HITPLAYER-L

and

SPX oem board

Digital audio networking interfaces

MP2 / MP3

Edition: 01/08

Compatible with firmware package : ECI3256.B

Language: English

Reference: AZT30803.DOC

AZTEC RADIOMEDIA
www.aztec-radiomedia.com
Groupe Digigram

Audionetworking solutions …

CONTENTS

VERSION RECORD

Date	Author	Object of the modification	Pages		
			Mod.	**addition**	Supp.
JULY 2000	GM	Creation		All	
APRIL 2001	FD	Update	1-6;12;23;30; 31;37-45;53		
SEPTEMBER 2001	FD	Update	6;8;11;14;22;23 34;51-56;59	53-65	

Information to user...

Note 1

This equipment "HITPLAYER" has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Note 2

The user of the equipment "HITPLAYER-L" has to be aware that intentional or unintentional changes or modifications not expressly approved by AZTEC RADIOMEDIA could void the user's authority to operate the equipment.

Before starting...

Check that your package is complete

- A power cable for the 230V models: check that the model corresponds to the country in which you are located.

Country	Reference
Europe	CA022
USA	CA143

- An Ethernet 10BaseT network - RJ45 cable
- An RS232 ribbon cable (one-to-one) to allow the HITPLAYER-L to be configured in terminal mode (ASCII console), as required.

Following items are not delivered

- This documentation, reference AZT3080[v].PDF where [v] represents the version. Available on our web site: http://www.aztec-radiomedia.com/
- Hard disk Power supply and IDE cables
- Internal IDE hard disk
- IP2 reference manual for developers, réf. azt2502[v].PDF where [v] represents the version.): available on the Web site : http://www.aztec-radiomedia.com/

Following items are available as on demand options

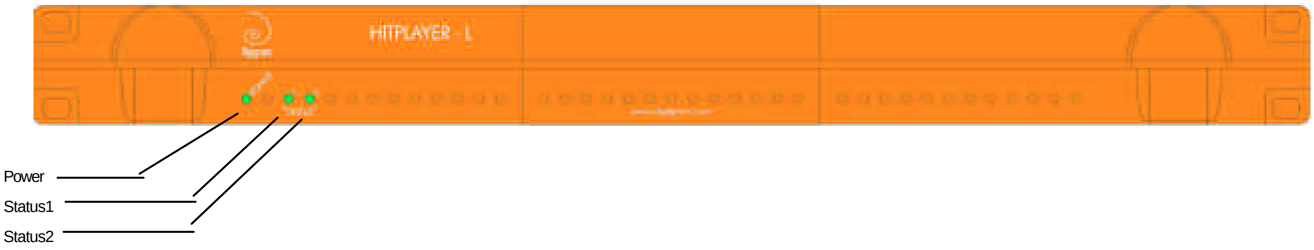
- IP2 reference manual
- Hard disk Power supply and IDE cables
- XLR outputs for professional usage
- XLR audio inputs for specific streaming server applications
- Opto-isolated logical inputs board

Description of the equipment

The front panel of the HITPLAYER –L

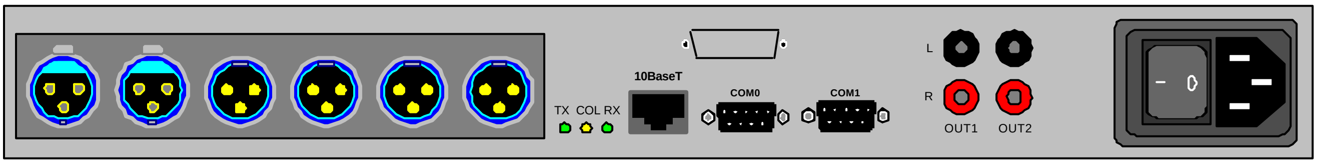
—

(fig.:eci3080.vsd)



PWR	Green	On / off indicator, always lit
Status1	Green	Indicates the operating mode of the OUTPUT1 of HITPLAYER: • Flashes when a Playlist is active. • Flashes 2 times when a file is played. • Lights up when line_out1 outputs line_in or a network audio stream.
Status2	Green	Indicates the operating mode of the OUTPUT2 of HITPLAYER: • Flashes when a Playlist is active. • Flashes 2 times when a file is played. • Lights up when line_out1 outputs line_in or a network audio stream.

HITPLAYER-L : rear panel and connectors



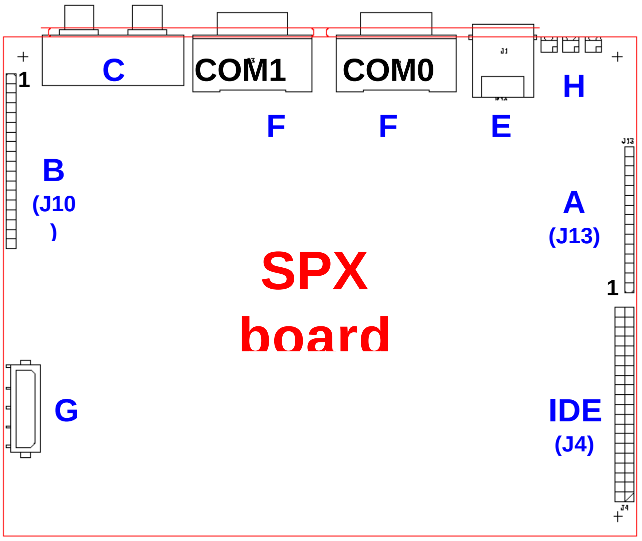
Note : XLR are mounted as on demand options.

TX	**Indicates that data is being received on the Ethernet network**
RX	**Indicates that data is being transmitted on the Ethernet network**
COLLISION	**Indicates the collision of data on the Ethernet network**
COM0	**SUBD9, console and application**
COM1	**SUBD9, application**
10BaseT	**10BaseT RJ45, twisted pairs**
Audio Stereo Outputs **LINE_OUT1 / LINE_OUT2**	**Cinch RCA outputs** **XLR as option**
Logic TLL Inputs/Outputs	**Available internally for specific usages**

🗎 SPX board inside HITPLAYER-L

Note : the SPX product separately as an OEM product.

🗎 SPX board



A	**TTL Inputs / Outputs**	8 TTL inputs / outputs. See below for pin out.
B	**Symmetrical audio out : LINE_OUT1, LINE_OUT2** **Symmetrical audio in : LINE_IN1**	See below for pin out.
C	**RCA outputs : OUT1, OUT2**	Audio out
E	**Ethernet RJ45 socket – 10BaseT**	Ethernet connector
F	**COM0 and COM1 serial ports**	COM0 : local console COM1 : external modem, PPP
G	**Power supply socket**	+5,00V mandatory
H	**3 external leds outputs for the back**	Ethernet TX, RX, COL
IDE	**IDE interface connector**	To plug the HITPLAYER-L Hard Disk if required

🗎 TTL input / outputs (connector A)

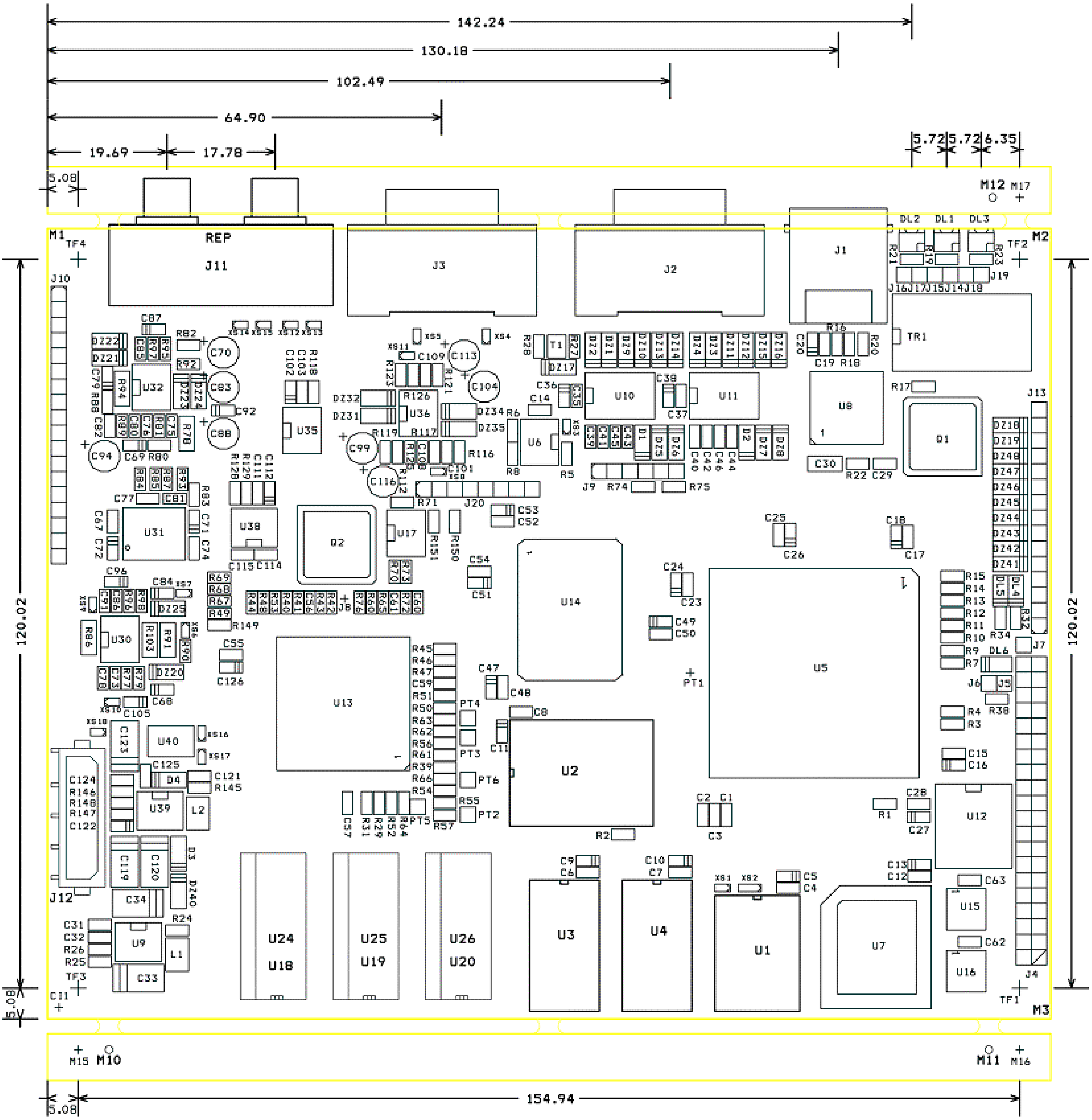
Pin 15	GND
Pin 14	Do not use
Pin 13	Do not use
Pin 12	DIG8
Pin 11	DIG7
Pin 10	DIG6
Pin 9	DIG5
Pin 8	DIG4
Pin 7	DIG3
Pin 6	DIG2
Pin 5	DIG1
Pin 4	Do not use

Pin 3	VCC (+5V)
Pin 2	Do not use
Pin 1	VCC (+5V)

Symmetrical audio connections (connector B)

Pin 1	LINE_OUT2_R+
Pin 2	LINE_OUT2_R-
Pin 3	GND
Pin 4	GND
Pin 5	LINE_OUT2_L+
Pin 6	LINE_OUT2_L-
Pin 7	LINE_OUT1_R+
Pin 8	LINE_OUT1_R-
Pin 9	GND
Pin 10	GND
Pin 11	LINE_OUT1_L+
Pin 12	LINE_OUT1_L-
Pin 13	LINE_IN_R+
Pin 14	LINE_IN_R-
Pin 15	GND
Pin 16	GND
Pin 17	LINE_IN_L+
Pin 18	LINE_IN_L-

SPX board dimensions



Specifications and advantages

Fast installation

The HITPLAYER-L is really easy to use. The 2 audio outputs are always available in the RCA connector format.

Fast configuration

communicating equipment that is easy to connect to a network. It is designed to be connected to any Ethernet IP 10BaseT network. Simply configure its IP address to activate its visibility on the IP network. Once connected to the network, you play with the HITPLAYER-L using simple network tools : your Internet browser (IE or Netscape) and any FTP (as Windows Explorer).

The IP2 simple operating system

The HITPLAYER-L operates on an operating system developed by AZTEC RADIOMEDIA, called IP2. With IP2 you handle your HITPLAYER-L providing line commands through Telnet, Html pages, FTP etc…

You do not need the IP2 user manual to handle HITPLAYER-L basics, everything is contained in this manual. However, if you wan to personalise your product with dedicated web pages, SNMP management, emails notification, logfile reporting, you may need the IP2 reference manual available on the web site of AZTEC RADIOMEDIA.

Simple Playlist management

HITPLAYER-L uses the M3U format to manage your program. This format is ASCII. Enhanced #commands using the M3U format enables you to change in real time the levels and provide crossfades and jingles insertions. Have a look on the extended M3U format proposed by Aztec-Radiomedia, where many “conditional play” features can be added to a playlist very easily, in addition of a fully configurable scheduler. Two independent playlists can be played on each of the HITPLAYER-L audio outputs : further in this document, we speak about PLAYER1 and PLAYER2.

HITPLAYER-L is able to handle playlist containing files, local audio sources and for the future firmware versions the ability to play URL’s and files projected on the network.

The scheduler build in the HITPLAYER-L is inherited from the IP2 system and brings all flexibility to the generation of Jingles in a process which may not be attached to the regular play list process.

Extended streaming functions

HITPLAYER-L integrates streaming functions that enables you to play audio received from the network as streams served by Shoutcast(*), Icecast(*), simple UDP or Soundwan servers. Refer to this manual for more information. Equipped with the audio input option, HITPLAYER-L can be configured to serve streams compatible to Shoutcast(*), simple UDP or SoundWan.

(*) this features are not currently available, and is a part of the functions roadmap of the product.

IDE hard disk compatible, FAT32 compatible

HITPLAYER-L accepts any kind of IDE hard disk or flashdisk. It provides IDE standard connection and a power supply which is dimensioned to supply regular types of hard disks.

CAUTION: IDE hard disk must contain a FAT32 active partition before being installed in the Hitplayer-L.

Power supply and IDE cables are available as on demand options.

Powerful remote control and file synchronisation with FTP

Thanks to the IP2 system, HITPLAYER-L can be remote controlled through many ways : Telnet, FTP, FTP synchronisation, HTTP, etc…

HITPLAYER-L handles a dated events log file, for each event you can configure whether or not an Email, SNMP trap or HTTP CGI request line has to be generated.

You can configure HITPLAYER-L to synchronize its playlist and music files from an image FTP server placed on the network or even on the Internet.

Serial port with PPP

HITPLAYER-L integrates a PPP client than enables it to connect to the Internet using an external modem.

Modem Access

When connected to a standard external modem, HITPLAYER-L accept incoming calls initiated for example by Hyperterminal and provides a VT100 human interface.

Security is insured by user/password. Several users, with different management rights can be configured.

The HITPLAYER-L specifications

General information	• Power supply: o Rack : 100 - 240 VAC 50Hz - 60Hz, IEC connector. o SPX Board : 5VDC(800mA) + 12VDC (100mA) • Consumption: 25W Max (Rack version) / 6W Max (SPX Board) • Dimensions: Rack 19" - 1U, depth 220mm / SPX board - 130 x 165 x 30 mm • Weight : Rack : 2600g without any options / SPX board : 160g • RCA audio line outputs / professional XLR as options on rack version only. • Operating conditions: Humidity: 0 - 95% / Temperature: 0 - 45°C.
Communication ports **COM0** **COM1**	• Ethernet 10BaseT RJ45 • Ethernet, TCP/IP (when no PPP client configured) • 2 V24 (RS232) ports: COM0 (console) and COM1 (PPP client and terminal server)
Line level audio outputs **LINE_OUT1** **LINE_OUT2**	• 2 stereo outputs without mixing and without crossfades • or 1 stereo output with mixing and crossfades • DSP, Resolution: 24 Bits, DSP56303 • Total harmonic distortion: 0.01% • Dynamic range: 105dB • Left / right channel crosstalk: 80dB • Difference of gain between left and right channel: 0.5dB • Frequency response: 20Hz to 20Khz (+/-0.5dB) • Output level: 0dBu
Line level audio inputs **LINE_IN**	• OPTION • 1 stereo input, LINE_IN • XRL connectors for both audio out and stereo input
Audio storage	• External Hard Disk IDE type, not included • FAT32
Digital TTL Inputs / outputs	• Available internally for specific usage • 8 TTL i/o, protected, common ground • Opto isolated inputs board available in option
Administration	• SNMP • MIB II
Configuration	• HTML pages, Telnet session or local RS232 port
File system	• UNIX, LINUX compatible • On board Flash : 4 Mb, about 1Mb reserved for the application firmware • Updating of the HITPLAYER-L firmware by FTP
History files	• On mains fault • All connections noted

	• HTML ASCII file (histo.txt and histo.html) • Audio Files played files can be logged

To connect the HITPLAYER-L

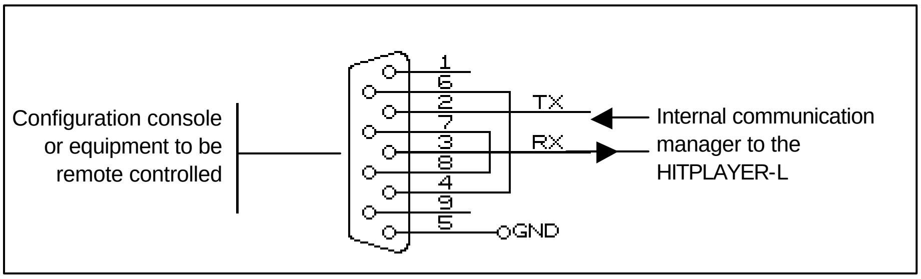
All the physical interfaces with the outside world are located on the rear panel of the HITPLAYER-L.

- The 230V power supply socket
- The on/off button
- The COM0 communication port (RS232, V24) that normally operates in console mode, but can also be used for the NETCOM service.
- The RS232 – V24 COM1 communication port for external modem
- The RJ45 connector, 10BaseT allowing access to the network. This access is isolated

Com0 port

COM0 is an RS232 DCE port (female), it is located on the rear panel of the HITPLAYER-L and is for configuration in console mode (9600,N,8,1).

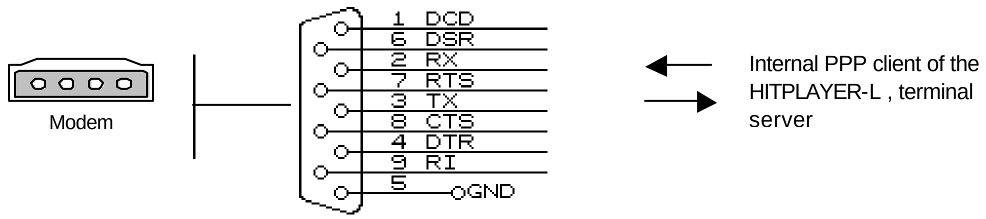
The connection to a console or computer is made via the ribbon cable (supplied with the equipment). There are no control signals on this port and the corresponding lines are connected in a loop as shown in the figure below.


Configuration console or equipment to be remote controlled
1
6
2
7
3
8
4
9
5
TX
RX
GND
Internal communication manager to the HITPLAYER-L

Note: the ground (gnd) of this communication port is electrically connected to the earth pin on the mains connector.

Com1 port

The COM1 port is an RS232 DTE port (male) and is there to be connected to an external modem for remote control and remote loading.



Ethernet port

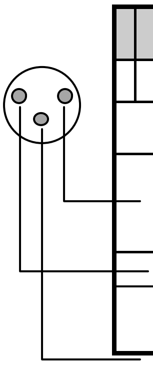
Isolated RJ45 connector for Ethernet 10BaseT network, Twisted pairs.

Electrical supply

IEC connector 100 / 240V/ 50hz.

Audio Line input LINE_IN (option)

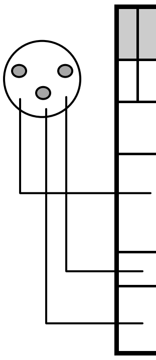
The audio input option is stereo and symmetrical :



Symmetrical Audio Input LINE_IN	
Female XLR connectors	
Pin	**Signal**
1	**Ground** of the HITPLAYER-L. It is recommended to connect the screen of the cable used to this pin, when possible.
2	**Audio (+)**
3	**Audio (-)**. This pin should be connected to pin 1 (ground) for an asymmetrical audio supply

Audio Line Outputs LINE_OUT1 and LINE_OUT2 (XLR option)

The audio outputs are symmetrical.



Symmetrical Audio Outputs	
Male XLR connectors	
Pin	**Signal**
1	**Ground** of the HITPLAYER-L. It is recommended to connect the screen of the cable used to this pin, when possible.
2	**Audio (+)**
3	**Audio (-).**This pin should be connected to pin 1 (ground) for an asymmetrical audio supply

🗐 HITPLAYER-L logical architecture

🗎 Warning…

Before tackling this section, be sure that you have read the IP2 system reference guide. This guide outlines the conventions relative to administering the HITPLAYER-L IP2 product.

Do not look in this manual for the way in which the network interface is configured: this is explained in detail in the IP2 system reference guide (ref. AZTEC RADIOMEDIA n°2502, pdf available on AZTEC RADIOMEDIA's Web site www.aztec-radiomedia.com).

The commands to be addressed to the HITPLAYER-L's command interpreter are always in the ASCII format (in plaintext) and can be entered via the COM0 port (with serial cable), Telnet port 23 (TCP connection), FTP with the quote site command or HTTP thanks to the overlay technique specified in the IP2 reference guide.

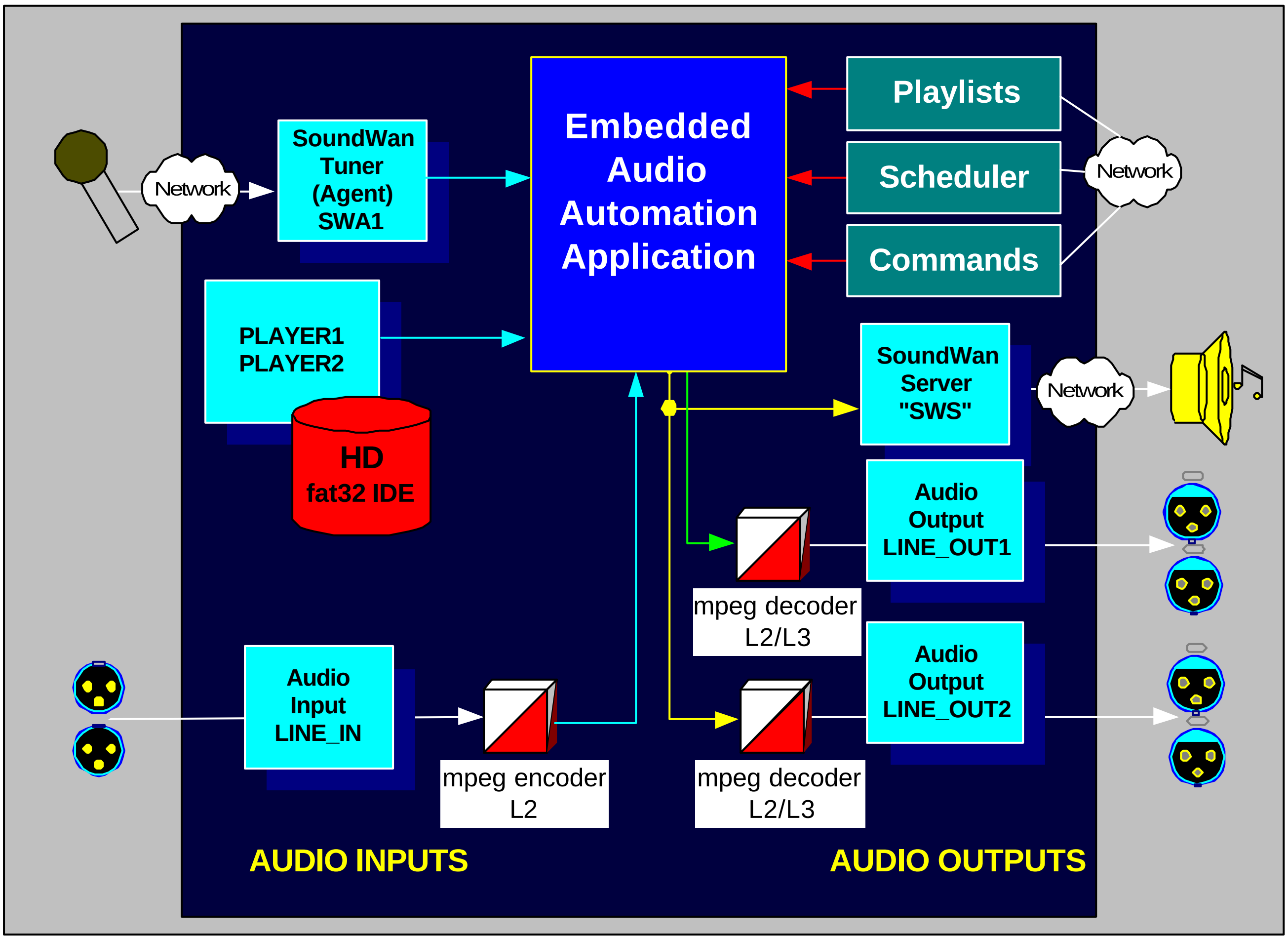
Thus, for example, it is possible to configure the state of the HITPLAYER-L's relays. This chapter describes the commands and the events likely to drive the HITPLAYER-L.

CAUTION: a good understanding of the IP2 system will allow you to make the most of the HITPLAYER-L's network functions and in particular, its embedded Web server and its ability to send emails, etc…

The on-line command **HELP.APPLI** addressed to the HITPLAYER-L displays the on-line help contained in the product.

🖹 Architecture description …

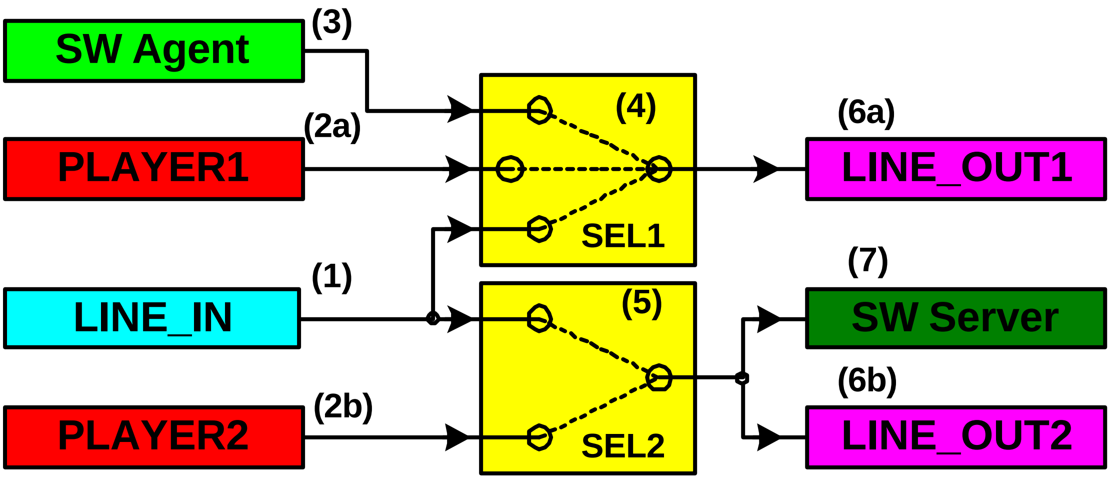
The following figure represents the logical architecture of the HITPLAYER-L. Several audio inputs and outputs are linked by an Audio Selector. The Audio Selector is driven by the HITPLAYER-L application which can be remote controlled by commands, or standalone using a playlist.



- Options mentioned in this synoptic :
  - o Audio input (no audio input by default)
  - o XRL connectors (Cinch RC by default)

🖹 Physical and logical Audio Inputs and Outputs

🖹 Block diagram from audio sources to audio outputs



Audio sources are listed on the left side of the synoptic, audio outputs are listed on the right side.

🖹 Physical audio inputs and outputs

These *physical* devices correspond to a physical audio connection (XLR or RCA/Cinch) on the Hitplayer-L :

- LINE_IN
- LINE_OUT1
- LINE_OUT2

🖹 Logical audio inputs and outputs

These *logical* devices correspond to a source of sound which is not a connector. :

- PLAYER1
- PLAYER2
- SoundWan Agent (SW Agent)
- SoundWan Server (SW Server)

🗎 Audio selectors, synoptic (4) and (5)

To route audio from inputs to outputs (logical or physical), there are 2 selectors **SEL1** and **SEL2**.

Selector SEL1 is feeding LINE_OUT1 physical output.

Selector SEL2 is feeding both with the same content LINE_OUT2 physical output and the SoundWan server.

The following commands enable to remote control the selectors. These commands are powerful and can be used using the scheduler or the playlist features of the HITPLAYER-L.

SEL1=SWA

SEL1=PLAYER1

SEL1=LINE_IN

SEL1? returns the current audio source selected on selector 1

SEL2=PLAYER2

SEL2=LINE_IN

SEL2? returns the current audio source selected on selector 2

🗎 General description of the Analog Audio Input : LINE_IN, synoptic (1)

One stereo input is available as an **option** in the HITPLAYER-L. This stereo input is directly attached to a MPEG coder.

🗎 Gain control of digital players PLAYER1 and PLAYER2

LINE_IN.GAIN=<-110/+18 dB> : adjust the audio level of the AUDIO IN with a correction gain given in dB.

🗎 Fade in / out on each audio output

Use the **LINE_IN.FADE=<s >,<g >**

where :

--> **s** is the absolute slope in dB/s up to 125dB/s

--> **g** the final gain value in dB (valid gains are those specified with the **LINE_IN.GAIN=** command)

Note : We advice the use of the fader command inside batch .CMD files. For the system to take in account the duration of the action of the fader, use the IP2system **WAIT=<time>** (time is explain in steps of 50ms) command after the FADE… command.

Configuration of the audio input is made as followed :

LINE_IN.CODER=<coder Type>,<bitrate>,<stereo/mono>,<sample_freq>

Codec Type	**MP2**	**MP3** **caution: Decoder only**	**GSM**
Possible bit rates (kbps)	32, 48, 56, 64, 80, 96, 112, 128, 160, 192	8,16, 24, 32, 40, 48, 56, 64, 80, 96, 112, 128, 144, 160, 176, 192	Not implemented in this version
Possible sample freqs (KHz)	32, 44.1 or 48 kHz	16, 22.05, 24, 32, 44.1 or 48 kHz	Not implemented in this version
Stereo	Stereo or Mono or Jstereo	Stereo or Mono or Dual	Mono only

CAUTION: Line_out 1 & 2 have the same sample frequency. Hitplayer-L can automatically adapts its outputs sample frequency as the two sources have the same sample frequency. For example, if player1 and player2 play files with the same sample frequency, the Hitplayer-L will adapts its sample frequency. If two sources are active, the Hitplayer will set its sample frequency at the value defined by the **LINE_IN.CODER.**

Description of AGC function on the analog audio input LINE_IN

The HITPLAYER-L has an Automatic Gain Control function acting on the audio input levels applied to the audio input LINE_IN. This function permits to adjust the input gains automatically.

AGC needs two parameters to work:

- **LINE_IN.L_AGC** : Level wanted at the codec input.
- **LINE_IN.AGC_GATE** : Gate level under which the AGC fixes.

AGC sets the input gains in order to compensate for the input signal fluctuations.
AGC will add/subtract up to 9 dB to the inputs gain.

AGC function Activation/Deactivation on the audio input on LINE_IN (if audio input option installed)

The command **LINE_IN.AGC=[ON|OFF]** allows to active / deactivate AGC. The command **LINE_IN.AGC?** displays the state of this function.

AGC level setting on the audio input on LINE_IN (if audio input option installed)

The command **LINE_IN.L_AGC=<0dB/-6dB>** sets the level that the AGC will have to maintain at the codec inputs. The command **LINE_IN.L_AGC?** displays the selected value.

🗎 AGC gate setting on LINE_IN (if audio input option installed)

In order to freeze the AGC function during audio blanks an **AGC_GATE** level can be set. When the input level is under the AGC_GATE level it freezes. The command **LINE_IN.AGC_GATE=<-50dB/-10dB>** sets this parameter. The command **LINE_IN.AGC_GATE?** displays the selected value.

🗎 Description of audio input detection on LINE_IN (if audio input option installed)

The HITPLAYER-L is able to monitor the audio input level. The following events are generated :

LINE_IN_LOST : no input signal or input level is extremely low.
LINE_IN_OK : audio input recovered or level is correct again.
To avoid untimely events the signal has to be low for more than **TO_DETECT** seconds before generating an LINE_IN_LOST event and high for more than **TO_RECOVER** seconds before generating an event LINE_IN_OK.

🗎 Low threshold setting on LINE_IN for audio detector

The command **LINE_IN.TL_DETECT=<-50dB/-10dB>|OFF** sets the low level threshold, When set to OFF the line in detection is deactivated. The command **LINE_IN.TL_DETECT?** displays the selected value.

🗎 Setting of TO_DETECT timeout for LINE_IN

The command **LINE_IN.TO_DETECT=<0s/240s>** sets the time_out before generating the LINE_IN_LOST event . Typing **LINE_IN.TO_DETECT?** command displays the selected value.

🗎 Setting of TO_RECOVER timeout for LINE_IN

The command **LINE_IN.TO_RECOVER=<0s/240s>** sets the delay before generating the LINE_IN_OK event . Typing **LINE_IN.TO_RECOVER?** command displays the selected value.

Players, synoptic 2a, 2b

2 players are implemented on the HITPLAYER-L : PLAYER1 and PLAYER2. These players are activated with **PLAY1=** and/or **PL1**= and **PLAY2**= and/or **PL2=** commands.

Players are able to read playlist and music files, decode them and generate the appropriate audio sources.

Refer directly to the corresponding chapter of this manual.

Gain control of digital players PLAYER1 and PLAYER2

PLAYER_OUT<n>=<-110/+18 dB> : adjusts the digital audio level of the player #n with a correction gain given in dB.

Turning OFF the players

PLAYER_OUT<n>=OFF to turn OFF the digital audio player #n. To turn ON again, just set a level or use the command below.

Turning ON the audio outputs

PLAYER_OUT<n>=ON to turn ON the digital audio player #n.

Analog Audio Outputs, synoptic 6a, 6b

Gain control of audio outputs LINE OUT1 and LINE OUT2

MASTER_OUT<n>=<-110/+18dB> : adjusts the audio level of the LINE OUT #n with a correction gain given in dB.

MASTER_OUT<n>_PAN=<-110/+110dB > : adjusts the balance of the audio output, in dB. Negative value means LEFT, positive values means right.

Note : all these parameters are linked. If you touch one, the others may change. HITPLAYER-L takes care of out of range possibilities.

Turning OFF the audio outputs

MASTER_OUT<n>=OFF to turn OFF the analog audio output #n. To turn ON again, just set a level or use the command below.

Turning ON the audio outputs

MASTER_OUT<n>=ON to turn ON the analog audio output #n.

Stereo / Mono

MASTER_OUT<n>_STEREO=ON

MASTER_OUT<n>_STEREO=OFF

Set ON or OFF the stereo mode of the audio output #n. Default value is ON.

Mixing the 2 audio outputs LINE_OUT1 and LINE_OUT2

The 2 stereo audio outputs can be mixed using the following commands :

MIXER=ON

MIXER=OFF

MIXER? to read the state.

When mixer is turned ON the audio output LINE_OUT1 supplies the mixed signal. The LINE_OUT2 is disable and Soundwan Server is disabled.

Fade in / out on each audio output

Use the **MASTER_OUT _FADE<n>=<s >,<g >**

where :

a) n is the output concerned

b) s is the absolute slope in dB/s up to 125dB/s

c) g the final gain value in dB (valid gains are those specified with the **MASTER_OUT<n>=** command)

Note : We advice the use of the fader command inside batch .CMD files. For the system to take in account the duration of the action of the fader, use the IP2system **WAIT=<time>** (time is explain in steps of 50ms) command after the FADE… command.

Crossfade between audio outputs LINE_OUT1 and LINE_OUT2

This function are directly integrated in the PLAYERs command sets. Refer to the appropriate chapter.

SoundWan server (logical audio output), synoptic (6)

The SoundWan server is used mainly to feed one or several SoundWan audio tuners on the network.

SoundWan is the protocol to be used for Intercom systems.

The following properties, events and methods defines the SoundWan server behaviour.

The SoundWan server is attached logically to PLAYER #2 of the HITPLAYER-L then implicitely the sound delivered on the audio output #2 is the same as the sound broadcasted on the network by the SoundWan server.

SWS=ON,<ch> : method which turns ON the SoundWan server on the network, on channel number <ch>, from 1 to 255

SWS=OFF : method which turns OFF the SoundWan server

SWS? : returns whether or not the server is ON (with the corresponding channel) or OFF.

SWS.IP=<ip address>:<port>

Defines the destination address of the SoundWan channel, it can be an unicast, multicast or broadcast address. Default value is 0.0.0.0:9000 (UDP broadcast on port 9000). **SWS.IP?** to read the value.

SWS_ON : this IP2 event appears when the SoundWan server is turned ON

SWS_OFF : this IP2 event appears when the SoundWan server is turned OFF

Caution: SWS must be restarted to make effective any parameter change.

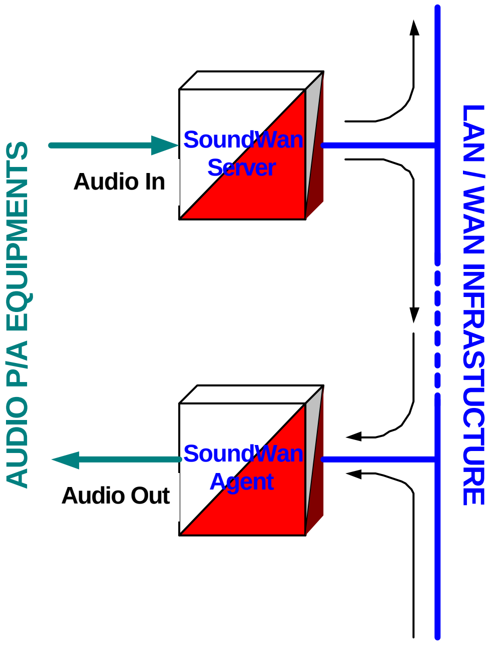
🗎 SoundWan Agent (SWA) (logical audio input), synoptic (3)

The SoundWan tuner listens to the network for any SoundWan audio channel that may be present and is automatically turned if the detected channel matches with the agent configuration.

The following figure represents a SoundWan agent and a SoundWan server placed on a network.

The following steps will help you to understand and configure the SoundWan agent.

1. **INTRO** : The SoundWan server generates an audio channel on the network. The channel number is specified on the server side. If it's a HITPLAYER-L that is generating the SoundWan stream, then refer to the previous chapter.



2. **UDP PORT SETTING:** Soundwan uses UDP broadcast streaming, so like the Soundwan Server the UDP Port of the agent must be set.

   To set the UDP port: **SWA1.PORT=<port>**

   To get the current setting : **SWA1.PORT?**

3. **STARTING THE AGENT** : The SoundWan agent is configured to listen to a certain number of channels. HITPLAYER-L SoundWan agent listens up to 16 SoundWan channels.

   To turn ON the SoundWan agent : **SWA1.ENABLED=ON,[multicast ip adress]**

   Multicast address must be specified only if the Soundwan server uses a Multicast adress.

   To turn OFF the SoundWan agent : **SWA1.ENABLED=OFF**

To get the current state : **SWA1.ENABLED?**

4. Following events are generated when the SoundWan agent is turned ON and OFF : **SWA1_ON** and **SWA1_OFF**
5. **LIST OF ACCEPTABLE CHANNELS** : A list of channels to be monitored by the Agent is defined using the **SWA1.LST(<n>)=<ch>[,<gain>]** command where n represent the rank of the channel (1 to 255) in the list. Default gain-offset is 0 in (dB). Given gain is used (when specified) when analog output is fed by the SoundWan agent, the gain-offset is set to the MASTER_OUT1 gain.
   To read the full list : **SWA1.LST?**
   To read element n of the list : **SWA1.LST(<n>)?**

6. **PRIORITY BETWEEN CHANNELS** :The first channel defined in the list (**SWA1.LST(0)**) has the highest priority. The last channel defined in the list (**SWA1.LST(15)**) has the lowest priority.

7. **PRIORITY** : It's the channel received with the highest priority that is decoded by the SoundWan agent.

8. **VOLUME** : Each channel defined in the list has an associated gain-offset setting in DB by the following command: **SWA1.LST(<n>)=[<ch>][,<gain>]** . This gain-offset applies only if an analog audio output is used on the HITPLAYER-L.
   To reading the gain configured for each SoundWan channel of the list: **SWA1.LST(<n>)?**
   To read the full list settings : **SWA1.LST?**

9. **STATUS** : current status of the SoundWan agent is obtained using the following command : **SWA1.CH_CUR?** Which gives the current decoded SoundWan channel.

10. **EVENTS ON INCOMING CHANGE OF STREAMS** : when the agent detects a new stream to be decoded or not anymore stream to decode, then events **SWA1_CHANGE** , **SWA1_ON** and **SWA1_OFF** are generated by the system with, in associated parameters, the value of the (new) channel decoded or OFF if not anymore channel to be decoded.

Caution: SWA must be restarted to make effective any parameter change.

🗎 HTTP file reader (request for comments, * not implemented yet)

Possible implementation in future versions.

This function is implicitely built in the PLAY commands. Then the logical audio input associated to this function can be considered as a file.

🗎 Streaming audio client (request for comments)

Possible implementation in future versions.

This function is implicitly built in the PLAY commands. Then the logical audio input associated to this function can be considered as a file.

🗐 The PLAYER application inside the HITPLAYER-L

🖹 Two Players inside one HITPLAYER-L… why?

The HITPLAYER-L handles 2 simultaneous and independent players : PLAYER1 and PLAYER2. Each player is remote manually by individual **PLAYn=** commands or through the Playlist concept initiated by the **PLn=** commands, n represent the number of the player (1 or 2).

PLAYER1 and PLAYER2 are feeding analog audio outputs respectively LINE_OUT1 and LINE_OUT2.

🖹 The PLAY command

🖹 PLAY command syntax

Players are named PLAYER1 and PLAYER2. When no figure 1 or 2 is given in the command to indicates for which player the command is sent, PLAYER1 is assumed to be the default.

Generic syntax of the PLAY command is :

PLAY[<n>]=<source>

(*) WARNING : not implemented : request for comments.

<source> : any valid source such as the one presented below :

- **local file name**
  <source>=/<file path>
  local file is stored inside the flash memory

- **ATA file name (external hard disk):**
  <source>=/ata/<file path>
  ATA stands for IDE hard disk extension

- **http file name (*) :**
  (*) : not implemented, request for comments
  <source>=http://<ip>[:<port>][(<domain name>)]/<file path>
  http://123.161.4.56[:80][(www.aztec-radiomedia.com)]/toto.mp3

Provide the Domain Name inside ( ) when multiple domains are hosted by the http server.

- **http Shoutcast stream name (*) :**

  (*) : not implemented, request for comments

  <source>=http://<ip>:<port>

PLAY<n>? command returns the current source being played. When <n> is specified it reports to PLAYER #1.

Note : when the file extension is .M3U or .PLS, the **PLAY=** command acts the same as the **PL=** command.

How to get the current position of a file being played

This section just apply to files being played.

POS1? , **POS?** : indicates the proportion of the file already played in % for player #1
POS2? : indicates the proportion of the file already played in % for player #2
NOTE : when no progress in % can be evaluated (eg: SWA1 or LINE_IN inputs played), '?' is returned.

How to stop or pause a PLAY ?

The **STOP1** and **STOP** commands are equivalent and stop the current play operation on PLAYER1
The **STOP2** command is the same for PLAYER2.

The **PAUSE1** and **PAUSE** commands are equivalent and pause the current play operation on PLAYER1
The **PAUSE2** command is the same for PLAYER2. When PAUSE is executed in a playlist, timers are continuing and have priority on the PAUSE action : then if MAX_TIME is defined for a playlist item, it's regardless whether on not a PAUSE have been executed when this playlist item was playing.

The **CONT1** and **CONT** are equivalent and resume a PAUSE for a PLAY or a PLAYLIST operation that has been previously paused on PLAYER1. The **CONT2** command is the same for PLAYER2.

Jingle insertion in a playlist with the OVERPLAY and OVERPAUSE commands

Principle

A jingle insertion can be inserted at any time during the time a Playlist is running.

The Jingle insertion as it is specified here is not done with mixing and just insert a source called “Jingle” between an existing play of a source called “A”.

This is done using the following command called “overplay” :

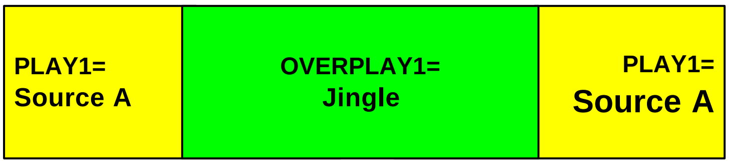
OVERPLAY[<n>]=<*source*>

OVERPAUSE[<n>]=<*source*>

The syntax of this command is exactly the same as the **PLAY=** command.

This command is equivalent to the following steps and commands :

1. Pauses the current source if it’s a file and only if the request comes from OVERPAUSE=
2. Perform an equivalent PLAY<n>=***<jingle source>*** at the nominal value
3. Restart the initial source if paused



Note 1 : OVERPLAY and OVERPAUSE do not perform a mix between the 2 files being played

Ending an OVERPLAY or OVERPAUSE

Use **OVERSTOP[<n>]** commands to stop explicitly a file currently overplaying. If the STOP command is sent during the time a file is overplayed, then the STOP command applies to the overplayed file.
When an OVERSTOP command is sent, then the main file being played comes back.

Note 1 : for remote automation, do not use STOP to stop a jingle which is overplayed as you are never sure on which item the STOP command is applied (main play or overplayed item).
Note 2: During an Overplay, if the initial file played was is in PAUSE state, then the initial file is put in PLAY mode again automatically.
Note 3: The right manner to use overplay is to let it ends automatically when the Jingle file ends, then the main play restart or appears again.

Mixing, crossfades, faders

🗎 Turning the Mixer ON / OFF

This is one of the powerful feature of the HITPLAYER-L. It is achieved by linking the 2 players output together with the **MIXER=ON** command.

WARNING: When **MIXER=ON** LINE_OUT1 outputs the signals coming from PLAYER1 and PLAYER2 of the HITPLAYER and LINE_OUT2 is inactive.

When **MIXER=ON** , then 2 two players are joined and no explicit player number must be used in commands. HITPLAYER handles automatically for you the assignment of the player numbers, searching the first player free for operation.

🗎 Crossfade between items in a PLAYLIST

a) Directives concerning individual items (files in the playlist) :

#FADE_START:<duration in seconds>

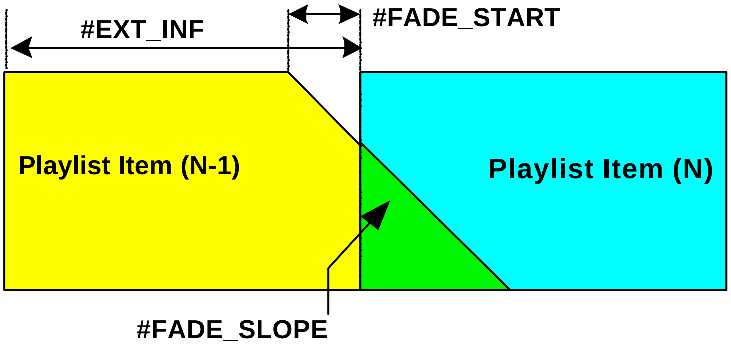
Defines when to start fade out regarding the starting point of the next playlist item. This parameter only apply to one item.

#FADE_SLOPE:<decreasing slope expressed in dB/s (positive value)>

b) Directives concerning all items (files) in the playlist (default values) :

#PL_FADE_START:<duration before stop in seconds>

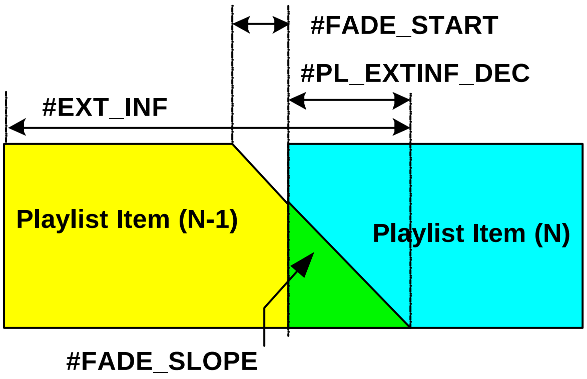
#PL_FADE_SLOPE:< decreasing slope expressed in dB/s (positive value)>



🗎**Three different ways to manage crossfades:**

A. Using the three directives below. In this case you'll have to modify all the EXTINF values in order to set the different crossfades durations, this case is shown on the figure above.

B. Using the directive **#PL_EXTINF_DEC:<value to substract in seconds>**. This directive will substract automatically n seconds to all EXTINF values of the playlist. In this case illustrated by the following figure:



Example:

If you add these three directives at the beginning of your playlist you'll have crossfades at each end of file without changing the EXTINF generated by your playlist editor.

#PL_FADE_START:5

#PL_FADE_SLOPE:2

#PL_EXTINF_DEC:5

C. Using the three following IP2 Commands:

PL_FADE_START:<duration before stop in seconds>, default value is 0

PL_FADE_SLOPE:< decreasing slope expressed in dB/s (positive value)>, default value is 0 (no fade)

PL_EXTINF_DEC:<value to substract in seconds>, default value is 0 (EXTINF is not decreased)

These commands change the default values of the three crossfade parameters. These default values are ignored when the playlist contains the associated directives that specify these three parameters. Using these three commands permits to have some crossfades between the playlist items without any modification of the playlist file. You can overwise these default value by using in the playlist the directives described above.

🗎 Jingle insertion in a playlist

INSERTPLAY=<item to play>[,<relative gain (dB)> ,<fade start slope (dB/s)>,<fade end slope (dB/s)>]

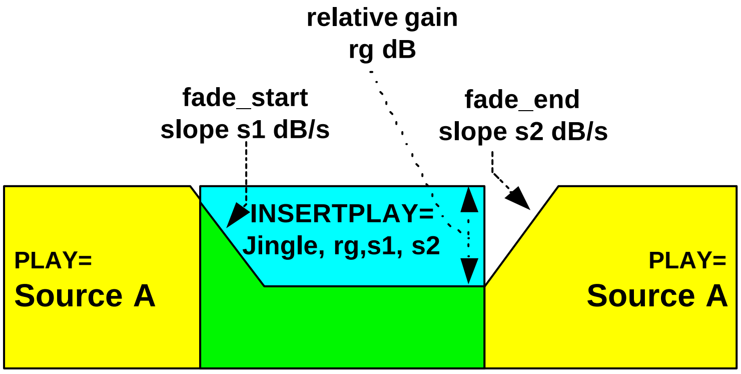
a) item to play is the file to insert

b) relative gain is the gain to apply to the source A, from 0 to -100dB

c) fade start slope in dB/s, from 1 to 100 dB/s

c) fade end slope in dB/s, from 1 to 100 dB/s

This command asks the HITPLAYER to insert a file during a first one is already playing. Automatic assignment of the player number is then performed to play the second file in the same time than the previous one. The *fade in* and *fade out slopes* concerns only the source A. If relative gain is set



🗎 Playlists

🗎 Introduction to Playlists inside the HITPLAYER-L

HITPLAYER-L manages playlists following 2 different classical formats :

.PLS format

.M3U format

These 2 formats are really the standard on the market and handle by many many softwares and sharewares that are managing playlists and audio automation on PC's. These formats are fully ASCII, and playlists can be created using a simple text editor like notepad.

Juste type **+pls +playlist** or **+m3u +playist** on a search engine like www.altavista.com and you will find many software that will help you to create playlist.

For advanced automation needs, AZTEC RADIOMEDIA has designed an extended set of instructions to be used inside the .M3U format that will enable your HITPLAYER-L to become a real standalone radio and sound automation system.

This extended set of M3U instructions enables you to :

- insert jingles mixed to the current program
- make crossfades between 2 files
- make conditional plays of audio sources (files, audio input, url(*), etc…) according the date, time, day it is
- make conditional plays of audio sources (files, audio input, url(*), etc…) according the number of plays that already occurred
- make conditional plays of audio sources (files, audio input, url(*), etc…) according the number of plays per hour / day / week
- lunch a new playlist
- execute any command or command file before and after the audio source is played

(*) : not implemented, request for comments

The next paragraphs will lead you to understand

- the M3U regular playlist format
- the extended M3U playlist format supported by the HITPLAYER-L and B
- the .PLS regular playlist format

When played with the PLAYLIST command, a playlist starts immediately.

Playlist commands

The following commands has to be used for playlist operation :

PL1=<playlist file name>, **PL=<playlist file name>** activates playing a sound file according to a playlist for player #1

PL1?, **PL?** indicates which playlist is active for player #1

PL2=<playlist file name>: activates playing a sound file according to a playlist for player #2

PL2?: indicates which playlist is active for player #2

PL_STOP1, **PL_STOP** :stops playing a sound file or a playlist for player #1

PL_STOP2 : stops playing a sound file or a playlist for player #2

PL_PAUSE, **PL_PAUSE** pause of the play of a playlist Item for player #1

PL_STOP2 : same for player #2

PL1_SKIP , **PL_SKIP** : moves on to the next file/source in the playlist for player #1

PL2_SKIP : moves on to the next file/source in the playlist for player #1

PL1_POS? , **PL_POS?** : Indicates the audio source (file, url, etc…) that is currently being played on player #1
PL2_POS? : Indicates the audio source (file, url, etc…) that is currently being played on player #2

PL1_NEXT? , **PL_NEXT?** : Indicates the name of the next file to be played on player #1
PL2_NEXT? : Indicates the name of the next file to be played on player #2

🗎 Mixing capabilities and crossfades inside Playlists

Mixing capability (for Jingle and Crossfade functions) are available when **MIXER=ON.**

When these conditions are met, the PLAYERS numbers are not anymore relevant. The number attribution is performed automatically by the internal application. Then no explicit reference to PLAYER1 or PLAYER2 has to be made using PLAY, OVERPLAY and PLAYLIST commands. Just avoid to specify the player number inside these commands.

Jingle Insertion capabilities

A Jingle is generally much shorter than a regular song file. Then we advice to use the **#CMD_AFTER_START:** directive inside a linear .M3U playlist to perform an **OVERPLAY[<n>]=…** command corresponding to the Jingle.
For more details about the OVERPLAY command, see before in the chapters concerning the PLAYERs operation.

🗎 Playlist events

All events are described at the end of this document.

🗎 .M3U Playlist format supported by the HITPLAYER-L

The .M3U format supported by the HITPLAYER is extremely powerful and is specified in the annex of this document.

NOTE: Some examples of playlist are available on our website: http://www.aztec-radiomedia.com/

🗎 PLS Playlist format supported by the HITPLAYER-L

HITPLAYER-L supports the regular .PLS playlist format.

🖹 File and playlist synchronization with an FTP server on the Internet

The following example will explain practically how to tell the Hitplayer to synchronize itself every night with an FTP server located on your LAN or the Internet, and then reload the playlist at 4:58 AM just before 5. This example needs the scheduler functionalities to work, Scheduler functions are described in the IP2 reference manual for developers, réf. azt2502[v].PDF where [v] represents the version.): available on the Web site : http://www.aztec-radiomedia.com/

Step 1 : Configure the scheduler by adding the following lines or creating the schedule.txt file (table below).

Step 2 : Change the time trigger according your own preferences

Step 3 : Transfer the schedule.txt file in the / directory of your Hitplayer (use an FTP client to do that)

```
‘Playlist global parameters
#SCH_CONSOLE=1
#SCH_ENABLED=1
#SCH_SCAN_PERIOD=20
#SCH_LOG=0
‘------------------------------------------------------------------------------------
--------
‘Trig every day at 04:05 AM and ask the server to synchronize with an FTP server on
the web
#TRIG * * * * 4 5 * * *
FTP_SYNCHRONIZE=ftp.aztec.fr,anonymous,hello@ip2.fr,/ata,/images/mp3,RETRIEVE,KEEP
‘------------------------------------------------------------------------------------
--------
‘Trig every day at 04:58 AM and ask the Hitplayer to reload the playlist
#TRIG * * * * 4 58 * * *
pl=s.m3u
```

Commands and events memo

Useful commands to start quickly…

More commands and all commands available in the IP2SYSTEM reference manual and the NETCOM reference manual, especially for developers.

The command HELP displays the following list

```
*** HITPLAYER-L : HELP COMMANDS ***

HELP.APPLI                    : Application specific help commands
HELP.BASIC                    : BASIC Interpreter commands help
HELP.DNS                      : DNS client commands help
HELP.EVENTS                   : Events commands help
HELP.FILE                     : File system commands help
HELP.FTP               : FTP server commands help
HELP.FTP_CLIENT               : FTP client commands help
HELP.HTTP_CLIENT              : HTTP client commands help
HELP.HISTO                    : Log file commands help
HELP.MAIL              : E-mail client commands help
HELP.MULTICAST                : Multicast group commands help
HELP.NETCOM                   : NETCOM help commands
HELP.NETWORK                  : Network commands help
HELP.PPP               : PPP commands help
HELP.SCHEDULER                : SCHEDULER rules and commands help
HELP.SNMP                     : SNMP agent commands help
HELP.SYSTEM                   : System commands help
HELP.TIMERS                   : Timers commands help
HELP.TELNET_CLIENT            : Telnet client commands help
HELP.UDP                      : UDP client/server commands help
HELP.USERS                    : Login and password table commands help
HELP.WEB                      : Web server commands help
HELP.SNTP                     : SNTP commands help
```

The command HELP.APPLI displays the following list

```
*** HITPLAYER-L : OTHER HELP COMMANDS ***

HELP.AUDIO_INPUT              : Display Audio Input help commands
HELP.AUDIO_OUTPUT             : Display Audio Output help commands
HELP.AUDIO_SELECTOR           : Display Audio Selector help commands
HELP.M3U                      : Display .M3U #directives
HELP.PLAYER                   : Display PLAYER help commands
HELP.PLAYLIST                 : Display PLAYLIST help commands
HELP.SWA                      : Display SoundWan Agent help commands
HELP.SWS                      : Display SoundWan Server help commands
HELP.DIG               : DIG commands help
```

Commands usually used:

MY_NAME[=<name of the HITPLAYER>] [?]	Decoder type
IP[=<x.x.x.x>] [?]	IP address configuration
MASK[=<x.x.x.x>] [?]	IP MASK subnet configuration
GATEWAY[=<x.x.x.x>] [?]	Gateway's IP address (IP address of the router)

MAC?	Displays the Ethernet MAC address
SN?	Displays the serial number
MTU[=<mtu value>] [?]	Defines the MTU value

Commands related to the audio input “LINE_IN”

The command HELP.AUDIO_INPUT displays the following list

```
*** HITPLAYER-L : HITPLAYER-L AUDIO INPUT COMMANDS HELP ***

[*]?LINE_IN                          : Display LINE_IN configuration and status
LINE_IN.GAIN=OFF|ON                  : Turn OFF|ON analog audio input
LINE_IN.GAIN=<-110..+18>             : Adjust LINE IN audio level (dB)
LINE_IN.GAIN?                        : Display LINE IN audio level (dB)
LINE_IN.FADE=<s>,<g>                 : Fade LINE IN In/Out and store final gain
                                              s = step in dB/sec
                                              g = final gain (-110dB..+18dB)
LINE_IN.CODER=<coder Type>,
          <bitrate>,
          <stereo/mono>,
          <sample_freq>              : Set MPEG coder configuration
LINE_IN.CODER=1                      : LINE_IN.CODER=MP2,192,STEREO,48
LINE_IN.CODER=2                      : LINE_IN.CODER=MP2,192,MONO,48
LINE_IN.CODER=3                      : LINE_IN.CODER=MP2,128,STEREO,44.1
LINE_IN.CODER=4                      : LINE_IN.CODER=MP2,128,MONO,44.1
LINE_IN.CODER=5                      : LINE_IN.CODER=MP2,112,MONO,32
LINE_IN.CODER=6                      : LINE_IN.CODER=MP2,96,MONO,32
LINE_IN.CODER?                       : Display MPEG coder configuration
LINE_IN.AGC=ON|OFF                   : Enable/Disable AGC
LINE_IN.AGC?                         : Display AGC state
LINE_IN.L_AGC=<-18..0>               : Set AGC level (in dB)
LINE_IN.L_AGC?                       : Display the AGC level
LINE_IN.AGC_GATE=<-50..-10>          : Set AGC gate (in dB)
LINE_IN.AGC_GATE?                    : Display AGC Gate
LINE_IN.TL_DETECT=<-50..-10>|OFF     : Set the low level threshold (in dB)
LINE_IN.TL_DETECT?                   : Display the low level threshold
LINE_IN.TO_DETECT=<0..240>           : Set time-out (sec.) before LINE_IN_LOST event
LINE_IN.TO_DETECT?                   : Display time-out before LINE_IN_LOST event
LINE_IN.TO_RECOVER=<0..240>          : Set time-out (sec.) before LINE_IN_RECOVER event
LINE_IN.TO_RECOVER?                  : Display time-out before LINE_IN_RECOVER event
```

Commands related to audio outputs “LINE_OUT1” and “LINE_OUT2”

The command HELP.AUDIO_OUTPUT displays the following list

```
*** HITPLAYER-L : HITPLAYER-L AUDIO OUTPUT COMMANDS HELP ***

[*]?MASTER_OUT                       : Display MASTER_OUT configuration and status
MASTER_OUT<n>=OFF|ON                 : Turn OFF|ON analog audio output #n
MASTER_OUT<n>_STEREO=OFF|ON          : Turn OFF|ON stereo mode of audio output #n
```

MASTER_OUT<n>=<-110..+18> : Adjust audio level of LINE OUT #n (dB)
MASTER_OUT_FADE<n>=<s>,<g> : Fade In/Out LINE OUT #n
s = step in dB/sec
g = final gain (-110dB..+18dB)
MASTER_OUT<n>_PAN= : Adjust the balance of audio output. Negative
value means left, positive value means right (dB)
PLAYER_OUT<n>=OFF|ON : Turn OFF|ON the Player #n
PLAYER_OUT<n>=<-110..+18> : Adjust MP3/MP2 Player #n audio level (dB)
MIXER=ON|OFF : Enable/Disable the mixing of 2 audio output

🖹 Events related to LINE_IN

LINE_IN_OK	Signal is present on LINE_IN
LINE_IN_LOST	Silence detected on LINE_IN

🖹 Commands related to the Audio selectors

The command HELP.AUDIO_SELECTOR displays the following list

```
*** HITPLAYER-L : HITPLAYER-L AUDIO SELECTOR COMMANDS HELP ***

[*]?SEL                          : Display audio selector configuration
SEL1=SWA                         : Set SoundWan Agent to feed LINE_OUT1
SEL1=PLAYER1                     : Set Player 1 to feed LINE_OUT1
SEL1=LINE_IN                     : Set LINE_IN to feed LINE_OUT1
SEL1?                            : Display the current audio source selected on selector 1
SEL2=PLAYER2                     : Set Player 2 to feed LINE_OUT2
SEL2=LINE_IN                     : Set LINE_IN to feed LINE_OUT2
SEL2?                            : Display the current audio source selected on selector 2
```

🖹 Playlist Directives

The command HELP.M3U displays the following list

```
*** HITPLAYER-L : M3U PLAYLIST FILE FORMAT ***

[#<directive>[:param>]]                : Playlist directive line
:<IP2_COMMAND>                         : IP2 command to be executed
<file.mp3>                       : MP3 file to be played
<file.mp2>                       : MP2 file to be played

Global playlist directives (default value is in {}):
#ID3_OUT:<1|{0}>                       : Enable/disable MP3 Tag display
#PL_CONSOLE:<1|{0}>                    : Enable/disable playlist debug messages
#PL_LOG_FILE:<1|{0}>                   : Enable/disable playlist log messages
#PL_DURATION_MAX:<{0}..65535>          : Max Playlist duration in sec(0=forever)
#PL_MAX_PLAYS:<{0}..65535>             : No of times Playlist executed(0=infinite)
#PL_FADE_START:<0..20>                 : Set fade out start time (sec)
                                       : The default value is defined by the
                                       : PL_FADE_START=<0..20> IP2 command
                                       : and can be displayed by the
                                       : PL_FADE_START? command.
#PL_FADE_SLOPE:<0..100>                : Set fade out slope (dB/sec, 0=Off)
                                       : The default value is defined by the
                                       : PL_FADE_SLOPE=<0..100> IP2 command
                                       : and can be displayed by the
                                       : PL_FADE_SLOPE? command.
#PL_EXTINF_DEC:<0..255>                : Set the playlist PL_EXTINF_DEC value (sec).
                                       : The default value is defined by the
                                       : PL_EXTINF_DEC=<0..255> IP2 command
                                       : and can be displayed by the
                                       : PL_EXTINF_DEC? command.

Playlist item directives (IP2 commands, MP2 or MP3 files)
```

```
  => must be placed before the playlist item

#EXTINF:<{0}..65535>,<title>                    : Max time to play, title=text descriptor
#CMD_AFTER_START:<{0}..65535>;<cmd>             : Execute IP2 command n secs after start
#INTERVAL_MIN:<[0]..65535>                      : Min interval (sec) between plays of item
#FADE_START:<{0}..20>                   : Set fade out start time (sec) for this item
#FADE_SLOPE:<{0}..100>          : Set fade out slope (dB/sec) for this item

#RANGE:<YY> <MM> <DD> <WD> <HH> <MN> <SS> : Scheduling
     YY : Years (00 for 2000)  e.g.: 0-8 ---> from 2000 to 2008
     MM : Months (1 to 12)     e.g.: *    --> for any month
     DD : Dates (1 to 31) e.g.: 1-4,6 -> day 1 to 4 and 6
     WD : Weekdays (1=Monday to 7=Sunday)
     HH : Hours (0 to 23)
     MN : Minutes (0 to 59)
     SS : Seconds (0 to 59)
```

Commands related to the PLAYERS

The command HELP.PLAYER displays the following list

```
*** HITPLAYER-L : PLAYER COMMANDS HELP ***

PLAY[n]=<file>                  : Play an audio file (MP3, MP2, M3U, PLS)
STOP[n]                         : Stop current audio file
POS[n]?                         : Display current position of played audio file
PAUSE[n]                : Pause current audio file
CONT[n]                         : Continue current audio file
OVERPLAY[n]=<file>              : Play <file> over the current play (no mix)
OVERPAUSE[n]=<file>             : Pause the current play and start <file>
OVERSTOP[n]                     : Stop current overplay audio file

INSERTPLAY=<file>[,level(-100..{0})[,slope_down(1..100)[,slope_up(1..100)]]]

                                : Mix <file> with current play (only available if MIXER=ON)
                                        file      = MP2/MP3 file to mixe with current play
                                        level      = Audio level offset (dB)
                                        slope_down = fade out slope (dB/sec)
                                        slope_up   = fade in slope (dB/sec)

INSERTSTOP                      : Stop file mixing
GET_ID3=<filename>              : Extract and display ID3 Tag
(n = player number, if not specified, means player #1)
(for HITPLAYER-B : n is always 1)
```

Commands related to the PLAYLIST

The command HELP.PLAYLIST displays the following list

```
*** HITPLAYER-L : PLAYLIST COMMANDS HELP ***

[*]?PLAYLIST                    : Display Playlist configuration and status
PLn=<file>                      : Load and start a playlist
PLn_CONT                        : Continue playlist
PLn_CUR?                        : Display current item playing
PLn_DISP?                       : Displays all active playlist content
PLn_INFOS?                      : Displays all active playlist parameters
```

```
PLn_ITEMS?                        : Displays all items of the active playlist
PLn_NEXT?                         : Information about the next item to play
PLn_PAUSE                         : Pause playlist
PLn_POS?                          : Current position inside the playlist
PLn_SKIP                          : Skip the playlist to next item
PLn_STOP                          : Stop playlist
PLn_STATUS?                       : Display playlist file
PLn[_NEXT]_ALBUM?                 : Displays [next] ID3 album
PLn[_NEXT]_ARTIST?                : Displays [next] ID3 artist
PLn[_NEXT]_COMMENT?               : Displays [next] ID3 comment
PLn[_NEXT]_SONG?                  : Displays [next] ID3 song
PLn[_NEXT]_YEAR?          : Displays [next] ID3 year
PL_EXTINF_DEC=<{0}..255>          : Set the #PL_EXTINF_DEC default value.
PL_EXTINF_DEC?                    : Displays the #PL_EXTINF_DEC default value.
PL_FADE_START=<{0}..20>           : Set the fade out start time default value (sec).
PL_FADE_START?            : Displays the fade out start time default value.
PL_FADE_SLOPE=<{0}..100>          : Set the fade out slope default value (dB/sec, 0=Off).
PL_FADE_SLOPE?                    : Displays the fade out slope default value.
(n = playlist number, if not specified, means playlist #1)
(for HITPLAYER-B : n is always 1)
(default value is in {})
```

Events related to Playlists

PL1_END	Appears each time a playlist ends.
PL2_END	
PL1_ITEM_PLAY	Appears each time there is a new audio source (MP2 or MP3 file) which is started.
PL2_ITEM_PLAY	
PL1_ITEM_END	Appears each time there is an source (MP2, MP3) or IP2COMMAND which ends.
PL2_ITEM_END	

Commands related to the SoundWan agent “SWA1”

The command HELP.SWA displays the following list

```
*** HITPLAYER-L : SOUNDWAN AGENT HELP COMMANDS ***


HELP.SWA                                  : Display this help
[*]?SWA                                   : Display SWA1 status and configuration
SWA.ENABLED=<ON>|<OFF>[,Multicast Addr.]  : Enable/Disable SoundWan Agent
SWA.ENABLED?                              : Display SoundWan Agent current state
SWA.PORT=<port number>                    : Defines the port number. Must be over 6000
SWA.PORT?                         : Display the port number
SWA.LST(<n>)=<ch>[,gain]                  : Configure the Index <n> of the channel list
                                      with gain in dB between -30 and +30
SWA.LST(<n>)?                             : Display the Index <n> of the channel list
SWA.LST?                                  : Display all channel list
SWA.CH_CUR?                               : Display current decoded channel
SWA.BUF_SIZE=<value>                      : Set SoundWan Agent buffer size (byte)
SWA.BUF_SIZE?                             : Display SoundWan Agent buffer size
```

Events related to the SoundWan Agent “SWA1”

SWA1_ON	Agent turned on
SWA1_OFF	Agent turned off
SWA1_CHANGE	Channel detected by the agent has changed

Commands related to the SoundWan SERVER “SWS”

The command HELP.SWS displays the following list

```
*** HITPLAYER-L : SOUNDWAN SERVER HELP COMMANDS ***


HELP.SWS                             : Display this help
?SWS                                 : Display SoundWan Server Status and configuration
SWS=<ON,ch>|<OFF>                    : Set SoundWan Server on channel <ch>
SWS?                                 : Display SounWan Server settings
SWS.IP=<ip address>:<port>           : Set destination address of the SoundWan channel
SWS.IP?                              : Display destination address of the SoundWan channel
SWS.STATUS?                          : Display SounWan Server status
SWS.IP_PRECEDENCE=<0..7>             : Set Precedence for SWS IP datagram
SWS.IP_PRECEDENCE?                   : Display Precedence for SWS IP datagram
SWS.IP_DELAY=<0..1>                  : Set Delay for SWS IP datagram
SWS.IP_DELAY?                  : Display Delay for SWS IP datagram
SWS.IP_THROUGHPUT=<0..1>             : Set Throughput for SWS IP datagram
SWS.IP_THROUGHPUT?                   : Display Throughput for SWS IP datagram
SWS.IP_RELIABILITY=<0..1>            : Set Reliability for SWS IP datagram
SWS.IP_RELIABILITY?                  : Display Reliability for SWS IP datagram
```

Events related to the SoundWan Server “SWS”

SWS_ON	SWS server is turned ON
SWS_OFF	SWS server is turned OFF

ANNEXES

M3U extended format proposed by Aztec Radiomedia

This extended format is proposed by Digigram / Aztec Radiomedia, for any remark : mailto:support@aztec-radiomedia.com .

M3U regular playlist format

The basic M3U format always begins with a first line that indicates the format of the Playlist, then the remaining part of the Playlist indicates which files must be played. Two lines are used for each file. The first one always begins with **#EXTINF:** , followed by the **maximum duration** of the file (or item) in seconds and then a full stop followed by the comment associated to this text. The next line indicates the name of the audio source (file, url, etc…) to be played.

When no duration is specified (implicit duration), the item plays until its natural end occurs.

```
#EXTM3U
#EXTINF:347,Love of my life - Santana - Supernatural
Love of my life.mp3
#EXTINF:157,Let The Children Play - Santana - The Very Best Of
LetTheChildren.mp3
#EXTINF:332,No One To Depend On - Santana - The Very Best Of
NoOneToDependOn.mp3
#EXTINF:433,One Chain - Santana - The Very Best Of
OneChain.mp3
#EXTINF:257,Oye Como Va - Santana - The Very Best Of
OyeComoVa.mp3
#EXTINF:273,Samba Pa Ti - Santana - The Very Best Of
SambaPaTi.mp3
#EXTINF:243,She's Not There - Santana - The Very Best Of
ShesNotThere.mp3
#EXTINF:298,Smooth - Santana - Supernatural
```

M3U extended format for mixing, jingles and advanced playlists

As you can observe, each of file specified in the .M3U playlist can or is preceded by a line which starts with the character "#" followed by EXTINF or EXTM3U. These kind of lines are called **"#DIRECTIVES"**, in this document.

HITPLAYERs from Aztec Radiomedia (Digigram Group) handles a complete set of directives that can follow the "#" sign on a line.

Local and Global directives

- ***Type Local #DIRECTIVES*** which concerns only a dedicated item to play. These directives must always be placed in lines just BEFORE the file name (or url) to of the item to be played (or the command to be sent to the command interpreter of the Hitplayer).

  Important : the first line that follows #directives lines is considered to be the file name or url of the item to be played.

- ***Type Global #DIRECTIVES*** which generally concerns the playlist in itself or a default properties for all item properties.

  Important : the order in which these global directives are placed is not relevant. They can be placed anywhere in the playlist file.

```
#EXTM3U
#<DIRECTIVE1 related to item A:parameters>
#<DIRECTIVE2 related to item A:parameters>
#<DIRECTIVE… related to item A:parameters>
#<DIRECTIVEn related to item A:parameters>
item A : <file name> or <url> or <:IP2COMMAND>
#<DIRECTIVE1 related to item B:parameters>
#<DIRECTIVE2 related to item B:parameters>
#<DIRECTIVE… related to item B:parameters>
#<DIRECTIVEn related to item B:parameters>
item B : <file name> or <url> or <:IP2COMMAND>
```

Comments inside lines

Any character that appears after ‘ in a line is considered as a comment and ignored.

Item to be played in a playlist	**<item path>** Any line starting with a character which is not : ‘ : space #

IP2 command inside a PLAYLIST	**:<full command name>** Any playlist line beginning with ‘:’ is considered as a command to be addressed to the internal command interpreter of the HITPLAYERs. Any command can be sent to the interpreter. A command is then considered as an ITEM to play. **Example** : with this feature you can specify the HITPLAYER to output the sound applied to its analog input (tuner, CD, live show) during a certain time or until a certain time. **See also** : **#CMD_AFTER_START**

Playlist loopback and maximum playlists loops	**#PL_MAX_PLAYS:<n>** **Type : Global** Add the directive where n represents the maximum number of times a playlist must loop and play before it stops. By default, this value is set to infinite (n=0)

🗎 **Setting the maximum duration of a playlist**	**#PL_DURATION_MAX:<d>** **Type : Global** D is expressed in seconds. By default, there is no time limitation to a playlist.

🗎 **Debugging the PLAYLIST process**	**#PL_CONSOLE:<n>** **Type : Global** n=1 : console enabled, n=0 ; console disabled Numerous information come out from the console port (COM0) for debug. You can use also the command **PL<v>_CONSOLE=0|1** to perform this action manually.

🗎 **Get a log of files and events that occurred during the play of a playlist**	**#PL_LOG_FILE :<V>** **Type : Global** V=1 : enabled , V=0 : disabled No events written in the log file if this directive does not appear somewhere in the playlist. **Log files** : **/histo.txt** or its html image **/histo.html**

🗎 **Outputs ID3 tag information on a console port (COM0)**	**#ID3_OUT:<v>** **Type : Global** v=1 : enabled v=0 : disabled

🗎 **Defines the fadeout time before the next playlist item starts**	**#PL_FADE_START:<v>** **Type : Global** **Important :** this directive apply only when the hitplayer enables mixing, otherwise its default value is set to 0. **(MIXER=ON)** V is expressed in seconds and is computed taking the duration specified with #EXT_INF local parameter. If no #EXT_INF value is defined, no fadeout can be performed See also **#FADE_START:<v>**

🗎 **Fadeout slope for crossfades**	**#PL_FADE_SLOPE:<v>** **Type : Global** This directive provides the default value for the FADE_SLOPE parameter. See also **#FADE_SLOPE:<v>**

🗎 **Conditional plays upon current time / date**	**#RANGE:<YY> <MONTH> <DAYM> <DAYW> <HOURS> <MIN> [<SEC>]** **Type : Local** This playlist directive indicates range(s) in the corresponding playlist item is authorized for play. Space is considered as the separator between items. Do not place any space between ':' and the beginning of the condition string. **Example** : **#RANGE:2000,2001 1-12 1-8,20-31 1-4 8-13 0-30 *** Playlist item authorized in year 2000 and 2001, any month in the year, only from days 1 to 8 and 20 to 31 of each month, from Monday to Wednesday, between 8 AM and 1PM only.

🗎 **Setting the maximum numbers of plays for each individual items (files) in the the PLAYLIST**	**#MAX_PLAYS:<MAXIMUM NUMBER OF PLAYS>** **Type : Local** The number of PLAYS counter of each individual items (eg music files) in a playlist is reset (to 0) each time the playlist is started. Once it reaches the value given in the MAX_PLAYS directive, the corresponding item in the PLAYLIST will be ignored when the playlist is processed. **Example** : **#MAX_PLAYS=1** hello.mp3 The item hello.mp3 will be played only one time, even when the playlist loops on itself.

🗎 **command insertion during a play especially for jingles**	**#CMD_AFTER_START: <seconds after starts>;<IP2 COMMAND>** **Type : Local** This very powerful directive enable to stack regular HITPLAYER-L interpreter commands (IP2 commands) during the play of a dedicated audio source (file, url,…). The command given inside the directive is put on a command buffer that is analysed permanently during the play of the dedicated file. As soon as the specified is reached (when calculation can be performed), the associated command is executed. A Jingle insertion can be programmed inside the playlist simply by declaring the appropriate OVERPLAY command.

🗎 **Setting the minimum duration between 2 plays of an individual file in a playlist**	**#INTERVAL_MIN:<D >** **Type : Local** D is expressed in seconds. By default, there is no minimum interval between 2 plays of one dedicated item.

🗎 **Specific title and maximum duration with the #EXTINF directive**	**#EXTINF: [<duration in seconds>][,<item title>]** **Type : Local** **This directive contains 2 parameters : Duration max in seconds, and Title** **Duration max** Duration to wait before to go to the next item in playlist. **Title** The Title of the current object played by the PLAYER #n can be read out using the commands **PL[<n>].TITLE?** and **PL[<n>]_NEXT.TITLE?**

🗎 **Defines the fadeout time before the next playlist item starts**	**#FADE_START:<v>** **Type : Local** **Important :** this directive apply only when the hitplayer enables mixing, otherwise its default value is set to 0. **(MIXER=ON)** V is expressed in seconds and is computed taking the duration specified with #EXT_INF. See also **#PL_FADE_START:<v>**

🗎 **Fadeout slope for crossfades**	**#FADE_SLOPE:<v>** **Type : Local** This directive provides the default value for the FADE_SLOPE parameter. See also **#PL_FADE_SLOPE:<v>**

Playlist Editors

****

Winamp player : www.winamp.com (PLS and M3U formats)

Tip: When you generate a playlist file (*.m3u) with Winamp and if you want WINAMP to specify the EXTINF for each item, **select** in the menu:

--> OPTIONS

--> PEFERENCES

--> OPTIONS

the ***"Read titles on load"*** option.

****

Freeamp player : www.freeamp.org (PLS)

****

PowerGold editor : www.powergold.com (M3U)

****

To use the COM0 and COM1 ports in NETCOM mode

The operation of the communication ports in NETCOM mode is described in the "NETCOM IP2 functionalities" manual ref. ECI25032.PDF. This manual is available in PDF format on AZTEC RADIOMEDIA's Web site.

The commands **COM0?** and **COM1?** display the configuration of the HITPLAYER's communication ports and inevitably the configuration of the NETCOM service that may be assigned to these ports.

HITPLAYER-L SNMP MIB

Under development.

Installing an IDE Hard Disk inside the HITPLAYER

Before connecting the Hard Disk

Disconnect the power supply cable of the Hitplayer- L before opening it.

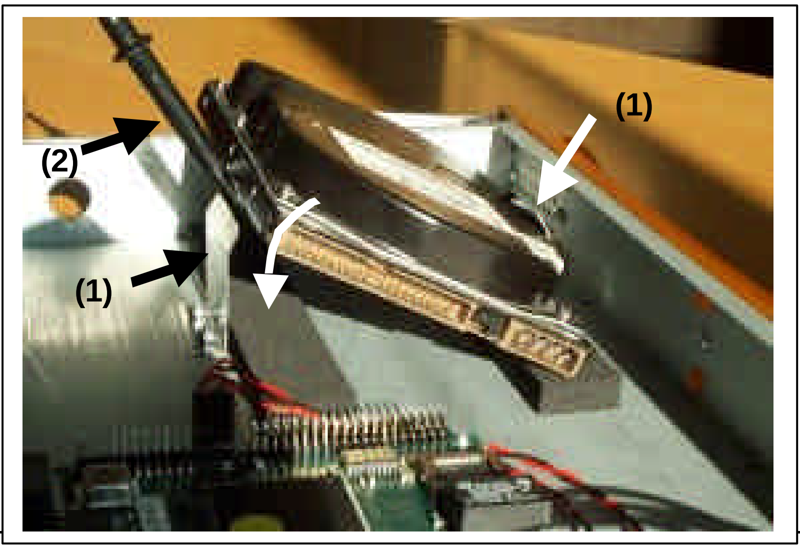
Before installing the hard disk be sure it contains a FAT32 active partition. If not, create a new FAT32 partition by using your computer's operating system utility "fdisk".

Installing the Hard Disk

There are two models of hard disk support. The first model was a clip system as show in section A. The second model is a screwed platinum system as shown in section B.

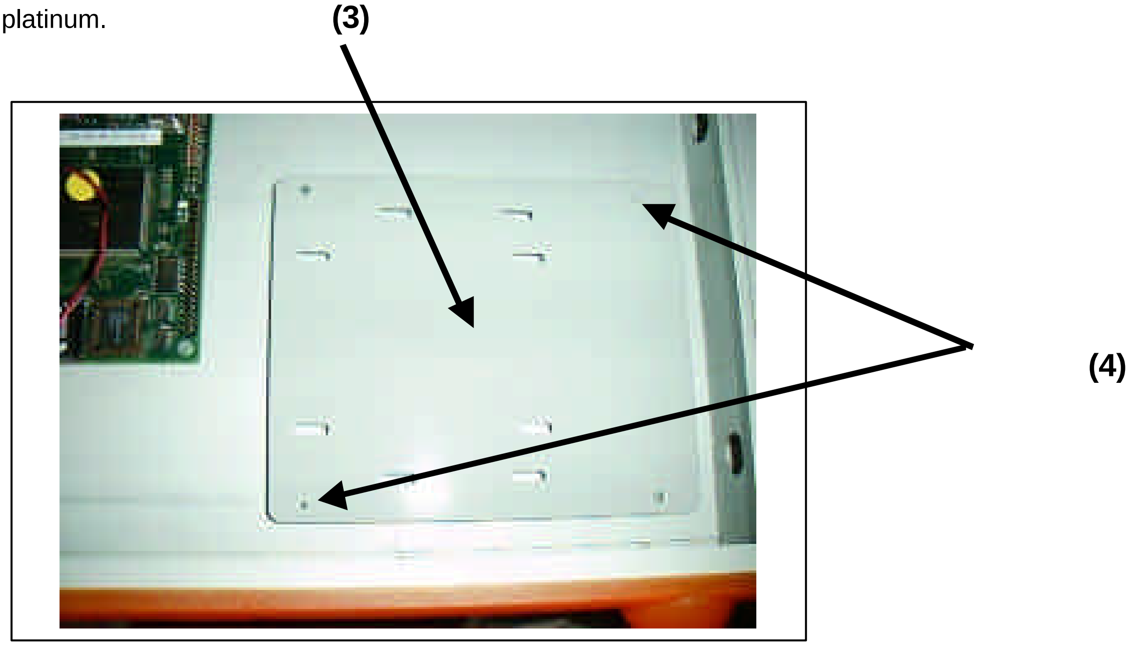
SECTION A

To install the hard disk insert it between the two clips(1) using a screwdriver(2) for example.

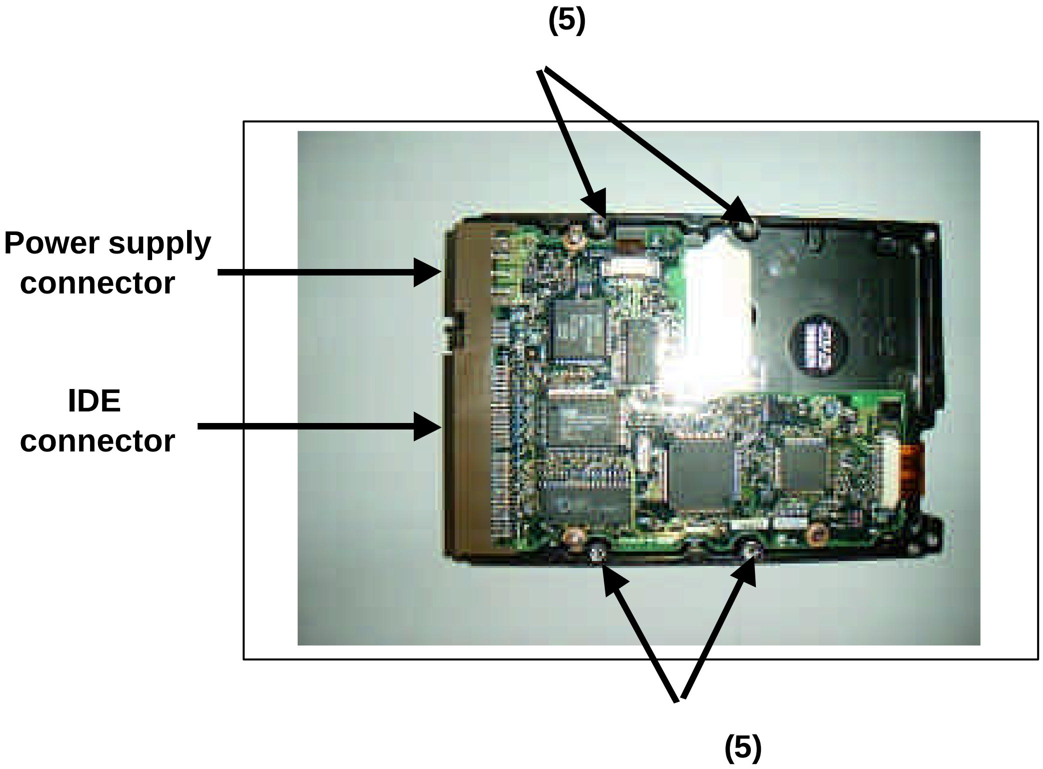


SECTION B

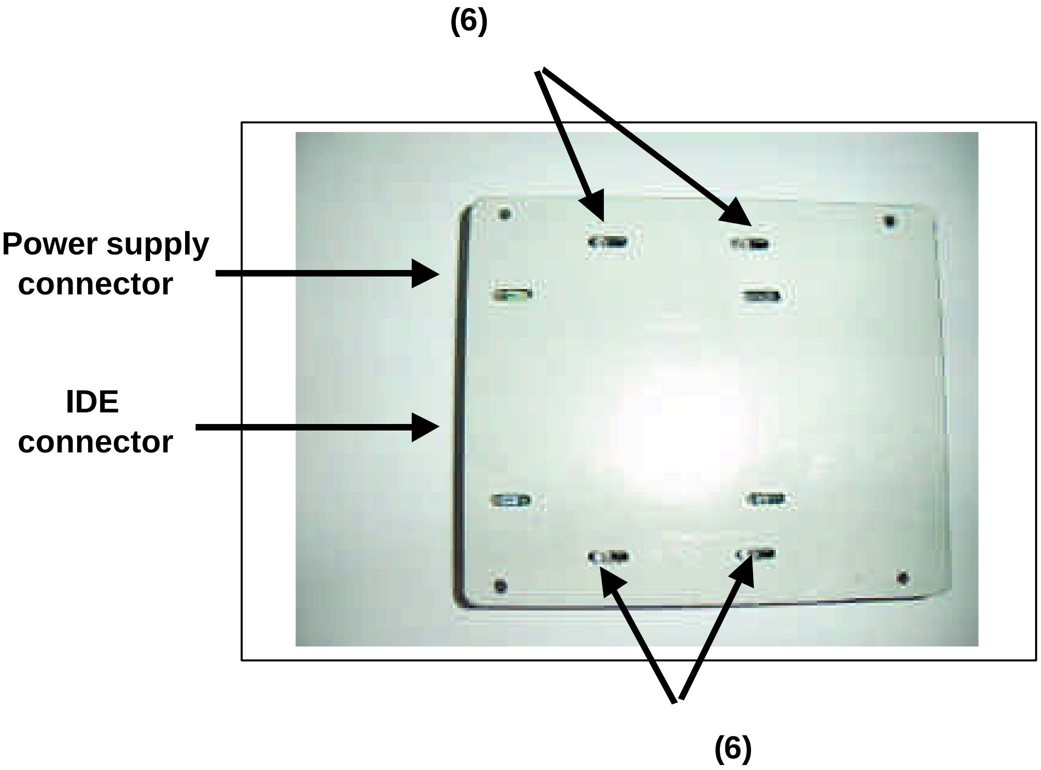
To install the hard disk remove the platinum (3) from the Rack. For this you have to remove the two screws (4) and unclip the platinum.



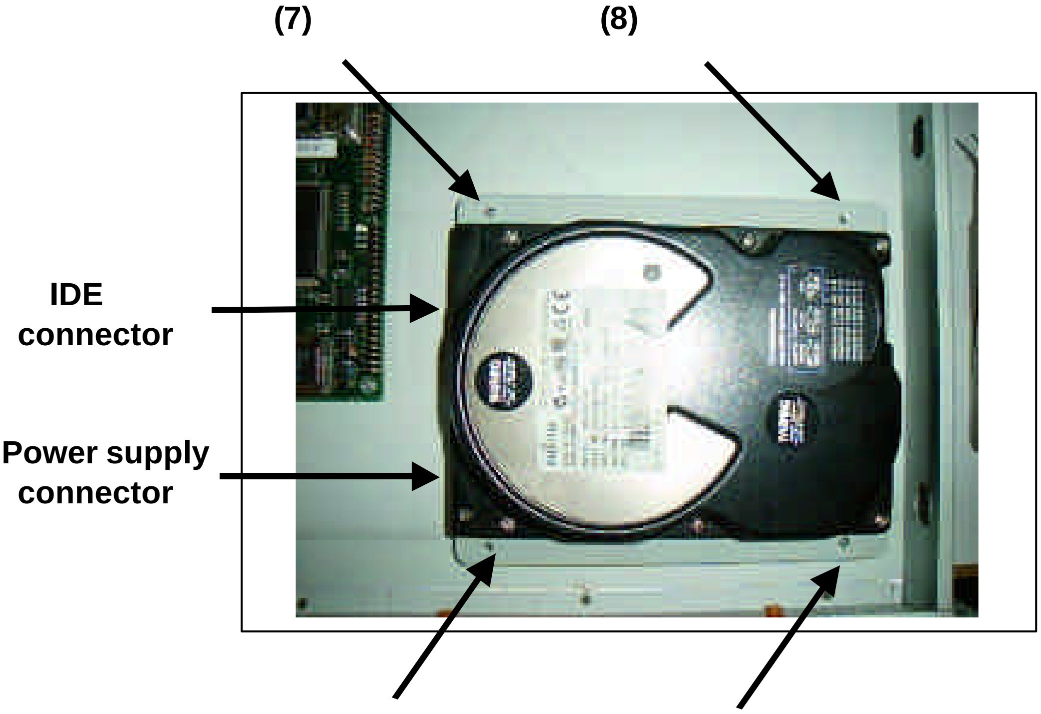
Then place your hard drive with the bottom side towards you, you should see the holes (5) for the screws:



Then place the platinum on your hard drive and place the four fixing screws (6):



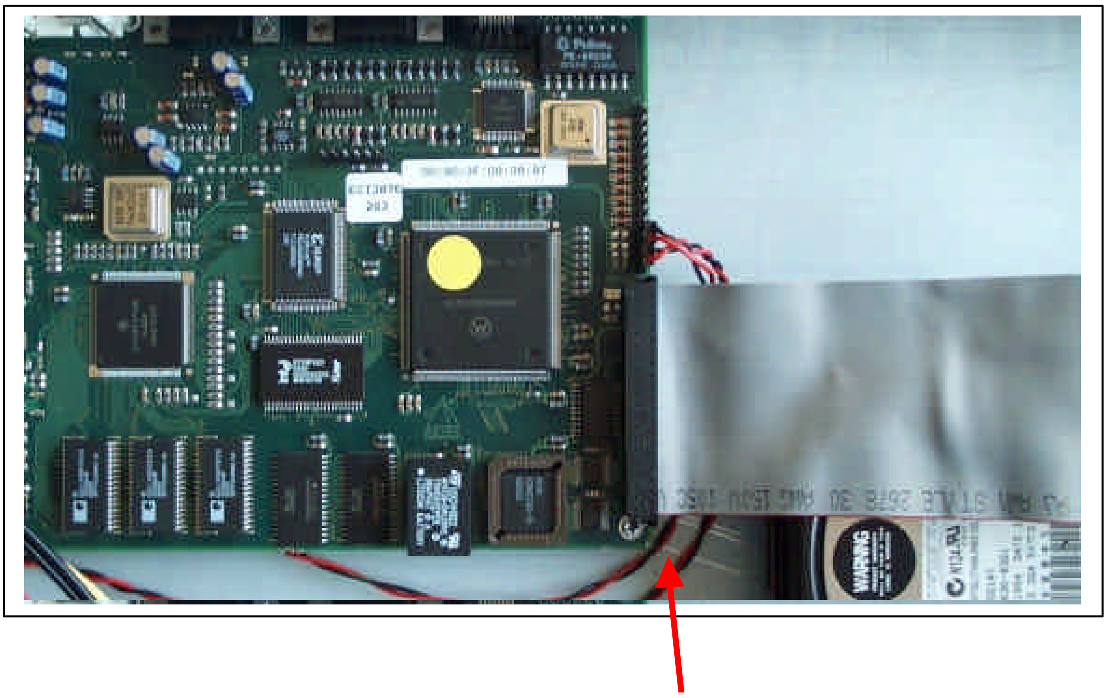
Clip the platinum and hard-drive set in the rack by using the two clips (7), then fix the set by using two screws (8):



(8) **(7)**

🖹 Connecting the IDE cable to the board

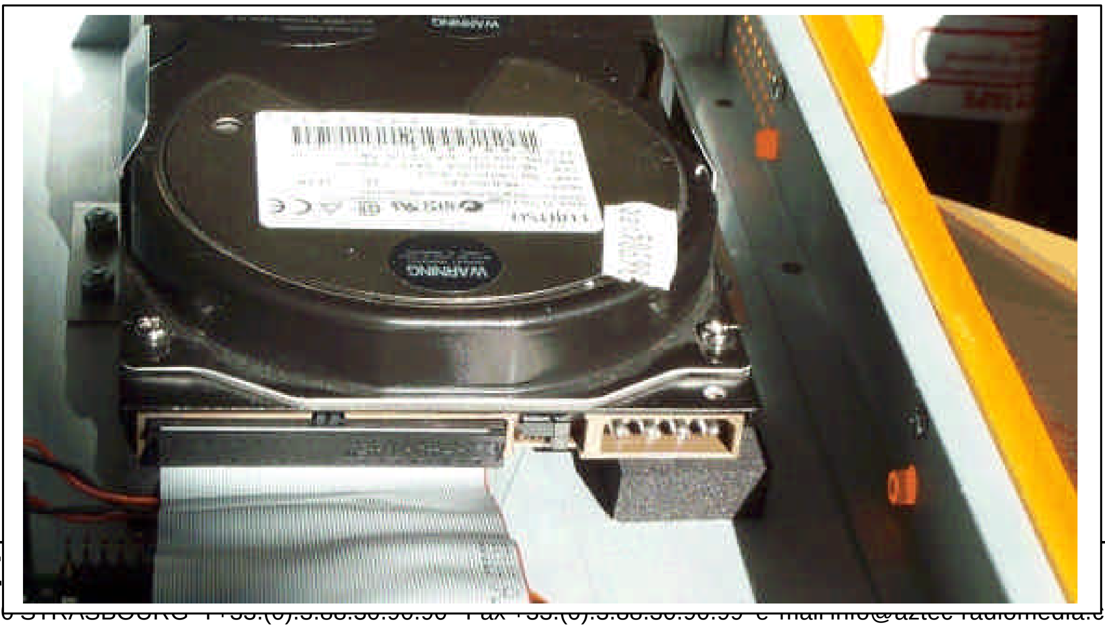
Then you'll have to connect the IDE cable between the board and the hard disk. The pin 1 of the IDE connector (J4) is marked with a "1" on the board. The pin 1 of the IDE cable is generally at the side coloured red and must be plugged on the pin one of the board connector.



pin 1 of the IDE cable

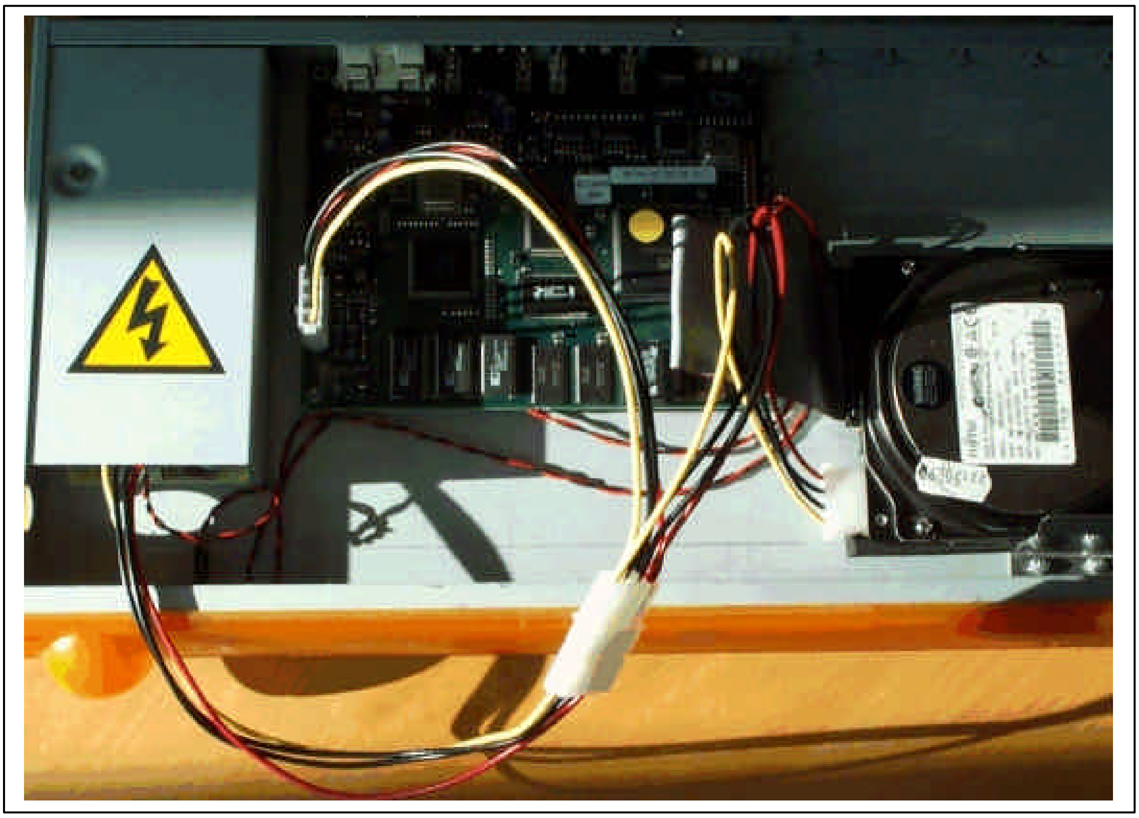
🖹 Connecting the IDE cable to the hard disk

After that, connect the IDE cable to the hard disk.



🗎 Connecting the hard disk power supply cable

Disconnect the power supply plug of the SPX board. Connect the Hitplayer power supply plug with the power supply extension cable. Then connect one of the two female plugs into the SPX power supply socket and the other plug into the hard disk power supply connector.



Close the Rack with its Cover, make sure the hard disk jumpers are in "master" position. After the power up the Hitplayer-L will automatically recognize the hard disk. If the hard disk isn't formatted used the followed command: **FORMAT ATA**.

The Hitplayer-L is now ready to store all kind of files you want.

IMPORTANT: In order to optimise the processing time it's recommended to store the audio files in several subdirectories. If you store too much files in the same directory the Hitplayer will take much more time to find, open and play the audio files. We recommend up to 500 files/folders by directory.

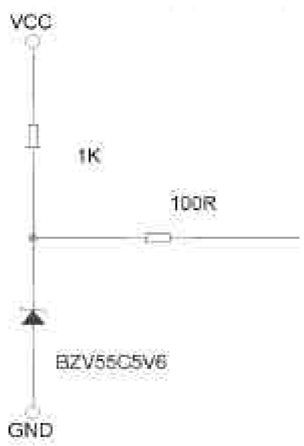
🗎 To use the logic inputs / outputs "DIGn" of the HITPLAYER-L board

8 logical TTL inputs / outputs are reachable on the HITPLAYER-L mother board.

The DIG1 to DIG8 are both inputs and outputs : TTL.

Very important : ON state is performed when the inputs presents 0 volts measured between the input and the ground. When an input is tight to ground, it gets the ON state. OFF state is performed when a) the inputs are left open **and** b) set to OFF.

The equivalent circuit of the 8 inputs DIG1 to DIG8 is given below :



🗎 To view all the info related to the digital input/output

The command **?DIG<n>** displays the list of parameters and states related to the digitals I/O.

🗎 To define a name to the digital input/output DIG<n>

The command **DIG<n>.NAME=<name>** attributes a name to the digital I/O DIG in plaintext. This allows this DIG to be "labelled" so that it can be remotely identified, without any doubt. Reread with **DIG<n>.NAME?**

🗎 To nominate the "ON" and "OFF" state of the logic I/O DIG<n> in plaintext

It is possible to attribute a name for each state of the DIG I/O:

DIG<n>. STATUS_ON=<ON state description of the DIG>

DIG<n>. STATUS_OFF=<OFF state description of the DIG>

(text descriptor in 16 characters max.)

By attributing a name to each state of DIG1, it is possible to display the actual state of the connected equipment in plaintext and in user language: the command **DIG<n>.STATUS?** is provided for this purpose: this command is very interesting when it is overlaid in Web pages served by the HITPLAYER-B to provide the user with a clear meaning of the state of this DIG I/O.

Example: if **DIG1.STATUS_ON=ALARM TEMPERATURE**, the command **DIG1.STATUS?** returns "**ALARM TEMPERATURE**" if the input DIG1 is in the ON state.

To monitor the input DIG<n> with associated events

When monitoring is enabled, the following IP2 events can be generated (for more details concerning the notion of events associated to IP2, read the corresponding chapter in the IP2 system documentation ref. ECI2502):

DIG<n>_ON : the input has just been enabled. To make things clear, the value programmed in DIG<n>.STATUS_ON is reminded in the descriptor field that accompanies the event in the history file, and consequently in the associated emails or SNMP traps.

DIG<n>_OFF : the same applies for deactivation, with the descriptor DIG<n>.STATUS_OFF

The command **DIG<n>.CHK_TO=<time before alarm (s.)>** can be used to determine the time after which the appearance or disappearance of the mains voltage must generate a **DIG<n>_ON** or **DIG<n>_OFF** type event. The value of the parameter must be from 0 to 9999 seconds.

The command **DIG<n>.CHK_STATUS?** returns ON or OFF according to the current state of the input DIG<n> monitoring. It should be noted that the value of this variable differs from DIG<n>? since it includes the notion of duration. The value DIG<n>.CHK_STATUS? changes on the appearance of the associated event, whilst the value DIG<n>? changes as soon as the corresponding input DIG<n> changes state.

To activate, deactivate and read the state of DIG<n>

The commands **DIG<n>=ON** and **DIG<n>=OFF** respectively activate and deactivate the output DIG<n>. You can reread the binary state of the DIG<n> with the command **DIG<n>?**.

CAUTION: To use **DIGn** as an **INPUT** you should set it to OFF using the command **"DIG<n>=OFF"**.

To send a pulse on DIG<n>

The command **DIG<n>_PULSE=<pulse duration d>** can be used to toggle the state (inversely) of the DIGn I/O for a time **d** expressed in seconds (0 < d <= 60).

Caution: the command DIG1_PULSE inverts the state of the I/O DIG1 for **d** seconds, which signifies that the state of DIG1 must be positively known or defined beforehand.

To save the state of the DIG<n> output

If you want the state of the digital I/O DIG<n> to be saved and recovered when the HITPLAYER-L is turned back on, you must enable the save mode for this I/O by entering the command **DIG<n>.SAVE=ON**. To disable: **DIG<n>.SAVE=OFF**

DIG I/O summary

The command **HELP.DIG** displays the following list

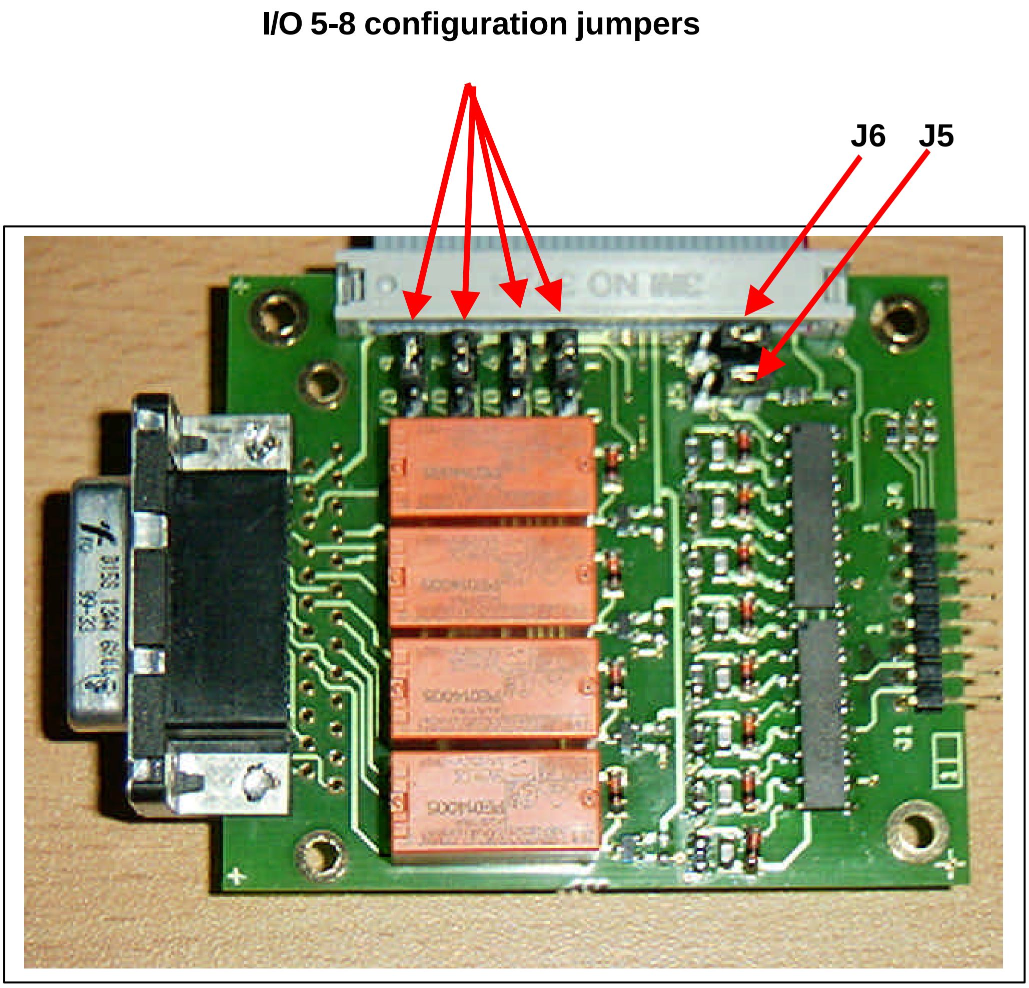
```
*** HITPLAYER-L : DIG HELP ***


[*]?DIG                  : Display config and status for all DIG
[*]?DIGx                 : Display config and status for DIGx
DIGx.NAME=<str>                  : Set alias for DIGx
DIGx=ON|OFF                      : Enable/Disable DIGx output
DIGx?                    : Display tested state of DIGx input
DIGx.STATUS_ON=<str>             : Set alias for DIGx On status
DIGx.STATUS_OFF=<str>            : Set alias for DIGx Off status
DIGx.STATUS?                     : Display alias for current tested DIGx status
DIGx.CHK?                        : Display current supervision status
DIGx.CHK_STATUS?         : Display alias for current supervision status
DIGx.CHK_TO=<val>                : Set the delay time for DIGx supervision (0-9999s)
DIGx.PULSE=<val>                 : Output a pulse on DIGx (0-60s)
DIGx.SAVE=ON|OFF         : Enable/Disable non-volatile saving of DIGx state
DIGx.TO=<val>                    : Set maximum time for DIGx On state (0-60mins)
INIT.DIG                         : Initialize config of all DIG
```

Events

DIG<n>_ON Digital i/o turned to ON
DIG<n>_OFF Digital i/o turned to OFF

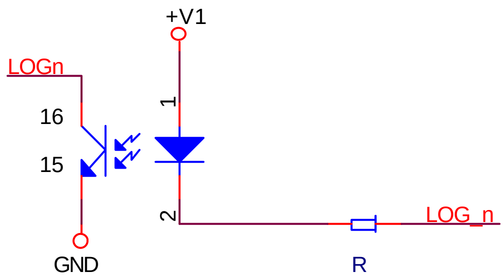
🗎 To use the Optional Opto-isolated GPIO board (ref. AZT3533).

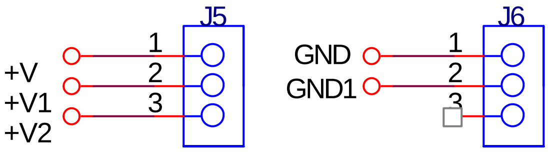


🗎 Technical description:

- 8 TTL logic inputs: common ground, selectable internally or externally
- 4 relays 24V / 1A, 1T contacts, if these relays are used, only 4 inputs will be available.
- SUBD15 connector

🗎 Input stage schematic:





+V : Internal power supply

+V1 : Logical input stage power supply

+V2 : External power supply (up to 12VDC)

GND : SPX board ground

GND1 : External Ground signal connected to SUBD15

Jumper	Function	Position 1-2	Position 2-3
J5	Input stage supply	Auto supply	External supply
J6	Input stage Ground	Connected to SPX Ground	Externally connected

🗎 Driving the inputs with relay contacts

- Place the jumpers J5 and J6 in 1-2 position.
- The logical stage are internally power supplied (+V1 connected to +V) and the SPX ground signal GND is connected to a pin of the SUBD15.
- In this mode you just have to connect the GND1 signal to your common pin relay and the R or T pin to Input n.

CAUTION: in this mode if there's an overvoltage on the SUBD15 pins the SPX board could be damaged.

🗎 Driving the inputs with relay contacts or logical signals using the Opto-isolated protection

- Place the jumpers J5 and J6 in 2-3 position.
- The logical stage are now externally power supplied (+V1 connected to +V2) and the SPX ground signal GND is disconnected to the GND1 pin of the SUBD15.
- In this mode you have to supply the input stages with an external power supply using +V2 pin of the SUBD15 connector.
- The relays contacts should connect the logical inputs with the external ground signal to make active the considerate inputs.

- This mode allows to drive the inputs with logical signals which could take these two values: 0 Volt or +V2 volts.

In this mode the SPX board is protected against overvoltage.

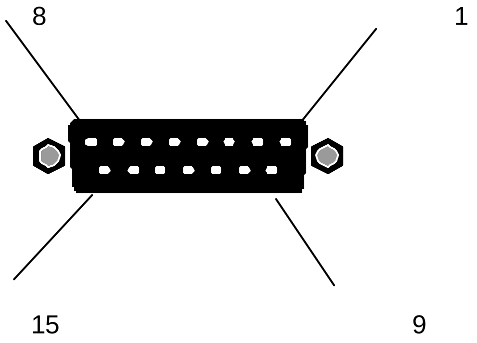
Inputs / Outputs 5 to 8

Inputs 1 to 4 are always in input mode. But I/O5, I/O6, I/O7 and I/O8are configurable in **input or output**. Each one is configurable independently. To configure these for I/O there are four jumpers labelled on the PCB I/O 5, I/O 6, I/O 7 and I/O 8.

The configuration is described bellow:

Jumper	Function	Position I	Position O
I/O 5	Input / Output configuration	I/O 5 is in input mode	I/O 5 is in output mode
I/O 6	Input / Output configuration	I/O 6 is in input mode	I/O 6 is in output mode
I/O 7	Input / Output configuration	I/O 7 is in input mode	I/O 7 is in output mode
I/O 8	Input / Output configuration	I/O 8 is in input mode	I/O 8 is in output mode

🖹 SUBD15 Pinout

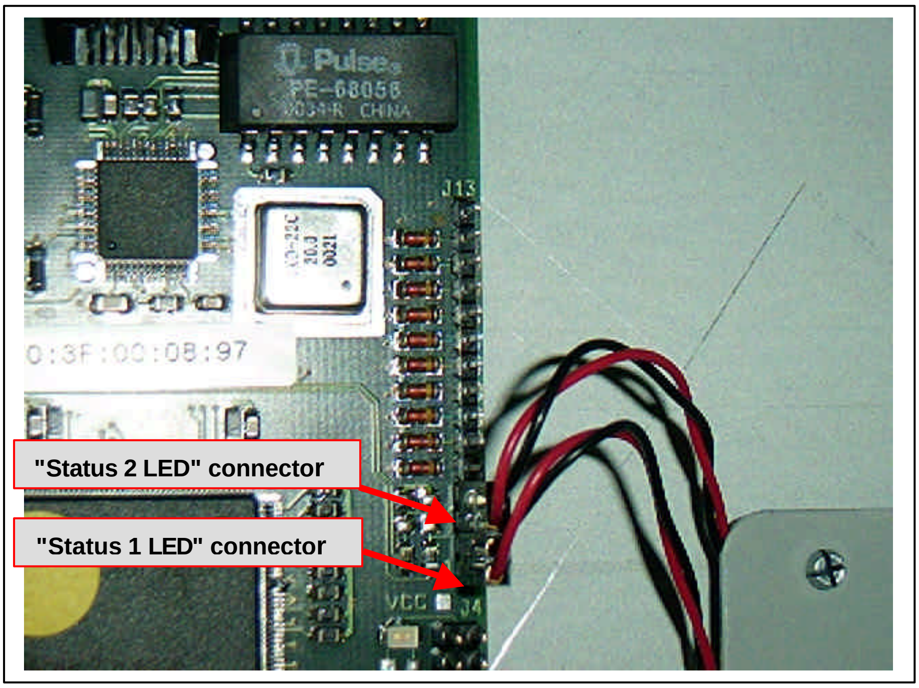


SUBD15 PIN N°	
1	Logical input n° 1
2	Logical input n° 2
3	Logical input n° 3
4	Logical input n° 4
5	Logical input n°5 *
	Rel5_T **
6	Logical input n°6 *
	Rel6_T **
7	Logical input n°7 *
	Rel7_T **
8	Logical input n°8 *
	Rel8_T **
9	+V2
10	GND1
11	GND1
12	REL5_C
13	REL6_C
14	REL7_C
15	REL8_C

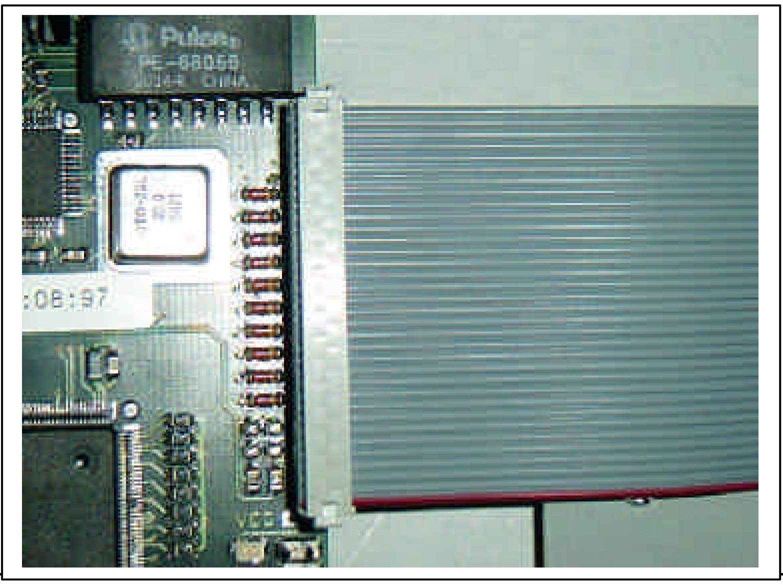
*** I/O jumper is in I position** **** I/O jumper is in O position**

🗎 Opto-isolated inputs optional board installation

- **First disconnect the power supply cable of the Hitplayer-L**
- Open the Hitplayer – L
- Before unplugging the LED cable connectors note in witch order they are connected
- Remove the Hitplayer – L leds cables from the SPX Board:

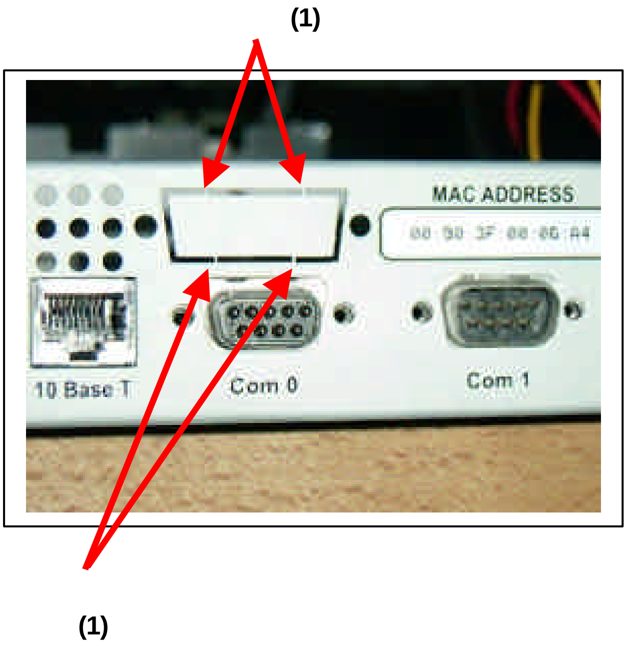


- Then connect the HE10-34 to the SPX Board (connector J13) as shown:

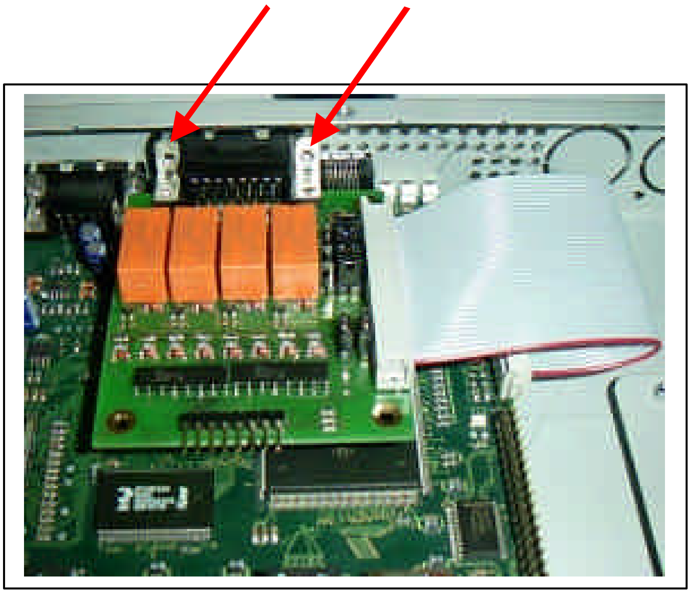


As J13 is a single row connector, the HE10-34 is double Row connector, so just connect pin 1 or 2 of the HE10-34 connector to the pin 1 of J13 connector.

- Then remove the pre–cut cover for the SUBD15 at the rear of the rack by cutting the clamps (1) as shown:



- Fix from the outside of the Rack the SUBD15 with two screws



- Reconnect the two LEDS cable connectors to J1 connector