

FAA Interfacility and Radar Simulator (FIRS)

JVN Communications, Inc.
Copyright © 1999 – 2005

User's Manual

1st Ed. Rev 4
Aug 16, 2005

Prepared by:

JVN Communications, Inc.
707 White Horse Pike, unit B6
Absecon, NJ 08201

Table of Contents

1.0 SYSTEM REQUIREMENTS	4
1.1 SOFTWARE	4
1.1.1 <i>Operating System</i>	4
1.1.2 <i>Graphical User Interface (GUI)</i>	4
1.2 HARDWARE	4
1.2.1 <i>Processor/Motherboard</i>	4
1.2.2 <i>Memory</i>	4
1.2.3 <i>Disk Storage</i>	4
1.2.4 <i>Video</i>	4
1.2.5 <i>Interface Adapters</i>	4
2.0 OVERVIEW	5
3.0 FIRS SOFTWARE	6
3.2 APPLICATION SOFTWARE	6
3.3 SIMULATION REQUIREMENTS.....	6
3.3.1 <i>FIRS Input</i>	6
3.3.2 <i>FIRS Output</i>	6
3.3.3 <i>Script File</i>	7
3.3.4 <i>Modules</i>	7
3.4 ATC SIMULATION OBJECTS.....	7
3.4.1 <i>Radar Recording</i>	7
3.4.2 <i>Radar Playback</i>	7
3.4.3 <i>Host</i>	7
3.4.4 <i>Arts</i>	7
3.4.5 <i>Radar Planned Position Indicator (RAPPI) Display</i>	7
3.4.6 <i>Interfacility Monitor</i>	7
3.4.7 <i>Host Central flow Simulation</i>	8
3.4.8 <i>ETMS</i>	8
3.4.9 <i>ADSB Recording</i>	8
3.4.10 <i>ADSB Playback</i>	8
4.0 FIRS HARDWARE	8
4.1 HARDWARE CONFIGURATIONS.....	8
4.1.1 <i>SDL RISCCom 8si</i>	8
4.1.2 <i>Sunhillo ICA (Formerly Emulex 486 DP)</i>	9
4.1.2 <i>Sunhillo ICA (Formerly Emulex 486 DP)</i>	10
5.0 PROCEDURES	12
5.1 SYSTEM INITIALIZATION.....	12
5.2 SHUTTING DOWN THE SYSTEM.....	13
5.3 CREATING A RADAR RECORD SCENARIO.....	13
5.4 CREATING A RADAR PLAYBACK SCENARIO.....	13
5.5 CREATING A HOST SCENARIO.....	14
5.6 CREATING AN ARTS SCENARIO.....	14
5.7 CREATING A ADSB RECORD SCENARIO.....	14
5.8 CREATING A ADSB PLAYBACK SCENARIO	15
5.9 CREATING A RAPPI DISPLAY.....	15
5.10 RUNNING A SCENARIO.....	15
5.11 RESTARTING A SCENARIO.....	15
5.12 EXITING A SCENARIO.....	16
5.13 RECORD AND PLAYBACK USING THE PATCH PANEL.....	16

6.0 UTILITIES 18

- 6.1 OFFLINE UTILITIES 18
- 6.2 FIRS UTILITIES 19
 - 6.2.1 *FIRS Automated Scenario Tool (FAST)* 19
 - 6.2.2 *Record* 19
 - 6.2.3 *Playback* 24
 - 6.2.4 *Manager* 25

7.0 TROUBLESHOOTING THE SYSTEM26

- 7.1 BOOT MESSAGES 26
- 7.2 TROUBLE SHOOTING GUIDE..... 26

8.0 APPENDICES27

- 8.1 APPENDIX A – SDL 27
- 8.2 APPENDIX B – SUNHILLO (EMULEX) 27
- 8.3 APPENDIX C – RDI 29
- 8.4 APPENDIX D – SIMULATION OBJECTS..... 29
- 8.5 APPENDIX E – SCRIPT FILE EXAMPLES..... 39
- 8.6 APPENDIX F – MODULES..... 40
- 8.7 APPENDIX G – SERIAL INTERFACES..... 40

9.0 CABLE DIAGRAMS42

10.0 HARDWARE SPECIFICATIONS50

1.0 SYSTEM REQUIREMENTS

1.1 Software

1.1.1 **Operating System**

The Linux Operating System Kernel version 2.0.35 to 2.0.38 is required. Version 2.4.20 is the highest version supported.

1.1.2 **Graphical User Interface (GUI)**

The X window system version 11 Revision 6 (X11R6) or higher is required for many of the tools and utilities.

1.2 Hardware

1.2.1 **Processor/Motherboard**

An Intel Pentium 100 MHz or higher processor with an AT, ATX, or passive-backplane motherboard is required.

1.2.2 **Memory**

A minimum of 32 Megabytes of memory is required to run the FIRS software.

1.2.3 **Disk Storage**

A minimum of 50 Megabytes of disk storage space is required for installation of the FIRS software. Any pre-recorded scenarios or surveillance files require additional space. Short-range radar surveillance files can be quite large, 13-15 MB per ASR-9 for 1 hour @ 650 targets.

1.2.4 **Video**

Minimum
8 bit
640 X 480 resolution

1.2.5 **Interface Adapters**

An Ethernet network interface card is recommended, but not required for stand-alone units. An Ethernet connection to the Internet (or the FAA Technical Center LAN) is especially useful for remote debugging and configuration by JVN.

A SCSI interface card is needed when using magneto-optical disks for storage of surveillance files and it is also necessary when using the Lynx file system driver for reading CDR data.

Serial interface cards such as the SDL RISCCom 8/si, the RDI U.S. Air force, the Computer Modules modem pump card, or the Emulex 486p PCI card. One ISA or PCI slot per interface card is required.

2.0 OVERVIEW

The FAA Interfacility and Radar Simulator (FIRS) is an economical and robust test tool, based on a PC hardware platform designed around industry proven hardware, running the Linux operating system. FIRS hardware and software can be easily configured to support a variety of Air Traffic Control (ATC) real-time simulations and test functions. FIRS can be used to debug external and internal cabling, assess hardware and software performance, assess system capacity and measure target throughput response time.

The FIRS can emulate multiple, full duplex, ATC data communications interfaces (data format, speed, and electrical characteristics). This allows the FIRS system to simulate, record and playback Interfacility and/or NAS Host communications, short range Airport Surveillance Radar's (ASR), and the long range Air Route Surveillance Radar's (ARSR), in a real-time environment. The FIRS system comes with offline utilities to convert ASR-9, ASR-8, and ARSR messages from CDR or LYNX data recordings into the FIRS file system format.

By utilizing a SCSI interface, FIRS can be configured with high capacity storage devices like magneto optical storage devices. This gives the customer the flexibility of removable media and the capacity to efficiently store, analyze, manipulate and retrieve large amounts of data. The economy and flexibility of the FIRS system make it an invaluable tool to aviation research facilities and government agency support groups.

The communications cards, operating system and associated applications and drivers are pre-installed and tested by JVN Communications prior to shipment. The FIRS tool is a turnkey system. External cabling is the customer's responsibility. Serial interface (RS-422/RS-232) cables that terminate at the FIRS system are required to be DB-25 female connectors. JVN Communications can supply external cabling at customer's request.

3.0 FIRS SOFTWARE

3.1 Operating System

The FIRS system runs under the Linux 2.0.x operating system. Kernel version 2.4.20 is the highest supported version. Linux is a free Multi-user, multi-tasking, POSIX compliant operating system.

3.2 Application Software

The FIRS application software is object oriented. Each modeled ATC object encapsulates the functionality of major ATC systems. Each ATC object utilizes communications channels to simulate real-world ATC interfaces. Custom device drivers written by JVN Communications handle all ATC communications protocols. These device drivers are dynamically loaded into the Linux Kernel at boot time.

JVN communications has designed the FIRS software tool to be easily configured to suit a variety of ATC test scenarios. A scenario is defined in an ASCII text script. The scenario script file is used by the application Xtest to run the ATC scenario. The script contains commands to create ATC objects and assign communications devices to each object. Each object is independent and may be added to or deleted from a script without affecting any other object. All objects have access to the master Simulation time to provide synchronization between objects. The most commonly used ATC object's record and playback radar data and simulate Host communications. The FIRS software tool provides a full-fledged scripting capability based on the tcl language and utilizes X-windows for a user-friendly graphical user interface from which all ATC simulations can be run.

3.3 Simulation Requirements

In order to run a simulation there are four general requirements. They are FIRS Input and Output, Script File and Modules.

3.3.1 *FIRS Input*

A FIRS simulation requires input. This input may be live data received from a communications card or input files. The input files must be pre-processed ahead of time and included in the script file. They may include any of the following:

- *Radar Data* - The radar file consists of timed stamped messages. This file is typically obtained by recording radar data with a FIRS tool. Other methods include converting a pre-recorded file, obtained by other means, to the FIRS format via an off-line utility, or by using a target generator such as the FIRS Radar Simulation Software (RSS) utility, to create a pre-recorded file.
- *Interfacility Data* – All flight plans, amendments, and cancellations are placed in an ASCII file, with the source field modified to indicate the time to send the message. All data will be converted to EBCDIC and sent along with internally generated parity and LRC bytes.
- *Weather Data* - The weather file may consist of ASR-9/ASR-11 six level weather messages and status control messages from the local Surveillance and Communications Interface Processor (SCIP). This file is obtained using the same methods as the radar file.

3.3.2 *FIRS Output*

The FIRS simulation requires an output device, which may be a communications card, a file or a communications card in a remote FIRS tool.

3.3.3 **Script File**

Script files create and configure the necessary ATC objects to run simulations. Depending upon the scenario, various combinations of ATC objects are incorporated into a script. Script files may be created or edited using a text editor to customize scenarios to meet particular requirements.

3.3.4 **Modules**

Modules are loadable device drivers that are added to or removed from memory when the system is booted up. The FIRS Air Traffic Control modules (ATC) contain the code that allows the communications cards to send and receive data in an ATC scenario.

Typing in the command *lsmmod* from a xterm window will print to the screen the modules that are currently loaded into memory. With the module commands, *rmmmod* and *insmod*, modules can be unloaded and loaded without rebooting. At system initialization the modules that are configured in the *rc.local* script will be loaded into memory.

Located in the */etc/rc.d* directory is the *rc.firs* script that contains the commands that automatically load the ATC modules.

3.4 **ATC Simulation Objects**

There are online manual pages that give detailed usage for each simulation object. For example, typing *man RadarRec* will output to the screen the manual pages for the RadarRec ATC object.

(See Appendix D – Simulation Objects)

3.4.1 **Radar Recording**

The **RadarRec** object is a record object. It will receive radar data from a specified communications card and record, time-stamped radar messages into a file. The time-stamp if not set by the user in the record script will default to the system time. The RadarRec object can be created as a short range or long range radar. The recorded file can be played back by RadarSim object. RadarRec will record all received radar messages to a file. This file is the input file for other simulations like RadarSim, Qars and Host.

3.4.2 **Radar Playback**

The **RadarSim** object is a playback object. It plays back a pre-recorded radar file, optimally starting from a user specified start time. It functionally acts as ASR or ARSR output. It will transmit radar data from an input file, through a specified communications card.

3.4.3 **Host**

The **Host** object emulates an ARTCC host. It requires radar input from a file or communications card. The Host object supports Interfacility data communications, either a live terminal system or simulated FIRS Arts object. All Host-Arts Interfacility messages are supported, including flight plan, amendment, cancellation, track initiate, track update, and track accept in both directions, and VFR flight plans. Multiple Host objects may be created in a single scenario.

3.4.4 **Arts**

The **Arts** object emulates a Terminal system. It requires input from a file or communications card. The Arts object supports Interfacility communications to a Host system, either a live ARTCC host system or a simulated FIRS Host object. The Arts object also supports Arts-Arts processing.

3.4.5 **Radar Planned Position Indicator (RAPPI) Display**

qars creates a RAPPI display to view beacon targets of a designated radar.

3.4.6 **Interfacility Monitor**

The *ifrec* object monitors interfacility

3.4.7 **Host Central flow Simulation**

cenflowsim object

3.4.8 **ETMS**

etms object

3.4.9 **ADSB Recording**

The *AdsbRec* object is a record object. It will receive ADSB data from a specified network card and record, time-stamped ADSB messages into a file. The time-stamp if not set by the user in the record script will default to the system time. The recorded file can be played back by *AdsbSim* object. *AdsbRec* will record all received *Adsb* messages to a file. This file is the input file for *AdsbSim*.

3.4.10 **ADSB Playback**

The *AdsbSim* object is a playback object. It plays back a pre-recorded ADSB file, optimally starting from a user specified start time. It will transmit ADSB data from an input file, through a specified network card.

4.0 **FIRS HARDWARE**

The FIRS system can be configured in portable, desktop and rack mount configurations. This versatility makes the FIRS system ideal for many ATC system design, test and support functions. Because FIRS utilizes industry standard hardware it is easily integrated into the customer's existing ATC equipment.

FIRS hardware supports most major ATC interfaces. The following is a general list of communication interface adapters (see Hardware Configurations) that the FIRS system utilizes for ATC simulations.

The *RISCom 8si*, *DCP-486p* and *RDI* cards support radar and interfacility simulations.

A maximum of two radars or eight interfaces per card is available.

The following ATC radars and interfacility interfaces are supported.

- **ASR9** – Synchronous, RS-422 serial.
- **ARSR** – Synchronous, RS-232 serial.
- **Interfacility** – Synchronous, RS-232

The *Syscon* and *Sabtech* cards support SRAP/SCIP to ARTS simulations. The following ATC interfaces are supported.

- **SRAP** – NTDS (MIL-STD-1397C)
- **SCIP** – NTDS (MIL-STD-1397C)

4.1 **Hardware Configurations**

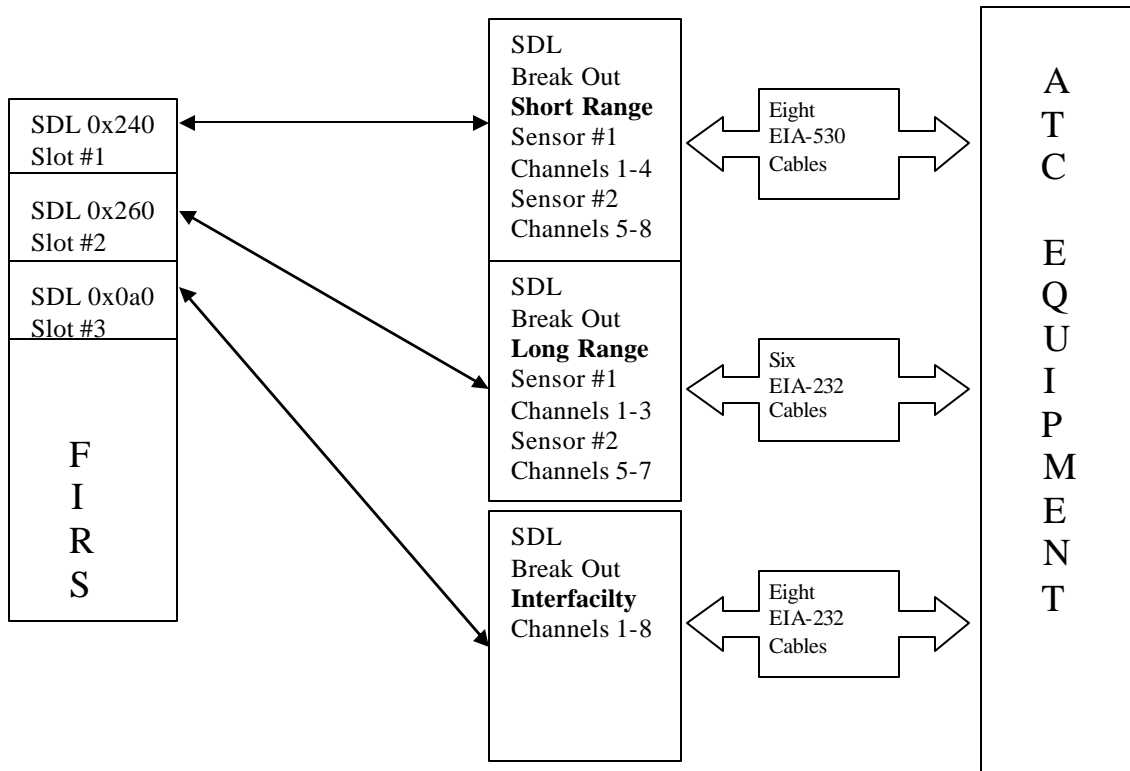
4.1.1 **SDL RISCom 8si**

The FIRS/SDL interface to the ATC equipment is accomplished through the SDL 100 pin AMP cable, hardware adapter and break out box. The standard serial interface from the ATC equipment is connected to an SDL hardware adapter and terminated at the FIRS SDL break out box.

Before installing an SDL card, each card must be configured with a unique I/O base address (See Appendix A – SDL)

Install the SDL interface card in an available ISA slot.

Diagram 4.1: Typical FIRS/SDL – System Level Block



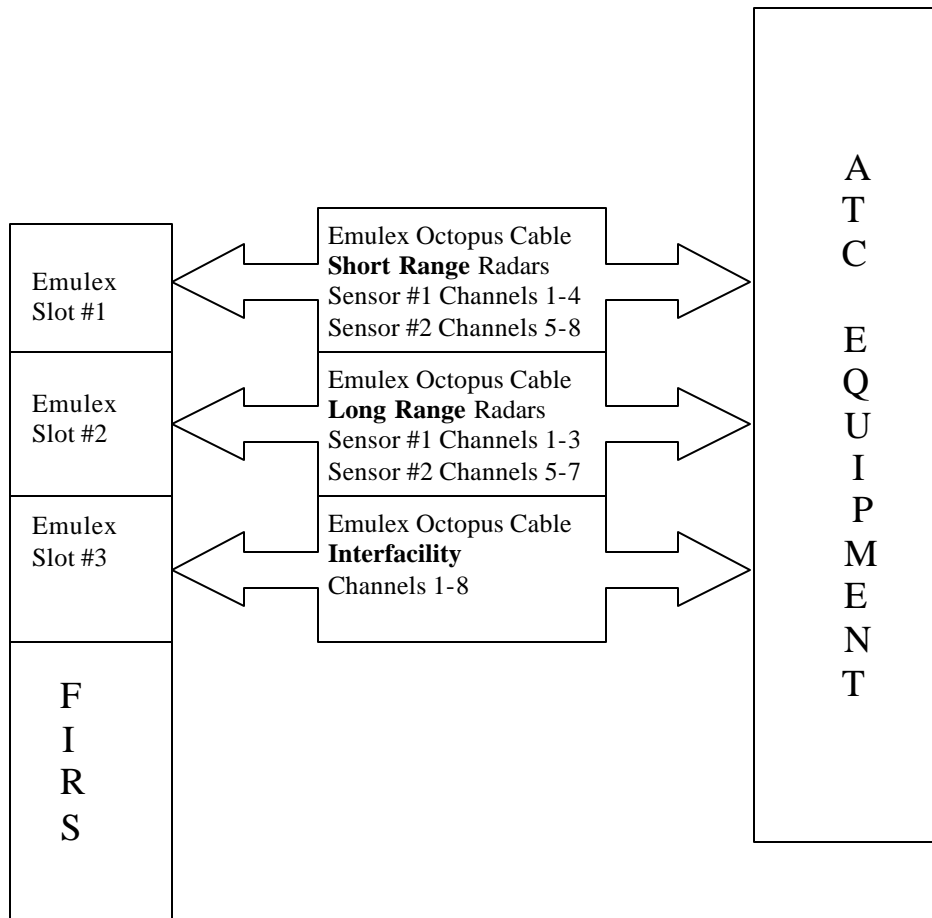
4.1.2 Sunhillo ICA (Formerly Emulex 486 DP)

The FIRS/Emulex interface to the ATC equipment is accomplished through an octopus cable. The standard serial interface from the ATC equipment is terminated at the Emulex octopus cable harness.

Determine available PCI slots. Before installing card, make sure the card has been configured, by JVN Communications, for transparent mode operation and that each of the eight channels are configured for DTE, DCE or split clock mode, depending on port function; e.g. playback, record or interfacility. (See Appendix B – Emulex)

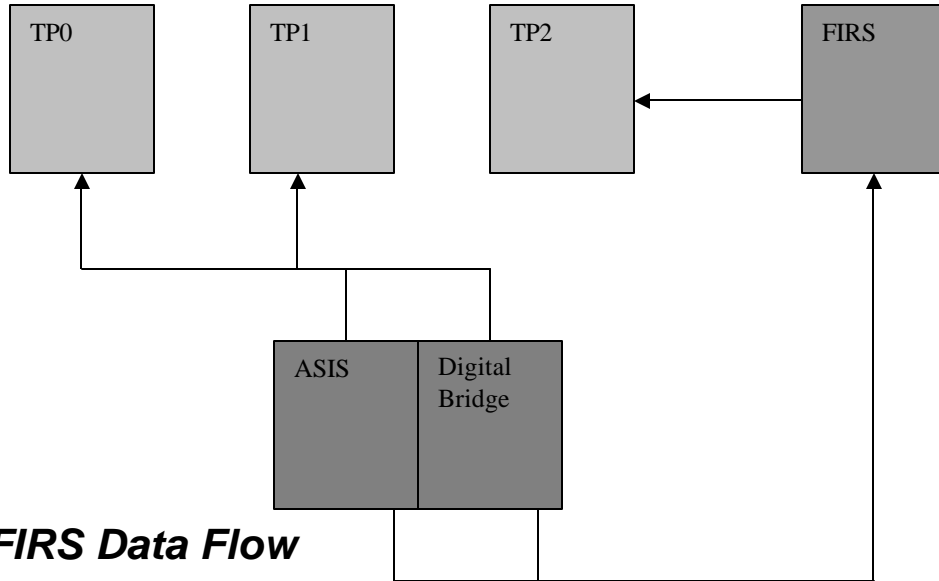
Install the Emulex card in an available PCI slot.

Diagram 4.2: Typical FIRS/Emulex – System Level Block

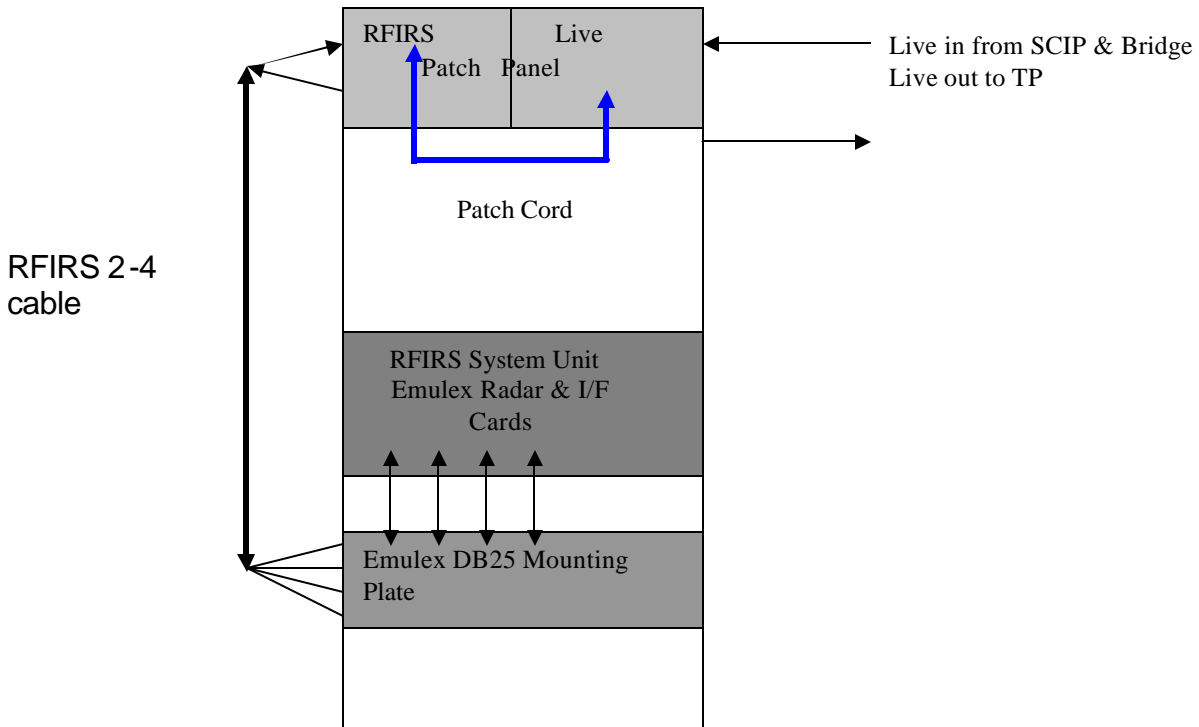


Hardware Diagram

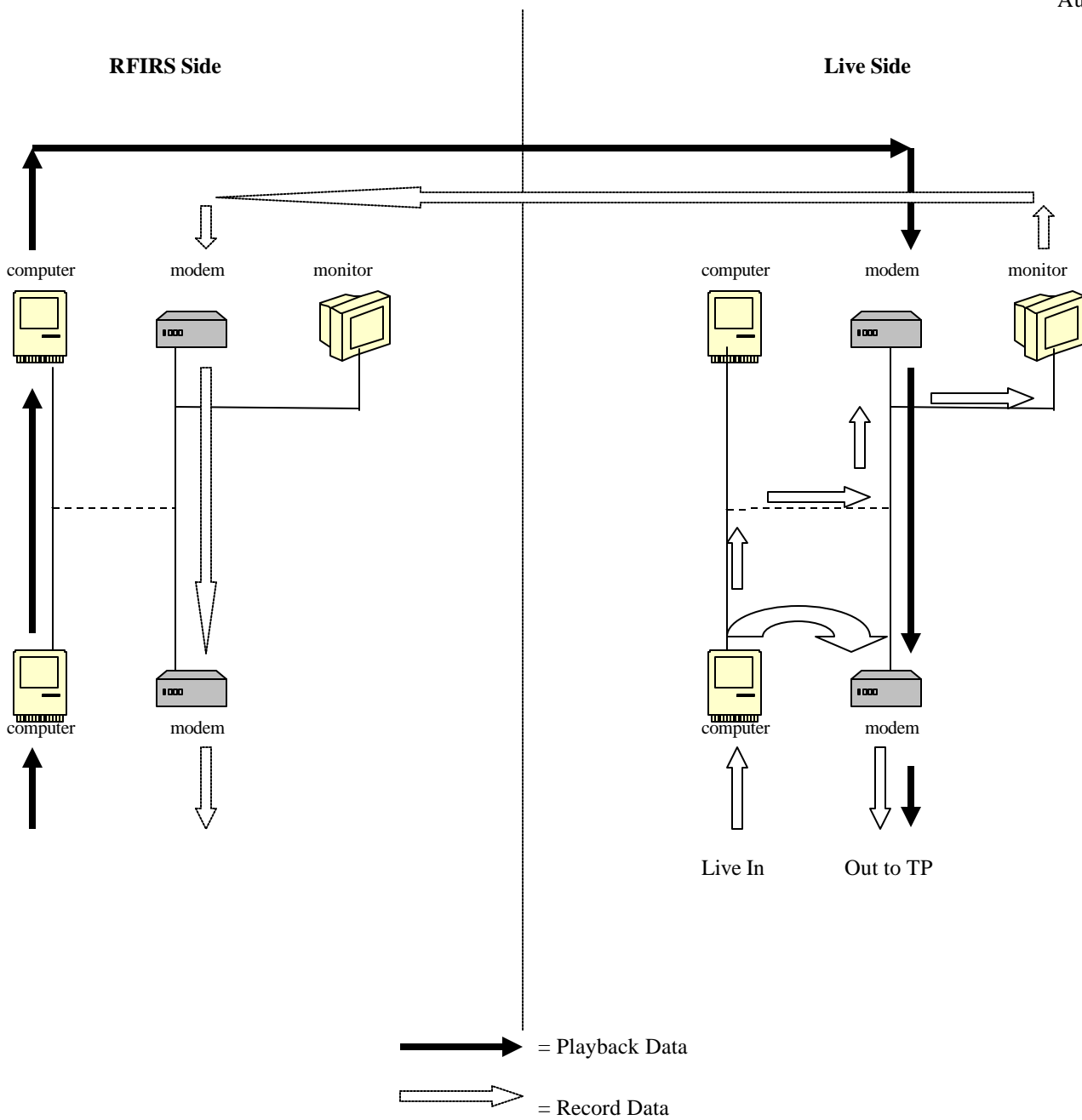
FIRS Installation Overview



FIRS Data Flow



Patch Panel Data Flow



5.0 PROCEDURES

5.1 System Initialization

A cold boot or power up causes the system to output boot messages to the screen. After the operating system is loaded the system prompts the user to login.

Logged in as root, a warm boot is accomplished by typing in the command "**shutdown -r now**". After the boot messages are printed to the screen the login prompt should appear. Another method to perform a warm boot is to simultaneously depress the Ctrl-Alt-Del keys.

5.2 Shutting Down the System

Logged in as the root user or super user, from the root directory simply type in the command **halt**. Linux will now clear buffers and halt the system so you can power down the system without corrupting any data. Typing in the command "**shutdown -h now**" will also initiate a graceful halt.

5.3 Creating a Radar Record Scenario

Using a text editor create a script file. The script file must have at a minimum the following two lines:

1. **set SimTime** <time>
2. **RadarRec** <object name> <options>

The **RadarRec** object will record short range and long range radars.

The parameters **-dev** and **-srv** must be entered using the convention described below. The parameter **-dev** identifies the channels on the communications card that receives data to record. Radar's *srr0* (short-range) and *lrr0* (long-range) are the first logical radars. There are four channels per short-range radar and three channels per long-range radar. A maximum of two radars per card may be received. The parameter **-srv** identifies the path and surveillance file that the radar data will be recorded to.

RadarRec test0 **-srv** <srv file directory path>/test.srv **-dev** dev/srr0

RadarRec test1 **-srv** <srv file directory path>/test.srv **-dev** dev/srr1

(See Appendix E – Script Files)

5.4 Creating a Radar Playback Scenario

Using a text editor create a script file. The script file must have at a minimum the following two lines:

1. **set SimTime** <time>
2. **RadarSim** <object name> <options>

The **RadarSim** object will playback short range and long range radars from a pre-recorded file. The parameters **-dev** and **-srv** must be entered using the convention described below. The parameter **-dev** identifies the channels on the communications card that are to transmit data for playback. Radar's *srr0* (short-range) and *lrr0* (long-range) are the first logical radars. There are four channels per short-range radar and three channels per long-range radar. A maximum of two radars per card can be transmitted. The parameter **-srv** identifies the path and surveillance file that the radar data will be transmitted from.

RadarSim test1 **-srv** <srv file directory path>/test.srv **-dev** /dev/srr0

RadarSim test0 **-srv** <srv file directory path>/test.srv **-dev** /dev/srr1

(See Appendix E – Script Files)

5.5 Creating a Host Scenario

Using a text editor create a script file. The script file must have at a minimum the following two lines:

1. *set SimTime* <time>
2. *Host* <object name> <options>

The *Host* object emulates an ARTCC Host system. It supports interfacility communications either a live terminal system or FIRS simulated. It requires radar input from a file or communications card. The parameters *-artsid*, *-dev* and *-srv* must be entered using the convention described below.

The parameter *-artsid* identifies the terminal system, where *ttt* is the three letter Arts identifier. The parameter *-dev* identifies the port(s) on the communications card that will support the interfacility communication. A maximum of eight interfacility interfaces per card is supported. The parameter *-srv* identifies the path and surveillance file or radar receiving device.

Host *zcl -artsid ttt -dev/dev/if0 -srv <srv file directory path>/test.srv* or radar receiving device

(See Appendix E – Script Files)

5.6 Creating an ARTS Scenario

Using a text editor create a script file. The script file must have at a minimum the following two lines:

1. *set SimTime* <time>
2. *Arts* <object name> <options>

The *ARTS* object emulates a terminal system. It supports interfacility communications to an ARTCC, either a live ARTCC system or FIRS simulated ARTCC system. It requires radar input from a file or communications card. The parameters *-artsid*, *-dev* and *-srv* must be entered using the convention described below.

The parameter *-artsid* identifies the terminal system. The parameter *-dev* identifies the port(s) on the communications card that will support the interfacility communication. A maximum of eight interfacility interfaces per card is supported. The parameter *-srv* identifies the path and surveillance file or radar receiving device.

Arts *ttt -artccidzcl -dev/dev/if0 -srv <srv file directory path>/test.srv* or radar receiving device

(See Appendix E – Script Files)

5.7 Creating a ADSB Record Scenario

Using a text editor create a script file. The script file must have at a minimum the following two lines:

3. *set SimTime* <time>
4. *AdsbRec* <object name> <options>

The *AdsbRec* object will record short range and long range radars.

The parameters *-dev*, *-srv* and *-port* must be entered using the convention described below. The parameter *-dev* identifies the network card that receives data to record. The parameter *-srv* identifies the path and asterix file that the ADSB data will be recorded to. The parameter *-port* identifies the UDP port which corresponds to specific ADSB GBT sensor.

AdsbRec *test0 -srv <srv file directory path>/test.srv -deveth0 -port 5555*
AdsbRec *test1 -srv <srv file directory path>/test.srv -deveth1 -port 3045*

(See Appendix E – Script Files)

5.8 Creating a ADSB Playback Scenario

Using a text editor create a script file. The script file must have at a minimum the following two lines:

3. **set SimTime** <time>
4. **AdsbSim** <object name> <options>

The **AdsbSim** object will playback ADSB sensor data from a pre-recorded file. The parameters **-dev** and **-srv** must be entered using the convention described below. The parameter **-dev** identifies the IP and UDP port on network that are to transmit data for playback. The parameter **-srv** identifies the path and asterix file that the ADSB data will be transmitted from.

```
AdsbSim test1 -srv <srv file directory path>/test.srv -devudp:172.26.100.255/3045
```

```
AdsbSim test0 -srv <srv file directory path>/test.srv -devudp:172.26.100.100/5555
```

(See Appendix E – Script Files)

5.9 Creating a RAPPI Display

Using a text editor create a script file. The script file must have at a minimum the following two lines:

1. **set SimTime** <time>
2. **qars** <object name> <options>

The **qars** object creates a RAPPI display to view beacon and search targets of designated radar.

```
qars testdsp -srv <srv file directory path>/test.srv or radar receiving device
```

(See Appendix E – Script Files)

5.10 Running a Scenario

In order to run a simulation the system must have at least one communications card configured with a unique I/O base address, the correct modules loaded and a script file with the correct objects and associated input files.

1. Log on using a valid user name and password.
2. Start X windows by typing **startx**.
3. Create a xterm window by clicking on the xterm icon located in the bottom tool bar.
4. Change to the directory where the scenarios script file is located by typing **cd <directory name>**.
5. Type **xtest <scriptfile>**.
6. The “JVN Communications” window will appear. Once all objects have been created, the start button will be enabled.
7. Click on the start button in the “JVN Communications” window.
8. From the six virtual Desktop buttons located at the bottom right corner of the screen, click on the button in the top row, middle column.
9. A RAPPI display appears; this is where the radar targets are displayed. The targets are identified by their associated beacon code.
10. There is a series of other tabs at the top of the window; the function being displayed is indicated by the darker text in the tab. Click on the other tabs to display the desired function, for example interfacility communications can be displayed by clicking on the interfacility tab.

5.11 Restarting a Scenario

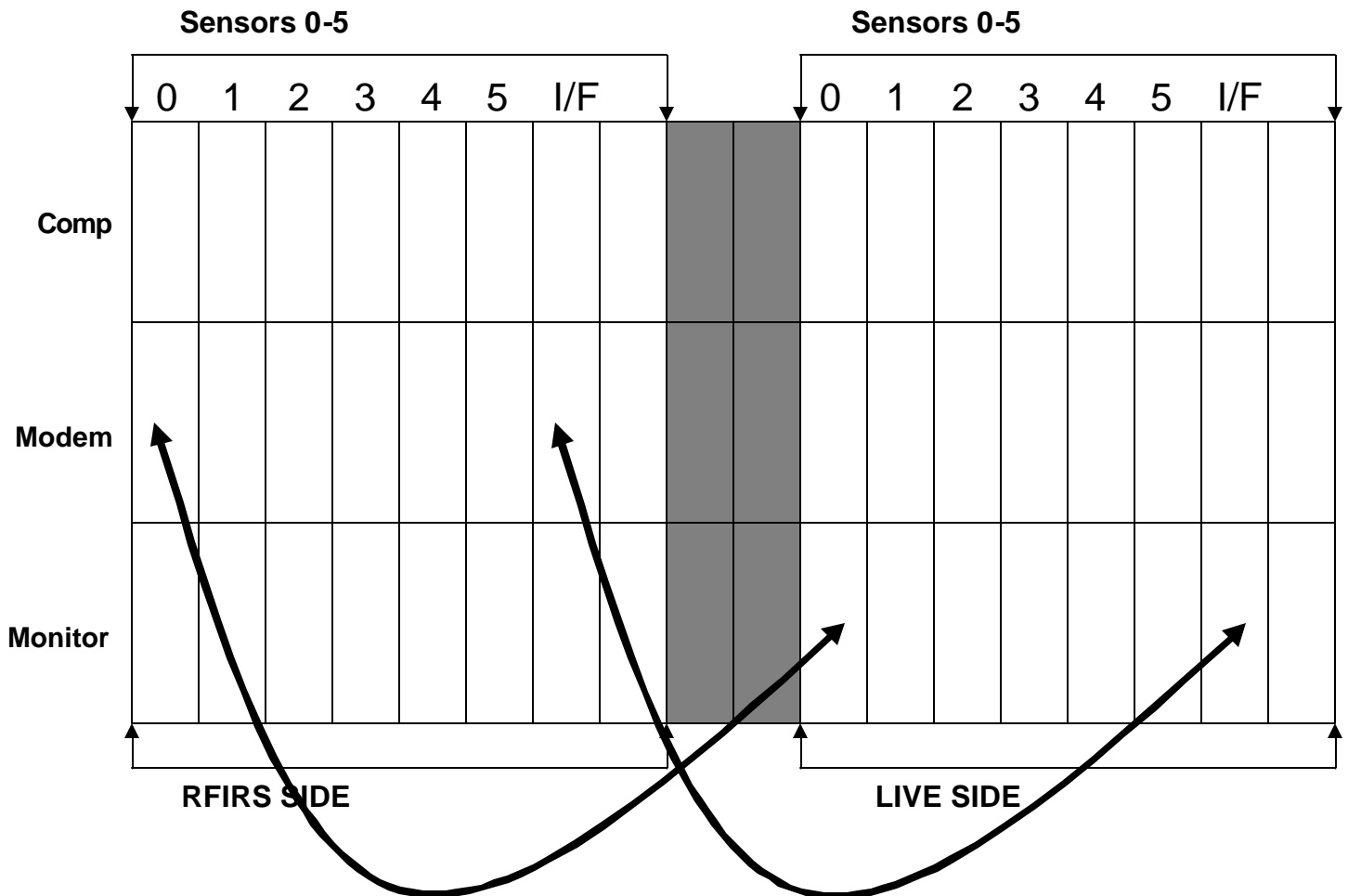
1. From the virtual Desktop buttons in the bottom right hand corner, click on the button in the top row, first column.
2. Click on the exit button in the “JVN Communications” window.
3. From the xterm window type **xtest <scriptfile>**
4. Click on the start button in the “JVN Communications” window.
5. Continue from step eight, from the above procedures.

5.12 Exiting a Scenario

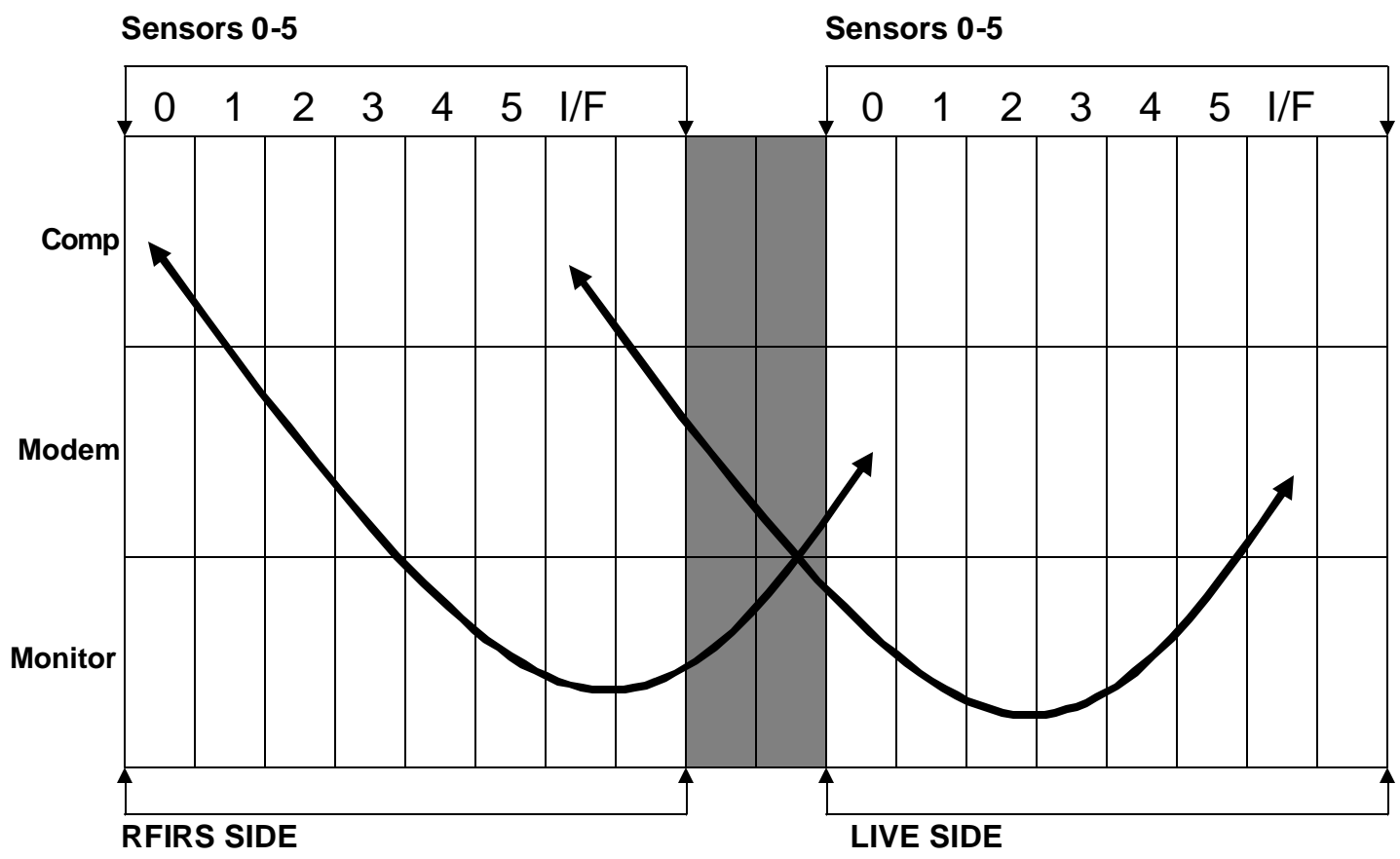
1. From the virtual Desktop, click on the top left button.
2. From the “JVN Communications” window click on the exit button.
3. Close the JVN Communications Radar/Interfacility Testing window by clicking on the X in the top right corner of the window.
4. Simultaneously depress the Ctrl-Alt-Backspace keys to exit X Windows.

5.13 Record and Playback Using the Patch Panel

Recording one short range sensor and interfacility



Playing back one short range sensor and interfacility



6.0 UTILITIES

The FIRS conversion utilities convert ASR-11, ASR-9, ASR-8, ARSR and interfacility data files from Continuous Data Recordings (CDR) or the LYNX file system from ARTS systems to the FIRS file system.

Once files are converted to the FIRS file system format, offline utilities allow the user to analyze and manipulate data on the FIRS system.

Simply typing in the utility command name will print to the screen, usage information for that command.

6.1 Offline Utilities

azflag	- Sets a test IRQ flag for a particular azimuth.
bcnflag	- Sets a test IRQ flag for a particular beacon code.
bcnhist	- Summarizes the history of a particular beacon code.
blinkled	- Blinks LED on Emulex card. Used to determine logical order of I/O adapter.
cdr3a	- Converts a CDR radar file recorded from an ARTSIIIA to a FIRS srv file.
cdr605	- Converts a CDR radar file recorded from an ARTSIIE to a FIRS srv file.
cd2srv	- Converts a CD format file to FIRS srv format.
Clrflag	- Clears the flag field.
findtime	- Determines the start and stop time of a FIRS srv file.
if3a	- Converts a CDR interfacility file recorded from an ARTSIIIA to a FIRS interfacility file.
if605	- Converts a CDR interfacility file recorded from an ARTSIIE to a FIRS interfacility file.
modes	- Determines if beacon source was Modes or IBI.
msglist	- Sequential list of time stamped radar messages in hex format.
msgsum	- Decoded list of time stamped radar messages.
setflag	-
setpar	-
smfix	- Fix sector mark quality.
smqual	- Analyze sector mark quality.
srcbox	- Lists search messages within a given range and azimuth.
srv2cd	- Converts a FIRS srv file to a CD format file.
timeslice	- Extract, by time, a segment of an srv file.
trklist	- Lists decoded beacon target information.
imte	- Converts integrated magnetic tape emulator file to FIRS srv file.
ntbak	- Extracts data from Windows NT tape backups and converts to srv file.
orr2si8	- Extracts data from Host ORRE tape and converts to FIRS srv file.
starscdr	- Extracts CDR data from DLT tape and converts to FIRS srv file.
monitor	- Displays system device usage and availability.
firsver	- Displays FIRS software version.

6.2 FIRS Utilities

6.2.1 FIRS Automated Scenario Tool (FAST)

FIRS playback, recording and scenario management can be done with the click of one of the JVN buttons below.



6.2.2 Record

To record live radar click the record button. This will scan your computer for available recording devices, such as emulex cards. Once the (FAST) finds you available devices it will prompt you with a GUI menu to select radar devices and a scenario name.



The first thing you should set is your scenario name. Select the Scenario Name box and type the name you would like to call your recording. If you do not set a recording name, (FAST) will use the timestamp name that appears in the box by default.

The timestamp name is used to avoid writing over an existing recording. This time is drawn from the systems set time. The name you select will later become a directory. This directory is where FIRS will place the data you recorded.



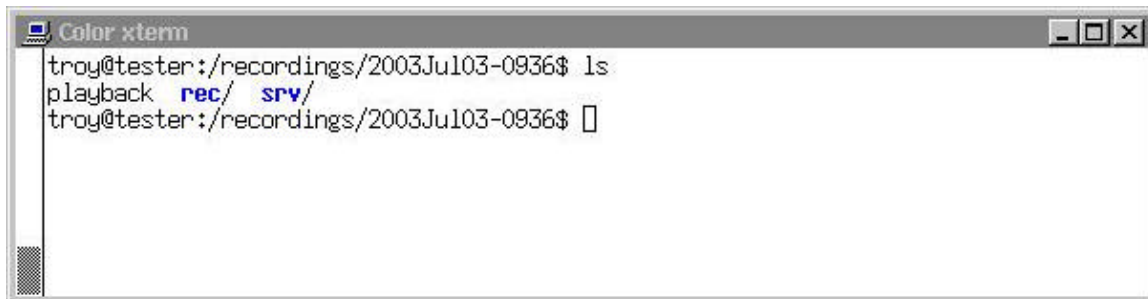
```

Color xterm
troy@tester:/recordings$ ls
2003Jul03-0936/  alllongrange/  allshortrange/
troy@tester:/recordings$ 

```

The next step is to select a device to record on. You will find a list of devices on the left starting with 0. Your system should be connected in the same order as your site's adaptation. For example, if SDF radar is your site's first sensor, then device 0 should be SDF and so on. Once you click your enable button you will be able to set options like radar name, radar type and data polarity. You can enable as many radars as you need.

When you select the Radar name, FAST uses that name to label the actual radar data file (.srv) file. This file is placed in a directory called "srv". This "srv" directory resides under the "scenario name" directory. FAST uses your input to, not only record, but also to build you a playback file. The playback file is also placed under the "scenario name" directory. The file is called playback and will be needed for future playback.

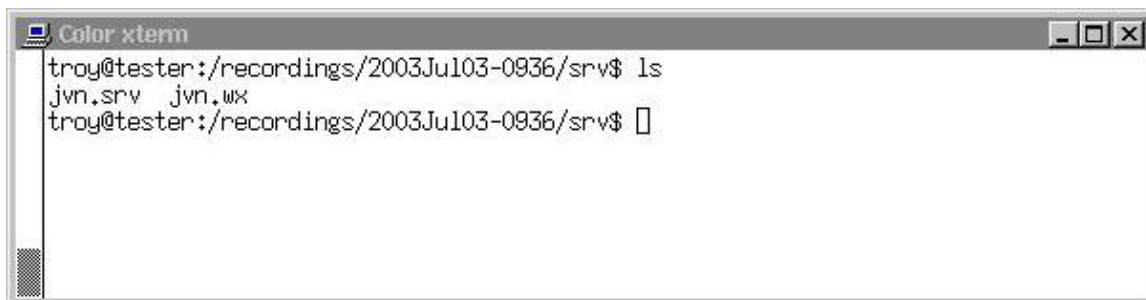


```

Color xterm
troy@tester:/recordings/2003Jul03-0936$ ls
playback  rec/  srv/
troy@tester:/recordings/2003Jul03-0936$ 

```

In the "srv" directory you will find the actual radar data file. These files will all end in ".srv" for surveillance files and or ".wx" for weather files. When you playback a scenario these file will be the output files used by (FAST).



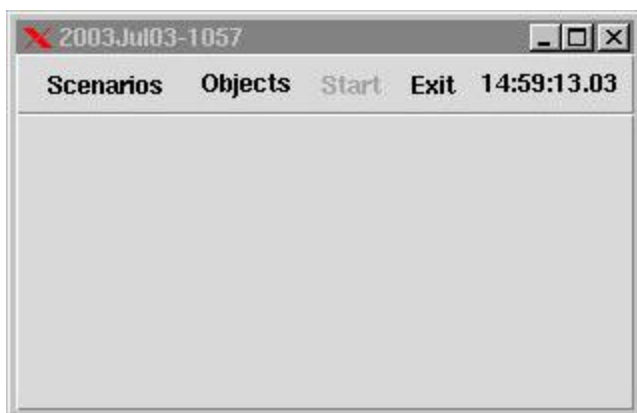
```

troy@tester:/recordings/2003Jul03-0936/srv$ ls
jvn.srv jvn.wx
troy@tester:/recordings/2003Jul03-0936/srv$

```

In the example above “**jvn**” was the name used in the “Radar Name” box.

Now that you have enabled your device and have chosen the radar names, you can begin recording by clicking on the “Start Recording” button. You will see a small icon appear with a clock in the upper right hand corner. If the clock is running you are recording. Now if you look down at the bottom right hand corner of the screen, you can select the virtual desktop window that houses your radar display window.



Click in the window to display your radar message counts. This is a helpful display window to determine radar flow and quality. You will see four channels in blue; these represent your four channels of radar. Across from the channels are green numbers that should be increasing as the recording goes on (a good sign you are recording data successfully). Above the green number you will find yellow message types. These represent the types of radar and weather data being recorded. To the right in red you will find error messages. These numbers should remain zero unless the data on the given channel has a problem although it is not unusual to have a small number of errors on some channels. If you find you are experiencing a large number of errors (nearly as much as the clean data) there is a problem (possibly with cable connection or script errors) and you should contact your technician.

ResetStats 15:05:01.70

	TOT	BCN	SRC	BRTQC	SRTQC	STAT	WX	MARK	INV	MALF	SYNC
Chan 1	0	0	0	0	0	0	0	0	0	0	0
Chan 2	0	0	0	0	0	0	0	0	0	0	0
Chan 3	0	0	0	0	0	0	0	0	0	0	0
Chan 4	0	0	0	0	0	0	0	0	0	0	0

In the FIRS Recording Setup you have some additional options. When you enable a device you can also select the **rappi** button. This is not needed to record, but gives the user a more graphical display of targets and weather. If you do not select the **rappi** option, the display will not be visible during playback of the recorded file.



At the bottom of the Recording Setup GUI you will find a “Playback Options” button. You can select this feature to add an interfacility host to your playback file that will be created. If you select this option you will need to know your ARTS ID and give a valid Host ID.



You will also see a help button. Click this button for help with patching. Before recording any radar you will need to have the proper patch cord inserted into the correct ADC patch panel module, these modules should be labeled per your site adaptation. For more help on patching please refer to the “Hardware Diagrams” section.

6.2.3 Playback

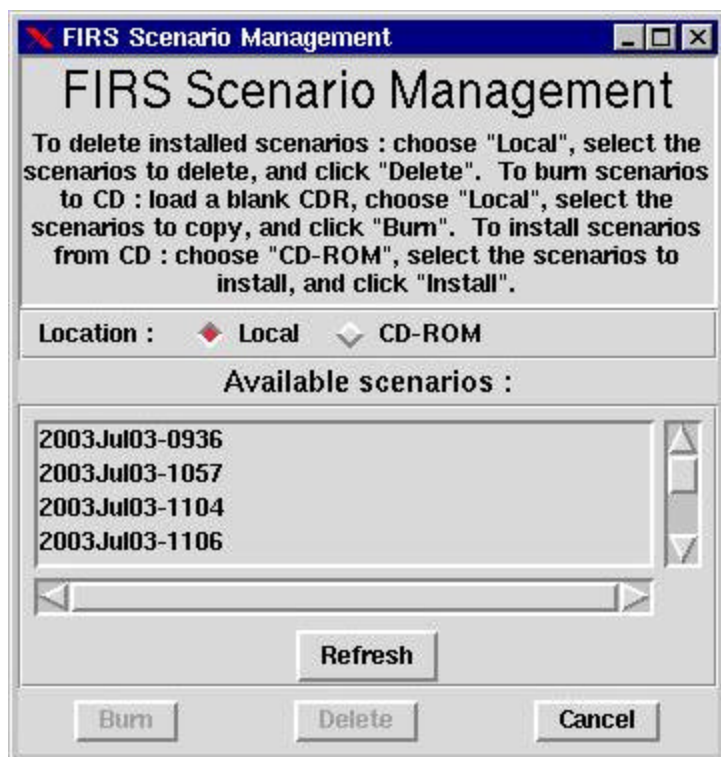
After you have some recorded or extracted radar data you can begin to playback a scenario. Simply click on the playback button and a FIRS Playback Setup GUI should open on the screen. Select a scenario from the list of available scenarios and click Load Scenario. (FAST) searches the /playback directory for available scenarios that the user named during the record process.



Make sure you have the proper patch connections (remember you can use the help button) then click the **Start** button to begin playback. You should then have data flowing to your TP. Click on the virtual display window at the bottom righthand corner of the screen to verify data flow from the FIRS tool. If you selected the **rappi** option during the recording, you will be giving a rappi during playback to view targets on screen.

6.2.4 *Manager*

Scenario Manager will help you manage your recordings. The File Manager can also be used to copy from a CD to the hard drive or even burn your recordings from the hard drive to a CD.



Use the “Location” button to get to your hard drive “Local” or you CD “CD-ROM” Be sure to place your cd into the drive before clicking CD-ROM button. Please note: the “Local” button will only search the /recordings directory. You can select a single file or multiple files to burn. Simply place a blank CD-R into the CD-ROM drive and click “Burn”. When CD is finished, the box below will appear:



You can also delete unwanted files from your hard drive to free up hard disk space. Just select local and select the files, then select delete. The selected files will be erased from the hard drive permanently. Note: Remember the “Local” button will only remove files located in the **/recordings** directory.

7.0 TROUBLESHOOTING THE SYSTEM

7.1 Boot Messages

Boot messages are written to the screen as the system is in the process of booting up. In a normal boot, most messages written to the screen are status messages indicating what devices have been recognized and initialized. If problems occur error messages are written to the screen such as *device or resource busy*, *error initializing hardware*, or *module already exists*. When a problem arises, the boot messages should be one of the first things looked at. Boot messages can be dumped to the screen any time by typing the command *dmesg*. The boot messages will then be dumped to the screen in their original sequence during boot time.

7.2 Trouble Shooting Guide

1. Did the scenario load?
 - If yes go to step #2.
 - Is the “start” button in the “JVN Communications” window is highlighted, indicating that the scenario is loaded and ready to run?
 - If no, observe the error message in the xterm window, output to the screen in the first line of messages, and go to step #5.
2. Determine if the FIRS/ATC interface is configured correctly.
 - Is the serial interface card terminated to the correct ATC equipment? For example, card 0x240 configured as an interfacility interface should be terminated to an EIA-232 interfacility interface.
 - Refer to appendix G and make sure the correct adapters are being used.
 - Make sure the cables and their pin assignments are correct for the type of interface you are using. Refer to appendix G for EIA-232 or SDL-530 cable pinouts.
3. If cables are correct, use an oscilloscope to determine if the correct signals are present. To isolate a recording problem look at the receive data pins (232-pin 3, EIA-530 pins 3 & 16) on the cable end that terminates at the FIRS. To isolate a playback problem look at the receive data pins (232- pin 3, EIA-530 pins 3 & 16) on the cable end that terminates at the ATC equipment. If correct signals are present go to step #4.
4. Check the script file associated for the scenario and determine if the correct objects and SimTime are being loaded.
 - If problem still exists call JVN technical support.
5. **ERROR** - **SOLUTION**
 - *device or resource busy* – There may be another simulation running using the same device. Look for another simulation window and exit from it. (Refer to appendix F – Modules).
 - *no such device* – Refer to appendix F and the sections titled *Linux* and *Modules* and check that the correct modules are loaded or configured correctly.

Problem may exist with serial interface card, check address switch settings and reseat the card.

Type the command *lsmod* to determine what modules are being loaded into memory and check that the correct modules are loaded or configured correctly.

Use module commands to remove and insert the correct modules or edit the *rc.local* file to reflect your scenario.

8.0 APPENDICES

8.1 Appendix A – SDL

Figure 1 – SDL RISCCom 8/si card layout

I/O Base address	I/O Base Address Switch							
	8	7	6	5	4	3	2	1
0x220	On	Off	Off	On	On	On	Off	On
0x240	On	Off	Off	On	On	Off	On	On
0x250	On	Off	Off	On	On	Off	On	Off
0x260	On	Off	Off	On	On	Off	Off	On
0x2A0	On	Off	Off	On	Off	On	Off	On
0x2B0	On	Off	Off	On	Off	On	Off	Off
0x300	On	Off	Off	Off	On	On	On	On
0x320	On	Off	Off	Off	On	On	Off	On

Do not use interrupts (remove all jumpers).

Each card must have a unique I/O Base Address.

Each card must use a unique I/O Port (0xD0, 0xD4, 0xD8, 0xDC, 0xE0, 0xE4)

8.2 Appendix B – Sunhillo (Emulex)

Switch Settings

Located on the solder side of the card, are a number of eight-position switches. A 2-channel card has two switches, a 4-channel card has four switches and an 8-channel card has eight switches.

DTE Mode (default mode)

In DTE mode, the Emulex interface adapter receives both of the transmit and receive clocks from the attached communications device.

An Emulex interfacility device connecting to a DCE device requires the Emulex interface adapter to be configured in DTE mode. To select the DTE mode, set each eight-position switch to the following configuration.

DTE Switch Settings

Position #	On	Off
1	*	
2	*	
3	*	
4	*	
5		*
6		*
7		*
8		*

DCE Mode

In DCE mode, the Emulex interface adapter drives the transmit and receive clocks for the attached communications device.

An Emulex Interfacility device connecting to an ATC device requires the Emulex interface adapter to be configured in DCE mode. To select the DCE mode, set each eight-position switch to the following configuration.

DCE Switch Settings

Position #	On	Off
1		*
2		*
3		*
4		*
5	*	
6	*	
7	*	
8	*	

Split Clock Mode

In split clock mode, the Emulex interface adapter drives the transmit clock for the attached communication device and receives the receive clock from the attached device.

An Emulex radar device requires the Emulex interface adapter to be configured in split clock mode. To select the split clock mode, set each eight-position switch to the following configuration.

Split Clock Switch Settings

Position #	On	Off
1		*
2		*
3	*	

4	*	
5	*	
6	*	
7		*
8		*

8.3 Appendix C – RDI

RDI ADDRESS SWITCH SETTINGS

Address Space	A23-A20	A19-A16	A15-A12
D0000-D1FFFh	0000	1101	0000
D2000-D3FFFh	0000	1101	0010
D4000-D5FFFh	0000	1101	0100
D6000-D7FFFh	0000	1101	0110
D8000-D9FFFh	0000	1101	1000
DA000-DBFFFh	0000	1101	1010
DC000-DDFFFh	0000	1101	1100
DE000-DEFFFh	0000	1101	1110
	0 = off 1 = on	0 = off 1 = on	0 = off 1 = on

8.4 Appendix D – Simulation Objects

- RadarRec(n) RadarRec(n)

NAME

RadarRec - Create Radar record object within a FIRS (Stan-dalone Interfacility and Radar Simulator) scenario.

SYNOPSIS

RadarRec name -dev input_device -srv surveillance_file [-wx wx_file] [-rs232 [0..1]] [-invert [0..1]]

DESCRIPTION

RadarRec creates a Radar record object which will record all received radar messages to disk during a FIRS scenario. The recorded files are suitable for playback by the RadarSim object.

Timestamps

If the global variable "SimTime" is defined, timestamps will start at that time when the Sim is started; otherwise, the platform's system time at the Start event will be used.

Options

-dev Specifies the radar receive device to use. Radar receive devices are named /dev/srr0 thru /dev/srrN, where N is one less than the number of radar receive devices (each device uses 4 channels, or 2

devices per card). Devices present on another FIRS tool may be specified by prefixing "hostname:" before the device.

-srv Specifies the filename to which surveillance msgs will be recorded.

-wx Specifies the filename to which wx msgs will be recorded.

-rs232 If this option is set to 1, line driver outputs will be forced to rs232.

-invert If this option is set to 1, data logic levels will be inverted.

CREATION ACTION

On creation, a window is created in the 3rd column of the virtual X-display. The radar device is initialized (is ready to begin recording), and the record file is opened.

START ACTION

Radar reception begins when the start command is received, and will continue until exit. The internal time is set to the value of the global variable SimTime; if SimTime is not defined, the system time of the start event is used. Message statistics will be updated in the window, and represent the messages received and recorded.

SEE ALSO

SIRS(n), Host(n), Arts(n), RadarSim(n)

- RadarSim(n) RadarSim(n)

NAME

RadarSim - Create Radar playback object within a SIRS (Standalone Interfacility and Radar Simulator) scenario.

SYNOPSIS

```
RadarSim name -dev Output_device -srv Surveillance_file
[-wx Wx_file] [-rs232 [0..1]] [-invert [0..1]] [-rate
baud_rate] [-type [lrr..srr..asr11]]
```

DESCRIPTION

RadarSim creates a Radar playback object which will functionally act as a radar during a FIRS scenario. The RadarSim requires a surveillance file, which consists of a list of time-stamped radar messages to be played back.

Options

- dev Specifies the output radar transmit device to use. Radar transmit devices are named /dev/rdrtx0 thru /dev/srrN, where N is one less than the number of radar receive devices (each device uses 4 channels, or 2 devices per card). Devices present on another SIRS tool may be specified by prefixing "hostname:" before the device.
- srv Specifies the filename from which surveillance msgs will be generated.
- wx Specifies the filename from which wx msgs will be generated.
- rs232 If this option is set to 1, line driver outputs will be forced to rs232.
- invert If this option is set to 1, data logic levels will be inverted.
- rate Allows user to specify a non-standard baud rate. Must be a multiple of 2400.
- type Perform some radar specific tuning. lrr -- don't activate 4th serial channel; ignore any wx file; don't generate SCIP control msgs. asr11 -- force all msgs to channel 0; don't generate SCIP control msgs; mask out flags field on all msgs.

CREATION ACTION

On creation, a window is created in the 3rd column of the virtual X-display. The radar device is initialized (starts idling), and the radar file position is set based on the value of the global variable SimTime.

START ACTION

Radar transmission begins when the start command is received, and will continue until exit. Message statistics will be updated in the window, and represent the messages uploaded to the device, not the number of messages transmitted over the wire.

SEE ALSO

SIRS(n), Host(n), Arts(n), RadarRec(n)

- Host(n) Host(n)

NAME

Host - Create simulated Host object within a SIRS (Standalone Interfacility and Radar Simulator) scenario.

SYNOPSIS

```
Host name [-dev interfacility_device] [-srv radar_input]
[-artsid artsid] [-fp flight-plan file] [-wx wx_input]
[-adaptation filename] [-map mapList] [-rec record_file]
[-rxclock rate] [-txclock rate] [-msgs msgs] [-delay ms]
[-if_timeout secs] [-if_retries num] [-new_NAS_if [0/1]]
[-ho_range range] [-ho_file file] [-utc [0/1]]
[-dynamic_ecids [0/1]] [-cenrap Irrfile] [-cenrap_quiet
[0/1]]
```

DESCRIPTION

Host creates a Host object which will functionally act as a HOST system during a SIRS scenario. The Host object assumes that the system it is communicating with is a Terminal ATC system. The Host requires a radar input, which can be either live data through a /dev/rdrxx device, or from a pre-recorded file. The Host optionally uses a flight-plan file, which is used to generate and send FP, AM, and CX messages over the interfacility link. Each FP message is dynamically assigned and ECID and transmitted over the line. Each CX and AM msg in the flight-plan file has a bcn code to allow the Host to associate a pre-assigned ECID to them. The Host object also generates the following Interfacility data messages dynamically during the scenario: TI, TU, TA, and TR. These messages are sent based on radar input and received interfacility messages.

CREATION ACTION

On creation, a "tabbed notebook" type window is created in the 2nd column of the virtual X-display. The interfacility device is initialized (starts idling), and the radar file position is set based on the global variable SimTime (unless a radar device is specified). The fp file position is set based on SimTime. The map files are read into an internal table and the first map specified is displayed in the rappi window.

START ACTION

Host processing will be scheduled to run 4 times per second (default). Internal processing performed can be generally described as:

1. interfacility input processing
2. interfacility timer/retransmission processing
3. VFR processing
4. Flight Plan output processing
5. Radar track update / TI generation
6. TU generation
7. Cenrap processing

In addition, the Host object will respond to several

asynchronous events, such as relaying ARTS-ARTS messages, and responding to user control msgs.

Options

- dev Specifies the interfacility device to use. Interfacility devices are named /dev/if0 thru /dev/ifx, where x = 8*number of if cards in system. (each device uses 1 channel, or 8 devices per card). Default is /dev/null.

- srv Specifies the filename or device from which surveillance msgs will be received. Either a filename or a radar receiving device (/dev/rdrxx0 thru /dev/rdrxx) may be specified. Default is /dev/null.

- fp Specifies the flight plan file to download over the if line. The flight plan file is a plain ASCII file containing a list of time-stamped FP, CX, and AM messages. The time stamps are embedded within the msg header, for example: ZCD1455105 would be sent at simtime 14:55:10.5. The last 3 digits of the header field will be dynamically overlaid with an assigned ECID before going out over the line. All occurrences of the characters :: in this file are replaced with an LRC PREPARE character (0xb3) and an LRC checksum before going out over the line. All occurrences of the character * are replaced with an EOM character (0xb1) before going out over the line. Default is none.

- wx Specifies the filename from which wx msgs will be received. If the -srv option specified a device, this option will be ignored, and srv device will be used as the wx source. The arts object will adjust the wx msg timestamps to agree with the srv file being used. This allows the use of a separately recorded wx file, with arbitrary start time. Default is none.

- adaptation
 Specifies an adaptation file to be used. The adaptation file is a plain ASCII file which specifies fixes are to be used for ARTS-ARTS processing. Each line in the adaptation file is of the format
 FIX OWNER1 OWNER2. FIX is a 3-character ATC fix found in FP messages. OWNER1 and OWNER2 are 3 character IDs specifying the owner of this fix. Usually, OWNER1 and OWNER2 are specified to be two ARTS ID's, specifying any Flight Plans with FIX to be handled as an ARTS-ARTS transfer.

- artsid
 3-character id of the system on the remote end of the interfacility link. This is used for sending TR's.

- map A list of map files, enclosed in double quotes if there is more than one. This is used for rappi display purposes only. The maps must be in NOAA form, with each line segment specified by a pair of Lat/Lon points. The Lat/Lon center of the map and desired rotation may also be specified - see an existing map file for examples.
- rec A recording file where all IF msgs and status msgs will be stored in ASCII format. If this option is not specified, a default record filename of name.rec will be used.
- txclock
Specifies the speed of the transmit clock to be used for the interfacility device. A value of 0 configures the card to accept transmit clock from the modem (this option only works with 8Si-232 cards, and Emulex PCI cards which have been jumpered for that mode. Default value is 2400.
- rxclock
Specifies the speed of the receive clock to be used for the interfacility device. A value of 0 configures the card to accept receive clock from the modem. A non-0 value will cause the receive clock to be generated using an internal crystal. Default value is 0.
- delay The delay in ms between each Host processing loop. The default is 250 ms, meaning the Host processing will run 4 times per second.
- msgs Number of flight plans to download per processing loop. Default is 5, combined with a default rate of 4 HZ, limits outgoing FPS to 20 per second.
- new_NAS_if
Use the new NAS IF protocol.
- cenrap
Specifies a long-range radar file to be used for cenrap processing, and enables cenrap output.
- cenrap_quiet
Turns off logging of cenrap msgs. Cenrap processing will continue.
- ho_range
Set the default handoff range for arrivals. Default is 53 nmi.
- ho_file
Specify a file describing handoff ranges by azimuth wedge, for arrivals. The format of the file is ASCII lines consisting of three space separated columns: start ACP, end ACP, ho range.

- if_timeout
Set the timeout parameter (in seconds) for message retransmission. Default is 5 seconds.
- if_retries
Set the retries parameter for message retransmission, before the message is removed from the retransmit queue. Default is 8.
- utc Use universal time for IF msg times. The coordination time field specified in the flight plan file is interpreted as an offset from the start time. This option alleviates the need to set the system time on the Terminal ATC system under test.
- dynamic_ecids
Setting to 0 forces the ECIDS specified in the flight plan file to be used, instead of the Host dynamically creating them. Use this option with care!
- max_tus
Set maximum number of TUs allowed per TU msg. Default is 6.
- enhanced_timestamps
IF msgs will be logged to a precision of .01 seconds, and the time of transmission is returned from the card, instead of the time of download to the card.

FILES

/dev/ifx
The interfacility device

/dev/rdrxx
The radar input device

SEE ALSO

SIRS(n), Arts(n), RadarSim(n), RadarRec(n)

- Arts(n) Arts(n)

NAME

Arts - Create simulated Arts object within a SIRS (Standalone Interfacility and Radar Simulator) scenario.

SYNOPSIS

Arts name -dev Interfacility device -srv Radar filename or Radar device -starttime starttime -artccid ARTCCID [-wx filename or devicename] [-adaptation filename] [-map mapList] [-rec record file]

DESCRIPTION

Arts creates an Arts object which will functionally act as an ARTS system during a SIRS scenario. The Arts requires a radar input, which can be either live data through a /dev/rdrxxx device, or from a pre-recorded file. The Arts object will maintain an interfacility data link.

Options

-dev Specifies the interfacility device to use. Interfacility devices are named /dev/if0 thru /dev/ifx, where x = 8*number of if cards in system. (each device uses 1 channel, or 8 devices per card)

-srv Specifies the filename or device from which surveillance msgs will be received. Either a filename or a radar receiving device (/dev/rdrxx0 thru /dev/rdrxxx) may be specified.

-wx Specifies the filename from which wx msgs will be received. The wx file is used only for display purposes. If the -srv option specified a device, this option will be ignored, and srv device will be used as the wx source.

-adaptation

Specifies an adaptation file to be used. Currently, the adaptation file specifies fixes that are to be used for ARTS-ARTS processing. The arts object uses this file to determine which flights are to be handed off to the adjacent ARTS/STARS system, and which are to be handed off to the

ARTCC.

-starttime

The start time for the object. If a radar filename was specified, the artssim will position itself in the file based on this starttime. Starttime may be specified in either "HH:MM:SS" or "HHMMSS" formats.

-artccid

3-character id of the system on the remote end of the interfacility link. This is used for sending TR's.

-rec

A recording file where all IF msgs and status msgs will be stored in ASCII format. If this option is not specified, a default record filename of name.rec will be used.

FILES

/dev/ifx

The interfacility device

/dev/rdrxx

The (optional) radar input device

SEE ALSO

SIRS(n), Host(n), RadarSim(n), RadarRec(n)

- qars(n) qars(n)
(To be Supplied)

- ifrec(n) ifrec(n)
(To be Supplied)

- etms(n) etms(n)
(To be Supplied)

8.5 Appendix E – Script File Examples

Script file usage

[Optional - brief description] *The # sign signifies a comment or disables a Line, it is not executable*

set SimTime 00:01:10

set useful temporary variables

set rdrp <location of data directory>

<ATC object type> <object name> <options >

<ATC object type> <object name> <options >

Example #1

Radar Playback & Host

2 ASR's and 2 ARSR's

1 interfacility

global SimTime

set SimTime 14:30:00

set rdrp /scenarios/srv

set mp /scenarios/map

RadarSim lan -srv \$rdrp/lan.srv -dev /dev/srr0

RadarSim acy -srv \$rdrp/acy.srv -dev /dev/srr1

RadarSim qsr -srv \$rdrp/qsr.srv -dev /dev/lrr2 -rate 4800

RadarSim qla -srv \$rdrp/qla.srv -dev /dev/lrr3 -rate 14400

Qars landsp -srv \$rdrp/lan.srv

Host zcl -artsid ttt -dev /dev/ifo -srv \$rdrp/lan.srv -rec rdrp/zcl.rec

Example #2

Radar Recording with a RAPPI

Recoding of 2 ASR's and 2 ARSR's with a RAPPI display of lan

set SimTime 14:30:00

set rdrp /scenarios/srv

RadarRec lan -srv \$rdrp/lan.srv -wx \$rdrp/lan.wx -dev /dev/srr0

RadarRec acy -srv \$rdrp/acy.srv -wx \$rdrp/acy.wx -dev /dev/srr1

RadarRec qsr -srv \$rdrp/qsr.srv -dev /dev/lrr2

RadarRec qla -srv \$rdrp/qla.srv -dev /dev/lrr3

Qars landsp -srv /dev/srr0

8.6 Appendix F – Modules

Modules

exrdr2.o – The module that loads the device driver to playback and record radar data using the communications card.

8.7 Appendix G – Serial Interfaces

EIA-232

DB – 25 Female/Male

Connectors

Label: **EIA-232**

pin - 1 shield
 2 transmit data
 3 receive data
 7 signal ground
 17 receive clock
 24 external clock

EIA-530

DB – 25 Female/Male

Connectors

Label: **EIA-530**

pin - 2 transmit data A (positive)
 3 receive data A (positive)
 9 receive clock B (negative)
 11 transmit clock B (negative)
 14 transmit data B (negative)
 16 receive data B (negative)
 17 receive clock A (positive)
 24 transmit clock A (positive)

Sunhillo-ICA

DB – Male

Connector

Label: **Emulex-530**

pin - 2 transmit data A (positive)
 14 transmit data B (negative)
 3 receive data A (positive)
 16 receive data B (negative)
 9 receive clock B (negative)
 17 receive clock A (positive)
 12 transmit clock B (negative)
 15 transmit clock A (positive)

SDL 232/530 Hardware Adapter

DB – 25 male
Connector

DB – 25 Female
Connector

Label: SDL		Label: 232/530
pin - 4	-----	pin - 2
5	-----	3
6	-----	14
7	-----	7
8	-----	16
9	-----	9
15	-----	11
20	-----	24
22	-----	17

Emulex 530 Hardware Adapter

DB – 25 Female Connector Label: Emulex		DB – 25 Female Connector Label: EIA-530
pin - 2	-----	pin - 2
3	-----	3
7	-----	7
13	-----	
9	-----	9
12	-----	11
14	-----	14
15	-----	24
16	-----	16
17	-----	17

Emulex 232 Hardware Adapter

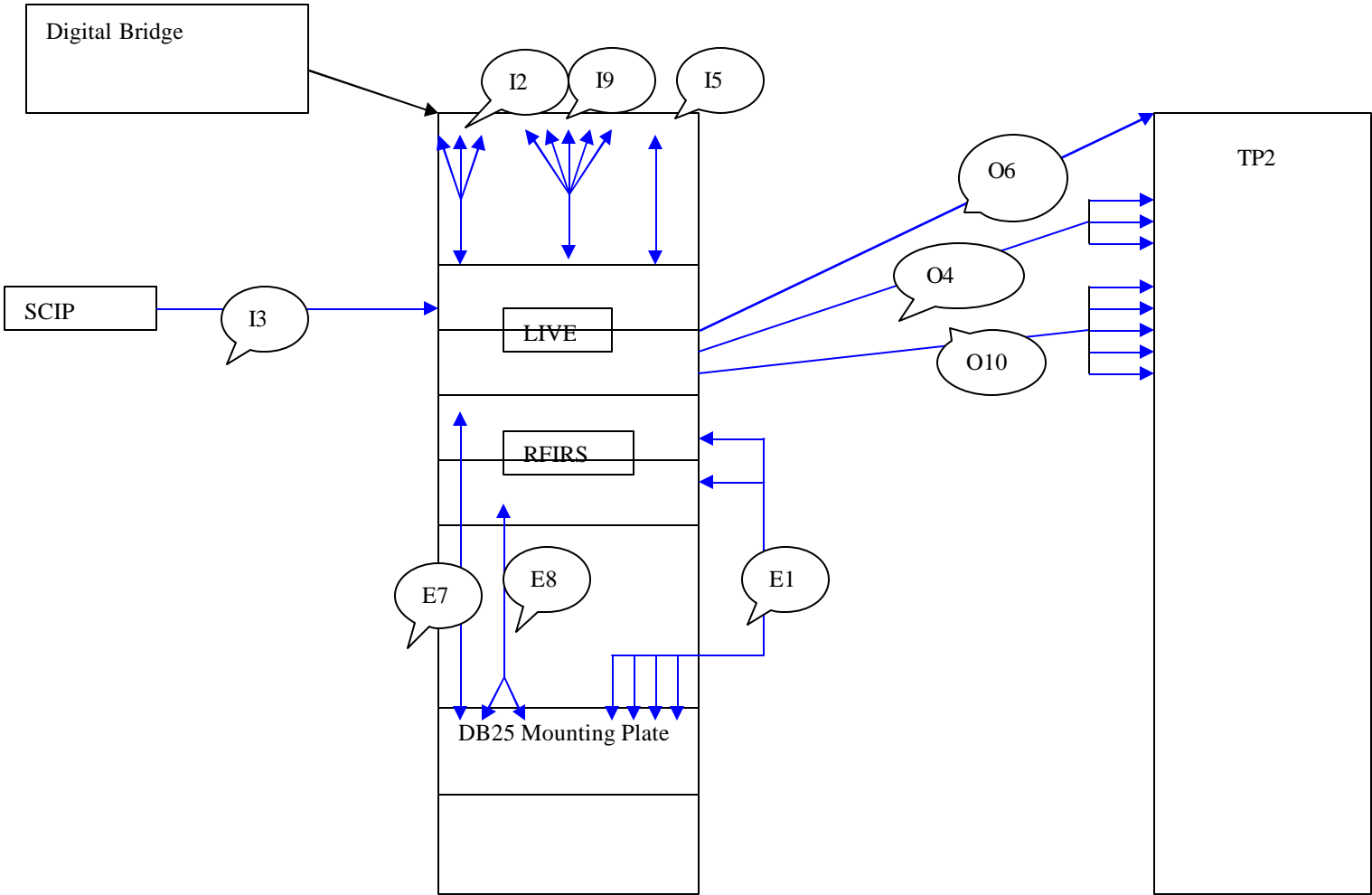
DB – 25 Female Connector Label: Emulex		DB – 25 Male Connector Label: EIA-232
pin - 2	-----	pin - 2
3	-----	3
7	-----	7
15	-----	24
17	-----	17

EIA-232/530 Crossover Adapter

DB – 25 Female Connector Label: EIA-232/530		DB – 25 Male Connector Label: EIA-232/530
pin - 2	-----	pin - 3
3	-----	2
7	-----	7
9	-----	11
11	-----	9
14	-----	16
16	-----	14
17	-----	24
24	-----	17

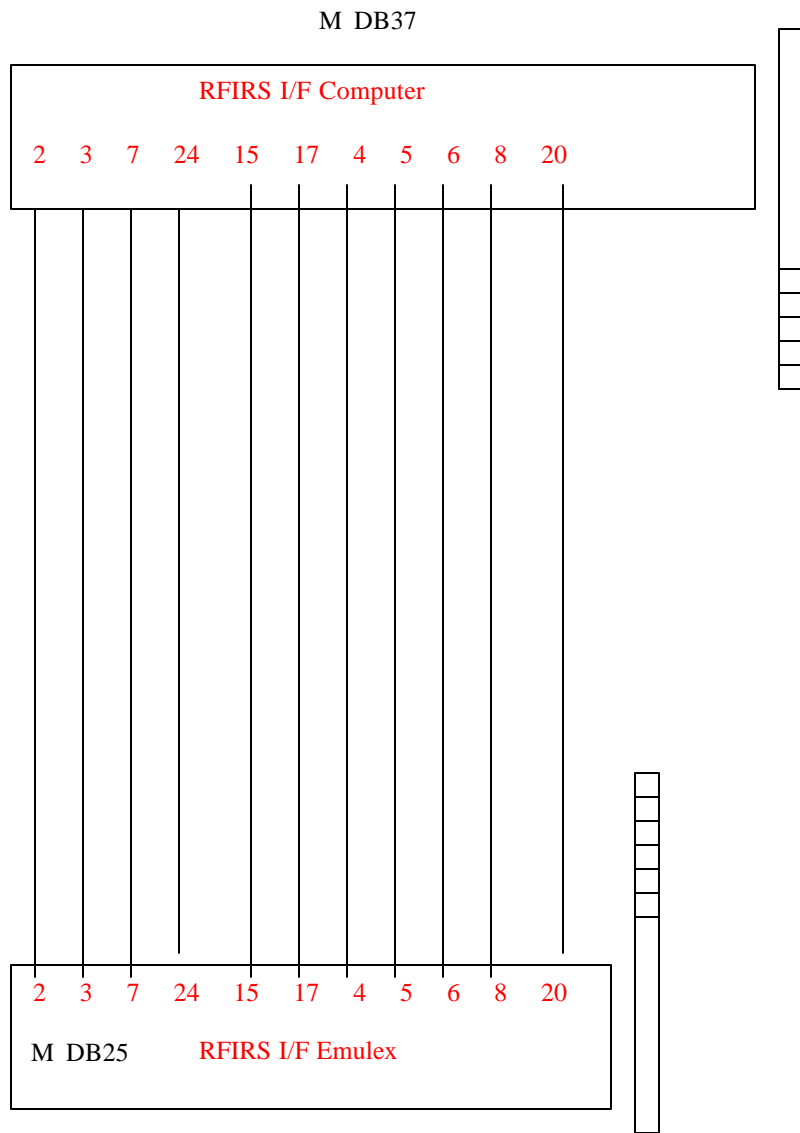
9.0 CABLE DIAGRAMS

RFIRS Cable Diagram



THIS PAGE INTENTIONALLY LEFT BLANK

E7 Cable
RFIRS I/F

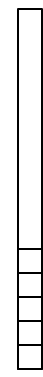
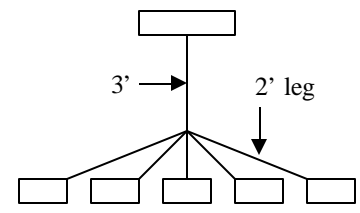
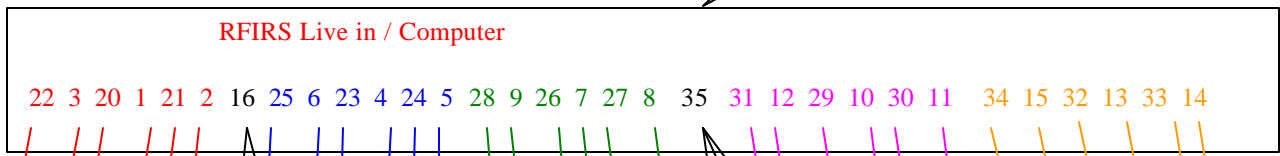


THIS PAGE INTENTIONALLY LEFT BLANK

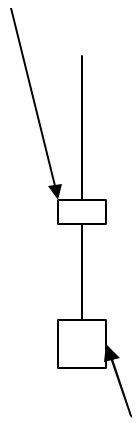
From RFIRS Patch Panel to TP2

o10 Cable

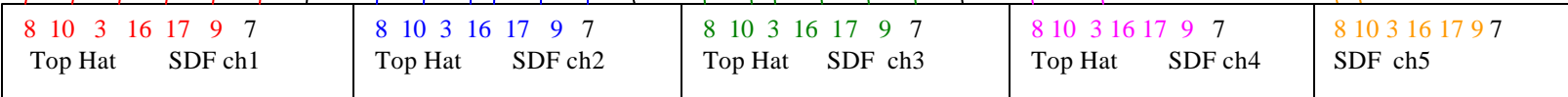
F DB37



Lock washer & nut



Extra long mounting Screw needed to mount on DB25 Mounting plate.



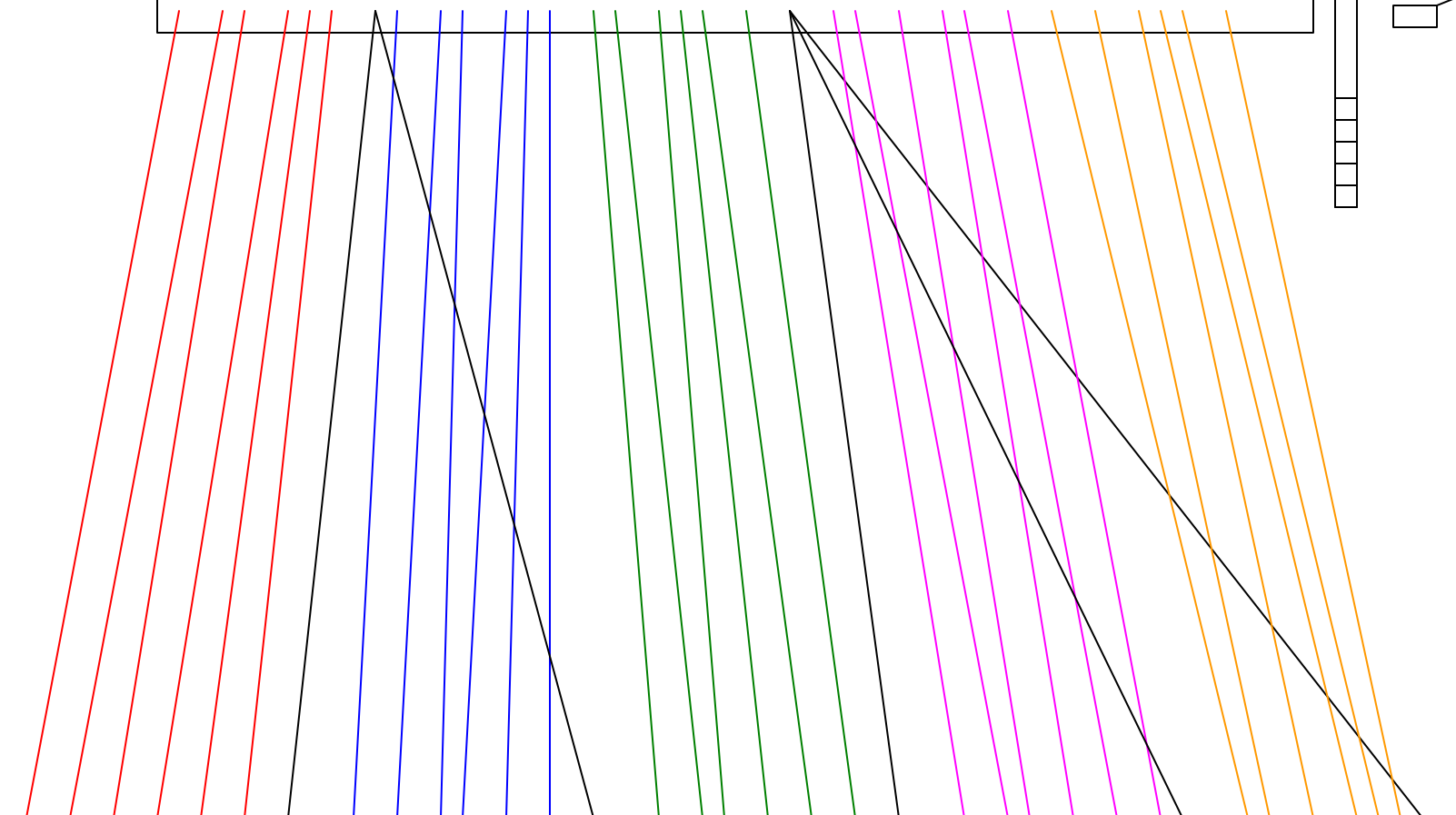
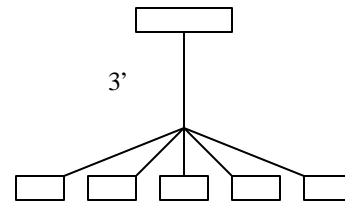
From RFIRS Patch Panel to TP2

o10 Cable

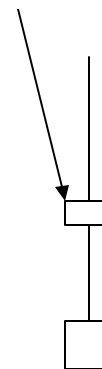
F DB37

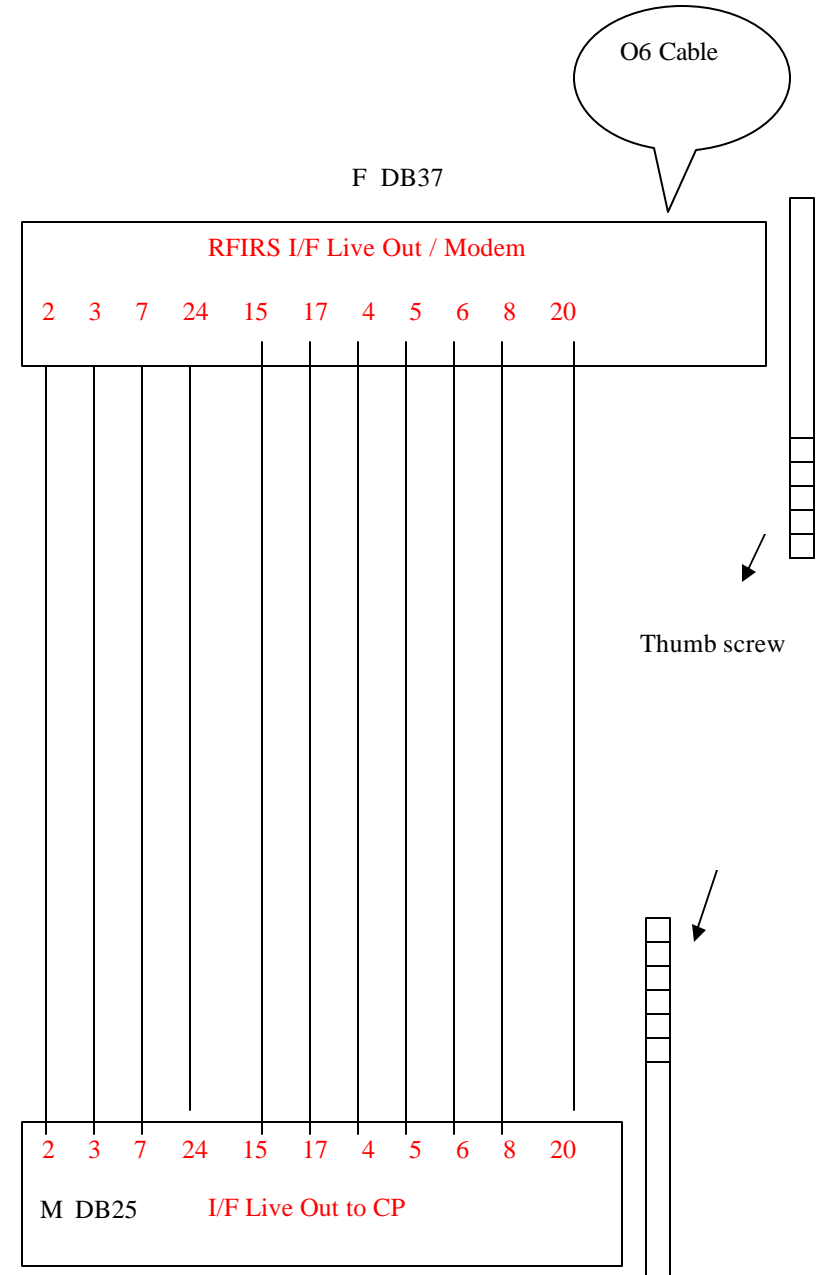
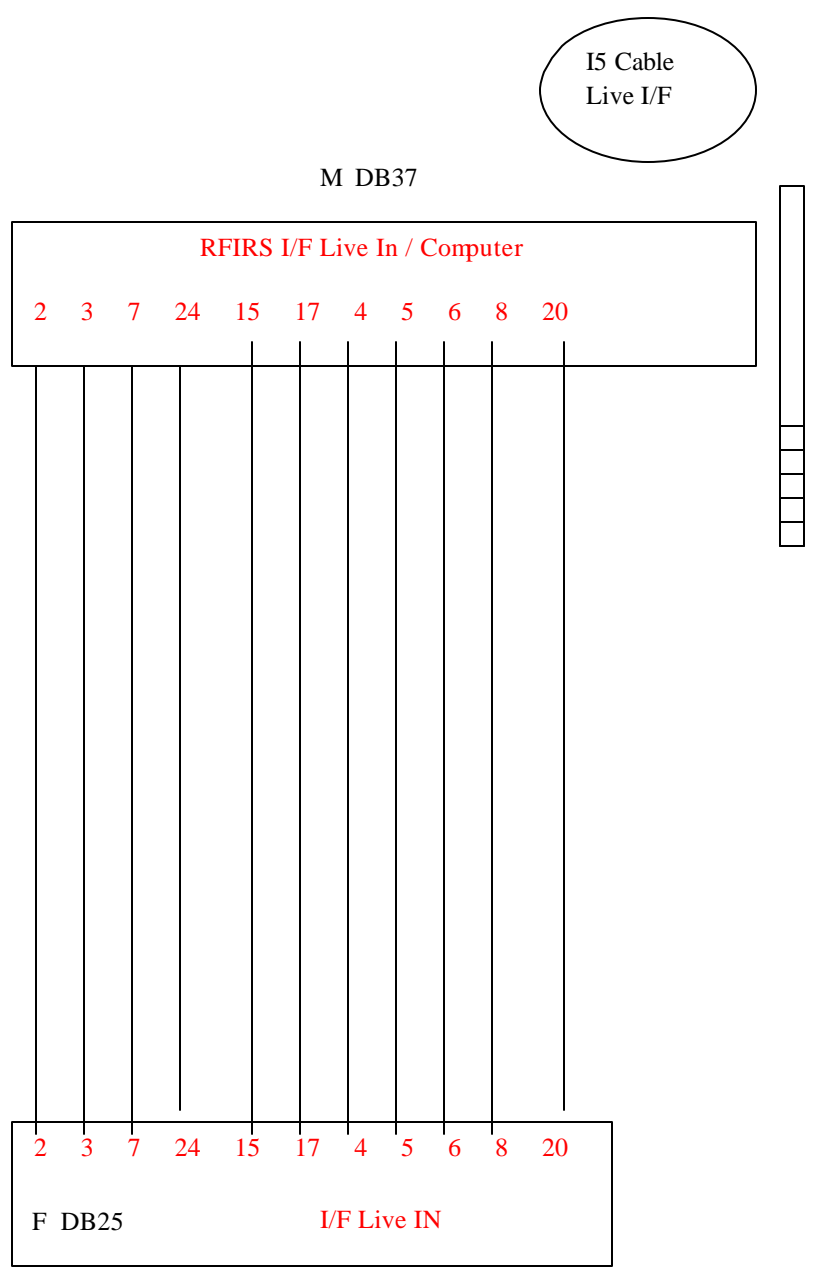
RFIRS Live in / Computer

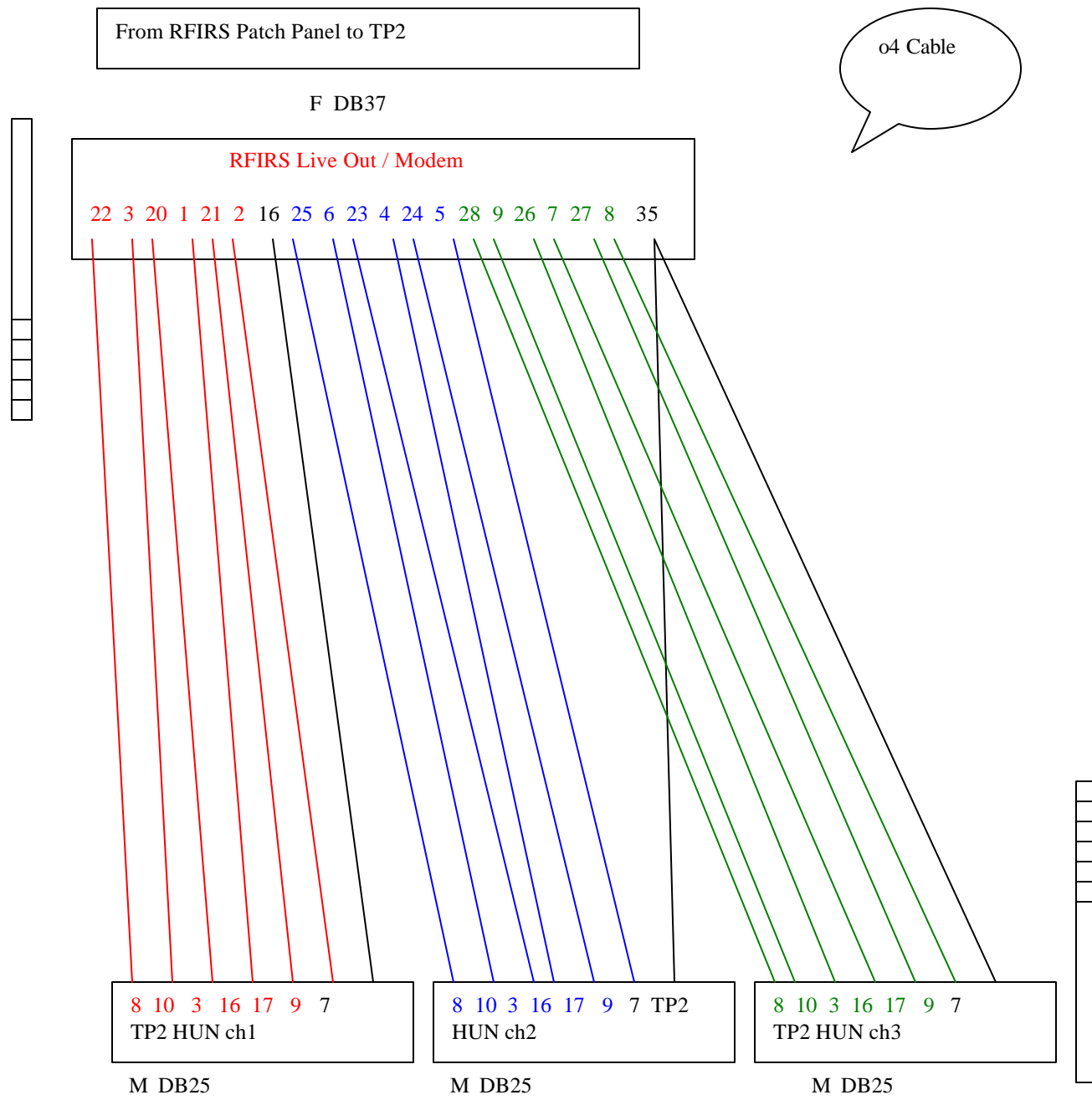
22 3 20 1 21 2 16 25 6 23 4 24 5 28 9 26 7 27 8 35 31 12 29 10 30 11 34 15 32 13 33 14



8 10 3 16 17 9 7	8 10 3 16 17 9 7	8 10 3 16 17 9 7	8 10 3 16 17 9 7	8 10 3 16 17 9 7
Top Hat SDF ch1	Top Hat SDF ch2	Top Hat SDF ch3	Top Hat SDF ch4	SDF ch5







10.0 HARDWARE SPECIFICATIONS

RFIRS Cable Pin-outs

