decimal number in the range 0 through 65535.

This command sets the destination UDP port addresses This is the port address of the DIS "service" within the destination devices that are used to filter incoming data packets by their UDP destination port address.

The default value is **DIS:TX_UDP_PORT= 6994**

DIS:RX_UDP_Port

Command string; **DIS:RX_UDP_PORT=**<Decimal Number>

Where <Decimal Number> =**YYY,YYY** a decimal number in the range 0 thru 65535.

This command sets the receiver UDP Port address. This is the port address of the DIS "service" within the DACS that will be used to filter incoming data packets by their UDP destination port address.

The default value is **DIS:RX_UDP_PORT=6994**

DIS:Protocol

Command string; **DIS:PROTOCOL=**<Decimal Number>

Where <Decimal Number>= **3, 4** or **5**

3 = DIS 2.0.3

4 = DIS 2.0.4

5= DIS IEEE

This command sets the revision level of DIS standard for inter-operability

The default value is **DIS:PROTOCOL=4**. (DIS 2.0.4)

DIS:Samples

Command string; **DIS:SAMPLES=**<Decimal Number>**,**[<Decimal Number>]

Where <Decimal Number>= **YYY.** Decimal number in the range 200 thru 480 representing, either a fixed value if a single entry, or a maximum value if followed by a second entry.

And where, [<Decimal Number>]= **YYY.** Optional second entry of a decimal number in the range 200 thru 480 representing a minimum value.

This command sets the maximum (and minimum) number of audio samples allowed in the signal PDU. If a single value is entered, the number of audio samples is fixed. This feature allows inter-operability with other manufacturers that do not provide field flexibility. This command is valid for all compression types.

The default values are **DIS:SAMPLES=200,480**

DIS:Mulaw_Samples

Command string; **DIS:MULAW_SAMPLES=** <Decimal Number>**,**[<Decimal Number>]

Where <Decimal Number>= **YYY.** Decimal number in the range 200 thru 480 representing, either a fixed value if a single entry, or a maximum value if followed by a second entry.

And where, [<Decimal Number>]= **YYY.** Optional second entry of a decimal number in the range 200 thru 480 representing a minimum value.

This sets the maximum (and minimum) number of audio samples allowed in the signal PDU for Mulaw compressed signals. If a single value is entered, the number of audio samples is fixed. This feature allows inter-operability with other manufacturers that do not provide field flexibility.

The default values are **DIS:MULAW_SAMPLES=200,480**

DIS:CVSD_Samples

Command string; **DIS:CVSD_SAMPLES=**<Decimal Number>**,**[<Decimal Number>]

Where <Decimal Number>= **YYY.** Decimal number in the range 200 through 480 representing, either a fixed value if a single entry, or a maximum value if followed by a second entry.

And where, [<Decimal Number>]= **YYY.** Optional second entry of a decimal number in the range 200 thru 480 representing a minimum value.

This sets the maximum (and minimum) number of audio samples allowed in the signal PDU for CVSD compressed signals. If a single value is entered, the number of audio samples is fixed. This feature allows inter-operability with other manufacturers that do not provide field flexibility.

The default values are **DIS:CVSD_SAMPLES=200,480**

DIS:Time_Out

Command string; **DIS:TIME_OUT=**<Decimal Number>[,<Decimal Number>]

Where <Decimal Number> = **YYYYY** Decimal number in the range 1 thru 32000.

The first number sets the maximum period in seconds that can elapse before an entity PDU is considered invalid. For example, if a transmitter PDU is timed out, the DACS presents the operator with the audio effects of no received carrier signal at that frequency.

The second number sets the maximum amount of time between new transmitter PDUs being sent out. The transmitter PDU gives all the transmitter frequency, power, and other transmitter information, and is sent out whenever any of the transmitter parameters change or when the transmitter object moves more than the PDU position delta, in meters. If no changes occur before the time out, a new transmitter PDU is sent anyway.

The default value is **DIS:TIME_OUT=5,2** (seconds.)

DIS:RX_Mode

Command string; **DIS:RX_MODE=**<Mode>

Where <Mode> = **OWN_ADDRESS** or,

<Mode> =**BROADCAST** or,

<Mode> = **MULTICAST** or,

<Mode> =**ALL_MULTICAST** or,

<Mode> =**PROMISCUOUS**

OWN_ADDRESS will read only those PDUs with the own-device Ethernet Physical Address in the header of the incoming transmission packet.

BROADCAST will read those PDUs with the own-device Ethernet Physical Address and Broadcast Address in the header.

MULTICAST will read those PDUs with the own-device Ethernet Physical Address, and Broadcast Address, and Multicast addresses in use.

ALL_MULTICAST will read all of the above packets plus all Multicast addresses.

PROMISCUOUS will read <u>every</u> packet and should only be used for debug purposes.

Note. On the DIS Control Page this field is toggled between these settings.

The default value is **DIS:RX_MODE =BROADCAST**

DIS:Site

Command string; **DIS:SITE=**<Decimal Number>

Where <Decimal Number> = **YYYY**.

This sets the Site Address for the own device. The Site Address is the same for all PDUs emanating from the DACS

The default value is **DIS:SITE=1**

DIS:Application, or Host

Command string; **DIS:APPLICATION (**or **Host) =**<Decimal Number>

Where <Decimal Number> = **YYYY**.

This sets the Host or Application Address for all PDUs emanating from the DACS.

The default value is **DIS:APPLICATION=1**

DIS:Exercise

Command string; **DIS:EXERCISE=**<Decimal Number>

Where <Decimal Number> =**YYYY**

This sets the network filter such that only DIS packets for this exercise are received through the network driver. A value of zero allows all exercises through this filter.

Set this to zero if the DACS is to support multiple exercises simultaneously.

Note: This feature is obsolete. Exercise ID should be set individually for each object through the world position object or the entity object.

The default value is **DIS:EXERCISE=0**

DIS:Exercise_Offset

Command string; **DIS:EXERCISE_OFFSET**=<Decimal Number>

Where <Decimal Number> = **YYYY**

The host computer for the simulation entity to which the DACS platform is connected has the ability to remotely set values for Exercise filter. This command identifies the number of bytes by which the Exercise number is offset from location zero in the Ethernet packet transmitted to the DACS from the simulator host.

A value of -1 disables the host control of the exercise filter.

Note: This feature is obsolete. Exercise ID should be set individually for each object through the world position object or the entity object.

The default value is **DIS:EXERCISE_OFFSET**= **-1**.

DIS:Subnet_Mask

Command string; **DIS:SUBNET_MASK=**<IP Address>

Where <IP Address> = **YYY.YYY.YYY.YYY** (Decimal number where each field can be 0 thru 255.)

This command sets the subnet mask bits for determining IP broadcast addresses.

The default is the correct subnet mask for the local IP address based upon device class.

DIS:CVSD_Encoding_Type

Note that this command string is subject to change. Contact ASTi for latest information.

Command string; **DIS:CVSD_Encoding_Type=**<Decimal Number>**,**<Decimal Number>

Where <Decimal Number>= **YYY.** Decimal number in the range 1 thru 255.

This command assigns the DIS Signal PDU audio data encoding type number to the two existing CVSD encoding schemes. The first decimal field assigns the encoding type number for the original CECOM SRM CVSD encoding method. The second field assigns the encoding type number for the CCTT SRM CVSD encoding method.

This feature is required to enable the DACS to interoperate with either the CCTT SRM or the original CECOM SRM as these two systems currently use a different CVSD encoding scheme but use the same value of 2 in the Signal PDU encoding type field.

The DACS uses a value of 2 (as specified in the DIS standard) and an abitrary value of 7 to define and differentiate the second encoding type. 7 is currently an unassigned encoding type number in the DIS standard.

To provide default interoperability with the original CECOM SRM encoding scheme use the following command line:

DIS:CVSD_Encoding_Type = 2, 7

To provide default interoperability with the CCTT SRM CVSD encoding schem use the following command line:

DIS:CVSD_Encoding_Type = 7, 2

The default value is:

DIS:CVSD_Encoding_Type = 2, 7

DIS:CCTT_CVSD_Encoding

Command string; **DIS:CCTT_CVSD_Encoding** = <Mode>

Where <Mode>= **ON** or **OFF**

This command is an alternative to the DIS:CVSD_Encoding_Type = command string. For Interoperability with CCTT versions of the SRM set

DIS:CCTT_CVSD_Encoding = ON

The default value is:

DIS:CCTT_CVSD_Encoding = OFF