Advanced Simulation Technology inc.
500A Huntmar Park Drive
Herndon, Virginia 20170 U.S.A.
Tel. (703)471-2104 • Fax. (703)471-2108
www.asti-usa.com

ASTi
Model Builder Basic
Training Course Manual

DACS

Model Builder

Document: DOC-01-MB-TC-1

Revision B (July 2005)

ASTi ASTi Model Builder Basic Training Manual

ASTi

500 A Huntmar Park Drive

Herndon, VA 20170

Table of Contents

1. Introduction and Agenda . . . 1
1.1 Summary . . . 1
1.2 Course Goals . . . 1
1.3 Lab Layout . . . 2
Figure 1: Training Room Network Layout ... 3
Figure 2: Training Room DACS Layout ... 4

2. Overview of DACS and Distribution Hardware . . . 5
2.1 DACS chassis . . . 5
Figure 3: 2U and 4U DACS ... 5
Table 1: Chassis Types ... 6
2.2 DACS Interface Cards . . . 6
2.2.1 Ethernet Interfaces ... 6
2.2.2 TDM Interface ... 6
2.2.3 8AFA Interface ... 7
2.2.3.1 Sample Rate ... 7
2.2.4 Interconnection ... 7
2.3 Remote Interface Unit (RIU) . . . 8
Figure 4: RIU Schematic ... 8
2.3.1 RIU Serial Ports ... 10
Figure 5: Connection Devices for RIU Serial Ports ... 10
2.4 TDM Ring Architecture (connecting TDM Interface to RUI(s)) . . . 11
Figure 6:TDM Ring with Constituent RIUs ... 11
2.5 DACS Ancillary Equipment . . . 12
2.5.1 Distributed Digital Interface (DDI) ... 13
Figure 7: DDI Units ... 13
Figure 8: Sample DDI Setup ... 13
2.5.2 Hand Held Terminal (HHT) ... 14
Figure 9: Radio HHT Unit ... 14
2.5.2.1 Radio HHT ... 14
2.5.2.2 Intercom HHT ... 14
2.5.2.3 HHT Basics ... 14
2.5.3 High Fidelity Panels ... 15
Figure 10: SINCGARS Radio ... 15
2.5.4 Headsets, Microphones and speakers ... 16
2.5.5 Cables ... 16
2.5.6 PTT Switches ... 17
Figure 11: PTT Switches ... 17
2.5.7 Touchscreen Softpanel ... 18
Figure 12: Touchscreen Softpanel ... 18
2.6 Telestra . . . 19

3. Overview of DACS Software 21
3.1 DACS: Model Builder Software 21
3.1.1 DACS Configuration File (Default.cfg) 21
3.1.2 Options File (.opt) 22
3.1.3 System Configuration File (Config.sys) 22
3.1.4 Path File (.pth) 22
3.1.5 Model File (filename.mdl) 22
3.1.6 Initialization File (.ini) 23
3.1.7 DLL File (.dll) 23
3.1.8 XML File (filename.xml) 23
3.1.9 ICD File (filename.icd) 23
3.1.10 Sound File(s) (name.au) 23
3.1.11 Model Builder Software Diagram 24
Figure 13: Model Builder Software Diagram 24

4. Model Builder Introduction (with hands-on exercises) 25
4.1 Model Builder Basics 25
4.2 Model Builder Status and Monitor Capabilities 25
4.3 Model Builder Screens 26
Table 3: Model Builder Screens 27
4.3.1 Model Builder Configuration Window 28
Figure 14: Model Builder Configuration Window 28
4.3.2 Model Builder Options Window 29
Figure 15: Model Builder Options Window 29
4.3.3 Model Builder Status Page 29
Figure 16: Model Builder Status Page 29
4.3.4 DIS Network Status Page 30
Figure 17: DIS Network Status Page 30
4.3.5 DIS Network Frequency of Transmitters 31
Figure 18: DIS Network Frequency of Transmitters 31
4.3.6 Model Builder Ethernet Control Window 32
Figure 19: Model Builder Ethernet Control Window 32
4.3.7 Model Builder Waveform DSPs Window (s) 33
Figure 20: Model Builder DSPs Window 1 33
4.3.8 Model Builder RIU Status Page 34
Figure 23: Model Builder RIU Status Page 34
4.4 Navigating in Model Builder 35
4.4.1 Function keys: 35
4.4.2 Moving through the menus 36
4.4.3 Accessing Objects in a List (Controls, Signals, Feeders, etc.) 37
4.4.4 Other Model Builder conventions 37

4.5 Creating, Saving and Restoring Models 38
4.5.1 Creating Models 38
Figure 24: Model Loading Screen 38
4.5.2 Saving Model 38
Figure 25: Save Model Screen 38
4.5.3 Restoring Models 39
Figure 26: Restoring Model Screen 39
4.6 Model Builder Object Types 40
4.6.1 Signal Objects 40
Figure 27: Signals Menu 40
4.6.2 Control Objects 41
Figure 28: Controls Menu 41
4.6.3 Feeder Objects 42
Figure 29: Feeder Objects 42
4.6.4 Functions Objects 43
Figure 30: Functions Objects 43
4.6.5 Sound File Objects 43
4.7 Hands-on Exercises 44
4.7.1 Generating and Routing Audio 44
Figure 31: Sine Object Screen 44
4.7.2 Generating and Routing Audio (continued) 45
Figure 32: Balancer Object Screen 45
Figure 33: Mixer Object Screen 45
4.7.3 Audio Input Interface 46
Figure 34: VOX Settings 46
4.7.4 Local Controls and Digital Inputs 47
Figure 35: RIU Controls 47
Figure 36: RIU Digital Inputs 47
4.7.5 Host Interface Example 48
Figure 37: Setting Host Controls 48
Figure 38: Setting Output Control 48
Figure 39: Setting the Receive Buffer 49
4.7.6 Review of Chapter 4 models 50
Diagram 1: 4.7.1 - Sine Wave Output 50
Diagram 2: 4.7.2 - Multiple Sine Waves Output to Different Highways 50
Diagram 3: 4.7.3 - Audio In/Audio Out via VOX Object 51
Diagram 4: 4.7.4 - Audio In/Audio Out via VOX Object and PTT 51
Diagram 5: 4.7.1 - 4.7.4 - Overall View of Previous Four Exercises 52
Diagram 6: 4.7.5 - Host Input Example with Boolean Input and Output 52

5. Networking DACS (DIS and Host Interface) 53
5.1 Introduction 53
5.2 DIS Interface 53
5.2.1 DIS Radio Basics 54
Figure 40: DIS Protocol Status Window *54*
5.2.2 DIS PDU Types 55
5.2.2.1 TX PDU 56
5.2.2.2 Signal PDU 56
5.2.2.3 RX PDU 56
5.2.2.4 Entity State PDU 57
5.3 Host Interface 57

6. Model Builder Advanced (with hands-on exercises) 58
6.1 Radio Communications 59
6.1.1 Radio Essentials 59
Figure 41: Radio Object Screen *59*
6.1.2 Radio Communication Guidelines 60
6.1.3 Crypto radios 60
6.1.4 Frequency Hopping radios 61
6.1.5 Communications Panel Object 62
Figure 42: Communication Panel Object *62*
6.2 Intercom Communications 63
6.2.1 Local Intercoms 63
6.2.2 Network Intercoms 63
Figure 43: Intercom Settings *63*
Figure 44: Network Intercom Screen *64*
6.3 Hands-on Exercises 65
6.3.1 Basis Radio Modeling 65
6.3.2 Crypto Radio Modeling 65
6.3.3 Frequency Hopping Radio Modeling 66
6.3.4 Using a Radio with the Communications Panel 66
6.3.5 Intercom Modeling 66
6.3.6 DIS Networking Modeling 67
6.3.7 Review of Chapter 6 Models 68
Diagram 7: Exercises 6.3.1 - 6.3.3 - Radio Model *68*
Diagram 8: Exercise 6.3.4 - Communication Panel for Operators 1 and 2 . . . . *69*

7. Troubleshooting with DACS and Model Builder 70
7.1 Startup Errors or Missing Screens/Functions in Model Builder 70
7.2 Model Issues 70
7.3 TDM/RIU Issues 71
7.4 DIS Interface Issues 71
7.5 Host Interface Issues 72
7.6 Radio/Intercom Reception and Transmission Issues 72
7.7 Voice Break-up Issues 72

8. Customer Specific material (if applicable) 73

9. Glossary of Terms and Acronyms 74

10. Class Notes 79

1. Introduction and Agenda

1.1 Summary

The heart of the ASTi Digital Audio Communication System (DACS) is the Model Builder sound model development toolkit. This software application transforms the DACS into a comprehensive development workstation for the creation, extension and tuning of sophisticated audio simulation models.

The ASTi Model Builder toolkit provides a simple “database-like” interface to sound model generation. Model configuration, interface, and model parameters can be changed quickly and easily in real time. No DSP Assembler or high-level language software knowledge is required to create or modify the model.

This training course will familiarize you with the layout of Model Builder and its menus, as well as the related hardware and its uses.

1.2 Course Goals

At the completion of this course you will understand how to:

- Set up a DACS communication ring (TDM ring)
- Configure Model Builder to interface with the TDM ring and all ancillary equipment
- Configure a DACS for control by a host PC
- Modify configuration files to ease startup and operation
- Open existing, create new, modify and save models
- Easily navigate the hierarchical Model Builder menus
- Build sound models using control, signal, feeder, and function objects
- Troubleshoot models using diagnostic screens and the Model Builder utilities

1.3 Lab Layout

All the equipment needed for the training course is provided in the ASTi training room. This includes:

- (2) DACS units, one 2U and one 4U both with Model Builder installed
- (2) Remote Interface Units (RIUs)
- (2) Telex headset/microphone units*
- (2) PTT buttons (one in-line, one foot-operated)
- (2) Fostex 6301B powered loudspeakers*
- (1) ASTi Telestra server for RMS network control
- (1) PC with Linux OS* for use with RMS
- All required PC peripherals and network hardware

*This is just what is in the training room. Other headset/mic units, loudspeakers and operating systems for the RMS PC will work just as well. Ask ASTi any questions you may have.

Two diagrams of the equipment connections in the training room are shown below. The first shows the Network Layout and the 2nd shows the DACS, TDM ring and peripherals.

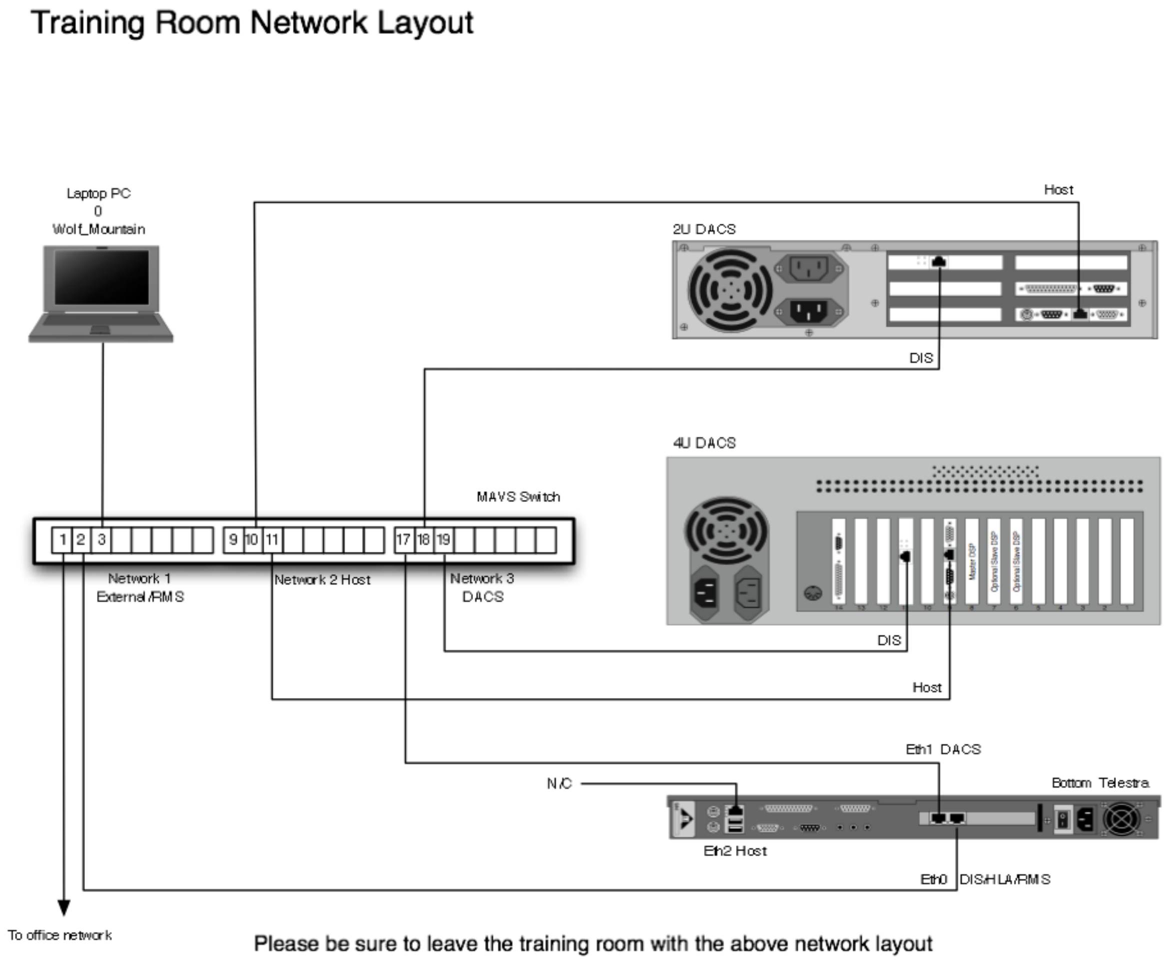


Figure 1: Training Room Network Layout

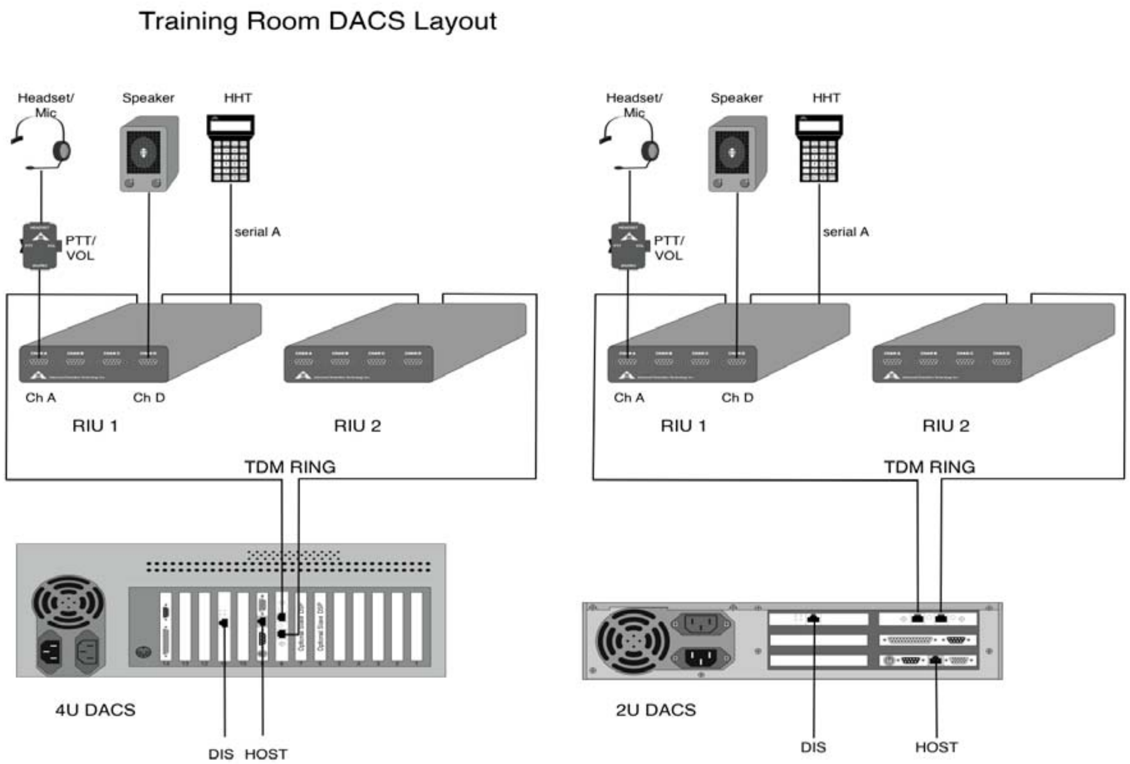


Figure 2: Training Room DACS Layout

2. Overview of DACS and Distribution Hardware

2.1 DACS chassis

There are 2 types of DACS hardware chassis:

Figure 3: 2U and 4U DACS

The 2U (top) and 4U (bottom) DACS processing nodes consist of a number of hardware components, and can be configured in many different ways depending on the requirements of the particular application. Components available include the main model processor, various audio processors such as Waveform Synthesizers, TDM controllers, Remote Interface Units, Distributed Digital Interfaces, network interfaces such as the Host Interface and DIS/HLA Interface, and general purpose I/O including Digital Inputs, Digital Outputs, and Analog Inputs.

The various components of a DACS are listed below:

- **Power Supply** - These vary slightly between platforms. Some are auto-sensing and some have a manual switch. The supply accepts 120V/60Hz, and can also be configured to accept overseas voltages such as 220V/50Hz.
- **Digital Signal Processing (DSP) Cards** - The number and type vary from DACS to DACS. Available types are Waveform Synthesizer or TDM
- **CPU/Memory Type** - These vary from DACS to DACS. Current systems (as of 07/18/05) are shipping with Intel 850 Mhz processors and 32M of RAM.
- **Hard Drive** - On most DACS, the hard drive is a fixed drive of at least 7GB of disk space. An optional removable drive is also available.
- **Floppy Drive** - A 3 1/2” Disk drive is available on all DACS units
- **Ethernet Interfaces** - All DACS include Ethernet interface on the motherboard. An additional network card is also available to provide second Ethernet interface.

The type of DACS chassis you purchase will vary based on your application. The matrix below helps to summarize some of the available configuration options for the various chassis types:

Chassis	#DSPs	Drive	Ethernet Interfaces
2U	1-2	Fixed or removable	1-2
4U	1-4	Fixed or removable	1-2

Table 1: Chassis Types

2.2 DACS Interface Cards

- (1) or (2) Ethernet interfaces
- At least (1) DSP card
- TDM (and associated RIUs)
- Waveform Synthesizer (a.k.a. 8AFA)

2.2.1 Ethernet Interfaces

The DACS comes standard with a DIS (Distributed Interactive Simulation) network interface card (NIC), which is used for voice traffic (radio, intercom, etc.) to and from other DACS or simulators on the network. An optional Host interface can be purchased to control state information such as frequencies, squelch, engine RPM, etc. These two interfaces will be discussed more in section 5. The host control and voice traffic functionality can be combined onto one interface, if the traffic load is fairly low and permitted under security guidelines.

2.2.2 TDM Interface

The TDM Controller forms a base driver for a Time Division Multiplex ring. This ring carries digital audio, local I/O and peripheral device data to and from Remote Interface Units (RIUs) populating the ring. The RIUs in turn provide the analog-to-digital interface at remote sites along the ring. See the next section for more information on the RIU. The TDM Controller has two RJ-45 connectors for connection of standard Category 5, UTP network cables. To form the TDM ring, cabling is run from the TDM controller card to the first RIU. Subsequent RIUs are daisy chained together with the ring eventually terminating back at the TDM Controller. This is explained in detail in section 2.4.

2.2.3 8AFA Interface

Waveform Synthesizers (8AFA) perform sound model digital signal processing and provide analog input and output interface for external equipment such as speakers, headsets, and microphones.

Each individual Waveform Synthesizer contains a single DSP, and performs all the signal processing associated with a specific, user-developed sound model. The Waveform Synthesizer provides up to eight channels of analog inputs and outputs (i.e., eight inputs and eight outputs) for interfacing to external audio equipment. Converters are 16 bit delta sigma with sample rate tracking anti-alias filters which provide excellent sound quality. The Interface to external audio equipment such as headsets, mics, amplifiers, etc. is via a 37 pin, female “D” connector located in the rear of the chassis.

2.2.3.1 Sample Rate

The user, through the Model Builder software environment, sets DSP sample rates on the Waveform Synthesizer. Sample rates range from 8kHz to 48kHz. Since the anti-aliasing filters are sample rate tracking, no hardware reconfiguration is required when sample rates are changed.

2.2.4 Interconnection

A high speed communication port connection is provided on the TDM and Waveform Synthesizer cards. This port provides the capability to link up to three cards together, enabling sounds and communications generated on one processor to be accessed by or mixed with the sounds and communications generated by another processor. The port connections all connect to what is know as the *global bus*. An example configuration is the connection of two Waveform Synthesizers. Connection of the two cards allows audio inputs and sounds generated on one card to be mixed into outputs on the other card. This high-speed communications link between Waveform Synthesizers is made through a 40 pin header on the top of each card or module. For multiple card-to-card connections, one 40 conductor ribbon cable is installed and connected to all cards that need to pass sound between them. Only cards of the same type (TDM-to-TDM or 8AFA-to-8AFA) are connected together using this high speed port. You can, however, have different card types in a DACS, they just cannot be connected via this global bus.

2.3 Remote Interface Unit (RIU)

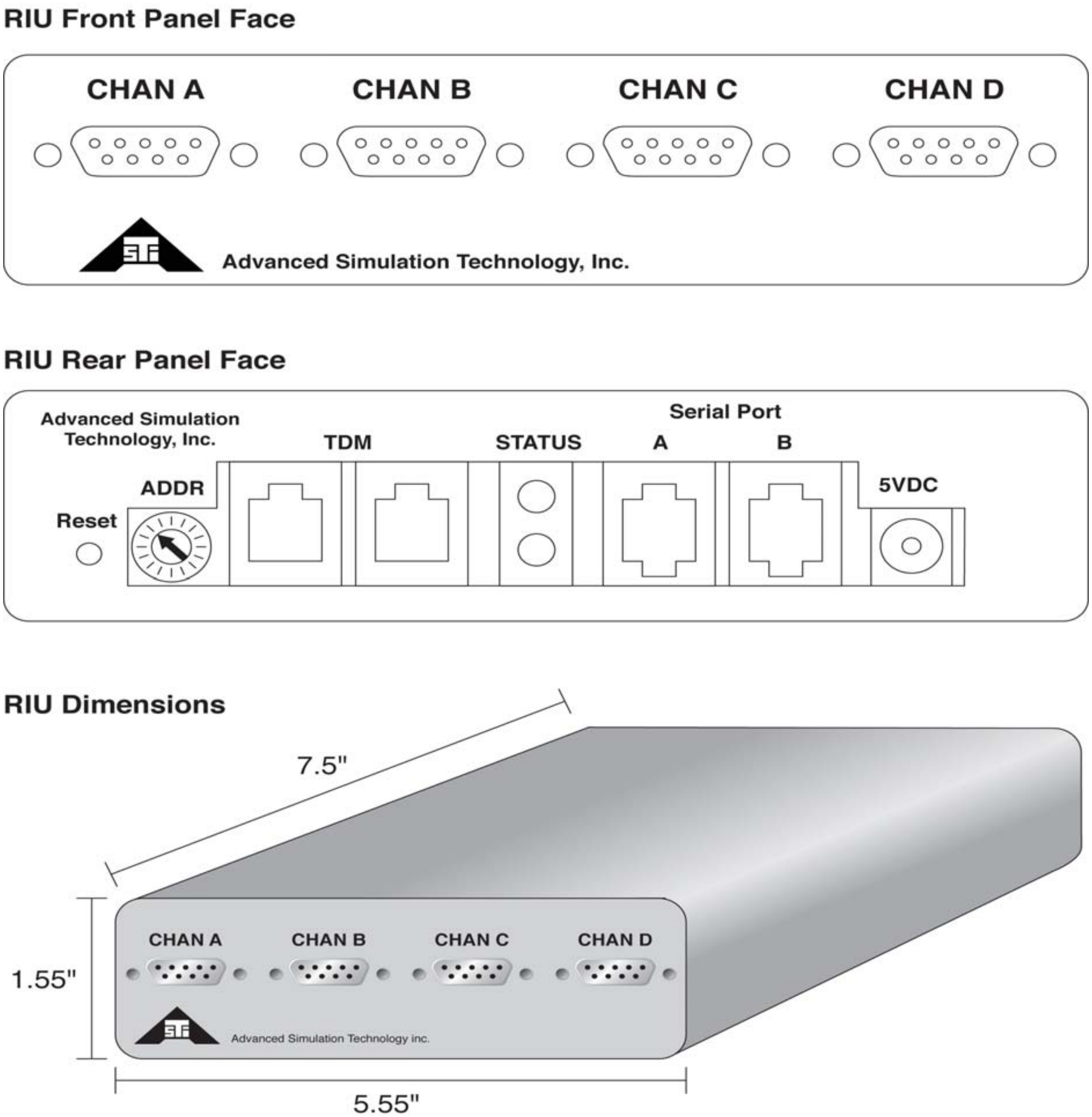


Figure 4: RIU Schematic

The RIUs are used in conjunction with TDM Controller cards to form a TDM-based digital communications network. RIUs are daisy chained together using standard category 5 UTP network cable to form a “TDM Ring”. Both ends of the ring are connected to a TDM Controller card located in the main DACS. Audio is sampled by the RIU and passed in digital format to the TDM Controller. Audio mixed and/or generated by the TDM Controller is also passed in digital format to the RIU. The RIU provides both audio input/output capability and some limited digital I/O capability. In addition, two serial ports are provided for connection of devices such as Hand Held Terminals and simulated or actual operator panels.

The RIU has 4 channels (labeled A, B, C and D). Each channel has an analog audio In/Out and a Digital In/Out as follows:

- Audio input circuit
  - Adjustable gain from 0 to 60 dB via hardware jumpers
  - Available +/- 5 V phantom power
  - Balanced input with 10 kΩ input resistance
  - Capacitively coupled input stages
- Audio output circuit
  - Balanced or single-ended operation
  - Power output:
  - 1 W RMS into 8 Ω balanced load
  - 250 mW RMS into 8 Ω unbalanced load
  - Output impedance typically 0.1 Ω into an 8 Ω load
- Digital input circuit
  - Contact-sensing, no power needed
  - Useful for PTT control
- Digital output circuit
  - Consists of an opto-isolate, solid state relay for switching power to external loads.
  - Maximum continuous current rating of 120 mA
  - Maximum power dissipation rating of 180 mW

See RIU v4.1 Technical Guide (ASSY-01-RIU04-MN-2, Rev. E) for complete RIU details.

2.3.1 RIU Serial Ports

Allow connection of user interface devices such as:

- High-fidelity simulated panels (i.e., SINCGARS)
- Hand-held terminals
  - Radio
  - Intercom
- Distributed Digital Interface (DDI)
  - Allows connection of discrete control circuits to Model Builder
  - Includes digital INs and OUTs, analog IN, keypad interface and display interface
  - Comes in single and double versions

Figure 5: Connection Devices for RIU Serial Ports

2.4 TDM Ring Architecture (connecting TDM Interface to RUI(s))

The TDM ring carries the digital audio and signals from the RIUs to the TDM DSP on the DACS. When creating a TDM ring there are 2 or more components:

- TDM DSP Card
- 1-14 RIUs

An example of a TDM ring is shown below:

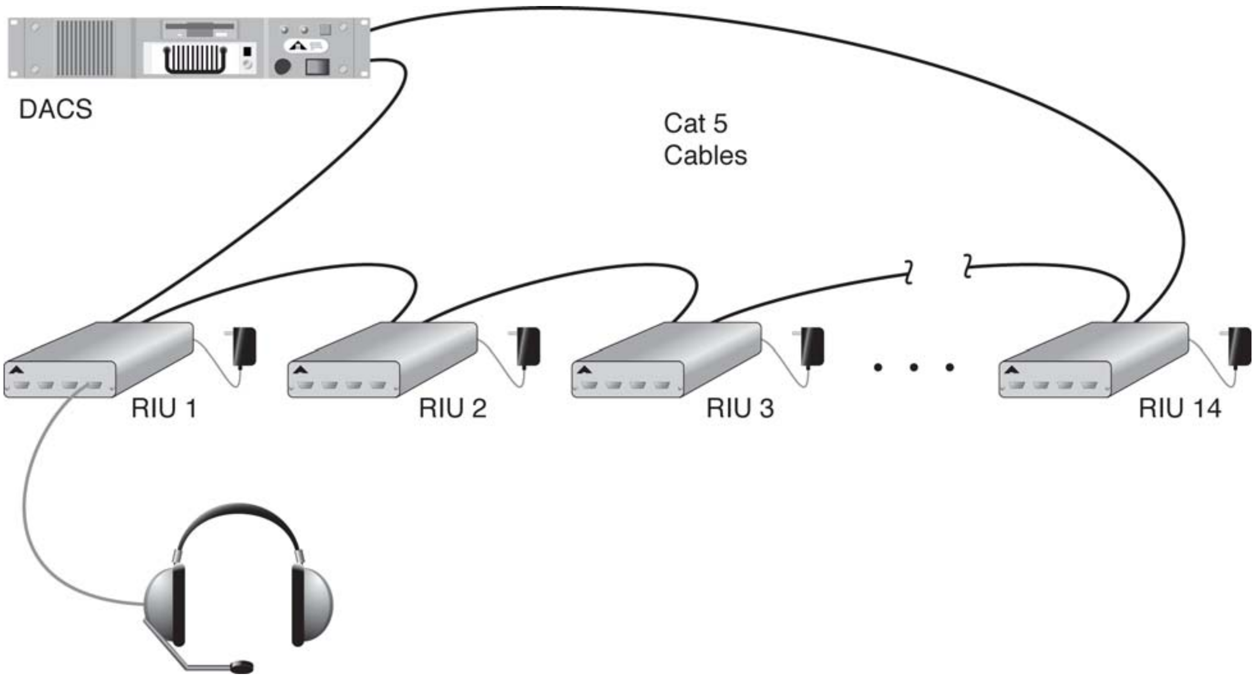


Figure 6:TDM Ring with Constituent RIUs

The TDM ring has 30 data slots available. Each RIU occupies one slot, and each independent audio channel occupies one slot. As the number of RIUs on the ring increases, the number of available independent audio channels increases. This translates to fewer operators per RIU as the TDM ring grows. The ASTi standard convention is to have 2 operators per RIU. The table below shows the correlation between the number of RIUs on a ring and the number of audio channels available per RIU:

Number of RIUs on TDM Ring	Number of Available Audio Channels per RIU
1-6	4
7	3
8-10	2
11-14	1

Table 2: RIU Numbers

Ring length is a function of the type of TDM card in the DACS, the sample rate of the model and the type of TDM cabling used. The TDM type is indicated by the version number listed in one of the Model Builder windows. Using Standard CAT 5 cable you can achieve a maximum TDM ring length of 500 feet. This assumes an audio sampling rate of 8 kHz. Higher frequencies will further limit the ring length.

2.5 DACS Ancillary Equipment

In addition to the DACS hardware and RIU there are several pieces of peripheral equipment that connect to the DACS system. These include:

- Distributed Digital Interface (DDI) units
- Hand held terminals
- High fidelity panels
- Headsets, microphones and speakers
- Cables
- PTT switches
- Touchscreen Softpanel

Below is a more comprehensive explanation of some of the peripheral equipment available.

2.5.1 Distributed Digital Interface (DDI)

This module connects directly to an RIU's serial port. DDIs are available in the two sizes shown below. The single-height DDI can support up to 8 digital inputs and 8 digital outputs while the double-height DDI can support 32 digital inputs, 8 digital outputs, and 4 analog inputs. Up to 7 DDIs may be connected to each RIU serial port for a maximum of 14 DDIs per RIU.

Figure 7: DDI Units

System expansion is easy using the DDI. For more connections to user hardware, simply snap another DDI into an RIU. A typical DDI Setup is shown below:

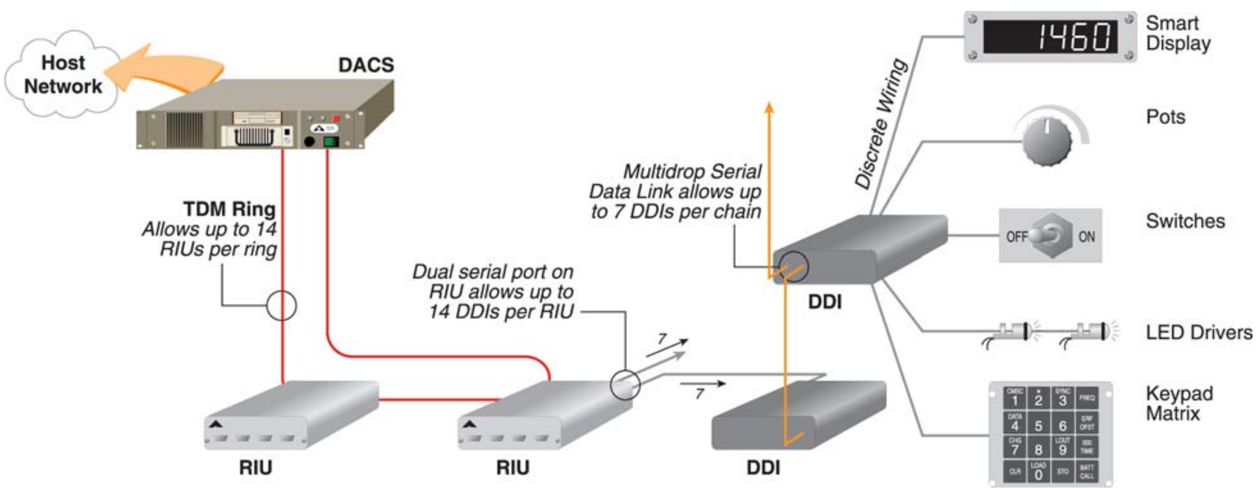


Figure 8: Sample DDI Setup

2.5.2 Hand Held Terminal (HHT)

Figure 9: Radio HHT Unit

The ASTi Hand Held Terminal system provides a highly flexible solution to multi-operator simulation requirements. There are two types of HHTs: the Radio version and the Intercom version.

2.5.2.1 Radio HHT

The Radio HHT unit provides the operator interface to the simulated radios. The functionality is designed to provide an intuitive display interface into a multi-radio environment and provide access to the key parameters of the individual radios themselves. The Radio HHT provides access to the key parameters of the individual radios such as frequency/net, radio volume, secure mode, as well as individual operator settings such as receive/transmit controls, master volume, and sidetone volume.

An initialization file also provides customization of default operator and system parameters.

2.5.2.2 Intercom HHT

The Intercom HHT is generally part of a preconfigured intercom system providing a customer-specified number of operators and communications buses.

The functionality is designed to provide an intuitive display interface into a multi-bus environment. The Intercom HHT provides access to the key parameters of the individual buses such as talk/listen settings, volume, duplex modes as well as individual operator settings such as master volume, sidetone volume and VOX/PTT mode.

An initialization file also provides customization of default operator and system parameters.

2.5.2.3 HHT Basics

The HHT interfaces to the serial port on the RIU, with the RIU–in turn–connecting to the DACS via the TDM ring. The initial settings for the HHT are defined in the .ini file, which resides on the DACS. This file sets up such things as: which Operators have an HHT, radio access for HHT (Tx, Rx, Tx_Rx), access rights to radio (can you change freq, etc.), radio frequencies, etc. The path file (.pth) on the DACS defines the routing between the serial interfaces. That is the mapping between the Operator, RIU and serial port.

Once the DACS Model Builder software is loaded, there are state machine (DLL) instances running for the HHTs to function correctly. For example:

If you press the volume button “+” on the HHT twice, the state machine (DLL) on the DACS gets the message via a pre-defined input port. (e.g., R1, byte 0). The DACS takes this input and applies it accordingly throughout the model where applicable. The modified output is then sent from the DACS. In this case, the gain on the RIU output is increased, and the volume number is adjusted by +2 on the HHT display.

2.5.3 High Fidelity Panels

In providing turnkey system solutions for our customers, ASTi can provide a variety of high-fidelity, simulated radio control panels.

From a complex radio such as SINCGARS to simple controls like an Action Alarm button, our custom panels are configured to meet your specifications. Whether you need a generic intercom control panel, or a fully operational simulated aircraft radio, our panels are your solution.

Not only do ASTi's custom panels look like the actual equipment, they operate just as fielded equipment, too. Working in conjunction with ASTi's Model Builder software, each panel is monitored by its own state-machine program. When you change a setting, or flip a switch, the associated state machine relays the panel's state to the Model Builder software, and the necessary effects (such as "Radio On," "Volume Change," or "Press-to-Talk Activated") are performed by the DACS platform immediately and efficiently to keep network bandwidth at a minimum.

Figure 10: SINCGARS Radio

The functionality for the high-fidelity, simulated SINCGARS panels is built directly into our Model Builder software. The Model Builder software responds to every switch setting, change and parameter with the necessary output. Analog sound output is modified to match the "switchology," intercom nets are subscribed to or excluded, and the panel's display shows the same information as the real SINCGARS radio.

2.5.4 Headsets, Microphones and speakers

ASTi offers commercial headsets and microphones for use with the DACS system. Our current offerings include (not a complete list):

HS-TX-PH-2R	Commercial, dual enclosed earcup, Telex
HS-TX-PH-44R	Commercial, dual lightweight earcup, Telex
HS-TX-PH-88R	Commercial, single lightweight earcup, Telex

While there is nothing special about these headsets, they have been tested with ASTi equipment and are known entities; that is, the voice and sound quality are known, where they are not always known with non-standard equipment. If using a non-standard headset, it is recommended that the end user gets ASTi to evaluate their headset.

Additionally, ASTi can provide headset and audio peripheral evaluation services either via support contract or as a service. In this case, ASTi will evaluate and test the customer's selected headset or audio peripheral with ASTi equipment, and generate a report listing optimum configuration and cabling recommendations and test results. The customer is required to provide headset/peripheral for evaluation, and a mating connector, if necessary.

ASTi currently offers a rugged, compact powered monitor speaker for use with the DACS hardware: ASTi part number A-PS.

Refer to the ASTi website (www.asti-usa.com) for pricing and ordering information.

2.5.5 Cables

ASTi offers a large assortment of cables to interconnect all of the DACS equipment. There are a variety of cables for the RIU to connect to a PTT, HHT, customer panel, headset, microphone, etc. Additionally, we offer a range of Waveform Synthesizer and Miscellaneous Cables. For a complete list of current cables, visit the ASTi website. Custom audio interface and cabling solutions are also available to meet specific applications; contact ASTi for details.

2.5.6 PTT Switches

ASTi carries a variety of PTT switches with differing functionalities from simple "nurse call" buttons to in-line PTTs with volume control or channel select that connect directly to the headset. PTT foot switches are also available. ASTi can even create realistic hand-held PTTs to match your high-fidelity simulated radio panel.

Figure 11: PTT Switches

2.5.7 Touchscreen Softpanel

The Touchscreen Softpanel features:

- Rugged industrial components for greater reliability
- Programmable displays and keys allowing customization to user application
- Color screen for improved visual clarity and operation
- Ethernet port for direct connection to local network
- Large display for increased flexibility, clarity, and usability

Figure 12: Touchscreen Softpanel

2.6 Telestra

The Telestra is ASTi's next generation communications system, and is based on the Linux OS. Several different software modules may be installed on the Telestra platform. The base software package comprises the Remote Management System (RMS). With ASTi's Telestra RMS and a simple web browser, the user can monitor and manage local and networked ASTi resources (equipment, software, models), ranging from stand-alone to multi-site, exercise-wide configurations. The familiar point-and-click, web-based interface makes RMS easy to use.

RMS allows the user to:

- Quickly ascertain status of Telestra and DACS equipment along with all supporting ASTi hardware locally and across the network
- View and load model and configuration files locally and remotely over the network
- Remotely backup and restore files
- Navigate Telestra and DACS file systems, download and upload individual files for archive, redistribution, and easy file allocation
- Monitor system hardware, software, models, configuration, status and performance
- Remotely manage most network set-up and configurations
- Remotely restart/reboot systems View online error and status reports
- View equipment and software version
- Configure HLA communications environment parameters; upload and install RTI, FED, RID and conversion files for HLA; specify Federate settings
- Monitor, configure, check status and server-query SATCOM, HF, ALE and Terrain Database services products

RMS also includes the following additional applications:

- Model Builder Model Documentation Tool: generates a click-through, interactive PDF format document with graphical layout of any DACS model, host ICD, unconnected objects report, audio I/O layout and more
- Model Builder Virtual Screen Utility: Remote terminal feature provides remote viewing and modification of system application software environment and user models for up to 12 DACS units per Telestra system (feature accessible via direct video and keyboard connection to the Telestra unit only)

Additional features:

- Password protection
- Multicast routing capability for network traffic management
- Searchable, contextual help system with troubleshooting suggestions
- Online "Getting Started" tutorials for RMS and HLA

Other software packages that may be installed on Telestra include:

- High Level Architecture (HLA)
- High Frequency (HF) Environment Server
- Automatic Link Establishment (ALE) Server
- Terrain Database Server
- SATCOM Modeling Server
- Model Builder Visual (Graphical version of Model Builder)

3. Overview of DACS Software

3.1 DACS: Model Builder Software

The DACS runs on a DOS operating system, with ASTi's Model Builder Runtime and Development Environment software being the heart of all audio processing. All work done in Model Builder is done in real-time, without any compiling or programming knowledge required. Model Builder requires certain files to operate. A list of these files is shown below:

Mandatory File Types:

- Default.cfg
- filename.opt (filename is project specific)
- Config.sys
- filename.mdl

Supporting File Types:

- .pth
- .ini
- .dll
- .au

Model Builder Generated File Types (automatically created during Model Builder save):

- filename.xml
- filename.icd

Details of the file types above are shown below in sections 3.1.1. through 3.1.10.

3.1.1 DACS Configuration File (Default.cfg)

Through the *.cfg configuration file you can:

- Set the master and slave designations in multi DSP platforms
- Designate the default models to load at startup
- Set DIS and Host interface networking parameters
- Set Model rates and DSP gains
- Enable cell communications for HHT or simulated radio panel operation
- Enable communication features such as Terrain and Tactical Data Link

The default.cfg file is stored in the following directory:

C:\MBUILDER\USER\MODELS

3.1.2 Options File (.opt)

The Options file is used to control the functionality of the DACS by specifying the number of credits and DSPs available. The options file is unique for each project, and keyed to the MAC address of the Host NIC.

The .opt file is stored in the following directory (note that the name of the file is based on the project name):

C:\MBUILDER\BIN

If more than one options file is present, then all options files are loaded, and Model Builder picks out the relevant bits. A single options file may contain the keys for several DACS.

3.1.3 System Configuration File (Config.sys)

The Config.sys file will set certain parameters on the DACS such as:

- Displays boot menu
- Enable/Disable RMS
- Network Settings (needed for RMS)
- DOS Settings

The config.sys file is stored in the following directory:

C:\

Note: This file is only modified when the DACS is under RMS control.

3.1.4 Path File (.pth)

This file controls the routing of cell data to/from RIUs, models and a state machine. This file is only required when an input/output device is connected to an RIU serial port. In general, the standard path files do not need to be modified. Contact ASTi for assistance if it becomes necessary to modify this file.

Standard path files for radio and intercom HHTs and SINCGARS DLLs are found in:

C:\MBUILDER\BIN

3.1.5 Model File (filename.mdl)

This file is created when you save a model within Model Builder. The .mdl file contains operator information, radios, intercoms, play sounds, etc. The various elements that make up a model will be covered later in this training course.

3.1.6 Initialization File (.ini)

This file is necessary when using HHT or SINCGARS and is used for:

- Radio Mapping of Operators
- Operator and Radio Names
- Initial Tx_Rx settings
- Initial Volume, Squelch and Sidetone settings
- Initial radio frequencies
- DIS parameters such as site, host, entity and radio IDs

Example files are found in:

C:\MBUILDER\BIN

File in use is found in:

C:\MBUILDER\USER\MODELS

3.1.7 DLL File (.dll)

This file is necessary when using HHTs or SINCGARS and provides the state machine or "brains" for these devices. The DLLs are installed with Model Builder and are invoked as necessary.

3.1.8 XML File (filename.xml)

Extensible Markup Language file, which is a representation of the model and Interface Control Document (ICD). It is used in conjunction with DocTool in Telestra RMS. It is created when a model is saved in newer versions of Model Builder. When used in conjunction with the DocTool, it helps the end user create a configuration map of the communications model.

3.1.9 ICD File (filename.icd)

The Interface Control Document file defines and controls input offsets, data types and UDP port number, etc. This file is created when the model is saved.

3.1.10 Sound File(s) (name.au)

These are sound files that are stored on the hard drive of the DACS, and can be referenced to play the specified sound on demand within the model. Sound files are stored in different directories based on their sampling rate. Below are the specified directories:

C:\MBUILDER\USER\SOUNDS8

C:\MBUILDER\USER\SOUNDS16

C:\MBUILDER\USER\SOUNDS22

3.1.11 Model Builder Software Diagram

The diagram below shows the overall inputs, outputs and flows that occur within the model builder software.

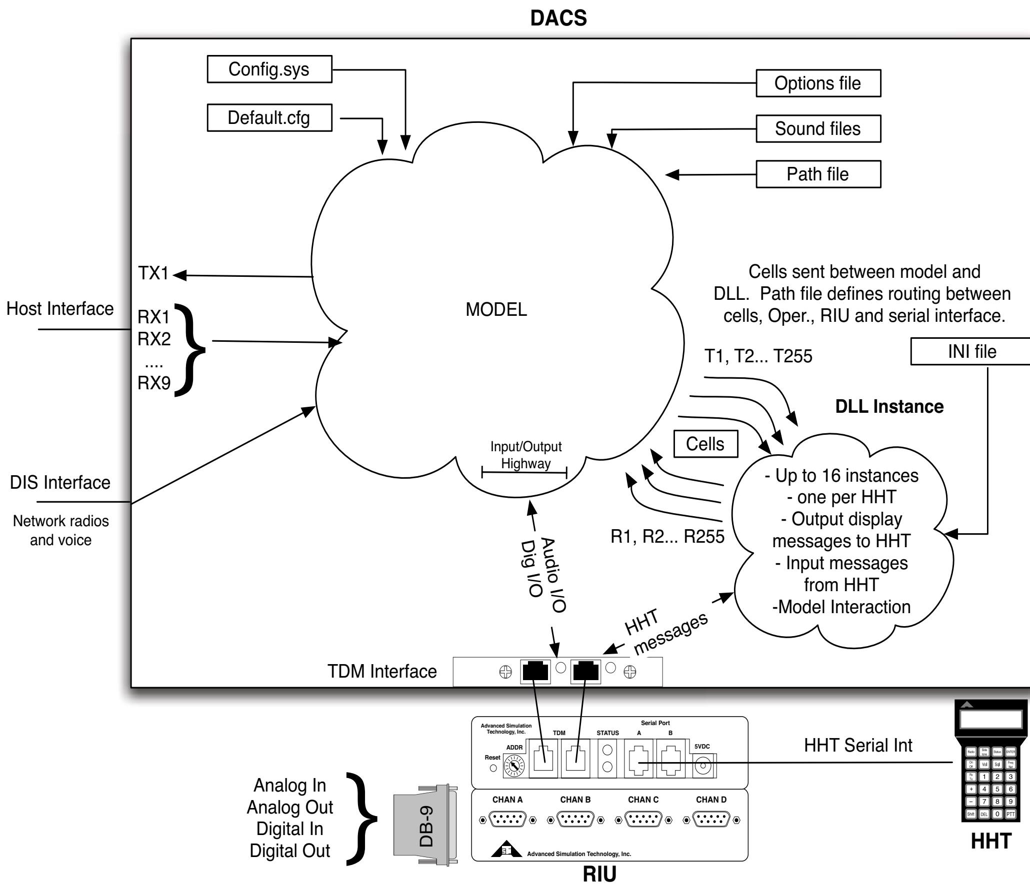


Figure 13: Model Builder Software Diagram

4. Model Builder Introduction (with hands-on exercises)

4.1 Model Builder Basics

Model Builder allows you to:

- Process and distribute audio at 8, 16, 22, 24, 32, or 48 kHz*
- Simulate radio communication complete with:
  - Ranging effects
  - Background noise
  - Voice break-up due to fading
  - Terrain restrictions
  - Encryption and frequency hopping
  - Jamming and in-band interference
  - Line of Sight (LOS) and Over The Horizon (OTH) propagation
- Simulate multi-bus intercom systems
- Create complex aural cue systems through synthesized audio

* The sample rate supported depends on the type of DSP card used.

4.2 Model Builder Status and Monitor Capabilities

With Model Builder you can:

- Determine the state of the host network and change the networking parameters
- Monitor cell traffic and activity (for HHTs and simulated radios)
- Determine the state of the DIS Network and change the networking parameters
- Monitor DSP state and view loading information
- Observe which RIUs are present and view the status of each RIU connected
- Monitor CPU loading

Some of the status and monitor screens in Model Builder are shown in the next few sections.

4.3 Model Builder Screens

Model Builder has a plethora of screens. The table below shows the screen layout available within Model Builder:

Level 1	Level 2	Level 3	Notes
Models	Model 1/2	Controls	Details shown in section 5.6 (Model Builder Object Types)
		Signals	Details shown in section 5.6 (Model Builder Object Types)
		Feeders	Details shown in section 5.6 (Model Builder Object Types)
		Functions	Details shown in section 5.6 (Model Builder Object Types)
		Soundfiles	Details shown in section 5.6 (Model Builder Object Types)
	Times		Model timing window that shows DSP, Network, cell timing information as total, peak and average.
	Memory		Available and used memory
	Options		Shows what is installed, credits, max DSPs and Channel width
	Pasteboard		Allows you to incorporate parts of other models into an existing or new model
DLL Status	Paths	Routes List	Setup by path file when model loaded
	Lost Cell Buf		
	Bad Cell Buf		
	DLL Status		DLL counter for put/get from HHT, sincgars, or other serial connections.
Host Interface (state information)	Ethernet Ctl		Network settings, set UDP ports for RX 1-9, Tx/Rx packets
	Ethernet Status		Network settings, packet types, stats and errors
	RX1 Buffer		RX1 Interface packet Ethernet display
	RX2 Buffer		RX2 Interface packet Ethernet display
	....		RX# Interface packet Ethernet display
	RX8 Buffer		RX8 Interface packet Ethernet display
	TX1 Buffer		TX1 Interface packet Ethernet display
DIS Network (Voice Networking)	Status		Network settings, Tx/Rx/Signal and Entity PDUs, and other UDP packet types, stats, errors, etc.
	Options		Protocol version, timeout, PDUs/packet, max samples, TTL, etc.
	Terrain		Terrain PDUs
	Freq of xmitters		Local and Network Tx PDU IDs for an exercise
	Transmitters		DIS network TX PDU list
	Local xmitters		Local to DACS
	Receivers		DIS network RX PDU list

Level 1	Level 2	Level 3	Notes
	Local receivers		Local to DACS
	Path Rx-Tx		Rx/Tx list for local and network
	Entities		DIS Network Entity PDU list
Local Network	Status		Frequency Tx PDUs count, local Tx PDU count and local Rx PDU count
	Terrain		Terrain occulting status window which shows terrain Tx/Rx PDUs and packets
	Freq of xmitters		Tx PDU for an exercise Local and Network based
	Local xmitters		Local to DACS
	Local Receiver		Local to DACS
	Path Rx-Tx		Rx/Tx list for local DACS
Waveform DSPs	Times		DSP frame count, frame time, memory %
	Gains		Audio input code gain in dB for each RIU/channel
	RIU status DSP#		RIU#, FW version, RIU overruns, errors and cell counts on a per DSP/RIU basis
OS Shell			Takes you to DOS shell without quitting Model Builder
Errors			List of startup Model Builder errors and warning messages
Status			List of what was loaded/configured during Model Builder startup: options, Ethernet settings, DIS, terrain, etc.
Restart	Restart		Restart Model Builder using current CFG file
	Reload		Restart Model Builder using new CFG file
	Restart remote = on		Restart with RMS capability
	Reboot		Reboot DACS

With RMS Telestra enabled on DACS the menu layout changes slightly. That is the DLL Status (Level 1 screen) is changed to the following below.

Level 1	Level 2	Level 3	Notes
Cell Interface	Remote Ethernet		Remote manager IP, Network settings, packet types, stats and errors
	Paths	Routes list	Setup y path file when model loaded
	Lost Cell Buf		
	Bad Cell Buf		
	DLL Status		DLL counter for put/get from HHT, singcars, or other serial connections.
	Cell Ethernet Status		Remote manager IP, Network settings, packet types, stats and errors

Table 3: Model Builder Screens

The subsequent sub-sections show some of the main submenus in detail.

4.3.1 Model Builder Configuration Window

- View/change freeze state
- View/change audio sample rate
- View/change audio highway width (# independent operators or audio outputs)
- View/change max number of RIUs on TDM ring (6 is min.)
- View/change the number of simultaneous receivers a given radio may process (important for net intercoms and AM radios)
- From here you have access to the file control, model creation, sound file creation and function creation menus

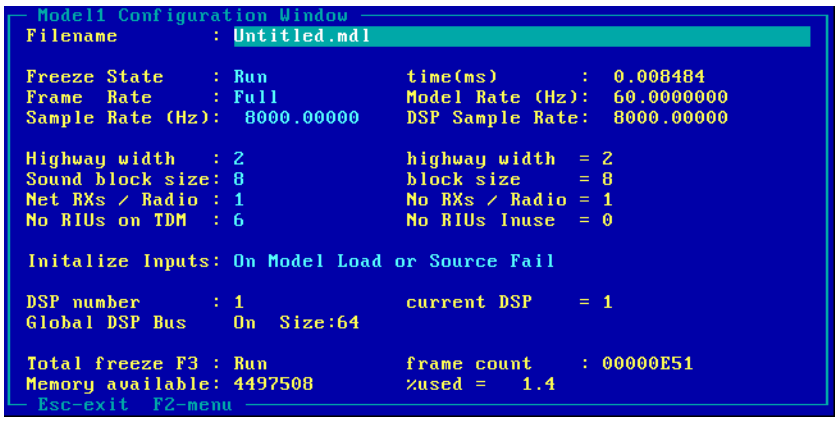
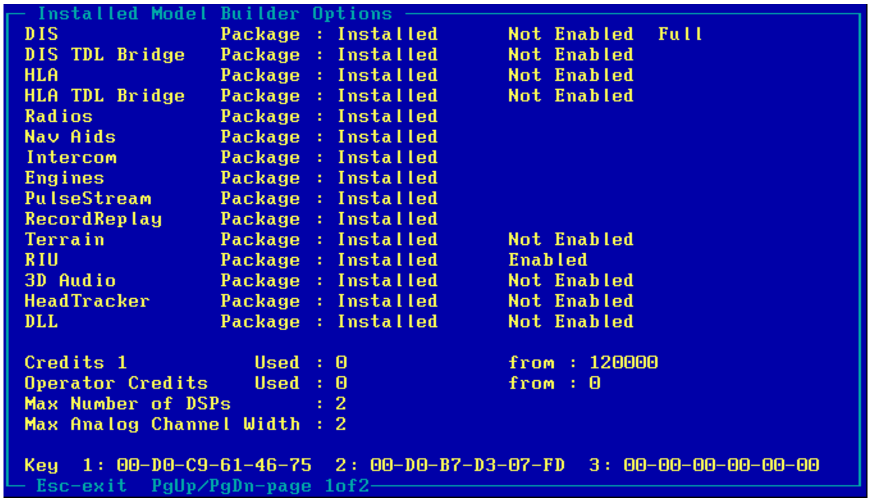


Figure 14: Model Builder Configuration Window

4.3.2 Model Builder Options Window

- View MAC address of DIS and Host Interface
- View Credits used/total
- View Max number of DSPs and Analog Ch width
- View which packages are installed (DIS, HLA, Terrain, etc.) and enabled

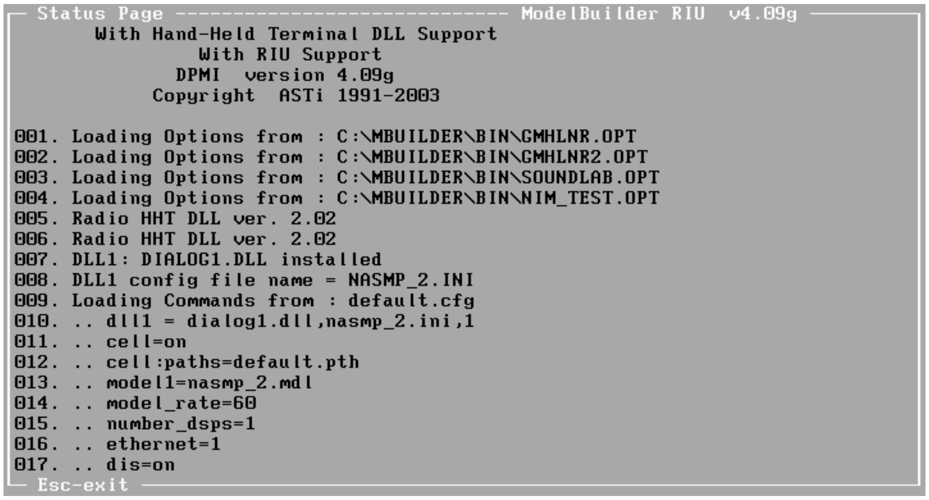


```
Installed Model Builder Options
DIS             Package : Installed      Not Enabled  Full
DIS TDL Bridge  Package : Installed      Not Enabled
HLA             Package : Installed      Not Enabled
HLA TDL Bridge  Package : Installed      Not Enabled
Radios          Package : Installed
Nav Aids        Package : Installed
Intercom        Package : Installed
Engines         Package : Installed
PulseStream     Package : Installed
RecordReplay    Package : Installed
Terrain         Package : Installed      Not Enabled
RIU             Package : Installed      Enabled
3D Audio        Package : Installed      Not Enabled
HeadTracker     Package : Installed      Not Enabled
DLL             Package : Installed      Not Enabled

Credits 1            Used : 0            from : 120000
Operator Credits     Used : 0            from : 0
Max Number of DSPs        : 2
Max Analog Channel Width  : 2

Key  1: 00-D0-C9-61-46-75  2: 00-D0-B7-D3-07-FD  3: 00-00-00-00-00-00
Esc-exit  PgUp/PgDn-page 1of2
```

Figure 15: Model Builder Options Window

4.3.3 Model Builder Status Page

- View which option file(s) were loaded
- View xxx.cfg commands loaded, where xxx is the configuration filename
- View Model Builder software version



```
Status Page ---------------------------- ModelBuilder RIU  v4.09g
      With Hand-Held Terminal DLL Support
               With RIU Support
             DPMI  version 4.09g
           Copyright  ASTi 1991-2003

001. Loading Options from : C:\MBUILDER\BIN\GMHLNR.OPT
002. Loading Options from : C:\MBUILDER\BIN\GMHLNR2.OPT
003. Loading Options from : C:\MBUILDER\BIN\SOUNDLAB.OPT
004. Loading Options from : C:\MBUILDER\BIN\NIM_TEST.OPT
005. Radio HHT DLL ver. 2.02
006. Radio HHT DLL ver. 2.02
007. DLL1: DIALOG1.DLL installed
008. DLL1 config file name = NASMP_2.INI
009. Loading Commands from : default.cfg
010. .. dll1 = dialog1.dll,nasmp_2.ini,1
011. .. cell=on
012. .. cell:paths=default.pth
013. .. model1=nasmp_2.mdl
014. .. model_rate=60
015. .. number_dsps=1
016. .. ethernet=1
017. .. dis=on
Esc-exit
```

Figure 16: Model Builder Status Page

4.3.4 DIS Network Status Page

- View IP address, mask, broadcast and ports for DIS Interface
- View Tx, Rx, Entity and Signal PDU counters
- View Tx/RX counts of various packet types (ARP, IGMP, UDP, etc.)
- View error counts, collisions and overflow counters
- Permanent settings are done through .cfg file

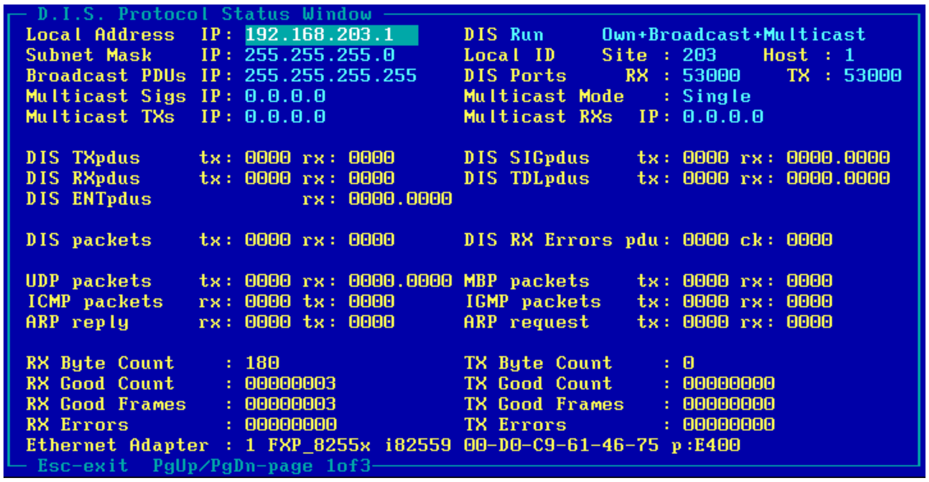


Figure 17: DIS Network Status Page

4.3.5 DIS Network Frequency of Transmitters

- View Site, Host, Entity and radio IDs for each object local to DACS
- View Site, Host, Entity and radio IDs for each network object received by DACS
- View Tx, Rx, On state of near and far end radios
- View frequency of near and far end radios
- One DIS exercise per page. Use +/- keys to move through pages or TAB to skip to next active exercise.

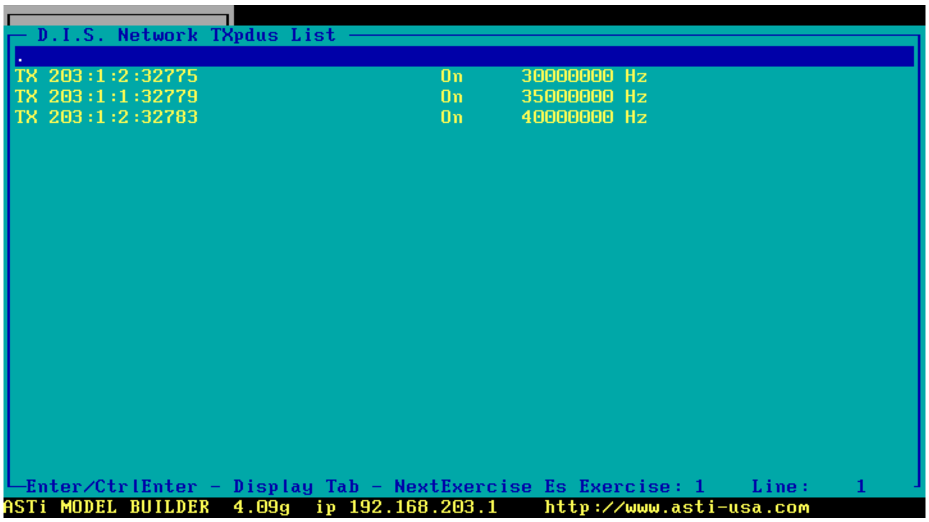


Figure 18: DIS Network Frequency of Transmitters

4.3.6 Model Builder Ethernet Control Window

- View/change local DACS IP address
- View/change IP address of DACS transmit destination
- View/change subnet mask IP address
- Packets received counter increments to indicate valid UDP packets being received on that UDP port
- Change UDP timeout values
- Enable/disable UDP checksums
- Change Endian format
- View/change UDP port
- View/change TX packet length
- Permanent settings are done in .cfg file

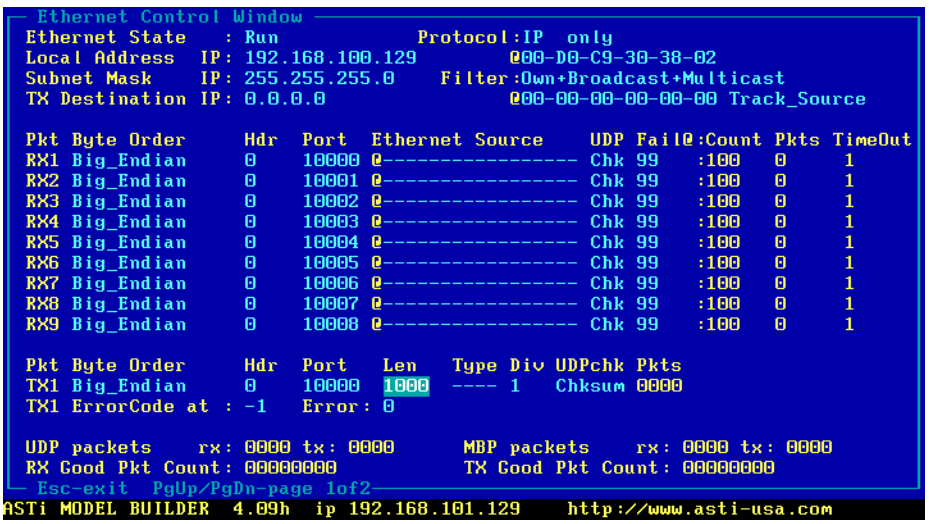


Figure 19: Model Builder Ethernet Control Window

4.3.7 Model Builder Waveform DSPs Window (s)

- Check percentage of DSP processing remaining
- Check for any DSP overruns
- See which RIUs are present on the TDM ring
- Check individual RIU status
- View/change RIU and 8AFA codec gains
- Check TDM Board version

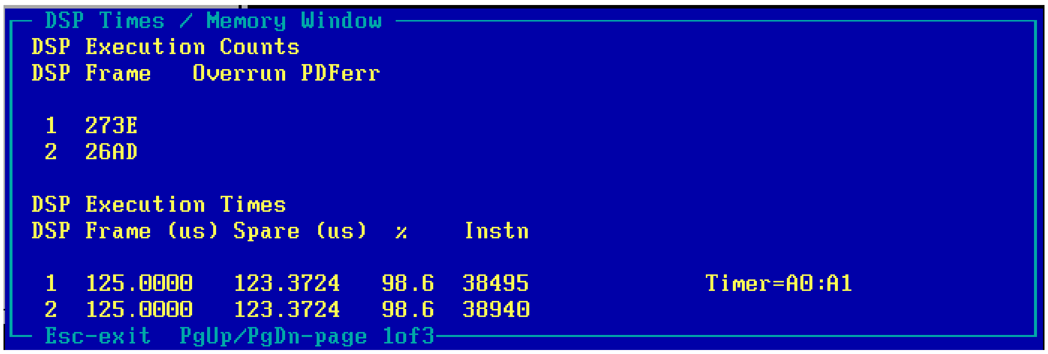


Figure 20: Model Builder DSPs Window 1

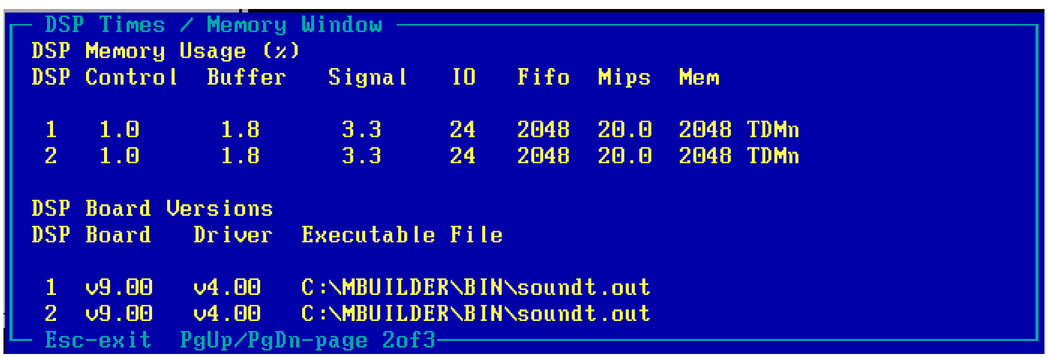


Figure 21: Model Builder DSPs Window 2

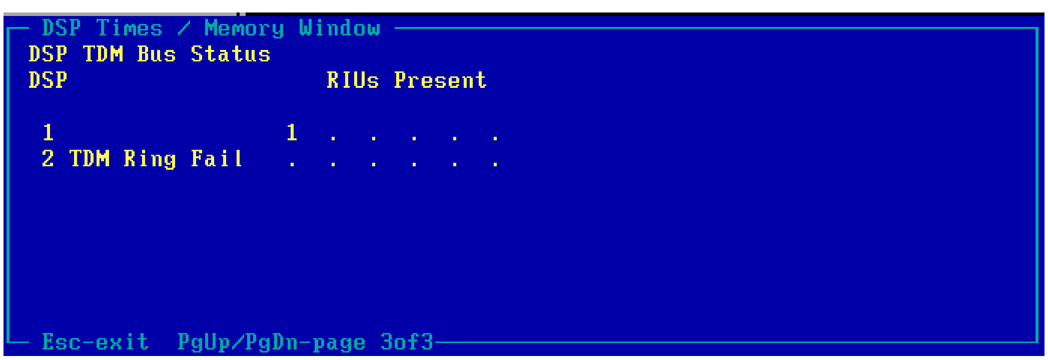


Figure 22: Model Builder DSPs Window 3

4.3.8 Model Builder RIU Status Page

- View status of each RIU individually
- See RIU firmware version
- Check for RIU overrun errors and checksum errors
- Verify that the RIUs are transmitting and receiving audio, data, and serial port information

NOTE: Your model MUST contain an RIU input object before the RIU status page will show that RIU. Without one, the RIU will still be seen in the RIUs present page, but not in the RIU status page.

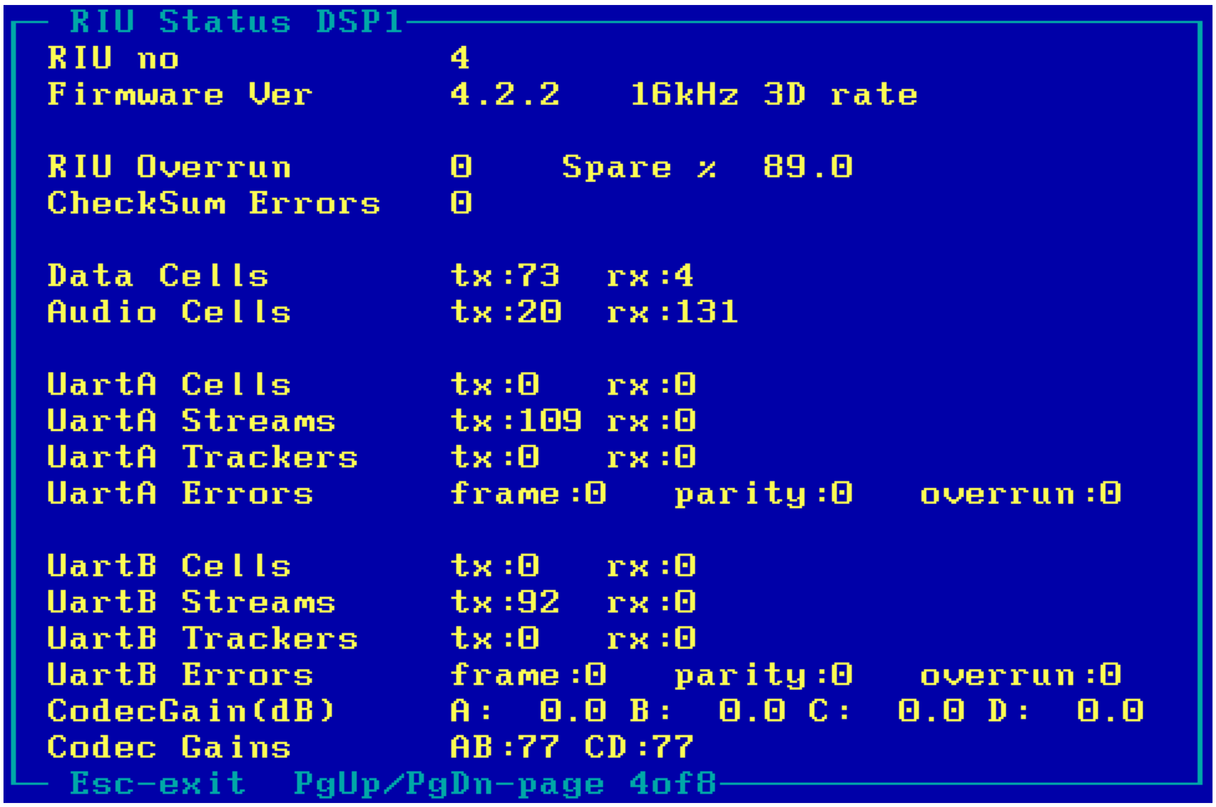


Figure 23: Model Builder RIU Status Page

4.4 Navigating in Model Builder

4.4.1 Function keys:

[F1]

The [F1] key cycles through various displays of information pertaining to objects in a list. For example: If viewing the Control list, pressing [F1] cycles through the object type, the Ethernet packet, packet offset, and any controls that are connected to all the objects in that list. Continuing to press [F1] brings you back around to the original screen.

[F2]

Pressing the [F2] key displays the pull-down menu for each of the Controls, Signals, Feeders, Functions and Soundfiles lists. If available, this key option will appear on the lower edge of each window.

The [F2] key is available in many individual objects, as well. For example: When viewing a Control object, pressing [F2] will display the contents of the buffer assigned to that object. This is much more convenient than backing out of the model, and referencing the Host Interface menu for buffer data.

[F3]

The [F3] key freezes the DSP and Ethernet card(s). The word “Freeze” will appear in the upper right corner of the screen when in this condition. Pressing [F3] again will release the freeze condition.

While in “Freeze”, you may still move through the model and add, delete and/or modify objects. The effects of changing the model will not be evident until the freeze condition is released.

HINT: Use the freeze condition if sound suddenly starts blaring out of speakers or headsets.

NOTE: If you are using RMS' virtual screen utility, pressing [F3] locally on the DACS' keyboard will break the Ethernet connection to the RMS server; the DACS will not respond to any remote commands until the freeze condition is released by pressing [F3] locally on the DACS' keyboard.

[F4]

The [F4] key marks objects in a list for copying or moving. More than one object can be marked at a time, and the resulting group may be copied, moved or deleted en masse.

[ESC]

Pressing the [ESC] key backs out of an object or list, bringing you to the next higher level in the model hierarchy. Pressing the [ESC] key repeatedly will eventually display the main menu.

4.4.2 Moving through the menus

There are two ways to navigate through Model Builder menus:

1. Shortcut Keys

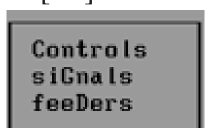
   A menu item's shortcut key is denoted by the first capitalized letter in the item name itself. For example: From the main menu, pressing [d] displays the “Dis network” submenu. Pressing [s] then displays the “Status” screen, etc.

   Another example: To access a model's “Signals” list from the main menu...

   a. Press the [m] key to access “Models”.

   b. In the submenu, press [Enter] to select “model1” (already highlighted).

   c. From the model screen, press [F2] to access the pull-down menu.



   d. Press the [g] key to select the “siGnals” option in that menu. Model Builder will then display the Signals list for that model. (For feeders press [d], for controls press [c]…)

2. Arrow Keys + [Enter]

   Any menu's items may be selected by using the arrow keys (up, down) to highlight that option and pressing [Enter]. The corresponding submenu or page will then appear.

Once the desired screen or object has been accessed, use the [PageUp] and [PageDown] keys to cycle through the multiple pages of some objects, where applicable. Multipage objects will display “Page X of Y” on the bottom edge of that window.

4.4.3 Accessing Objects in a List (Controls, Signals, Feeders, etc.)

1. Navigate to the desired list.
2. Use the arrow keys to highlight the desired object, and press [Enter].
3. Use the arrow keys to move among the various fields in that object.

Press [ESC] to move back to the list.

4.4.4 Other Model Builder conventions

- Use the +/- keys to toggle Boolean values on/off
- Use **page up/page down** to move through multi-page objects
- Values in **BLUE** may be changed
- Values in **YELLOW** show status, gains, results, etc. and are not directly modified
- Changing Values
  - Use arrow keys to highlight values

    -Use +/- to change value by 5%
  - Press **ENTER** to change from Blue to Red

    -Use # keys at the top of the keyboard to enter values

    -Press **ENTER** for value to be set

4.5 Creating, Saving and Restoring Models

4.5.1 Creating Models

Creating a new model in Model Builder is quite simple. From the main model configuration window, press [F2] to bring up the menu, then press [N] or scroll down to 'NEW' and press enter. A screen like the one below will appear.

Figure 24: Model Loading Screen

Then, press [Y] for yes and Model Builder will save the current model to a backup (.bak extension) and open a blank, untitled model.

4.5.2 Saving Model

To save a model, press [F2] from the model configuration window and press [S] or scroll down to save and press enter. Then press [Y] for yes. This saves the model with its current name. To save the model with a different name, use 'Save As' and specify a new file name. Below is the prompt you should get for entering a new name.

Figure 25: Save Model Screen

HINT: Save early and often!

NOTE: Model Builder does NOT ask you if you want to save if you exit without saving any changes. You are asked however, to verify that you want to quit. Exiting without saving automatically loses all the changes you made since the last save. This can be good or bad, just don't forget about it.

4.5.3 Restoring Models

Restoring a model allows you to revert to your last saved version of a model. This process saves the current model to a backup file and loads the last saved version of the model. This is the same as quitting Model Builder without saving, and then restarting and loading the last saved version. Below is the restore dialogue window.

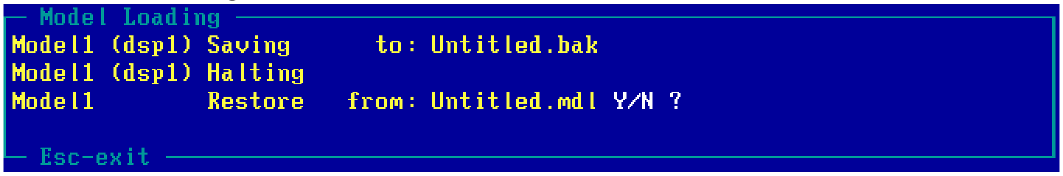


Figure 26: Restoring Model Screen

To load a file into Model Builder, just highlight the filename on the model configuration page, press [Delete] then [Enter] and a menu will pop up. This menu is a list of all the model files in your current working directory. Scroll down to the model you want, press [Enter] twice and answer yes by pressing [Y] and the desired model is loaded into Model Builder.

4.6 Model Builder Object Types

There are three types of main object types in model builder: Signal, Controls and Feeders. There are also functions and sound file objects available.

4.6.1 Signal Objects

A signal is a sound or message that is transmitted or received in electronic communications (or in this case communications simulation).

There are several signal types, as can be seen from the Signals menu in Model Builder below. Simple noise makers such as a Sine, Triangle, sawtooth, square or pulse waves are available for model development. Filters are available to remove unwanted audio and white noise. Radios and Intercoms are at the heart of Model Builder and allow the end user to model real radio parameters over a DIS network (HLA with Telestra option) or locally on a DACS. Audio can also be transmitted into and out of the RIU channels. Additionally, play sounds can be incorporated into models for realistic simulations of environmental, background, simulator, etc. noises. Communication Panels allow for emulation of aircraft, vehicle or shipboard communications control panels.

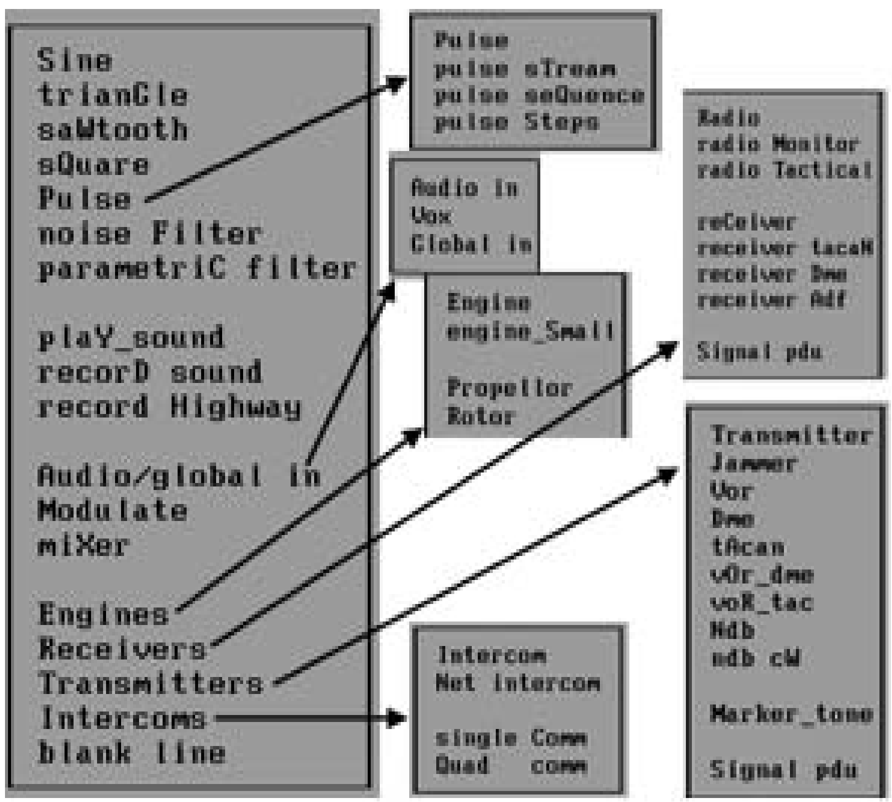


Figure 27: Signals Menu

4.6.2 Control Objects

Control objects produce numbers or on/off logic that are used to manipulate other parts of the model. Control objects include counters (which give incrementing numbers), functions, lookup tables, morse code generators, and other objects. In addition, control data sent to and from a host simulation application, to and from control panels, or to and from a state machine go through control objects. The following are examples of control objects:

- Booleans, integers, floats (In/Out of DACS)
- Counters
- Logic tables
- World positions (radios and intercoms)
- RIU I/O (DI/DO)
- DDI (a.k.a PIU) - DI, DO, AI (float)

Control objects interact with Model Builder through the Ethernet interface and local I/O variables used by a particular sound model. In general, the host interface is used to receive input objects from an external device (a cockpit, for example). The Interface can also output control objects to an external device.

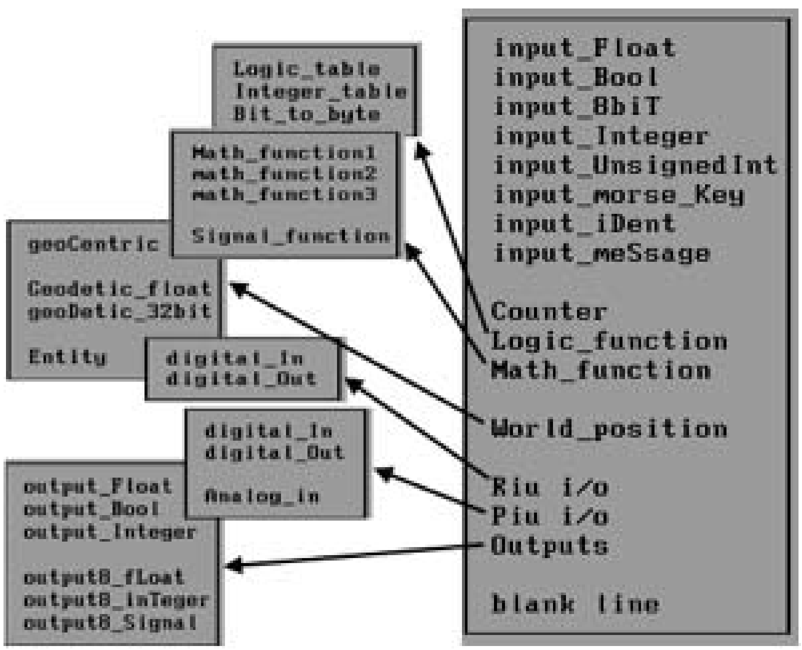


Figure 28: Controls Menu

4.6.3 Feeder Objects

Feeders are the objects that connect either a signal source or analog input through to the mixing highway. They are divided into:

- Basic feeders - provide connection for one signal onto one or more channels of the highway
- Mixing feeders - allow for simple selection of multiple signals onto a single highway channel
- Analog input feeder - provide summation of one or more analog input sources onto a single highway channel
- Analog output feeder - connects a series of highway channels to the analog output

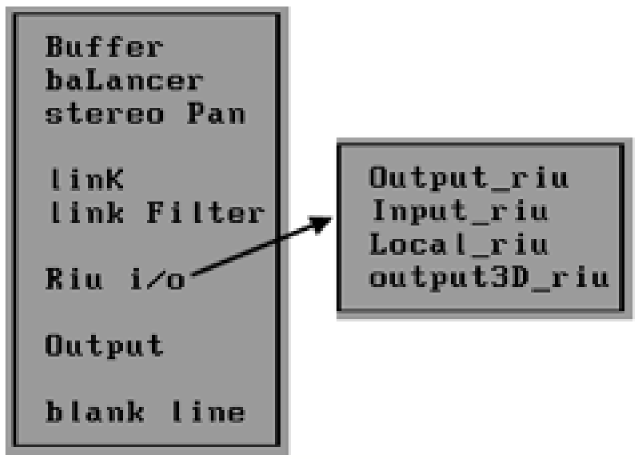


Figure 29: Feeder Objects

4.6.4 Functions Objects

The functions menu within Model Builder provides access to various mathematical functions, which may aid in model development. Function Objects, in combination with Control Objects, permit local manipulation of data within the model and hence increase the capabilities of Model Builder. Below is a copy of the Functions menu that shows the various objects available within Model Builder:

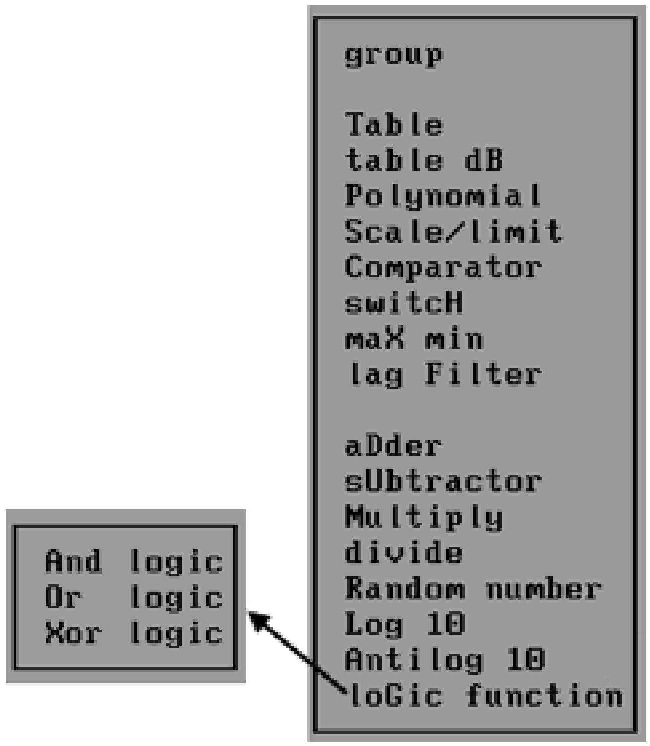


Figure 30: Functions Objects

4.6.5 Sound File Objects

The sound files list contains individual files or a group of files that the user has loaded into the model. Grouping allows the host to dynamically select files from within a predefined subset, using an integer file identifier. The “Message List” control object may be used to seamlessly stream together individual sound files in a group. Sound files are in the Next/Sun .au format. The sample rate of the sound file should match that of the model.

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.

4.7 Hands-on Exercises

4.7.1 Generating and Routing Audio

Purpose: This exercise will familiarize you with simple sound generation objects and how they are routed in Model Builder. After this exercise, you will be able to create a sound in Model Builder and output it to your headset.

1. From the 'Feeders' menu, add an Output object and set all gain fields to 1.0.
2. Add an Output_riu object, set all gain fields to 1.0 and set the highway to 1 and RIU number to 1.
3. Add a Buffer object and set all gain fields to 1.0.
4. Go to the 'Signals' menu and add a Sine object.

   a. Set gain field to 0.2 and set frequency to 500 Hz.

   b. Insert the buffer you created in the "Feeder" connection point.
5. You should now hear the 500 Hz sine wave in the headset and speaker.

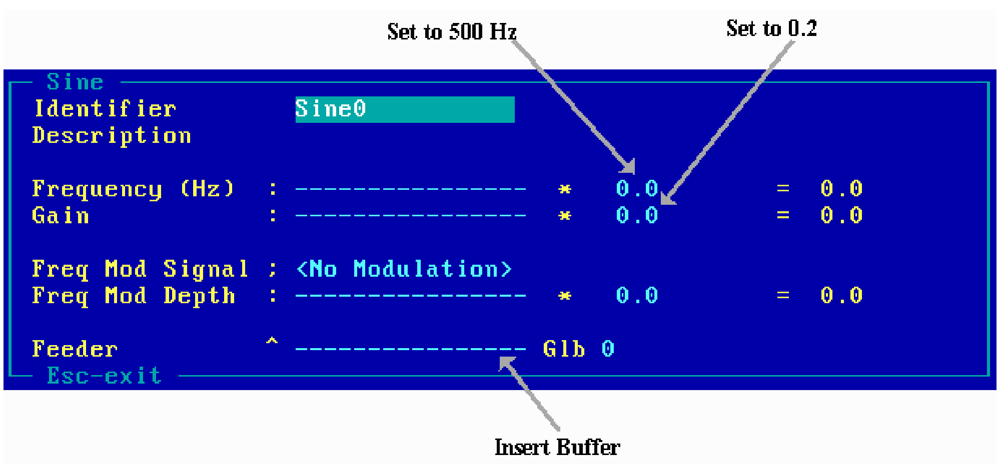


Figure 31: Sine Object Screen

6. Try changing the gain and frequency and listen to what happens.
7. Save this model as "471_sine.mdl."

4.7.2 Generating and Routing Audio (continued)

Purpose: This exercise will teach you how to use a balancer to send one signal to multiple highways and the mixer to send multiple sounds to one highway.

1. Now, from the 'Feeders' menu add a baLancer object (sends one signal to multiple highways).
2. Also add another Output_riu object that outputs from highway 2 to RIU 1 Ch. D.
3. Change the first output object to output only to RIU 1 Ch. A.
4. Set the balancer gain to 1.0.

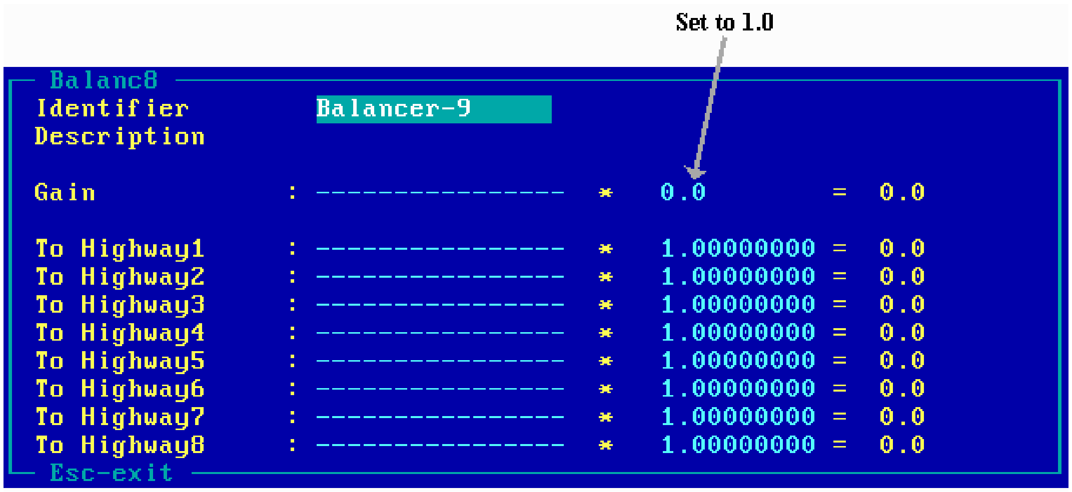


Figure 32: Balancer Object Screen

5. Insert the balancer as the output for the Sine object instead of the buffer.
6. Sound should now be coming from both Ch. A and D, and the gain can be changed from the baLancer object window.
7. Now add another Sine object, but at 250 Hz.
8. Add a miXer object as shown below with Sine_1 and Sine_2 as the inputs and the baLancer as the output. Don't forget to remove the baLancer as the output from Sine_1.

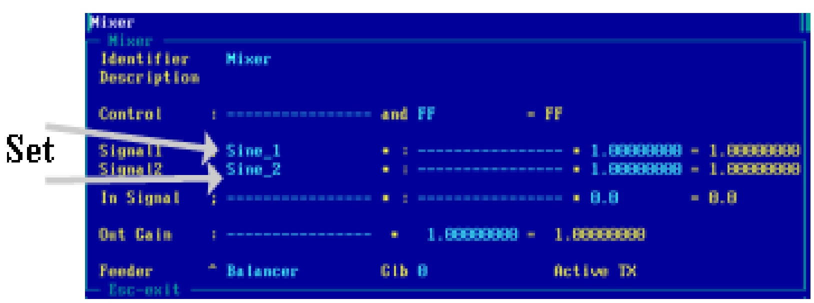


Figure 33: Mixer Object Screen

9. Try changing the gain of each signal in the mixer window and listen to the result.
10. Save this model as “472_sine.mdl.”

4.7.3 Audio Input Interface

Purpose: After this exercise you will be able to set up an RIU input and configure the VOX object to get audio into the model from your microphone.

1. From the 'Feeders' menu, add an Input_riu object. Set the gain to 1.0 and set the RIU number and highway to 1 and 1 respectively.
2. Add a Vox object from the 'Signals' menu. Set the Audio Input (AI) to 1 or set it to match the input highway used in step 1.
3. Add a Buffer, riu_Out, and Output object to the "Feeders" list and insert the buffer into the Feeder field in the Vox object. (You can use an existing buffer and output if you have one already).
4. You should set the gain of either the VOX, buffer, or RIU out to 0.5 or the audio will be very loud.

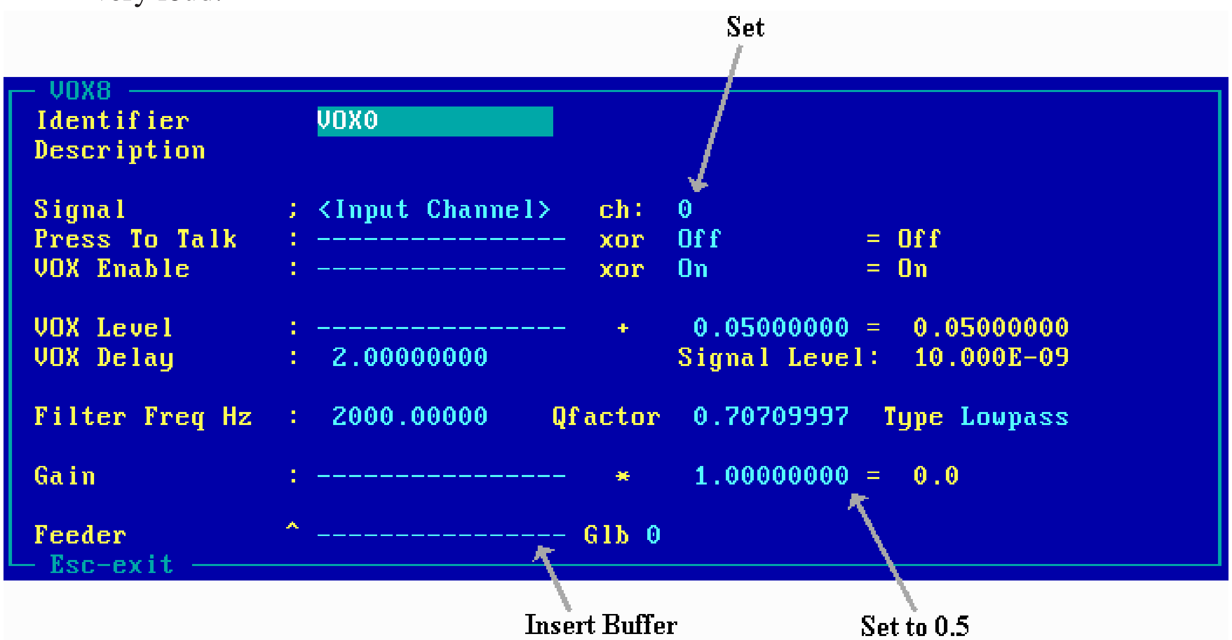


Figure 34: VOX Settings

5. Speak into the microphone. You should see the gain change from 0.0 to 1.00000 and the Signal Level will increase. You should also hear the audio in your headset.
6. Change the 'VOX Level' field and notice how as it increases, you must speak louder before your voice is transmitted (the object displays "Active TX" in the lower right corner).
7. Save this model as "473_vox.mdl."

4.7.4 Local Controls and Digital Inputs

Purpose: This exercise will familiarize you with the RIU digital inputs and how to use them. At the completion of this exercise, you will be able to use the RIU digital input with a PTT switch to control the input to a VOX object.

1. From the 'Controls' menu add a digital_In object from under RIU i/o.

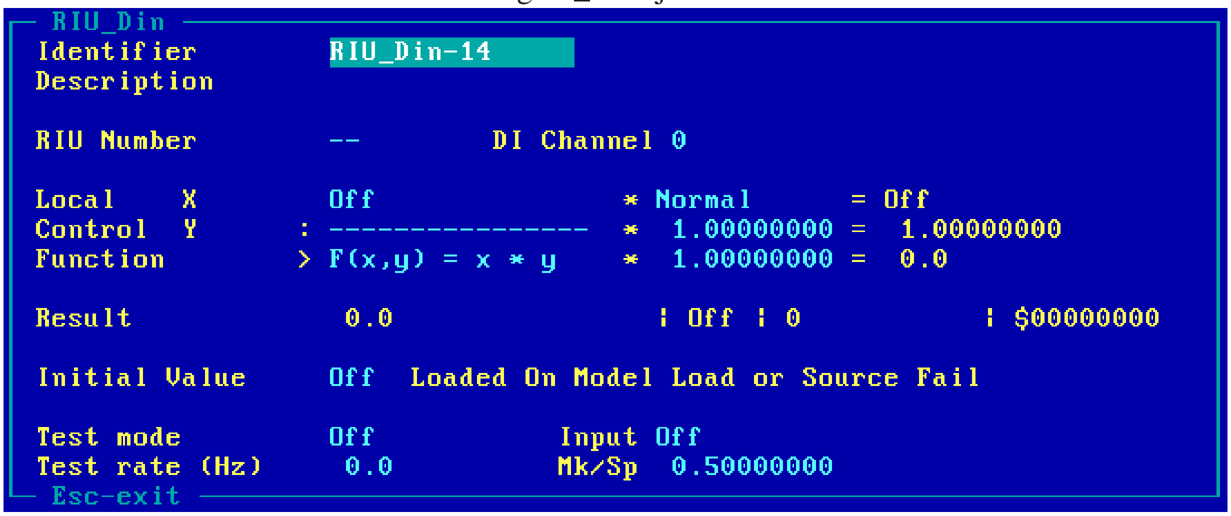


Figure 35: RIU Controls

2. Set the RIU number to 1 and the DI channel to A, B, C or D, by pressing the + key. (**NOTE**: In previous Model Builder versions the RIU audio channels are numbered 1-4, but the digital in/out channels are numbered 0-3.)
3. Press your PTT Switch. You should see the “Result” field change accordingly.
4. Now return to your VOX object and turn off the “Vox Enable” field.
5. Now insert your digital_in object into the “Press to Talk” field.

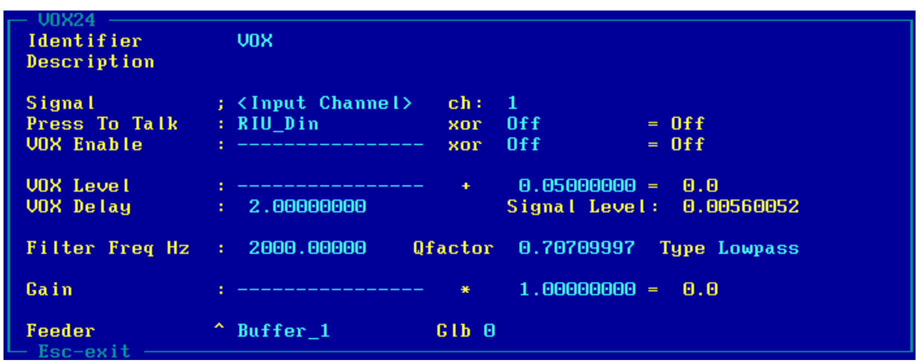


Figure 36: RIU Digital Inputs

6. When you press your PTT button and speak, the 'Gain' field should change and you should hear your voice in the headset.
7. Save this model as “474_PTT.mdl.”

4.7.5 Host Interface Example

Purpose: This exercise will help you understand the basics of using host controls in Model Builder.

1. To set up host controls, you need at least one DACS unit and a second computer to act as the host on the same network. The second computer can be another DACS or a PC.

   NOTE: Our example assumes you are using two DACS. The DACS transmitting host control data is DACS1. The DACS receiving the host control data is DACS 2. On DACS 1, set Local Address to 192.168.100.102 and TX Destination to 192.168.100.120. On DACS 2, set Local Address to 192.168.100.120 and TX Destination to 192.168.100.102.

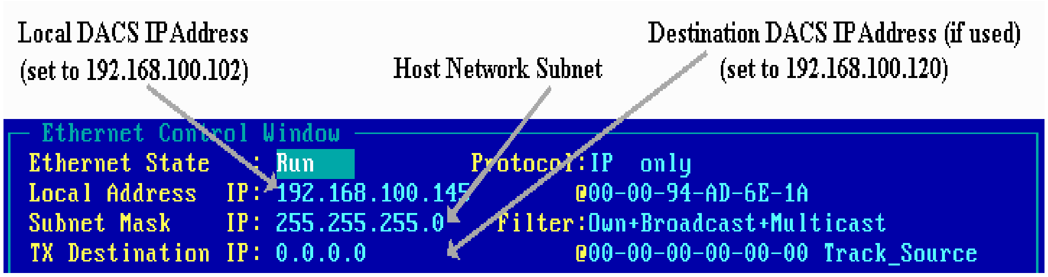


Figure 37: Setting Host Controls

2. Enter all information, shown above, in the Ethernet Control window of one DACS unit. Also make sure the Tx Length is set to 60 or greater. Substitute your own IP addresses and make sure the local IP's are unique.

3. Insert an input_morse_Key object in your model on DACS 1, this will be your host control input. The input_morse_Key object can be found on the controls list. Use the Test mode to set the value to SOS.

4. On DACS 1, add an output_Bool object as well. This will send the control signal to DACS 2.

5. In the output_Bool object on DACS 1, set the transmit buffer to TX1 and insert the input_morse_Key as the output control for this object.

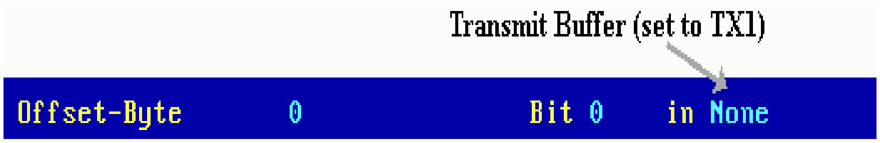


Figure 38: Setting Output Control

6. On DACS 2, add an input_Bool object in the 'Controls' list and set the receive buffer to RX1.

Figure 39: Setting the Receive Buffer

7. In the 'Signals' list on DACS 2, create a Sine object with a frequency of 500 Hz and set the input_Bool object as the gain control. You will see the Sine object gain changing from 1.0 to 0.0 along with the Morse input coming from the host.
8. Look at the Host Interface -> Ethernet Control so that you can monitor the Tx and RX packets from the DACS. The TX1 and RX1 fields on DACS 1 and DACS 2 respectively should be incrementing.
9. As an advanced exercise, create the necessary “Feeder” objects and listen to the Morse tone on DACS 2.
10. Save these models as “475_D1HT.mdl” and “475_D2HT.mdl” for DACS 1 and DACS 2 respectively.

4.7.6 Review of Chapter 4 models

- Feeder Objects bring audio into the model and route audio from the model.
- Signal Objects process the audio.
- Control Objects manipulate the Signals and Feeders. I.e. PTT, volume (gain) levels, VOX threshold, etc.

Below are diagrams for the models created in the previous sections:

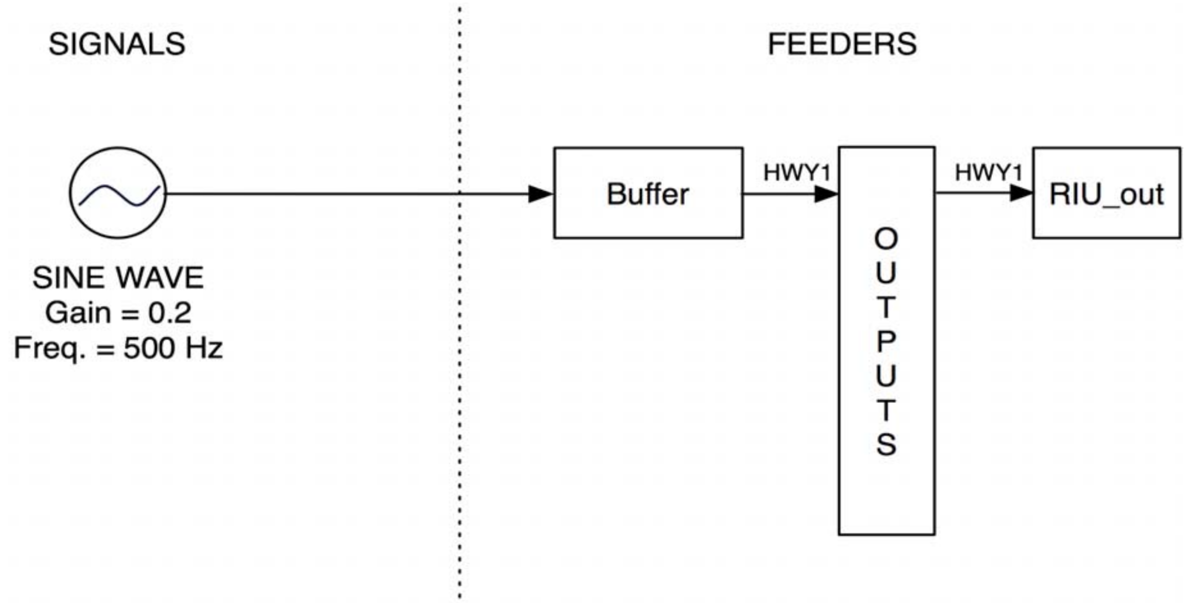


Diagram 1: 4.7.1 - Sine Wave Output

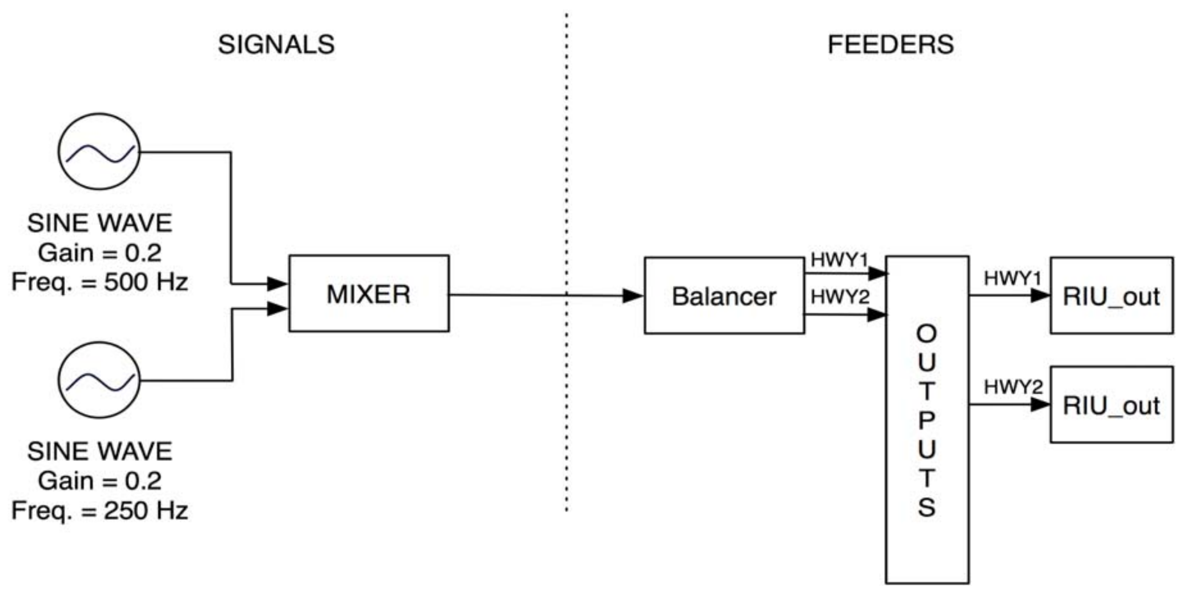


Diagram 2: 4.7.2 - Multiple Sine Waves Output to Different Highways

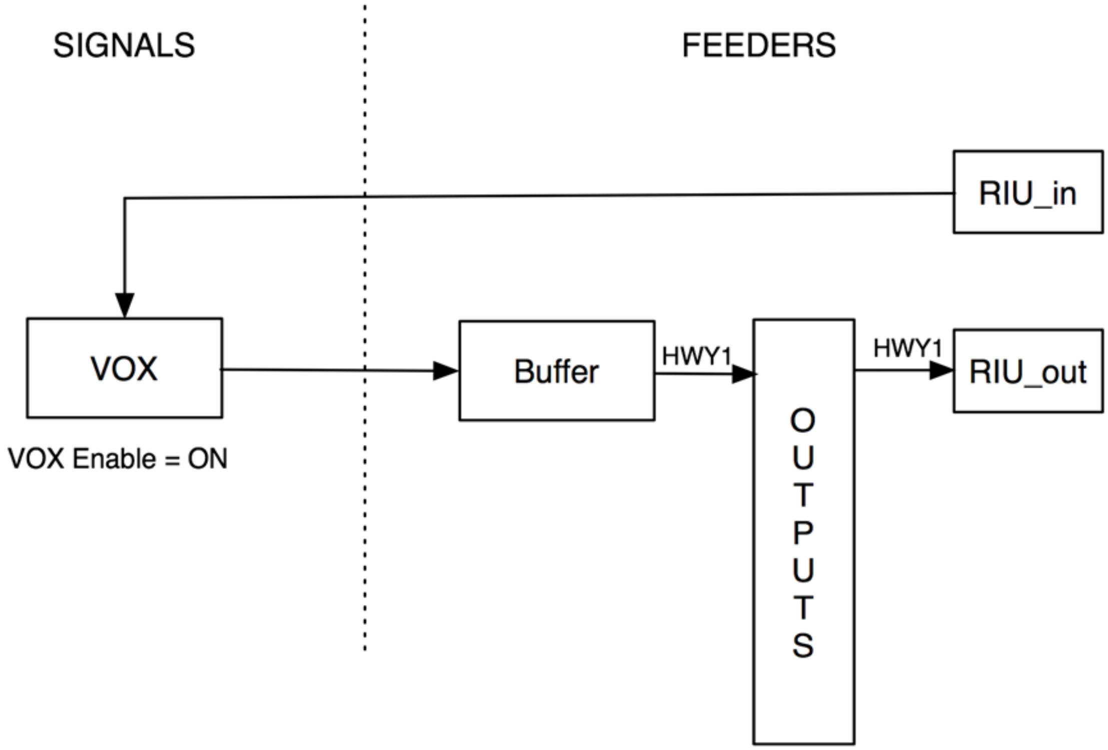


Diagram 3: 4.7.3 - Audio In/Audio Out via VOX Object

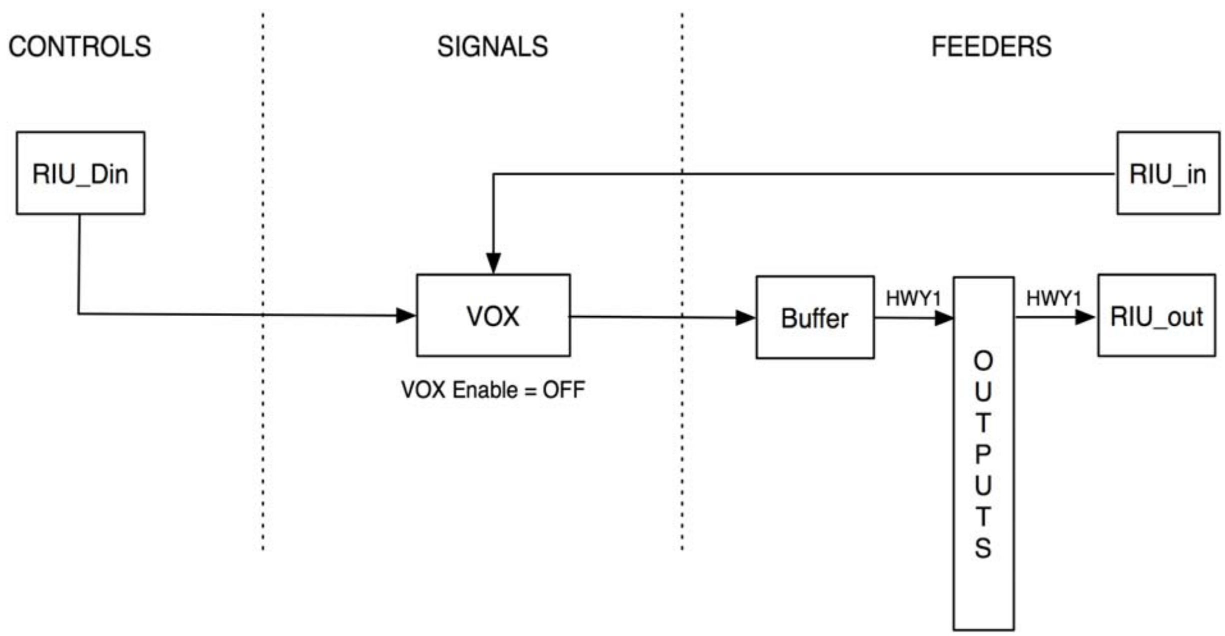


Diagram 4: 4.7.4 - Audio In/Audio Out via VOX Object and PTT

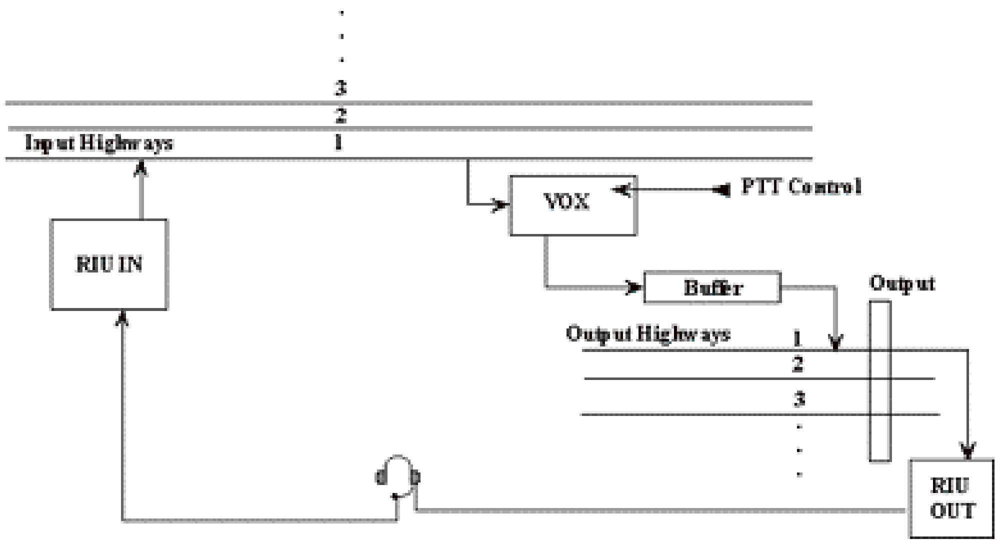


Diagram 5: 4.7.1 - 4.7.4 - Overall View of Previous Four Exercises

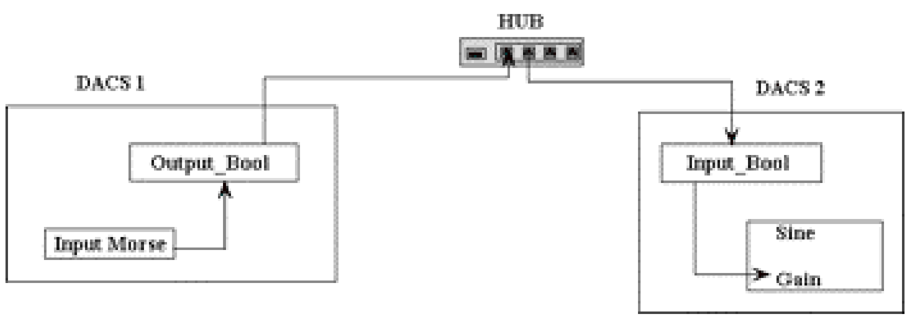


Diagram 6: 4.7.5 - Host Input Example with Boolean Input and Output

5. Networking DACS (DIS and Host Interface)

5.1 Introduction

The DACS can be purchased with one or two Ethernet Interfaces, which are used for one or two connection types:

1. DIS/HLA connection
2. Host connection

If one interface is available on the DACS, it is generally used for DIS/HLA or host traffic. You can, however, combine the DIS and HOST traffic onto a single interface if desired. In this case, you need to be careful of network loading and configuration in addition to any security requirements of the network. This solo interface will be located on the single board computer. If two interfaces are installed on the DACS, then the single board computer interface acts as the host connection, while the separate NIC is the DIS/HLA connection. The following two sections cover the DIS/HLA and Host interfaces in detail.

5.2 DIS Interface

The DIS interface carries the voice and data traffic, which is generated and received by Model Builder. In the snapshot below, you can see the Network settings, TX PDUs, RX PDUs, Signal PDUs and Entity PDUs in addition to various other interface counters. The DIS interface is also used in conjunction with the Telestra RMS for remote management. This feature allows a user to remotely manage multiple DACS from a single Telestra RMS.

5.2.1 DIS Radio Basics

Network configuration:

- To communicate over DIS, this feature must be enabled by ASTi in your CFG file.
- You need two DIS enabled network products (such as two DACS).
- Local IP address and Mask
- Broadcast IP address
- Tx and Rx UDP ports
- IGMP Parameters (multicast only)
- Checksum parameters and DIS PDU timeout values (Options screen)
- Many of the above settings are configured in the DIS status window (shown below)

Model configuration:

- Ensure the DIS IDs are unique and radios are in the same exercise.
- Match modulation type, frequency, BW, modulation, crypto state, Frequency Hopping/HQ settings, etc. for the radios.
- Model Builder decodes incoming audio based on signal PDU.
- Support of multiple simultaneous exercises (up to 255).

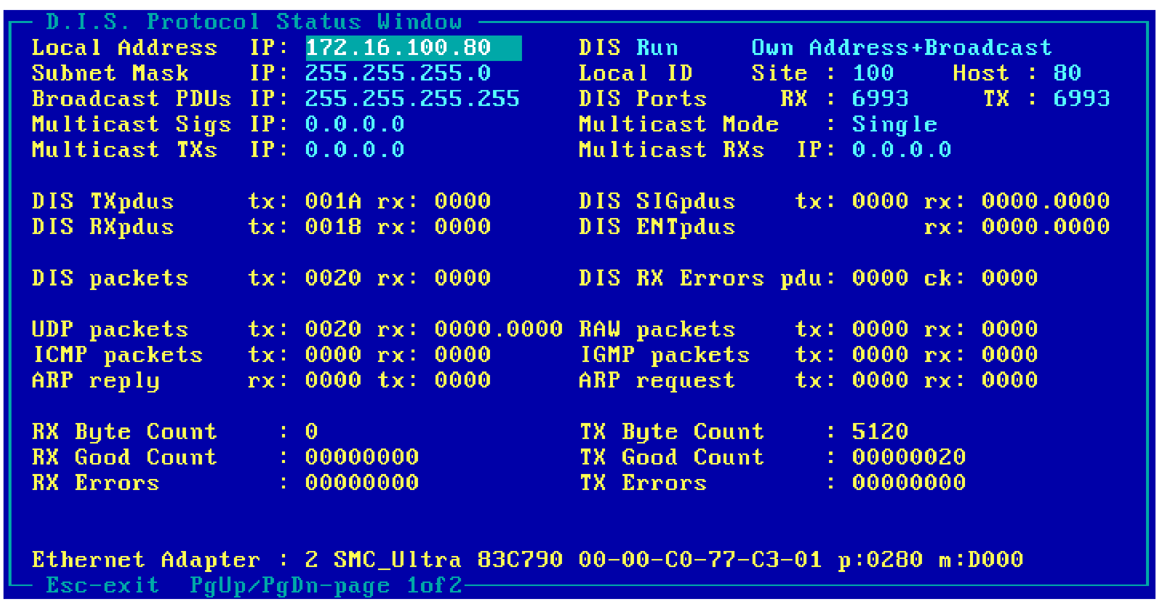


Figure 40: DIS Protocol Status Window

5.2.2 DIS PDU Types

While this is a course on the DACS and Model Builder software, it is important to understand the DIS interface traffic and its characteristics. When the DIS interface is used to carry model communications information the packet format is "IP|UDP|DIS_PDU" over standard Ethernet. The UDP port setting is configured through the default.cfg file. The DIS PDUs contain all of the pertinent information such as:

- Voice communication
- Tactical Data communication
- Radio Parameters (frequency, location, Tx power, DIS IDs, exercise #, etc.)
- Terrain request/reply

DIS IDs are broken down into site, host, entity and radio IDs. The 4 set string: **site:host:entity:radio** (for example: 10:20:30:40) must be unique for each radio on the network. While there is not a steadfast rule for setting up the IDs, one common scenario is to associate the site and host IDs with a physical location, the entity ID with a DACS and then have individual radio IDs for each radio instance.

There are four types of PDUs detailed in the following sections.

5.2.2.1 TX PDU

Transmitter PDUs are required for DACS to operate in a networked mode. They are both transmitted and received by DACS systems. The TX PDU is an informational PDU that is sent out periodically and contains information about:

- Site:Host:Entity:Radio IDs
- Radio frequency
- Location
- TX Power
- Exercise number
- Modulation
- Bandwidth
- Crypto parameters
- Frequency Hopping/HQ parameters
- State (On, Off, On_Tx or ACTIVE)

Within Model Builder, you can see all active TX PDU IDs that have been received by a DACS. Under Model -> Status -> Frequency of Tx you will see all local and remote DIS IDs on the network. The white items listed are remote and the yellow items are local to the DACS.

In short, Tx PDUs are a radio's (or other object in Model Builder) way of saying, “who, what and where I am.” Rx objects in Model Builder scan Tx PDUs to determine who is in range. Transmitter PDUs are put out periodically if the radio is stationary (five second default) or when the radio has changed state; that is moved, started/ended transmission, or changed parameters.

5.2.2.2 Signal PDU

Signal PDUs are required for DACS to operate in a networked mode. They are both transmitted and received by DACS systems. The signal PDU is a UDP packet, which contains voice information or data messages. When actively transmitting or receiving from a radio, for example, you will Rx/Tx a continuous packet flow during this time. The audio is encoded with the settings (mu-law, PCM, CVSD) in a given radio/intercom object.

5.2.2.3 RX PDU

The RX PDU is not necessary for Model Builder to run, however it is built into Model Builder as a standard feature. RX PDUs transmit receiver state information, such as the received power level. It is for informational purposes only, and does not cause the DACS to make any adjustments based on the values received. The Rx PDU says, “who I am in tune with” for each receiver, and whether or not they are actively receiving audio.

5.2.2.4 Entity State PDU

Entity state PDUs are not necessary for Model Builder to run and are not generated by the DACS. The DACS receives entity PDUs to obtain position information for radios when an "Entity Attach" object is used. For example, the radios in Model Builder can be moved by a Mod Semi Automated Force (SAF) entity generator.

5.3 Host Interface

The host interface is used to receive simulation state data for control of the local audio model(s). This host simulation computer will normally provide simulation state and sound model control parameters such as engine RPM, radio frequency, radio position, communications panel switch settings, etc.

This data is transmitted over an Ethernet network to the DACS and received via the host interface network card. Communications with the DACS are asynchronous. The host computer transmits packets at the host defined iteration rate. The DACS Ethernet hardware receives and buffers the packet in local memory. With each DACS model iteration, the Model Builder application software uses the latest receive packet for host input data.

Some model state data and system health parameters can also be transmitted back to the host. Packet transmission for data being returned to the Host can take place in each model iteration, or be reduced in frequency via system start up parameters. Data received is buffered and brought in to the model using various control objects available in the Model Builder development environment.

The user can inspect and modify packet data, as well as network modes and parameters through both on-screen displays and also through the start up configuration file. For transferring simulation state parameters, the DACS supports IEEE 802.3 standard UDP level protocol with IP addresses. The use of UDP facilitates the reception of state data from multiple simulation sources by selection of independent port numbers. The user will need to configure the DACS with the appropriate network settings to ensure proper network operation. The DACS can receive host data on up to nine UDP ports for Model Builder 4.09 and later.

6. Model Builder Advanced (with hands-on exercises)

This section will discuss radio simulation in Model Builder. This topic is somewhat difficult to follow at times, so it will be broken into basic and advanced sections, with an overview of special features and extras. This section will also cover network radio communication via DIS.

Overview:

- Creating radio objects
- Getting radios to communicate in the same model and over the network
- DIS Primer

Essential Controls:

- Mode
- Tuned Frequency
- World Position
- Tx Signal

Advanced Topics:

- Crypto
- Frequency Hopping and Havequick
- Ranging and background noise
- Voice encoding schemes
- Tactical data link
- Improving the fidelity:
    - Variable bandwidth
    - Jamming libraries
    - Terrain interface
    - HF modeling

6.1 Radio Communications

6.1.1 Radio Essentials

There are three essential parameters required in Model Builder when talking about radios. They are:

- Power/Mode: This must be given a value between 1 and 16 to select one of the radio modes.
- Tune Frequency: This must be set to a non-zero value. Remember, frequency is in Hertz. A value of 0 will turn the radio off.
- World Position: The radio must be connected to a world position control object, so it has a position to perform ranging calculations with and to identify the radio in the simulated environment.

These are the bare minimum requirements for properly simulating a radio. The radio object consists of nine to ten pages of parameters that control special features such as transmit signal, frequency hopping, crypto, internal noise, AGC, antenna gain, and other parameters. The first page of the radio object is shown below.

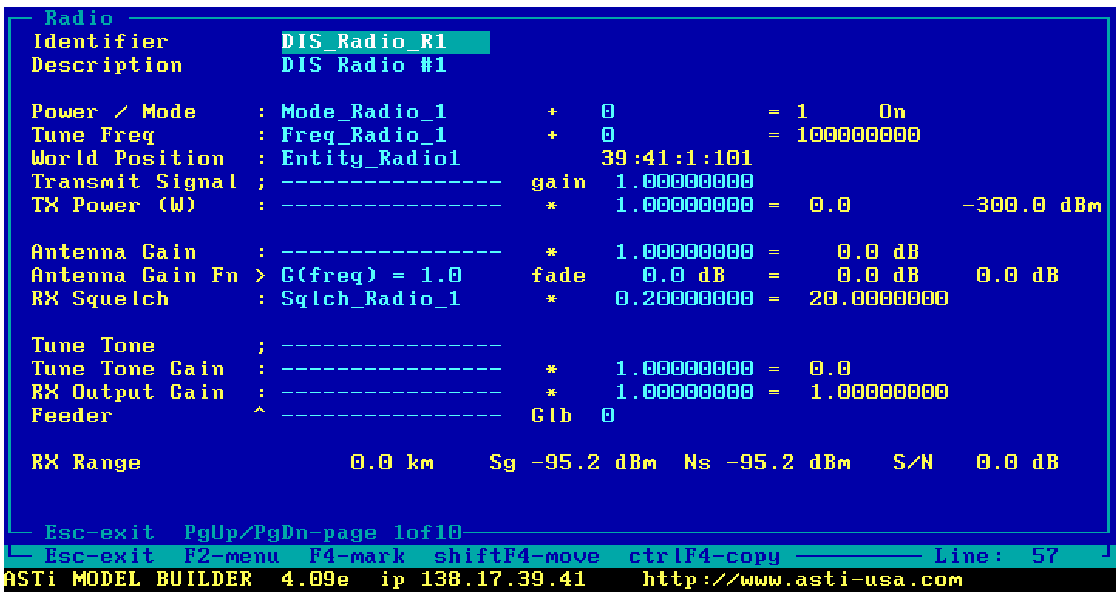


Figure 41: Radio Object Screen

6.1.2 Radio Communication Guidelines

In addition to the three essential variables that make a radio work, there are some guidelines that must be followed in order to make the radio communicate with other radios.

Each radio must:

- Have matching Tune Frequencies
- Be in-range (unless using rangeless radios or net intercoms)
- Have same modulation type
- Have matching crypto, Frequency Hopping or Havequick settings (if used)
- Have a unique ID
- Exist in the same DIS exercise or HLA federate

Once these parameters match, the radios should talk to each other without any problems.

Each radio can have up to 16 modes. These modes include parameters such as: BW, Antenna Gain, AGC, modulation, noise, duplex type, etc. Model Builder also comes with certain pre-defined modes such as: UHF, VHF, Sincgars, etc.

A few notes:

1. If an FM receiver detects two FM Transmitters, the strongest signal will win.
2. If an AM receiver detects two AM Transmitters, the signals will combine (in proportion to their received strengths) - so in essence you will get interference.
3. The Radio Antenna is isotropic. We do not model a directional antenna, however you can modify the gain to simulate an antenna type.
4. The most common Radio System type is generic.
5. A crypto radio can be set to receive plain and encrypted transmissions or just encrypted transmissions.

Complete the Basis Radio Modeling Exercise in section 6.3.1.

6.1.3 Crypto radios

To use crypto settings with a set of radios, the radios must have:

- Matching Crypto Key and Crypto System settings. If these settings do not match, then the signal used for Crypto Tone mismatch will be heard instead of the radio transmission. If the transmitter is set with a Crypto System of 0, this is considered a plain radio. A Crypto Key or Crypto System value of 65535 is used as a wildcard.
- DIS enumeration standard provides values for various Crypto System types (e.g. KY-58, KY-100).
- A user defined crypto library may be inserted into the radio. This is used to model pre-amble, post-amble and mismatch tones. For complete details refer to Application Note 47.

Complete the Crypto Radio Modeling Exercise in section 6.3.2.

6.1.4 Frequency Hopping radios

To use Frequency Hopping settings with a set of radios:

- The radios must be set to system type sgcar or SCGAR (mode 6 by default).
- The Net ID of each radio is non-zero.
- The remaining frequency hopping parameters may be set to enhance the fidelity of the simulation or when communicating with CCTT. These parameters are:
  - Hopset WordDay = # (0 is wildcard)
  - Lockout ID = # (0 is wildcard)
  - Frequency Hopping SyncTimeDay = # (0 is wildcard)
  - Transec key = 0 (0 is wildcard)
- When used, the frequency hopping extensions appear in the transmitter PDU.

In Frequency Hop mode, the tune frequency is fixed and is used for free space path loss calculations and ranging. Radios will be considered in tune if all the Frequency Hopping parameters match. The only exception is that a value of 0, except as the Net ID, acts as a wildcard.

The Frequency Hopping extensions are also applied when the system types are HQ, HQII or HQIIIA.

Complete the Frequency Hopping Radio Modeling Exercise in section 6.3.3.

6.1.5 Communications Panel Object

Communications Panel Objects allow for the following:

- The control operator has access to all radios/intercoms/signals from one object.
- The control of overall volume and sidetone level.
- It eliminates the need for separate output and input objects for every radio/intercom.
- The devices can share input and output in Model Builder through the communications panel.
- Operators can easily share radios and intercoms. Even though radios and/or intercoms are shared each operators interaction in independent.

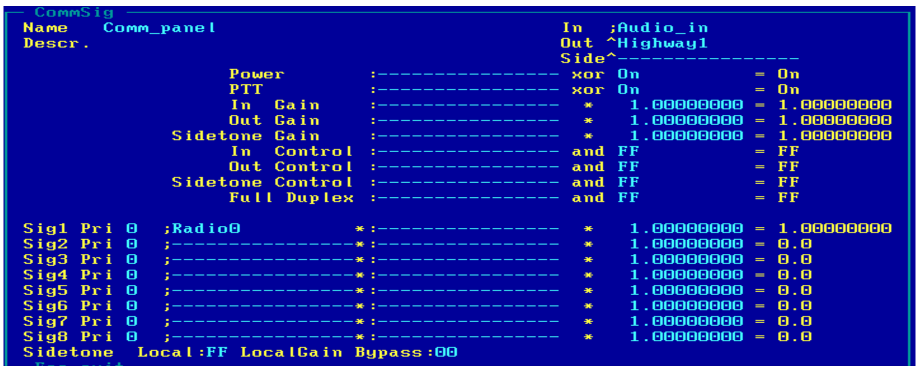


Figure 42: Communication Panel Object

Notes:

- You can establish a shared sidetone so operator one can hear operator two when he/she transmits.
- Both operators can interact with the radios independently for Tx_Rx selections and volume controls.

Complete the Communications Panel Modeling Exercise in section 6.3.4.

Hint:

- Your perspective is from the inside of the communications panel.
  - “IN” is audio into the communications panel directed to a given radio or intercom by the “In Control” selection.
  - “OUT” is audio from the communications panel directed to the operator headset based on the “Out Control” selection.
  - Based on the communications panel settings an operator may transmit on or receive from one, many or all of the radio and/or intercom objects.

6.2 Intercom Communications

The intercom signal is a special object for local (intra DACS) and networked intercom, interphone and Public Address communication simulation. Additionally, the intercom is useful for reliable full duplex audio that does not incorporate propagation effects. The two intercom types are described below:

6.2.1 Local Intercoms

Local intercoms reside on the DACS and simulate aircraft intercoms or are used between operators and instructors.

6.2.2 Network Intercoms

A networked intercom extends the local intercom capability between DACS over the LAN or WAN. Additionally, networked intercoms may be used for DIS communications when full fidelity radios are not required or desired.

Requirements:

1. Needs a DIS world position (simply for identification purposes)
2. Needs a channel number (1-100000)
3. Channel numbers must match to communicate

In general, an intercom object needs the following:

- Must be powered 'On'
- TX signal specified
    - Specific warning signal or vox input
    - Put in a communications panel for automatic routing of audio in
    - Warning gain greater than 0.0 (if warning signal is used)
- Output connection
    - Feeder
    - Put in a communications panel for automatic routing of audio out

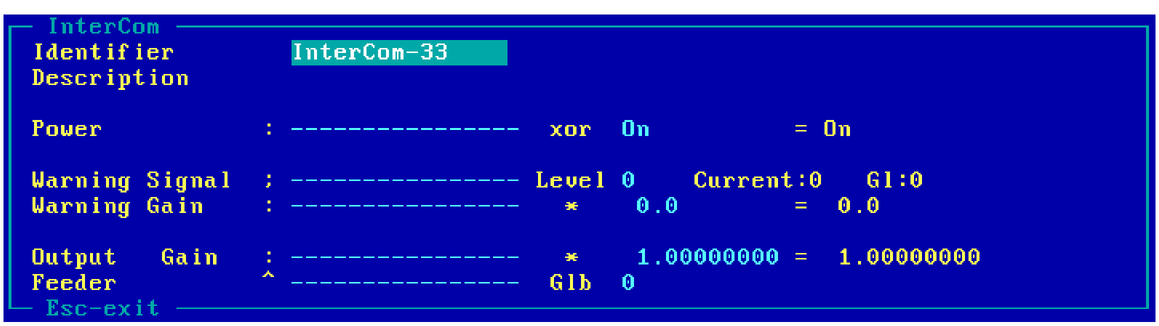


Figure 43: Intercom Settings

The network intercom is very similar to a network radio, only without all the extra radio in-tune features. The DIS setup rules for radios apply to the network intercom as well.

- The power must be on.
- The 'Comm Channel' fields on all intercoms must be the same. Valid values are from 1-100,000. This allows for multiple intercom busses to be set up and used.
- The source entity must be defined (only for DIS ID purposes).

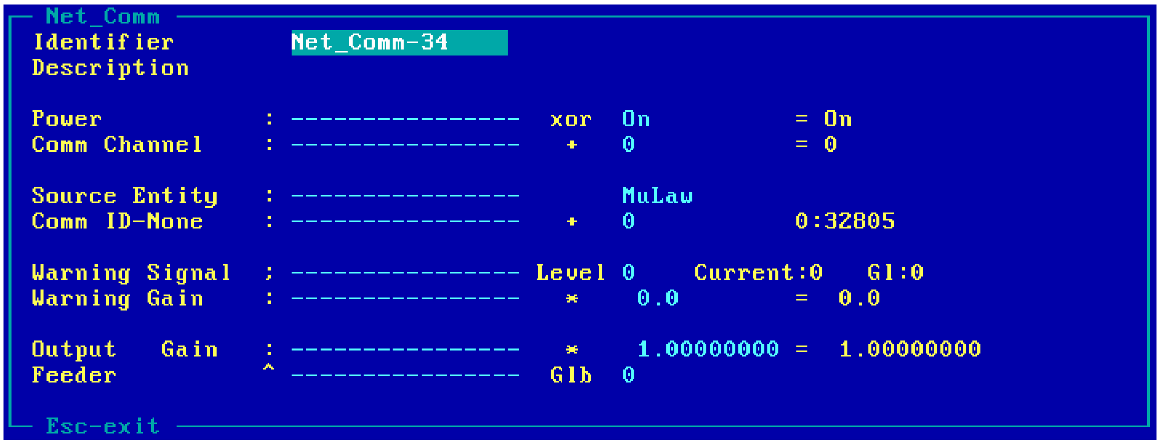


Figure 44: Network Intercom Screen

Complete the Intercom Modeling Exercise and DIS Network modeling exercise in section 6.3.5.

6.3 Hands-on Exercises

6.3.1 Basis Radio Modeling

Purpose: Familiarize the user with the radio object and what it takes to establish communications between several radios.

1. Add Radio object in 'Signals' menu (Receivers > Radio).

   a.Use Page Up and Page Down to move through the model so you can see what the various modes entail.

2. Change the Mode field to '1' and give the radio a tuned frequency (i.e. 30,000,000).
3. Add a World_position object in the 'Controls' menu.
4. Add an Audio_in object in the 'Signals' menu. Set it to receive input from a microphone.
5. Plug the World_position and Audio_in objects into the Radio page.
6. Make sure the Tx power and signal gains are set to a value greater than zero.
7. Select an output feeder and connect it to the radio.

Repeat these steps for a second radio, using a separate audio input and world position object. Make sure the frequency is the same, along with the other parameters discussed on the previous page, then communicate locally (on the same DACS) between them.

8. Save this model as “631_rad.mdl.”

6.3.2 Crypto Radio Modeling

Purpose: Demonstrate the use of crypto in radio simulation.

1. Use the radios from the last example.
2. Set the Crypto System and Crypto Key values to match (i.e. System=1, Key=1).
3. Communicate between the radios.
4. Set one radio with a Crypto Key value of 0 and communicate.
5. By default, an encrypted radio can receive from other encrypted radios (with matching settings) or from plain radios.
6. Radios can be set on a per mode basis as to whether or not the encrypted radio will hear plain transmissions.
7. Save this model as “632_cryp.mdl.”

Use crypto libraries for automatic playing of preamble, post amble, mismatch and other crypto sounds. For more information please read Application Note 47.

6.3.3 Frequency Hopping Radio Modeling

Purpose: To demonstrate the use of simulated Frequency Hopping.

1. Use the same radios from before except set the radio mode to six.
2. Set the Net ID on both radios to the same value (i.e. 99).
3. Communicate between radios. You should be able to transmit and receive on both.
4. Save this model as “633_FH.mdl.”
5. Change the NET ID of one radio to another value. You should not have transmissions between the radios after this change.

6.3.4 Using a Radio with the Communications Panel

Purpose: To familiarize you with the communications panel object and how to use it.

1. Add a single_Comm object from the 'Signals' menu (Intercoms > single Comm).
2. Set the 'In' and 'Out' fields at the top of the screen with the Audio_in and the output feeder from the previous exercise. (Remove the feeder from the VOX object.)
3. Insert the radio object into the first 'Signal' field at the bottom of the screen.
4. Make sure the "In" and "Out" gains are set to non-zero values.
5. You should now be able to communicate on the radio and use the controls in the communications panel instead of the controls in the radio and/or signals.
6. Save this model as “634_comm.mdl.”

You can also add Boolean control objects and float objects to control on/off states, and volumes in the communications panel.

6.3.5 Intercom Modeling

Purpose: To familiarize you with intercom objects in Model Builder.

1. Insert an Intercom object into Model Builder from the 'Signals' menu.

   a.(Intercoms >Intercom)
2. Change the 'Power' field to 'ON.'
3. Use the communications panel to route operator audio to/from the intercom bus.
4. Create a second operator input and communications pane.
5. Have the two operators communicate over the intercom bus.
6. Save this model as “635_intc.mdl.”

6.3.6 DIS Networking Modeling

Purpose: To familiarize you with very basic DIS Radio communications.

- Connect two DACS units together by the DIS interface. Make sure DIS is enabled in the configuration file and that each unit has an RIU, microphone, and headset.
- Set up a simple radio model on each DACS as described previously.
- In the DIS Protocol Status Window, make sure the TX and RX UDP ports match, and that each DACS has a unique local IP address and a valid broadcast IP address.
- Set the DIS exercise number to 1 on all world positions.
- Ensure that the world position is set to use "DIS."
- You should be able to communicate via DIS from one DACS to another.
- Save this model as "636_rnet.mdl."

Follow-up:

- Look at the DIS status page and observe the various counters.
- Look at the "Frequency of Transmitters" page so you can see the local and networked assets.

6.3.7 Review of Chapter 6 Models

The following list reviews the models covered in this chapter.

- Radios must have a frequency, mode and world position in order to operate.
- Radios must have a transmit signal in order to go Active Tx (may be through communications panel).
- Radio Parameters must match in order for them to communicate.
- Local intercoms apply to a single DACS.
- Network intercom are very similar to a network radio, only without all the extra radio in-tune features.

Below are diagrams for the models created in the previous sections:

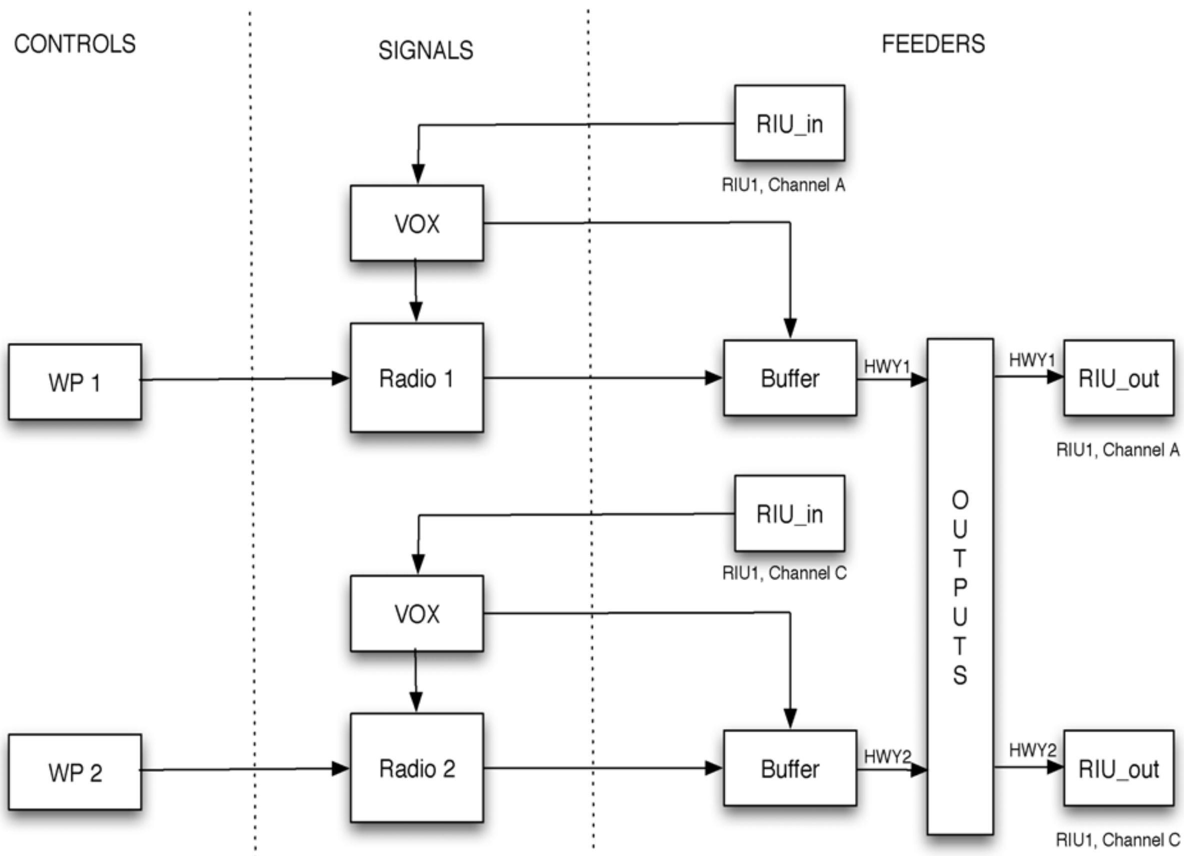


Diagram 7: Exercises 6.3.1 - 6.3.3 - Radio Model

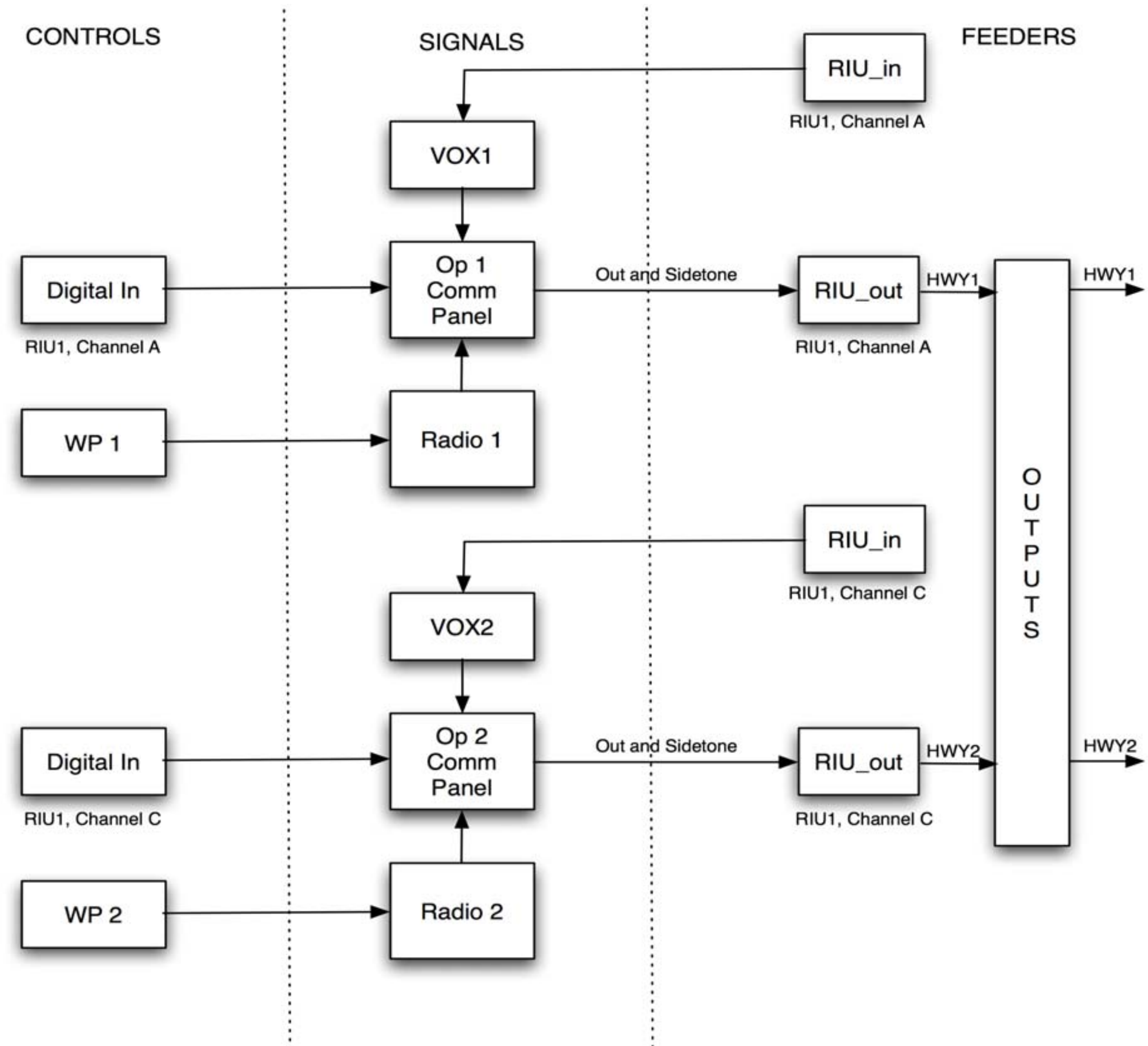


Diagram 8: Exercise 6.3.4 - Communication Panel for Operators 1 and 2

Notes: The radio diagram above is good for the plain, crypto, and Frequency Hopping radios. The only difference is in the radio object parameters. The crypto radio requires the system and crypto keys to be set. The Frequency Hopping radio requires the mode to be set to Frequency Hopping and the Net ID not equal to 1 (plus various optional parameters).

The communications panel diagram shows that a single communications panel object is used for one operator. As you can see, there is one place for audio in and out of the communications panel. This audio is sent and received from the communications panel signals, in this case just one radio. However, you can see how a single communications panel can be used to control multiple signals. The Tx control is done through the PTT.

For the DIS Network radios example, the most obvious change is the world position object must be set to DIS. This will cause the DACS to send the radio information out over the DIS network. In addition, the DIS network parameters must also be set up correctly. Such as DIS exercise number, IP addresses, Tx/Rx UDP ports, etc.

7. Troubleshooting with DACS and Model Builder

This section is broken into possible problems with the DACS and Model Builder, and the recommended solutions for each. This is in no way a complete list of all possible problems, but should help you get started in the right direction. Many of these screens and their contents have been covered in previous sections of the training course, but not from this perspective. For some of the individual screens, files, etc. referenced in this section you should consult other documents or other sections of this training manual for further information.

7.1 Startup Errors or Missing Screens/Functions in Model Builder

If you received startup errors when Model Builder launched, or you are missing certain functions (DIS, HLA, etc) that you expect to have, then most likely you have a file(s) configuration issue. The following should be checked:

- Config.sys (if using RMS)
- Default.cfg
- Option file
- Ini file if applicable
- Within Model Builder are there any error messages?
- Within Model Builder are there any unexpected or missing lines in the Status Page?

7.2 Model Issues

This is a broad question and could be any number of things, but at a basic level you should check the following:

- Check allocation of DSP and CPU resources to ensure no overruns are occurring
- Look for DSP error
- Check RIU status
- Check cabling
- Check Status LEDS on RIUs and DACS
- Balance model rate and frame rate

Try running the built in Model Builder utility such as the daily readiness test. This will help verify Audio In/Out, DI/DO, PTT, VOX, etc. and will force you to eliminate components in the system. The utility is described in detail below, and can be run from the C prompt by typing “dred_riu”.

- Dred_Riu Model Details
  - It requires a headset to be connected to the RIU under test.
  - Tests audio output from each channel, using a different tone for each.
  - Tests audio in and digital in for PTT switches.
  - Allows subjective evaluation of audio quality on each RIU channel.

If this utility runs successfully, we know that the hardware and Model Builder are good, and that the problem exists in the model or model configuration files specific to what you are trying to do.

7.3 TDM/RIU Issues

- Check TDM cabling
  - The Most common cause of failure is customer manufactured cabling. Try testing the RIU with a store-bought cable.
- Check RIU operation, gains, phantom power and LEDs
- Check headsets, cabling + grounding scheme
- Check RIU status in Model Builder
- Does the Dred_Riu test run OK?
- Is the path file correct? (Be careful not to change unless absolutely sure - Please contact ASTi at 703-471-2104.)

7.4 DIS Interface Issues

- Can you ping the DACS(s) in question (from external device)?
- Are the network settings in the configuration files (default.cfg and config.sys) correct?
- Are there any obvious errors on the DIS Status page?
  - No Tx/Rx/Ent/Sig PDUs
  - Packet errors
  - Collisions
  - FIFO errors, etc.
- Are the Ethernet LEDs on the DACS showing activity?
- Are there network components in between DACS(s) that could be causing issues?
- Can you see the Frequency of transmitters from your DACS and/or other DACS?

7.5 Host Interface Issues

- Can you ping the DACS(s) in question (from external device)?
- Are the network settings in the configuration files (default.cfg and config.sys) correct?
- Are there any obvious errors on the Host Interface Status page?
  - No Tx/Rx packets
  - Packet errors
  - Collisions
  - FIFO errors, etc.
- Are the Ethernet LEDs on the DACS showing activity?
- Are the Ethernet Control settings right?
  - Are UDP ports set up correctly?
  - Are you receiving packets on the specified UDP port?
  - Are you transmitting out on the specified UDP port?
  - Is the packet length defined to be equal to or greater than 60 bytes?
- Are there network components in-between DACS and host that could be causing issues?

7.6 Radio/Intercom Reception and Transmission Issues

- Are the radios/intercoms in the same exercise?
- Does the radio have the three basics so you can talk?
- Do you see far/near end radio in frequency of transmitters list?
- Do crypto/Frequency Hopping/HQ settings match?
- Are the Radios in range? (Do you see Rx from etc., on radio object?)
- Do parameters in the Tx PDU such as BW and modulation type match?

7.7 Voice Break-up Issues

- Check cabling and headsets
- Check CPU and DSP loading
- Check amount of network traffic
- Check if radios are going in and out of range

8. Customer Specific material (if applicable)

9. Glossary of Terms and Acronyms

8AFA	Waveform Synthesizer - previously called DSP
AAR	After Action Review
ABCCC	Airborne Battlefield Command and Control Center
ACBL	Audio Cable
AFB	Air Force Base
AFRL	Air Force Research Lab
AFV	Armored Fighting Vehicle
AI	16/32 Channel Analog Input board (discontinued)
AIU	Audio Interface Unit
ALE	Automatic Link Establishment
AMS	AWACS Modeling & Simulation system
ATC	Air Traffic Control; Air Training Command
ATP	Acceptance Test Procedure
ATRC	AEGIS Training and Readiness Center
AVCATT-A	Aviation Combined Arms Tactical Trainer - Aviation reconfigurable Manned Simulator
AWACS	Airborne Warning and Control System
AWE	Army Warfighting Experiment
BSC	Battle Simulation Center
C2V	Command & Control Vehicle
CATT	Combined Arms Tactical Trainer
CCTT	Close Combat Tactical Trainer
CDR	Critical Design Review
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CRR	Customer Repair Return
CSMET	Crew Station Mission Equipment Trainer
CVSD	Continuous Variable Slope Delta; a type of audio compression
DACS	Digital Audio Communications System
DDI	Distributed Digital Interface

DI	16 Channel Digital Input Board
DI/DO	8In/8Out Digital board
DIS	Distributed Interactive Simulation
DMSO	Defense Modeling and Simulation Office
DMT	Distributed Mission Training
DR	Discrepancy Report
DRED	Daily Readiness Test
DSP	Digital Signal Processor
DTED	Digital Terrain Elevation Data
EW	Electronic Warfare
FAA	Federal Aviation Administration
FFS	Full Flight Simulator
FOB	Freight On Board
FOM	Federation Object Model
FTX	Field Training Exercise
HF	High Frequency
HHT	Hand Held Terminal
HLA	High Level Architecture
HQ	Have Quick
HRTF	Head Related Transfer Function; 3D Audio reference
I/ITSEC	Interservice/Industry Training, Simulation and Education Conference; trade show (Orlando, FL)
ICD	Interface Control Document
IDM	Integrated Data Modem
IFF	Identify Friend / Foe
JANUS	An interactive, entity-based, tactical simulation used to train at Brigade and below (Battle-Focused Training)
JCSAR	Joint Combat Search and Rescue
JEFX	Joint Expeditionary Force Experiment
JNTF	Joint National Test Facility
JPATS	Joint Primary Aircraft Training System

JSAF	Joint Semi-Automated Forces
JSF	Joint Strike Fighter
JSTARS	Joint Surveillance and Target Attack Radar System
JTASC	Joint Training Analysis and Simulation Center
JTDL	Joint Tactical Data Link
JTIDS	Joint Tactical Information Distribution System
JTMD	Joint Theater Missile Defense
JTMDAO	Joint Theater Missile Defense Attack Operation
MB	Model Builder software
MCAS	Marine Corps Air Station
MCASMP	Marine Corps Aviation Simulator Master Plan
MLU	Mid-Life Update
MODSAF	Modular Semi-Automated Forces
MRD	Mission Rehearsal Device
MTBF	Mean Time Between Failure
Model	Collection of Objects constructed by user using the Model Builder application tool to build sound and/or communications simulation of their application. Models are run within the Model Builder application environment on the DACS. Each model is associated with a specific DSP card. One or more Models may reside on DACS depending on the hardware configuration. Models can interact locally to the DACS or across the network with other DACS models.
Model Server	Runs state machine software (legacy software and hardware)
MuLaw	G711 ITU Standard, a type of audio compression
Multicast Router	Device that bridges specific multicast address between different interfaces
NAS	Naval Air Station
NAWC/AD	Naval Air Warfare Center, Aircraft Division
NDIA	National Defense Industrial Association
NSWC	Naval Surface Warfare Center
NSWCDD	Naval Surface Warfare Center Dahlgren Division
NTSA	National Training and Simulation Association
OFT	Operational Flight Trainer
OneSAF	a composable, next generation Computer Generated Force

OMDT	Object Model Development Tool
OML	Object Model Library
OMT	Object Model Template
ORTT	Operations Room Team Trainer
PCA	Program Configuration Audit
PDR	Preliminary Design Review
PDU	Protocol Data Unit
PEO STRI	Program Executive Office - Simulation, Training and Instrumentation (formerly STRICOM)
PIU	Panel Interface Unit
PTT	Push To Talk (switch)
RID	RTI Initialization Data Editor
RIU	Remote Interface Unit
RMA	Return Material Authorization
RPR FOM	Real-Time Platform Reference Federation Object Model
RMS	Browser-based Local & Remote DACS Systems Management: The ASTi Remote Management System (RMS) gives the user a Gods-eye view and control of all ASTi devices and activity on the exercise-wide network via a standard web browser.
ROM	Rough Order of Magnitude
RTI	Runtime Infrastructure
RTI NG	RTI Next Generation (After Version 1.3)
SED	Shippers Export Declaration
SHINCOM	SHips INternal COMmunications
SimTecT	Simulation Technology and Training; trade show (Australia)
SINCGARS	Single-channel Ground and Airborne Radio System
SOF	Special Operation Force
SOM	Simulation Object Model
SOW	Statement of Work
STRICOM	Simulation Training and Instrumentation Command (see PEO STRI)
STOW	Synthetic Theater of War

Synapse	Plug-and-play link between live field radios and virtual communications networks
TACAN	Tactical Air Navigation
TACCSF	Theater Aerospace Command and Control Simulation Facility
TACOM	Tank-automotive & Armaments Command
TADIL	Tactical Digital Information Link
TDM	Time Division Multiplex
TDM Ring	Connection from DACS to RIUs and back to DACS; no longer than 300 feet total
Telestra	Native HLA radio simulation & communications module
THAAD	Theater High Altitude Air Defense
Ts & Cs	Terms and Conditions
UDP	User Datagram Protocol
UHF	Ultra High Frequency
UTD	Unit Training Device
VHF	Very High Frequency
VRR	Vendor Repair Return
WARSIM	Warfighters' Simulation
WST	Weapons System Trainer

10. Class Notes