LPM11162 Audio Module

www.lpelettronica.it

Firmware version 5.2R
Doc. Rev.0 (08/10/12)

Features

- Separate Analog/Digital Supplies
- Typical 3.3V Supply Voltage
- Advanced Flash Filesystem to store WAV files
- 95 seconds of total audio recording
- 11.025KHz-16bit-mono WAV files
- High quality voice/sounds playback
- UART for programming/playback (SERIAL-MODE)
- Input PIN reading for playback (KEY-MODE)
- Analog audio output for playback
- 5KHz audio bandwidth for optimum voice playback
- LOOP function for playback looping (KEY-MODE)
- Small footprint device (20x28mm)
- Low power consumption
- Applications: automatic distributors, robots, industrial controllers, entertainment, gadgets

Description

LPM11162 is a simple integrated solution to store and reproduce high quality audio wave files in electronic circuits. Typical applications are automatic distributors, robots, speaking devices, gadgets and in general all the devices where it is required to program and play audio wave files.
The audio files (11KHz-16bit-mono) can be programmed in the internal 2Mbyte flash memory that is provided with an advanced filesystem. Programming is done on the UART port using simple filesystem commands. The baudrate for communication is programmable with the AutoBaudrate feature (up to 115200 bps).
The audio files can be reproduced on the analog ouput in two modes: SERIAL-MODE - using serial commands, KEY-MODE - driving the input pins (pressing keys , driving relè or transistors, ecc).

Typical Application

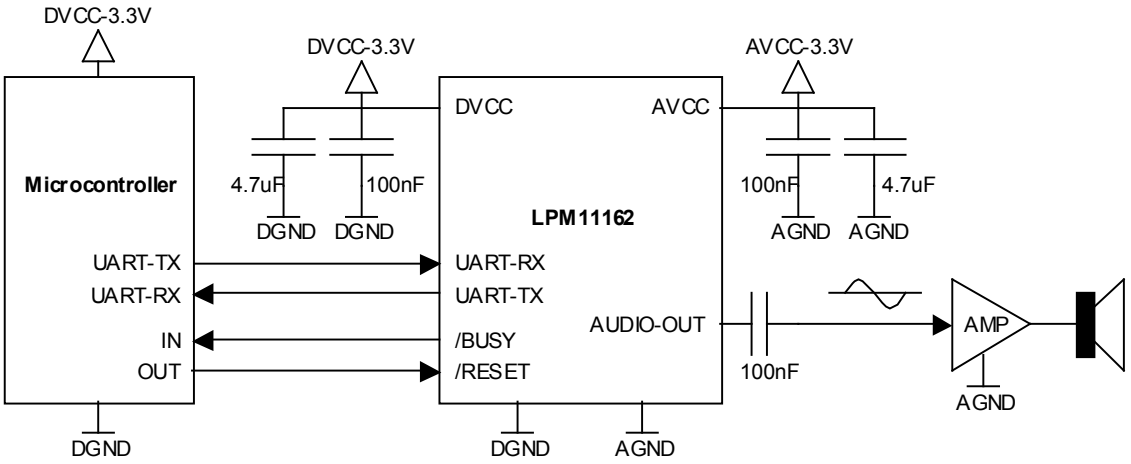


Figure 1 – Typical application schematic (SERIAL-MODE)

LPM11162

www.lpelettronica.it

Pin Diagram

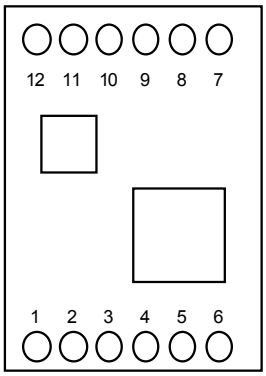


Figure 2 – Pin Diagram (Top View)

Pin Description

Pin	Pin Name	Pin Type	Description
1	**/BUSY** (SERIAL-MODE)	OUT	In SERIAL MODE this is an active low Busy pin. This signal is normally high, it is driven low by the module during a command execution (Busy condition). New commands are not accepted when the module is in Busy state.
	KEY2 (KEY-MODE)	IN	In KEY-MODE this pin is named KEY2 and it is used as input with other KEYx pins to play wave files on inputs status change.
2	**/RESET**	IN	Active low Reset pin. Low level is reset condition, high level is running condition.
3	**DV_{CC}**	POWER	Digital supply voltage.
4	**DGND**	POWER	Digital ground.
5	**UART-TX** (SERIAL-MODE)	OUT	In SERIAL-MODE this is the UART transmit pin.
	KEY1 (KEY-MODE)	IN	In KEY-MODE this pin is named KEY1 and it is used as input with other KEYx pins to play wave files on inputs status change.
6	**UART-RX** (SERIAL-MODE)	IN	In SERIAL-MODE this is the UART receive pin.
	KEY0 (KEY-MODE)	IN	In KEY-MODE this pin is named KEY0 and it is used as input with other KEYx pins to play wave files on inputs status change.

7	NC	-	Not connected
8	AV_{CC}	POWER	Analog supply voltage.
9	AUDIO-OUT	OUTPUT	Analog audio output signal (single ended).
10	NC	-	Not connected
11	AGND	POWER	Analog ground.
12	NC	-	Not connected

Table 1 – Pin Description

Absolute Maximum Ratings

Stresses exceeding the absolute maximum ratings may damage the device. The device may not function or be operable above the recommended operating conditions and stressing the parts to these levels is not recommended. In addition, extended exposure to stresses above the recommended operating conditions may affect device reliability. The absolute maximum ratings are stress ratings only.

Symbol	Parameter	Rating
DV_{CC}	Digital supply voltage	-0.3V to +4.0V
AV_{CC}	Analog supply voltage	0V to +5.5V
V_{IN}	Digital input pin voltage	-0.3V to +4.0V
T_A	Operating ambient temperature range	-20°C to +85°C
I_{OD}	DC maximum source/sink current from digital output pins	±10mA
I_{OA}	DC maximum source/sink current from analog output pin	±10mA

Recommended Operating Conditions

The Recommended Operating Conditions table defines the conditions for actual device operation. Recommended operating conditions are specified to ensure optimal performance to the datasheet specifications. It is not recommended exceeding them or designing to absolute maximum ratings.

Symbol	Parameter	Rating
DV_{CC}	Digital supply voltage	+2.7V to +3.6V
AV_{CC}	Analog supply voltage	+2.7V to +5.5V
V_{IN}	Digital input pin voltage	0V to DV_{CC}
T_A	Operating ambient temperature range	0°C to +70°C

Electrical Characteristics

Symbol	Parameter	Min	Typ	Max
I_{DCC}	Supply current on DV_{CC}: - Format - Programming - Play - Firmware update (LPM11162) - Firmware update (LPM11162 rev.1) - Idle	-	 30mA 20mA 20mA 50mA 13mA 20mA	-
I_{ACC}	DC maximum supply current on AV_{CC}	-	-	1mA
BW	Maximum analog output bandwidth	-	-	5KHz
V_{OUT}	Analog output voltage amplitude	$0V_{PP}$ (Volume 0%)	$1V_{PP}$ (Volume 50%)	$2V_{PP}$ (Volume 100%)
V_{IH}	High level input voltage on digital pin	0.8 DV_{CC}	-	-
V_{IL}	Low level input voltage on digital pin	-	-	0.2 DV_{CC}
V_{OH}	High level output voltage on digital pin	2.4V	-	-
V_{OL}	Low level outputvoltage on digital pin	-		0.4V

1 – Filesystem

1.1 – General Description

LPM11162 is designed with an internal filesystem that is able to store standard 11KHz-16bit-mono wave files. The UART interface makes possible to program/playback the wave files connecting LPM11162 to another device (for example a small microcontroller or a PC).

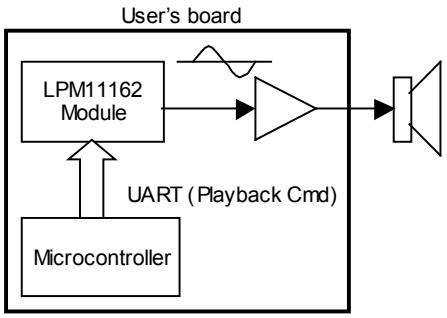


Figure 3 – Typical Programming and Playback block diagram (SERIAL-MODE)

LPM11162 is equiped with a 2Mbyte flash memory that with 11KHz-16bit-mono wave format allows to store a maximum of 95 seconds (maximum time for all the files). The maximum number of files is 512 and there is no limitation on the size of each file.

Parameter	Value
Max. number of files	512
Max. file name lenght	16 characters
Audio files sample rate	11025 Hz
Audio files data bits	16
Audio Channels	1 (mono)
Total flash memory size	2 Mbyte

Table 2 – Filesystem Specification

The filesystem is not case-sensitive for the file's names.

2 – KEY-MODE

2.1 – Introduction

There are cases where you don't have a microcontroller in your circuit, but would be nice to reproduce wave file if you have already programmed them on LPM11162.

This is the case of KEY-MODE, where 3 pins of LPM11162 are configured as input pins and the audio module is able to reproduce wave file when the pin's status changes. With KEY-MODE you can put LPM11162 in your circuit with a few of other components and reproduce audio wave files.

2.2 – Description

KEY-MODE is the default working mode of the module at startup.
After power-on the module's pins 1, 5, 6 are configured as input pins and the module consider them as "not pressed" in high voltage level. No wave files are played until one or more pins change state. If one ore more pins becomes low voltage level then LPM11162 will reproduce a wave file accordingly to the following table:

KEY2 (pin 1)	KEY1 (pin 5)	KEY0 (pin 6)	Wave File Name
L	L	L	File0.wav
L	L	H	File1.wav
L	H	L	File2.wav
L	H	H	File3.wav
H	L	L	File4.wav
H	L	H	File5.wav
H	H	L	File6.wav
H	H	H	No playback

Table 3 – Filesystem Specification

When all the KEY pins are high level the LPM11162 is not reproducing any audio file, while in all other pin's configurations there is an associated wave file to be reproduced. Of course each wave file can be reproduced only if it was previously programmed (see SERIAL-MODE for programming).
In a typical application each KEY pin can be connected to the supply voltage with a 100KΩ pull-up resistor. Then some external keys, or open-collector transistors, or relè can drive the KEY pins low to enable the playback of a specific wave file.

2.3 – LOOP function

The interesting LOOP function allows to reproduce a wave file in loop, so at the end of a file the playback will restart from the beginning.
This function is very simple to use: just select one of the input pin configurations from Table 3 in order to start the audio playback and maintain active the input pins. At the end of wave file playback the LPM11162 module will check again the input pins status and restart playback.

As an example of looping of a single wave file, select the playback of “File3.wav” by setting fixed levels on the pin 1 to GND and the pin 5, 6 to DVCC. After the LPM11162 is powered-on it checks for the input pin levels and start playing of “File3.wav”. When the file playback will finish the audio module will check again the input pins status and, if unchanged, it will restart playing “File3.wav” from the beginning.

2.4 – Inputs debouncing

When changing the status of LPM11162 input pins with switches, relè or other electromechanical device, some voltage bouncing can occur at the inputs of audio module. This situation is properly handled by the LPM11162 module with an internal debouncing time of 60ms.
It means that a status level change is considered valid only after the new status is stable for at least 60ms, that is a time longer enough to ensure a proper working with all electromechanical switching devices.

3 – SERIAL-MODE

3.1 – Introduction

In SERIAL-MODE the LPM11162 module is able to receive commands from the UART port in order to program audio wave files in the internal Flash memory and reproduce them on the analog audio output. This mode is the most flexible and the user can program/reproduce a large quantity of wave files sending simple serial commands from a host microcontroller or a PC (download the free LPM11162 ToolKit application).
SERIAL-MODE can be entered after power-on by sending the auto-baudrate character. When this character is received LPM11162 switches to SERIAL-MODE, configuring the pin 1 and pin 5 as outputs for /BUSY and UART-TX functions and configuring pin 6 as UART-RX.

3.2 – Writing files

Wave files can be programmed on LPM11162 with only 3 simple commands: File Open, File Write, File Close.
The File Write command permits to write the content of the audio files on the Flash of LPM11162.
Each File Write command permits to transfer a maximum of 256 bytes, so if the file size if greater than 256 bytes it is required to send some File Write commands in sequence for a complete file transfer.
The File Write commands are accepted only after the file has been opened with the File Open command in write mode to communicate to the module the name of the file that will be written. The file must be properly closed after all the data have been transferred by sending the File Close command.

For example suppose to write the file audio.wav that is a 612 bytes file, the sequence of commands is the following:

1) File Open command – write mode (audio.wav)
2) File Write command (first 256 data block)
3) File Write command (second 256 data block)
4) File Write command (last 100 data block)
5) File Close command

It is not possible to have two file with identical file name, so the File Open command will fail in case the file name is the same of another file that is already present in the Flash.

3.3 – Reading files

Wave files can be read from LPM11162 with only 3 simple commands: File Open, File Read, File Close.
The File Read command permits to read the content of the audio files on the Flash of LPM11162.
Each File Read command permits to request a maximum of 256 bytes, so if the file size if greater than 256 bytes it is required to send some File Read commands in sequence for a complete file transfer.
The File Read commands are accepted only after the file has been opened with the File Open command in read mode to communicate to the module the name of the file that will be read. The file must be properly closed after all the data have been transferred by sending the File Close command.

For example suppose to read the file audio.wav that is a 612 bytes file, the sequence of commands is the following:

6) File Open command – read mode (audio.wav)
7) File Read command (first 256 data block) – wait OK response containing read data
8) File Read command (first 256 data block) – wait OK response containing read data
9) File Read command (first 100 data block) – wait OK response containing read data
10) File Close command

3.4 – Formatting memory

The entire content of the Flash memory can be erased with the File Format command. This command erases all the files that are present in the Flash memory. During the format procedure the /BUSY line is driven low to indicate the module cannot receive other commands.
The duration of the format procedure can take up to several seconds (see Timing Specifications).

3.5 – Power-up sequence and auto-baudrate

In the figure below shows the power-up sequence with autobaudrate feature that allows to connect the module to different host devices without any hardware configuration.
After the module is removed from reset, the first operation to do is to send the auto-baudrate character 0x55 (ASCII "U"). When this character is received the module is able to understand the baudrate of the transmitter device and to configure itself to work at the same baudrate.
After the auto-baudrate character is received the /BUSY line is driven low to indicate that the module cannot accept commands. It sends a Module ID response and Module Firmware Version response, then the /BUSY line is driven high and the module is ready to receive commands.
For each command received the module replies with an OK/ERR response.

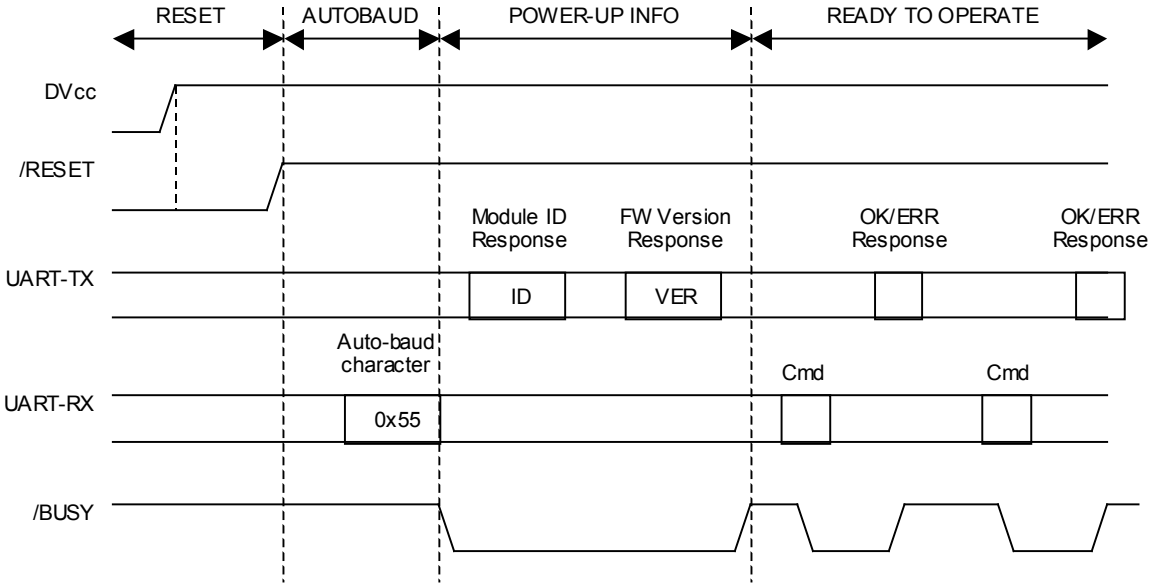


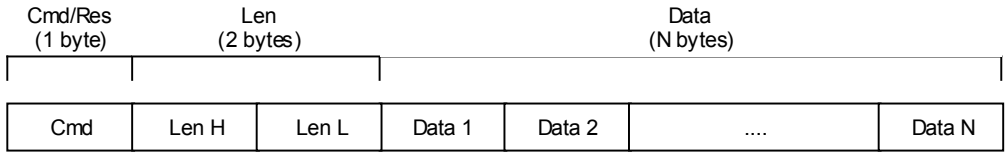
Figure 4 – Power-up sequence

3.6 – Busy state

After a command is received the module becomes Busy for the time required to process and execute the command. During this time the /BUSY signal is low to indicate that the module is able to receive news commands.
The unique command accepted in Busy state is the Check Busy command that allows to know via serial port if the previous command is completed or not.
The use of Check Busy command is an alternative to check the /BUSY line.
The /BUSY line is normally high and it is driven low by the module when the first character of a command is received to indicate that the command processing is going on. The line will reamain low for the time required to receive, process and execute the entire command. After execution the /BUSY line returns high and new commands are accepted.
Exception is made for Stop and Volume commands. After the Play command is received the module is considered Busy until the playback of the audio file is completed. During this time the /BUSY signal remains low, however the Stop and Volume commands can be sent in order to stop the file playback or adjust the volume level while playback is in progress.

3.7 - General Commands/Response format

The module's operations are controlled via a simple UART protocol where another device can send commands to the module and the module can reply with responses.
The commands are divided in Filesystem commands, Playback commands and Control commands. Using Filesystem commands another device can program wave files on the module and then using the Playback commands it is very simple to reproduce them.
For each command the LPM11162 module replies with an OK response if the command is accepted or with an ERR response otherwise.
All the commands/responses have the same general structure: the first byte indicates the command/responses type, the second and third bytes indicates the number of data bytes that will be sent into the command/response, then the data bytes are transmitted.
The general commands/responses structure is illustrated here:



Cmd/Res (1 byte):

- 0x01 – File Open command
- 0x02 – File Write command
- 0x03 – File Close command
- 0x04 – Format command
- 0x05 – File Read command
- 0x06 – File Delete command
- 0x07 – File Find command
- 0x08 – File Get Size command
- 0x10 – Play command

- 0x11 – Stop command
- 0x12 – Volume command
- 0x20 – Reset command
- 0x21 – Check Busy command
- 0x81 – Module ID response
- 0x82 – Module Firmware Version response
- 0x83 – OK response
- 0x84 – ERR response

Len (2 bytes):

- Number of data bytes in the command/response. Len is sent as high byte first.

Data 1…Data N (N bytes):

- Sequence of generic N data bytes. Data bytes can be present or not (Len=0) depending on the command/response.

Command	Description	Cmd	Len	Data
File Open	Open a new file on the module. Required before writing data into the file.	0x01	N+1 (N = number of characters in the file name).	RW byte + N file name characters.
File Write	Write data bytes into the file that is currently open for writing.	0x02	Number of data bytes to write in the file with current command.	Data bytes.
File Close	Close the file. Required after writing all data bytes into the file.	0x03	0x0000	None.
Format	Erase all the files that are currently present in Flash memory.	0x04	0x0000	None.
File Read	Read data bytes from the file that is currently open for reading.	0x05	Number of data bytes to read from file.	None.
File Delete	Delete a single file.	0x06	Number of characters in the file name.	File name characters.
File Find	Find a file in the Flash memory.	0x07	0x0001	0-First file 1-Next file
File Get Size	Get size of specified file.	0x08	Number of characters in the file name.	File name characters.
Play	Start playback of a selected file.	0x10	Number of characters in the name file to be played.	File name characters.

Stop	Stop playback of the file that is currently playing.	0x11	0x0000	None.
Volume	Set the playback volume level	0x12	0x0001	0-100
Reset	Reset the module.	0x20	0x0000	None.
Check Busy	Check if the module is busy after sending another command.	0x21	0x0000	None.

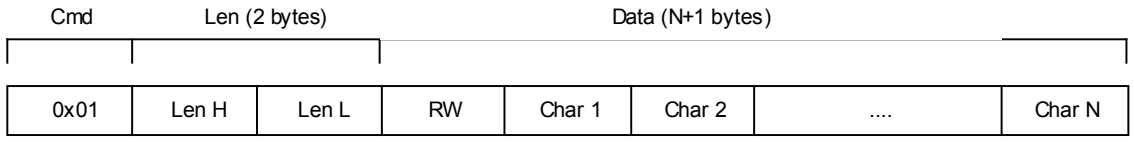
Table 4 – Commands Table

Response	Description	Res	Len	Data
Module ID	Send to the host device the module's ID.	0x81	0x0002	Module's ID
Module Firmware Version	Send to the host device the module's firmware version and date.	0x82	0x0009	Module's firmware version and date.
OK	Last command accepted.	0x83	Number of fata bytes required to reply to last command (see OK response details).	Data bytes.
ERR	Last command not accepted.	0x84	0x0000	None.

Table 5 – Responses Table

3.8 - File Open command

The first operation to access a file is send the File Open command. This command informs the module that a new file must be created with file name that is specified by the data bytes in the command.
The general format of File Open command is composed as follows:



Cmd (1 byte):
- 0x01 – File Open

Len (2 bytes):
- Number of data bytes, including RW byte and N characters in the file name.

RW (1 byte):
- 0x01 – Read mode
- 0x02 – Write mode

Char 1…Char N (N bytes):

- Sequence of N characters of the file name - no terminating char (N≤16).

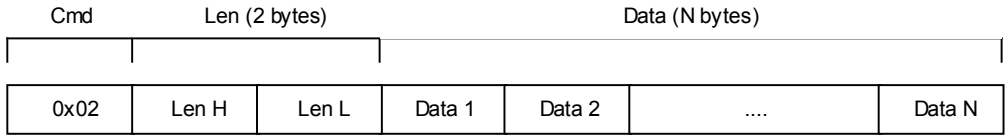
Here is shown an example of creating a new file (write mode) called “f1.wav”:

0x01	0x00	0x06	0x02	‘ f ’	‘ 1 ‘	‘ . ‘	‘w ‘	‘ a ‘	‘ v ‘

If the File Open command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. In both cases the response doesn’t contain any data so the Len field in the response is zero.

3.9 - File Write command

After a file has been opened in write mode (see File Open command) the File Write command can be used to write the file’s data bytes on the internal memory of the audio module.
The general format of File Write command is composed as follows:



Cmd (1 byte):

- 0x02 – File Write

Len (2 bytes):

- Number of data bytes in current command to write in the file (N≤256).

Data 1…Data N (N bytes):

- Sequence of N data bytes to write in the open file.

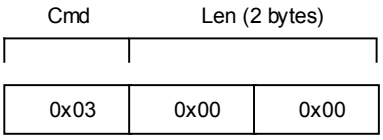
Here is shown an example of writing 4 bytes (0x0A, 0x0B, 0x0C, 0x0D) to an open file:

0x02	0x00	0x04	0x0A	0x0B	0x0C	0x0D

If the File Write command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. In both cases the response doesn’t contain any data so the Len field in the response is zero.

3.10 - File Close command

After finished to write all the data bytes to an open file, the file must be closed by sending the File Close command.
The general format of File Close command is composed as follows:



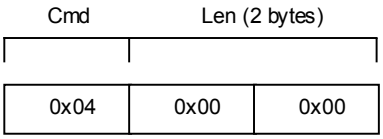
Cmd (1 byte):
- 0x03 – File Close

Len (2 bytes):
- Len is 0x0000 because no data bytes are present in the command.

If the File Close command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. In both cases the response doesn't contain any data so the Len field in the response is zero.

3.11 - Format command

The Format command is used to erase all the files present in the internal memory. This command is accepted only when there are no open files.
The general format of Format command is composed as follows:



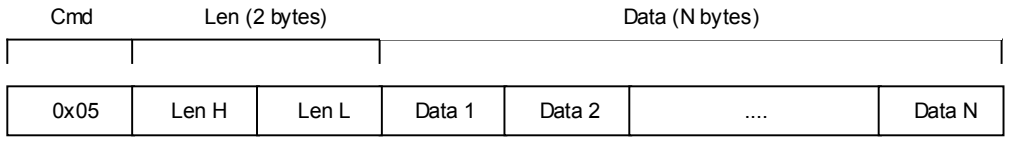
Cmd (1 byte):
- 0x04 – Format

Len (2 bytes):
- Len is 0x0000 because no data bytes are present in the command.

If the Format command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. In both cases the response doesn't contain any data so the Len field in the response is zero.
In case of an OK response the format requires some seconds to be completed and during this time the module is in Busy state, so cannot accept other commands. The Busy state is signaled by the /BUSY line but can also be checked via serial port by sending the Check Busy command that is the unique accepted during Busy state after a Format command.

3.12 - File Read command

After a file has been opened in read mode (see File Open command) the File Read command can be used to read the file's data bytes from the internal memory of the audio module.
The general format of File Read command is composed as follows:



Cmd (1 byte):

- 0x05 – File Read

Len (2 bytes):

- Number of data bytes to read from the file (N≤256).

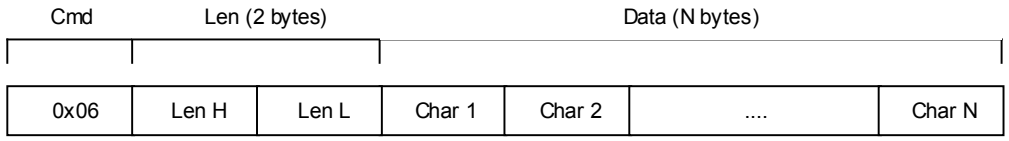
Here is shown an example of reading 4 bytes from an open file:

0x05	0x00	0x04

If the File Read command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. The OK response contains bytes read from file, see description of OK response for more details.

3.13 – File Delete command

The File Delete command allows to delete a single file from the internal memory.
The general format of File Delete command is composed as follows:



Cmd (1 byte):

- 0x06 – File Delete

Len (2 bytes):

- Number of characters in the file name (N≤16).

Char 1…Char N (N bytes):

- Sequence of N characters of the file name (no terminating char).

Here is shown an example of delete of file called “f1.wav”:

0x06	0x00	0x06	‘ f ’	‘ 1 ‘	‘ . ‘	‘w ‘	‘ a ‘	‘ v ‘

If the File Delete command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. In both cases the response doesn’t contain any data so the Len field in the response is zero.
In case of an OK response the delete time depends on the file size and it can take up to some seconds to be completed. During this time the module is in Busy state, so cannot accept other commands. The Busy state is signaled by the /BUSY line but can also be checked via serial port by sending the Check Busy command that is the unique accepted during Busy state after a File Delete command.

3.14 – File Find command

The File Find command allows to search files in the internal memory and get their names.
The general format of File Find command is composed as follows:

Cmd	Len (2 bytes)		Data
0x07	0x00	0x01	First/Next

Cmd (1 byte):

- 0x07 – File Find

Len (2 bytes):

- Len is 0x0001 because there is only 1 data byte for the First/Next value.

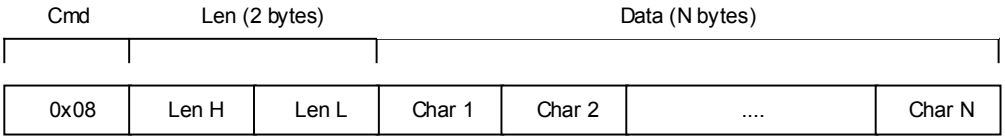
First/Next (1 byte):

- 1-Start a new file search from the beginning and find the first file.
  0-Continue search previously started and find next file.

If the File Find command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. The OK response contains the name of the file found, see description of OK response for more details.

3.15 – File Get Size command

The File Get Size command allows to get the size of a file from the internal memory.
The general format of File Get Size command is composed as follows:



Cmd (1 byte):

- 0x08 – File Get Size

Len (2 bytes):

- Number of characters in the file name (N≤16).

Char 1…Char N (N bytes):

- Sequence of N characters of the file name (no terminating char).

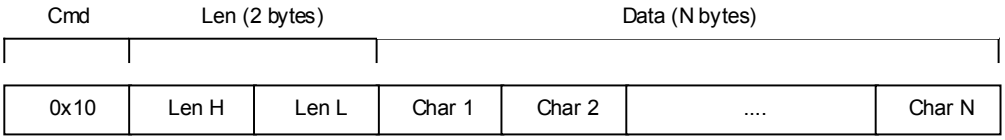
Here is shown an example of delete of file called “f1.wav”:

0x08	0x00	0x06	‘f’	‘1’	‘.’	‘w’	‘a’	‘v’

If the File Get Size command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. The OK response contains the size in bytes of the specificed file, see description of OK response for more details.

3.16 - Play command

The Play command starts playing an audio file that was previously written to the internal memory. The Play command for an audio file is accepted only if no other audio files are currently playing.
The general format of Play command is composed as follows:



Cmd (1 byte):

- 0x10 – Play

Len (2 bytes):

- Number of characters in the audio file name (N≤16).

Char 1…Char N (N bytes):

- Sequence of N characters of the audio file name (no terminating char).

Here is shown an example of playing the file "f1.wav":

0x10	0x00	0x06	' f '	' 1 '	' . '	'w '	' a '	' v '

If the Play command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. In both cases the response doesn't contain any data so the Len field in the response is zero.
In case of an OK response the Play continues until the wave file playback is completed and during this time the module is in Busy state. During Busy after a Play command the module can accept only the Stop, Volume and Check Busy commands.
The Busy state is signaled by the /BUSY line but can also be checked via serial port by sending the Check Busy command.

3.17 - Stop command

The play of an audio file can be stopped sending the Stop command.
The general format of Stop command is composed as follows:

Cmd	Len (2 bytes)	
0x11	0x00	0x00

Cmd (1 byte):

- 0x11 – Stop

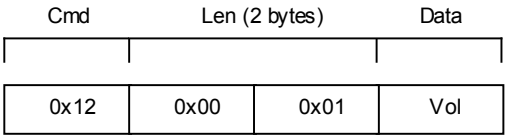
Len (2 bytes):

- Len is 0x0000 because no data bytes are present in the command.

If the Stop command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. In both cases the response doesn't contain any data so the Len field in the response is zero.

3.18 - Volume command

The Volume command allows to set the volume level for audio reproduction. The volume level is expressed in the range 0-100.
The general format of Volume command is composed as follows:



Cmd (1 byte):
- 0x12 – Volume

Len (2 bytes):
- Len is 0x0001 because there is only 1 data byte for the volume value.

Vol (1 byte):
- Volume level in the range 0-100 (default is 50).

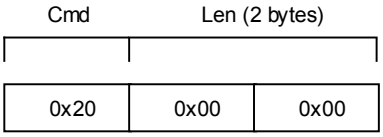
Here is shown an example of setting volume to 70%:

0x12	0x00	0x01	0x46

If the Volume command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. In both cases the response doesn't contain any data so the Len field in the response is zero.

3.19 - Reset command

The Reset command to reset the module by sending a serial command instead of driving the /RESET pin. It can be usefull for example when serial communication is lost in order to restart the system.
The general format of Reset command is composed as follows:



Cmd (1 byte):
- 0x20 – Reset

Len (2 bytes):
- Len is 0x0000 because no data bytes are present in the command.

There is no response to this command because the module is immediately reset.

3.20 – Check Busy command

The Check Busy command allows to check if the execution of the previous command is completed and can be used as an alternative to check the /BUSY line of the module.
The general format of Check Busy command is composed as follows:

Cmd	Len (2 bytes)	
0x21	0x00	0x00

Cmd (1 byte):

- 0x21 – Check Busy

Len (2 bytes):

- Len is 0x0001 because there is only 1 data byte for the Busy state.

If the Check Busy command has been accepted the module replies with an OK response, otherwise if not accepted it replies with an ERR response. The OK response contains the status of the module, see description of OK response for more details.

3.21 - Module ID response

Each module has a unique ID that permits to recognize the module. The module's ID is sent by the module to the host controller during the power-up sequence..
The general format of Module ID response is composed as follows:

Res	Len (2 bytes)		Data (2 bytes)	
0x81	0x00	0x04	ID 1	ID 2

Res (1 byte):

- 0x81 – Module's ID

Len (2 bytes):

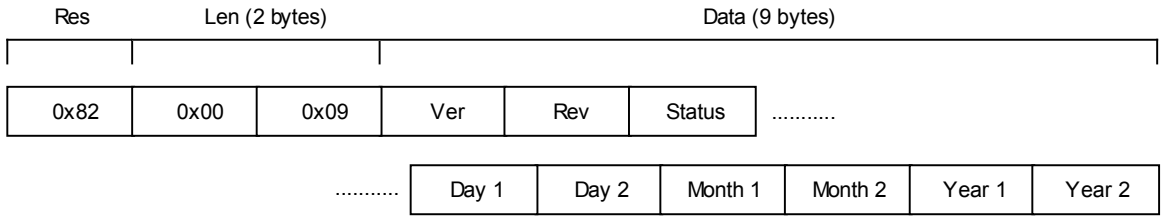
- Len is 0x0002 because the module's ID is 2 bytes long.

ID 1…ID 2 (2 bytes):

- Sequence of 2 module's ID bytes.

3.22 - Module Firmware Version response

The module's firmware version and release date are sent by the module to the host controller during the power-up sequence.
The general format of Module Firmware Version response is composed as follows:



Res (1 byte):

- 0x82 – Module's Firmware Version

Len (2 bytes):

- Len is 0x0009 because there are 3 bytes for the firmware version and 6 bytes for the date.

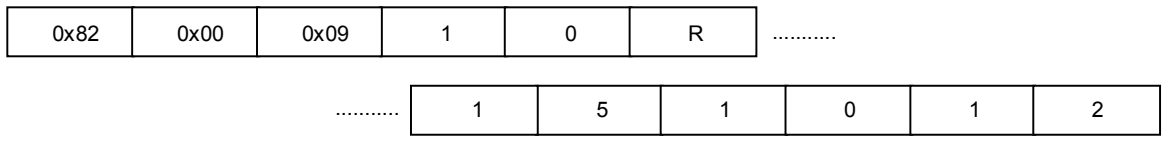
Ver...Status (3 bytes):

- The firmware version is composed by one Version character, one Revision character and one Status character. Each character is expresses in ASCII format, so for example the firmware version 1.0R will be expressed by the characters '10R' in the response.

Day 1...Year 2 (6 bytes):

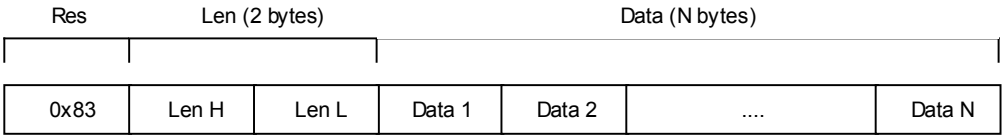
- The firmware release date is two bytes for the day, two bytes for the month and two bytes for the year. For example the date 15/10/09 is expressed by characters '151009' in the response.

Here is shown an example of firmware version 1.0R (15/10/12):



3.23 - OK response

The OK response is sent from the module when a received command is accepted.
The general format of OK response is composed as follows:



Res (1 byte):
- 0x83 – OK response.

Len (2 bytes):
- Number of data bytes to reply to last received command.

Data 1…Data N (N bytes):
- Sequence of N data bytes to reply to last received command.

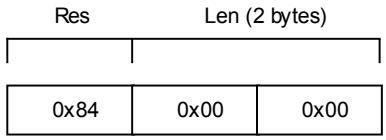
The following table shows the OK response data bytes returned for each command received:

Command received	OK response Len	OK response data
File Open File Write File Close File Delete Format Play Stop Volume	0x0000	None.
File Read	Number of data bytes required with File Read command	Sequence of N data bytes read from file.
File Find	Number of characters in the file name, included terminating char (N≤17).	Sequence of N characters of the file name, included terminating char.
File Get Size	0x0004	File size in bytes (first byte MSB – last byte LSB).
Check Busy	0x0001	0x01 – Module Busy 0x00 – Module Idle

Table 6 – OK response Len and Data after each command.

3.24 - ERR response

The OK response is sent from the module when a received command is not accepted.
The general format of ERR response is composed as follows:



Res (1 byte):

- 0x84 – ERR response

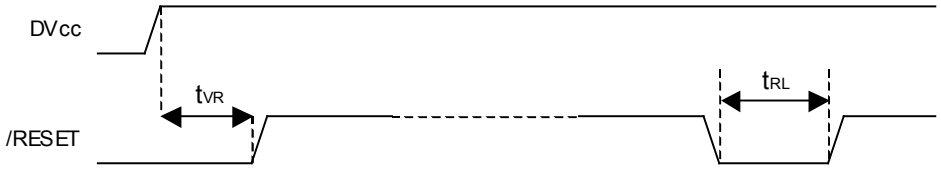
Len (2 bytes):

- Len is 0x0000 because no data bytes are present in the response.

4 – Timing Specifications

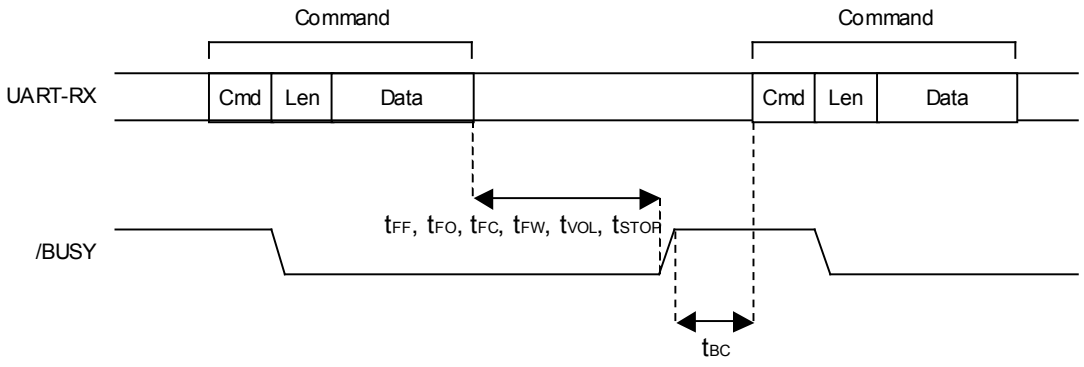
4.1 – Reset Timing Specifications

Symbol	Parameter	Min	Max
t_{VR}	Minimum time between DV_{CC} and /RESET high	1 ms	-
t_{RL}	Minimum /RESET low time	10 ms	-



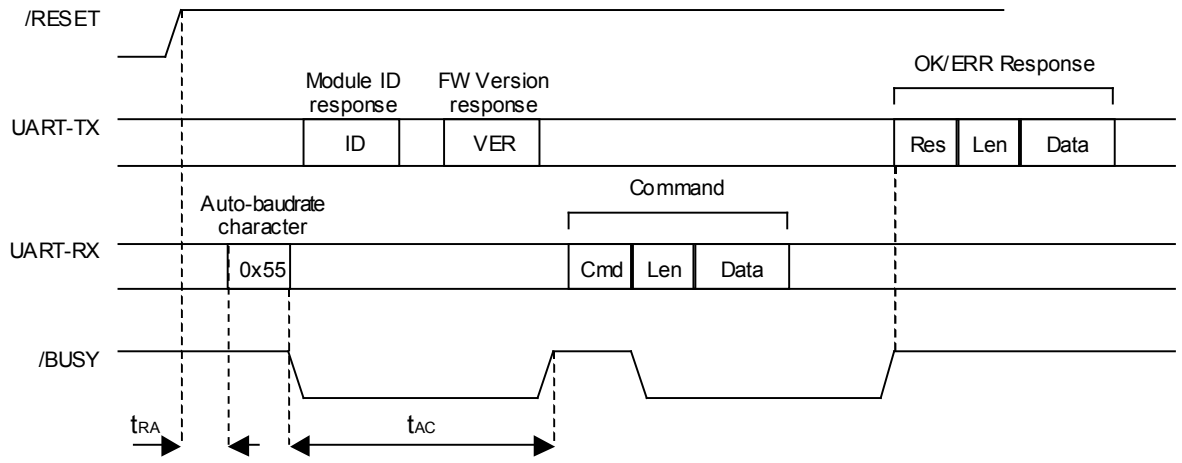
4.2 – Commands Timing Specifications

Symbol	Parameter	Min	Max
t_{FF}	Maximum File Format command time (/BUSY low)	-	65 sec
t_{FO}	Maximum File Open command time (/BUSY low)	-	40 µs
t_{FC}	Maximum File Close command time (/BUSY low)	-	25 µs
t_{FW}	Maximum File Write command time (/BUSY low)	-	2.5 ms
t_{STOP}	Maximum Stop command time (/BUSY low)	-	250 ms
t_{VOL}	Maximum Volume command time (/BUSY low)	-	25 µs
t_{BC}	Minimum time between /BUSY high and next command	200 µs	-

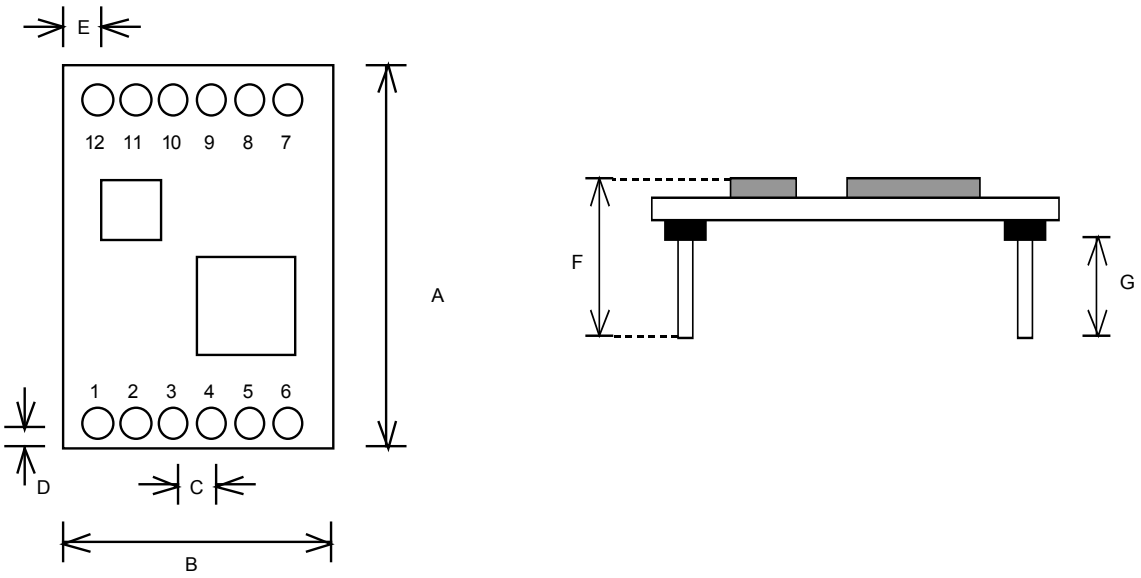


4.3 – Power-up Timing Specifications

Symbol	Parameter	Min	Max
t_{RA}	Minimum time between /RESET high and auto-baudrate character	600 ms	-
t_{AC}	Maximum time between auto-baudrate character and module ready to receive commands (1200 bps worst case)	-	150 ms
	Baudrate values	1200 bps	115200 bps



Packaging Informations



Dim	Millimeters	Inches
A	28.0	1.100
B	20.0	0.787
C	2.54	0.100
D	1.40	0.550
E	3.53	0.139
F	12.0	0.472
G	6.0	0.236

Figure 5 – Package Dimensions