

Software Draft

SD-002
RS232 Protocol



La Ville en Bois
44830 BOUAYE
☎ : 024078036

☎ : 02 40 78 22 44
✉ info@wsystem.com
<http://www.wsystem.com>



RS232 Protocol

v1.4

SD-002

Released list

| Released | Date | Description |
|----------|------------|--|
| 1.0 | 2004/06/29 | First released |
| 1.1 | 2004/10/20 | Add RD_STATUS, FILE_INFO, MEDIA_INFO... |
| 1.2 | 2004/10/26 | RD_STATUS minor change |
| 1.3 | 2004/12/23 | FILE_INFO major change, Ethernet functions added |
| 1.4 | 2005/01/12 | Add ID Ethernet + complete Module I/O Token |

Document validity

Player : SP603 / MP903 / AP303
Firmware : 3.17 or sup

Reference Documents

| Ref | Released | Description |
|--------|----------|----------------|
| SD-001 | 1.1 | Mapping EEPROM |
| SD-005 | 1.0 | Timer Messages |
| SD-003 | 1.0 | Tones tables |

Table of Contents

| | |
|--|-----------|
| 1. Electrical specifications..... | 4 |
| 2. General description | |
| 2.1 Overview..... | 4 |
| 2.2 Command Set Summary..... | 5 |
| 2.3 Mode | 6 |
| signification..... | |
| 2.4 Direction signification..... | 6 |
| 2.5 Writing access..... | 7 |
| 2.6 Reading access..... | 7 |
| 3. Complete command set | |
| 3.1 Transport / Status..... | 8 |
| 3.2 Tones | 11 |
| Control..... | |
| 3.3 RTC Informations..... | 13 |
| 3.4 EEPROM control..... | 15 |
| 3.5 Timer | 17 |
| Messages..... | |
| 3.6 Cmde Timer Messages Done..... | 19 |
| 3.7 File Informations..... | 20 |
| 3.8 Media Informations..... | 23 |
| 3.9 Command | 24 |
| IO..... | |
| 3.10 FTP Mode..... | 25 |
| 3.11 MAIL..... | 25 |

Note : ■ Changes with the previous released.

The purpose of this document is to specify the RS232 protocol to control the MP3 player. It is subject to change so, readers should check from time to time the new specifications released of this document available in the web site.

1. Electrical specifications

| Specification | Comment |
|---------------|------------------------------------|
| Support | Serial COM RS232 (TX/RX/GND) |
| Baudrate | 9600, 19200, 57600 bauds (default) |
| Data | 8 bits |
| Parity | No |
| Stop Bit | 1 |
| Handshaking | No |

2. General description

- 2.1 Overview

| FRAME | | | | | | | |
|--------|---------|--------|--------|--------|-----|--------|---------|
| TOKEN | ACK/ERR | Data 1 | Data 2 | Data 3 | ... | Data n | ACK/ERR |
| Byte 1 | Byte 2 | Byte 3 | Byte 4 | Byte 5 | ... | ... | ... |

- **TOKEN** : This byte is used to select a specific command and chose its direction (Read/Write).
- **ACK/ERR** : These 2 bytes provide a flow control, avoiding losing data during a transfert. An ACK byte means the command is accepted or no errors occured during a transfert.
- **DATA** : These bytes depend on the executed command (See each command description to get more informations).

| TOKEN | | | | | | | |
|------------------------------------|---|------------|----|--------------------------|----|----|-----|
| Name : TOKEN BYTE Size : 1 byte | | | | | | | |
| MSB | | | | | | | LSB |
| 0 | 0 | 0 | C3 | C2 | C1 | C0 | R/W |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit Name | | Bit Number | | Description | | | |
| R/W | | 0 | | Read/Write | | | |
| C3-C0 | | 1-4 | | Code Command (See below) | | | |
| x | | 5-7 | | Not Used | | | |

- **2.2 Command Set Summary**

| TOKEN CMDS | | | | |
|-------------------------|--------------|------------------|-------------|----------------|
| Code Cmd (C3-C0) | TOKEN | Direction | Mode | Command |
| '0001' | 0x02 | Write | Slave | WR_TRANSPORT |
| | 0x03 | Read | Slave | RD_STATUS |
| '0010' | 0x04 | Write | Slave | WR_TONES |
| | 0x05 | Read | Slave | RD_TONES |
| '0011' | 0x06 | Write | Slave | WR_RTC |
| | 0x07 | Read | Slave | RD_RTC |
| '0100' | 0x08 | Write | Slave | WR_EEPROM |
| | 0x09 | Read | Slave | RD_EEPROM |
| '0101' | 0x0A | Write | Slave | WR_TMSG_PROG |
| | 0x0B | Read | Slave | RD_TMSG_PROG |
| '0110' | 0x0C | Write | Slave | TMSG_END_PROG |
| | 0x0D | Read | Slave | RD_FILE_INFO |
| '0111' | 0x0E | Write | Master | IO_EXT |
| | 0x0F | Write | Slave | RD_MEDIA_INFO |
| '1000' | 0x10 | Write | Master | FTP_CLIENT |
| | 0x11 | Write | Master | MAIL |

| TOKEN FLOW CONTROL | |
|---------------------------|------------------|
| TOKEN Value | Byte Name |
| 0x01 | ACK |
| 0x1F | ERR |

| TOKEN FLAG (Only master) | |
|---------------------------------|--|
| TOKEN Value | Byte Name |
| 0x0D | FILE_CHANGE |
| 0x1C | PLAYER_READY |
| 0x1E | TICK Token (only used with I/O Module) |

Note : These Token are sended only by the player (master mode) with no protocol, to inform the host.
- FILE CHANGE : Sended just before playing a MP3 file. It's useful for the host to create a list of files played.
- PLAYER_READY : Sended when the player is ready after a RESET command, a reboot or power-on.

| TOKEN ID EXTENSION (At player start up) | |
|--|--------------------------|
| TOKEN Value | Extension module |
| 0x1D | I/O Module (Relay / DMX) |
| 0x1B | Ethernet Module |

Note : TOKEN 0x00 is not used.

- **2.3 Mode signification**

| Command Mode | Description |
|---|----------------------------|
| <p>Master Mode</p> <p>Only the player can send a TOKEN byte matching to the command to be executed. It means this kind of command can be executed under specific conditions.</p> | <p>Master</p> <p>Slave</p> |
| <p>Slave Mode</p> <p>Only the host can send a TOKEN byte matching to the command to be executed. It means this kind of command can be executed any time.</p> | <p>Slave</p> <p>Master</p> |

- **2.4 Direction signification**

| Direction | Description |
|---|--------------------|
| <p>Read</p> <p>The host read informations from the player</p> | |
| <p>Write</p> <p>The host send informations to the player</p> | |

- **2.5. Writing access**

| Step | Transfert | Comment |
|------|---|---|
| 1 | <pre> graph LR Host[Host] -- TOKEN --> Player[Player] </pre> | The host send a TOKEN byte to the player. |
| 2 | <pre> graph LR Player[Player] -- ACK --> Host[Host] </pre> | If the TOKEN byte is detected and the player is ready to process a command, it send back an ACK byte to the host. |
| 3 | <pre> graph LR Host[Host] -- DATA --> Player[Player] </pre> | The host send data command to the player (See each command) |
| 4 | <pre> graph LR Player[Player] -- ACK --> Host[Host] </pre> | The player send an ACK byte to inform the host the command has been executed without errors. |

Note : This figure shows the data flows in slave mode. In master mode, the player and the host are just permuted.

- **2.6 Reading access**

| Step | Transfert | Comment |
|------|---|---|
| 1 | <pre> graph LR Host[Host] -- TOKEN --> Player[Player] </pre> | The host send a TOKEN byte to the player. |
| 2 | <pre> graph LR Player[Player] -- ACK --> Host[Host] </pre> | If the TOKEN byte is detected and the player is ready to process a command, it send back an ACK byte to the host. |
| 3 | <pre> graph LR Player[Player] -- DATA --> Host[Host] </pre> | The player send data to the host (without flow control) (See each command) |
| 4 | <pre> graph LR Player[Player] -- ACK --> Host[Host] </pre> | The player send an ACK byte to inform the host the end of the transfert (The host should check this byte to be sure not to have lost any data). |

3. Complete command set

- 3.1 Transport / Status

| TRANSPORT / STATUS | | | | | | | |
|--|---|--------------|---|---|---|---|-----|
| Name : TRANSPORT / STATUS | | | | | | | |
| Mode : Slave | | | | | | | |
| Access : Read/Write | | | | | | | |
| Code CMDE : '0001' | | | | | | | |
| Description : Control and read the player's status | | | | | | | |
| MSB | | | | | | | LSB |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | R/W |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R/W | | Descriptions | | | | | |
| 0 | | WR_TRANSPORT | | | | | |
| 1 | | RD_STATUS | | | | | |

| WR_TRANSPORT | | | | | | | |
|--------------------------------|----|----|----|--------------------------------|----|----|-----|
| Valeur du TOKEN : 0x02 | | | | | | | |
| Data Number : 1 byte | | | | | | | |
| Direction : Write | | | | | | | |
| Description : Control register | | | | | | | |
| DATA | | | | | | | |
| MSB | | | | | | | LSB |
| R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value | | | | Action | | | |
| 0x01 | | | | PLAY | | | |
| 0x02 | | | | STOP | | | |
| 0x03 | | | | RESET | | | |
| 0x04 | | | | NEXT_FILE | | | |
| 0x05 | | | | PREVIOUS_FILE | | | |
| 0x06 | | | | NEXT_DIR | | | |
| 0x07 | | | | PREVIOUS_DIR | | | |
| 0x08 | | | | MODE USB | | | |
| 0x09 | | | | Reserved | | | |
| 0x0A | | | | PAUSE | | | |
| 0x0B | | | | FTP_SERVER ↔ MODE ETHERNET [1] | | | |

Note :

[1] : MODE ETHERNET : in FTP server mode (used with Ethernet Module). The player switch OFF all the data bus to enable the Ethernet module to access to the media (Hard drive, Compact Flash). To exit of this mode, the Ethernet module can send a RESET command to the player.

RS232 Protocol

v1.4

SD-002

RD_STATUS

Valeur du TOKEN : **0x03**
 Data number : 3 bytes
 Direction : Read
 Description : Status register

Byte 1

Name : CONFIG_STATUS
 Description : Return options menu

| MSB | | | | | | | | LSB | | | | | | | | | | | | | | | | | | |
|----------|--|---|--|--------|--------|---------------|----------------------------------|-------|----------|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|
| 0 | | 0 | | MODE_H | MODE_L | PSWD | TMSG | RS232 | AUTOPLAY | | 7 | | 6 | | 5 | | 4 | | 3 | | 2 | | 1 | | 0 | |
| Bit name | | | | Bit | Value | Default (Rst) | Description | | | | | | | | | | | | | | | | | | | |
| AUTOPLAY | | | | 0 | 1/0 | 0 | Autoplay (ON/OFF) | | | | | | | | | | | | | | | | | | | |
| RS232 | | | | 1 | 1/0 | 1 | RS232 remote (ON/OFF) | | | | | | | | | | | | | | | | | | | |
| TMSG | | | | 2 | 1/0 | 0 | Timer Messages (ON/OFF) | | | | | | | | | | | | | | | | | | | |
| PSWD | | | | 3 | 1/0 | 0 | Password (Options Menu) (ON/OFF) | | | | | | | | | | | | | | | | | | | |
| MODE_L | | | | 4 | 1/0 | 0 | Playing Mode (See below) | | | | | | | | | | | | | | | | | | | |
| MODE_H | | | | 5 | 1/0 | 0 | Playing Mode (See below) | | | | | | | | | | | | | | | | | | | |
| Not Used | | | | 6 | 0 | 0 | Not Used | | | | | | | | | | | | | | | | | | | |
| Not Used | | | | 7 | 0 | 0 | Not Used | | | | | | | | | | | | | | | | | | | |

Note : '1' value ⇔ ON / '0' value ⇔ OFF

| MODE_H | MODE_L | PLAYING MODE |
|--------|--------|--------------|
| 0 | 0 | Sequentiel |
| 0 | 1 | Single |
| 1 | 0 | Repeat |
| 1 | 1 | Random |

Byte 2

Name : PLAYER_STATUS REGISTER 1
 Description : Return the state of the player

| MSB | | | | | | | | LSB | | | | | | | | | | | | | | | |
|----------|----------|---------|-------|------|-------|---------------|----------------------------|-----|--|---|--|---|--|---|--|---|--|---|--|---|--|---|--|
| FILE_DIR | ROOT_DIR | OPTIONS | PAUSE | STOP | PLAY | USB | STANDBY | 7 | | 6 | | 5 | | 4 | | 3 | | 2 | | 1 | | 0 | |
| Bit name | | | | Bit | Value | Default (Rst) | Description | | | | | | | | | | | | | | | | |
| STANDBY | | | | 0 | 1/0 | x | '1' : Standby Mode | | | | | | | | | | | | | | | | |
| USB | | | | 1 | 1/0 | x | '1' : USB Mode | | | | | | | | | | | | | | | | |
| PLAY | | | | 2 | 1/0 | x | '1' : Playing | | | | | | | | | | | | | | | | |
| STOP | | | | 3 | 1/0 | x | '1' : Stop | | | | | | | | | | | | | | | | |
| PAUSE | | | | 4 | 1/0 | x | '1' : Pause | | | | | | | | | | | | | | | | |
| OPTIONS | | | | 5 | 1/0 | x | '1' : Options | | | | | | | | | | | | | | | | |
| ROOT_DIR | | | | 6 | 1/0 | x | '1' : Root / '0' Directory | | | | | | | | | | | | | | | | |
| FILE_DIR | | | | 7 | 1/0 | x | '1' : File / '0' Directory | | | | | | | | | | | | | | | | |

RS232 Protocol

v1.4

SD-002

| Byte 3 | | | | | | | |
|---|------------|--------------|----------------------|---------------------|---|-----|------|
| <i>Name</i> : PLAYER_STATUS REGISTER 2 | | | | | | | |
| <i>Description</i> : Return the state of the player | | | | | | | |
| MSB | | | | | | | LSB |
| 0 | 0 | 0 | 0 | 0 | 0 | ETH | MUTE |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit name | Bit | Value | Default (Rst) | Description | | | |
| STANDBY | 0 | 1/0 | x | '1' : Mute Mode | | | |
| ETH | 1 | 1/0 | x | '1' : Mode Ethernet | | | |
| Not Used | 2 | 0 | x | Not Used | | | |
| Not Used | 3 | 0 | x | Not Used | | | |
| Not Used | 4 | 0 | x | Not Used | | | |
| Not Used | 5 | 0 | x | Not Used | | | |
| Not Used | 6 | 0 | x | Not Used | | | |
| Not Used | 7 | 0 | x | Not Used | | | |

- **3.2 Tones Control**

| TONES | | | | | | | |
|---|---|---|---|---------------------|---|---|-----|
| Name : TONES | | | | | | | |
| Mode : Slave | | | | | | | |
| Access : Read/Write | | | | | | | |
| Code CMDE : '0010' | | | | | | | |
| Description : Control and read tones (Volume and balance) | | | | | | | |
| MSB | | | | | | | LSB |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | R/W |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R/W | | | | Descriptions | | | |
| 0 | | | | WR_TONES | | | |
| 1 | | | | RD_TONES | | | |

| WR_TONES | | | | |
|---|-------------------------------|----------|---------------|-----------|
| TOKEN : 0x04 | | | | |
| Data number : 2 bytes | | | | |
| Direction : Write | | | | |
| Description : Select a tone and set a value | | | | |
| DATA | | | | |
| MSB | | | | LSB |
| Tone | | Value | | |
| Byte 1 | | Byte 2 | | |
| TONE Value in decimal (Byte 1) | VALUE in decimal (Byte 2) [2] | | | TONE TYPE |
| | Min | Max | Default (Rst) | |
| 1 | 0 (-99dB) | 70 (0dB) | 0 (0dB) | VOLUME |
| 2 | 0 | 30 | 15 (0dB) | BASS |
| 3 | 0 | 30 | 15 (0dB) | TREBLE |
| 4 | 0 | 255 | 255 | MUTE [1] |

Notes :

[1] : Mute :

- 0 : Set Mute mode (decoding MPEG frames is not interrupted).
- Any others values : End of Mute mode.

[2] : Tones values : See '**SD-003 Tones Tables**' document to get more informations.

RD_TONES

TOKEN : 0x04
Data number : 3 bytes
Direction : Read
Description : Read 3 tones values (Volume/Bass/Trebles)

DATA

| Byte Value in decimal [1] | | | TONE TYPE |
|----------------------------------|------------|----------------------|------------------|
| Min | Max | Default (Rst) | |
| 0 | 70 | 0 (0dB) | VOLUME |
| 0 | 30 | 15 (0dB) | BASS |
| 0 | 30 | 15 (0dB) | TREBLES |

Note :

[1] : Tones values : See '**SD-003 Tones Tables**' document to get more informations.

- **3.3 RTC Informations**

| RTC | | | | | | | |
|--|---|--------------|---|---|---|---|-----|
| Name : RTC Mode : Slave Access : Read/Write Code CMDE : '0011' Description : Read/Write RTC informations | | | | | | | |
| MSB | | | | | | | LSB |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | R/W |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R/W | | Descriptions | | | | | |
| 0 | | WR_RTC | | | | | |
| 1 | | WR_RTC | | | | | |

| WR_RTC | | | | | | |
|--|--------|---------------|----------------|--------|--------|--------|
| Valeur du TOKEN : 0x06 Name : WR_RTC Data Number : 7 bytes Description : Write RTC Informations | | | | | | |
| DATA | | | | | | |
| MSB | | | | | | LSB |
| YEAR | MONTH | DATE | DAY | HOUR | MIN | SEC |
| Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 |
| Values in decimal [1] | | | Byte Field | | | |
| Min | Max | Default (Rst) | | | | |
| 0 | 59 | 0 | SEC | | | |
| 0 | 59 | 0 | MIN | | | |
| 0 | 23 | 12 | HOUR (24H/24H) | | | |
| 1 | 7 | 1 | DAY | | | |
| 1 | 31 | 1 | DATE | | | |
| 1 | 12 | 1 | MONTH | | | |
| 0 | 99 | 1 | YEAR | | | |

Note :

[1] RTC values are coded in 24/24 format (am/pm format is not supported).

RS232 Protocol

v1.4

SD-002

RD_RTC

Valeur du TOKEN : **0x07**
 Name : RD_RTC
 Data number : 7 bytes
 Description : Read RTC Informations

DATA

| MSB | | | | | | | LSB |
|-------------------|--------|---------------|----------------|--------|--------|--------|-----|
| YEAR | MONTH | DATE | DAY | HOUR | MIN | SEC | |
| Byte 6 | Byte 5 | Byte 4 | Byte 3 | Byte 2 | Byte 1 | Byte 0 | |
| Values in decimal | | | Byte Field | | | | |
| Min | Max | Default (Rst) | | | | | |
| 0 | 59 | 0 | SEC | | | | |
| 0 | 59 | 0 | MIN | | | | |
| 0 | 23 | 12 | HOUR (24H/24H) | | | | |
| 1 | 7 | 1 | DAY | | | | |
| 1 | 31 | 1 | DATE | | | | |
| 1 | 12 | 1 | MONTH | | | | |
| 0 | 99 | 1 | YEAR | | | | |

- **3.4 EEPROM control**

| EEPROM | | | | | | | |
|--|---|---|---|---------------------|---|---|-----|
| Name : EEPROM | | | | | | | |
| Mode : Slave | | | | | | | |
| Access : Read/Write | | | | | | | |
| Code CMDE : '0100' | | | | | | | |
| Description : Read/Write EEPROM informations | | | | | | | |
| MSB | | | | | | | LSB |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 | R/W |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R/W | | | | Descriptions | | | |
| 0 | | | | WR_EEPROM | | | |
| 1 | | | | RD_EEPROM | | | |

| WR_EEPROM | | | | |
|--|------------|-------------------|------------|-------------|
| Valeur du TOKEN : 0x08 | | | | |
| Name : WR_EEPROM | | | | |
| Data Number : 3 bytes | | | | |
| Description : Write a byte in EEPROM memory (32Kbytes) | | | | |
| DATA | | | | |
| MSB | | | | LSB |
| Adress High | | Adress Low | | Data |
| Byte 1 | | Byte 2 | | Byte 3 |
| Adress High | | Adress Low | | DATA |
| Min | Max | Min | Max | |
| 0x00 | 0x7F | 0x00 | 0xFF | |
| Byte to write in EEPROM (Max address 0x7FFF) | | | | |

| RD_EEPROM | | | | |
|---|------------|-------------------|------------|-------------|
| Valeur du TOKEN : 0x09 | | | | |
| Name : RD_EEPROM | | | | |
| Data Number : 2 bytes sended, 1 byte returned | | | | |
| Description : Read a byte in EEPROM memory (32Kbytes) | | | | |
| DATA | | | | |
| MSB | | | | LSB |
| Adress High | | Adress Low | | |
| Byte 1 | | Byte 2 | | |
| Adress High | | Adress Low | | |
| Min | Max | Min | Max | DATA |
| 0x00 | 0x7F | 0x00 | 0xFF | |
| Byte to read in EEPROM (Max address 0x7FFF) | | | | |

| Step | Read Transfert | Comment |
|-------------|---|--|
| 1 | <pre> graph LR Host[Host] -- TOKEN --> Player[Player] </pre> | The host send a TOKEN byte to the player. |
| 2 | <pre> graph LR Player[Player] -- ACK --> Host[Host] </pre> | If the TOKEN byte is detected and the player is ready to process a command, it send back an ACK byte to the host. |
| 3 | <pre> graph LR Host[Host] -- ADDRESS --> Player[Player] </pre> | The Host send the address (16 bits / 2 bytes). |
| 5 | <pre> graph LR Player[Player] -- DATA --> Host[Host] </pre> | The player return the byte read from EEPROM memory. |
| 6 | <pre> graph LR Player[Player] -- ACK/ERR --> Host[Host] </pre> | <ul style="list-style-type: none"> - Correct Address : ACK sended - Incorrect Address : <ul style="list-style-type: none"> > ERR byte returned > Data must be ignored |

- **3.5 Timer Messages (Reserved)**

| Timer Messages | | | | | | | |
|---|---|---------------------|---|---|---|---|-----|
| Name : EEPROM Mode : Slave Access : Read/Write Code CMDE : '0101' Description : Read/Write a Prog in Timer Messages | | | | | | | |
| MSB | | | | | | | LSB |
| 0 | 0 | 0 | 0 | 1 | 0 | 1 | R/W |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| R/W | | Descriptions | | | | | |
| 0 | | WR_TMSG_PROG | | | | | |
| 1 | | RD_TMSG_PROG | | | | | |

| WR_TMSG_PROG | |
|---|--|
| Valeur du TOKEN : 0x0A Name : WR_TMSG_PROG Data Number : 24 bytes sended Description : Write a Prog timer | |
| DATA | |
| See ' <i>SD-005 Timer Messages</i> ' document to get more informations about frames data | |


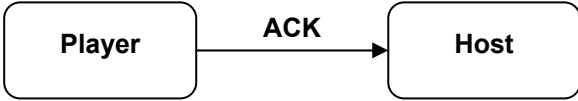

| RD_TMSG_PROG | |
|---|--|
| Valeur du TOKEN : 0x0B Name : RD_TMSG_PROG Data Number : 24 bytes returned + 1 byte sended (Prog Number) : 25 bytes Description : Read a byte in EEPROM memory (32Kbytes) | |
| DATA | |
| See ' <i>SD-005 Timer Messages</i> ' document to get more informations about frames data | |

Note : These functions are reserved for the Timer Messages program.

| Step | Read Transfert | Comment |
|------|---|---|
| 1 | <pre> graph LR Host[Host] -- TOKEN --> Player[Player] </pre> | The host send a TOKEN byte to the player. |
| 2 | <pre> graph LR Player[Player] -- ACK --> Host[Host] </pre> | If the TOKEN byte is detected and the player is ready to process a command, it send back an ACK byte to the host |
| 3 | <pre> graph LR Host[Host] -- PROG --> Player[Player] </pre> | The host send the prog number (1 byte) to select the prog to be read. |
| 4 | <pre> graph LR Player[Player] -- ACK/ERR --> Host[Host] </pre> | <ul style="list-style-type: none"> - Correct Prog Number : ACK - Incorrect Prog Number : <ul style="list-style-type: none"> > ERR byte sended > No data returned (end of the transfert). |
| 5 | <pre> graph LR Player[Player] -- DATA --> Host[Host] </pre> | If an ACK byte is received, the player returns the Prog frame (24 bytes). |
| 6 | <pre> graph LR Player[Player] -- ACK/ERR --> Host[Host] </pre> | The player send an ACK byte to inform the host the end of the transfert (The host should check this byte to be sure not to have lost any data). |

- **3.6 Cmde Timer Messages Done (Reserved)**

| TMSG_END_DONE | | | | | | | |
|---|---|---|---|---|---|---|-----|
| Name : TMSG_END_DONE Mode : Slave Access : Write Data number : 0 byte Code CMDE : '0110' (Token 0x0C) Description : Send a command to the player to finish the TMSG_PROG transferts. | | | | | | | |
| MSB | | | | | | | LSB |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Step | Command Transfert | Comment |
|------|--|---|
| 1 |  <pre> graph LR Host[Host] -- TOKEN --> Player[Player] </pre> | The host send a TOKEN byte to the player. |
| 2 |  <pre> graph LR Player[Player] -- ACK --> Host[Host] </pre> | If the TOKEN byte is detected and the player is ready to process a command, it send back an ACK byte to the host. |
| 3 |  <pre> graph LR Player[Player] -- ACK --> Host[Host] </pre> | The player send an ACK byte to inform the host the end of the transfert (The host should check this byte to be sure not to have lost any data). |

Note : These functions are reserved for the Timer Messages program.

- **3.7 FILE Informations**

| File Informations | | | | | | | |
|--|---|---|---|---|---|---|---|
| Name : File informations | | | | | | | |
| Mode : Slave | | | | | | | |
| Access : Read | | | | | | | |
| Data number : 264 bytes | | | | | | | |
| Code CMDE : '0110' (Token 0x0D) | | | | | | | |
| Description : Get File informations | | | | | | | |
| MSB LSB | | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Bytes Read | | | | | | | | | | | | |
|--|-------------|--|---|---|---|---|---|---|----|----|---------------|-----|
| MSB LSB | | | | | | | | | | | | |
| a | b | b | b | b | c | c | c | c | d | d | ... | d |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | ... | 264 |
| Bytes | Size (byte) | Values in decimal | | | | | | | | | Fields | |
| a | 1 | 1 – 255 (8-bit value) | | | | | | | | | FileName Size | |
| b | 4 | 0 – 2 ³² (32-bit value) LSB-MLSB-MMSB-MSB | | | | | | | | | File Size | |
| c | 4 | 32-bits value HEADER MPEG Info (MSB-LSB) | | | | | | | | | MPEG info | |
| d | 255 | ASCII bytes | | | | | | | | | FileName | |

Note :

- A "file" is a generic word which can be designed both a MP3 file and a directory (see below).
- FileSize is always a 32-bit value, independent of the FAT type (FAT16/32).
- FileName Size can't exceed 255 characters due to the FAT specifications (Long File Name).

| | | File type | |
|---|-----------|---------------------|------------|
| | | MP3 file | Directory |
| b | File Size | ≠ 0x00 | 0x00000000 |
| c | MPEG info | Exist (MPEG header) | 0x00000000 |

| MPEG INFORMATIONS | | | | | | | | | | | | | | | | |
|-------------------|---------------|-----------|--|--------|-------|-------|--------|----------|------|----------|--|--|--|--|--|--|
| Sign | Length (bits) | Positions | Description | | | | | | | | | | | | | |
| a | 11 | 31-21 | Synchronisation MPEG frame : all bits must be set ('1') | | | | | | | | | | | | | |
| b | 2 | 20-19 | MPEG version : 00 : MPEG 2.5 (V2) 01 : Rsvd 10 : MPEG 2 (V2) 11 : MPEG 1 (V1) | | | | | | | | | | | | | |
| c | 2 | 18-17 | Layer description : 00 : Rsvd 01 : Layer III (L3) 10 : Layer II (L2) 11 : Layer I (L1) | | | | | | | | | | | | | |
| d | 1 | 16 | Protection bit : 0 : Protected by CRC 1 : Not protected | | | | | | | | | | | | | |
| e | 4 | 15-12 | Bitrate index : Kbits/s | | | | | | | | | | | | | |
| | | | Bits | V1,L1 | V1,L2 | V1,L3 | V2,L1 | V2,L2&L3 | | | | | | | | |
| | | | 0000 | free | free | free | free | free | free | | | | | | | |
| | | | 0001 | 32 | 32 | 32 | 32 | 32 | 8 | | | | | | | |
| | | | 0010 | 64 | 48 | 40 | 48 | 48 | 16 | | | | | | | |
| | | | 0011 | 96 | 56 | 48 | 56 | 56 | 24 | | | | | | | |
| | | | 0100 | 128 | 64 | 56 | 64 | 64 | 32 | | | | | | | |
| | | | 0101 | 160 | 80 | 64 | 80 | 80 | 40 | | | | | | | |
| | | | 0110 | 192 | 96 | 80 | 96 | 96 | 48 | | | | | | | |
| | | | 0111 | 224 | 112 | 96 | 112 | 112 | 56 | | | | | | | |
| | | | 1000 | 256 | 128 | 112 | 128 | 128 | 64 | | | | | | | |
| | | | 1001 | 288 | 160 | 128 | 144 | 144 | 80 | | | | | | | |
| | | | 1010 | 320 | 192 | 160 | 160 | 160 | 96 | | | | | | | |
| | | | 1011 | 352 | 224 | 192 | 176 | 176 | 112 | | | | | | | |
| | | | 1100 | 384 | 256 | 224 | 192 | 192 | 128 | | | | | | | |
| | | | 1101 | 416 | 320 | 256 | 224 | 224 | 144 | | | | | | | |
| | | | 1110 | 448 | 384 | 320 | 256 | 256 | 160 | | | | | | | |
| | | | 1111 | bad | bad | bad | bad | bad | bad | | | | | | | |
| f | 2 | 11-10 | Sampling rate frequency index : Hz | | | | | | | | | | | | | |
| | | | Bits | MPEG 1 | | | MPEG 2 | | | MPEG 2.5 | | | | | | |
| | | | 00 | 44100 | | | 22050 | | | 11025 | | | | | | |
| | | | 01 | 48000 | | | 24000 | | | 12000 | | | | | | |
| | | | 10 | 32000 | | | 16000 | | | 8000 | | | | | | |
| | | | 11 | Rsvd | | | Rsvd | | | Rsvd | | | | | | |
| g | 1 | 9 | Pading bit : 0 : frame is not padded 1 : frame is padded | | | | | | | | | | | | | |

| | | | |
|---|---|-----|---|
| h | 1 | 8 | Private bit |
| i | 2 | 7-6 | Channel Mode : 00 : Stereo 01 : Joint Stereo (Stereo) 10 : Dual Mono (2 mono channels) 11 : Single channel (Mono) |
| j | 2 | 5-4 | Not used |
| k | 1 | 3 | Copyright : 0 : Audio is not copyrighted 1 : Audio is copyrighted |
| l | 1 | 2 | Original : 0 : Copy of original media 1 : Original media |
| m | 2 | 1-0 | Not used |

- **3.8 MEDIA Informations**

| Media Informations | | | | | | | |
|--|---|---|---|---|---|---|-----|
| Name : Media informations | | | | | | | |
| Mode : Slave | | | | | | | |
| Access : Read | | | | | | | |
| Data number : 11 bytes | | | | | | | |
| Code CMDE : '0111' (Token 0x0F) | | | | | | | |
| Description : Get informations about the media (Compact Flash, HDD, Size, File System) | | | | | | | |
| MSB | | | | | | | LSB |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Bytes Read | | | | | | | | | | |
|------------|--------|--|---|---|---|---|------------|---|----|----|
| MSB | | | | | | | | | | |
| f | a | a | a | a | a | a | s | s | s | s |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| LSB | | | | | | | | | | |
| Bytes | Offset | Values | | | | | Fields | | | |
| f | 1 | 0x01 = FAT16 / 0x02 = FAT32 | | | | | FAT type | | | |
| a | 2 | ASCII bytes | | | | | Media Name | | | |
| s | 8 | Media size (32 bits) LSB-LLSB-MMSB-MSB | | | | | Media Size | | | |

Note : Media Size is the number of sectors of the media. To get the correct size in bytes, just multiply this 32-bit values by 512.

- **3.9 Command IO (used in TMSG)**

| IO_EXT | | | | | | | |
|--|---|---|---|---|---|---|-----|
| <p>Name : IO_EXT Mode : Master/Slave Access : Write Data number : 2 bytes sended Code CMDE : '0111' (Token 0x0E) Description : <u>Slave</u> : Receive a command to launch a prog Timer. <u>Master</u> : Send a command to a device to switch relays.</p> | | | | | | | |
| MSB | | | | | | | LSB |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| IO_EXT – SLAVE Mode (Timer Messages Input) | | | | | | | |
|---|----|----|----|-----------------------------------|----|----|-----|
| <p>Name : IO_EXT/S Size : 1 byte Description : The player receive a data, which is the prog number to launch (Timer Messages)</p> | | | | | | | |
| MSB | | | | | | | LSB |
| R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value (R7-R0) | | | | Conditions | | | |
| 0x01 – 0xFF | | | | Prog Number in the Timer Messages | | | |

| IO_EXT – Master Mode (Timer Messages Output) | | | |
|--|------------|--|-----|
| <p>Name : IO_EXT/M Size : 2 bytes Description : The player send 2 data to an external device to control output relays.</p> | | | |
| MSB | | | LSB |
| Relay Number | | Value | |
| Byte 1 | | Byte 2 | |
| Value in decimal | | Byte Name | |
| Min | Max | | |
| 1 | 255 | Relay Number | |
| 0 | 255 | Value (proportional value : 0 > OFF, 1 > ON) | |

- **3.10 FTP CLIENT (used in Ethernet module)**

| FTP_Mode | | | | | | | |
|--|---|-----|---|-------------------------------|----|----|----|
| Name : FTP_CLIENT Mode : Master Access : Write Data number : 1 byte Code CMDE : '1000' (Token 0x10) Description : Send a command to switch in FTP (in client mode). | | | | | | | |
| MSB | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| MSB | | | | | | | |
| 0 | 0 | 0 | 0 | R3 | R2 | R1 | R0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value in decimal | | | | FTP_ADDRESS (R0-3) | | | |
| Min | | Max | | | | | |
| 1 | | 8 | | Up to 8 FTP ADR are available | | | |

- **3.11 MAIL (used in the Ethernet Module)**

| MAIL | | | | | | | |
|---|----|----|----|-----------------------------------|----|----|----|
| Name : MAIL Mode : Master Access : Write Data number : 1 byte send Code CMDE : '1000' (Token 0x11) Description : Send the mail number to send. | | | | | | | |
| MSB | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| DATA | | | | | | | |
| MSB | | | | | | | |
| R7 | R6 | R5 | R4 | R3 | R2 | R1 | R0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Value (R7-R0) in decimal | | | | Conditions | | | |
| 1 – 255 | | | | Mail Number in the Ethernet Board | | | |



www.wsystem.com

www.id-al.com