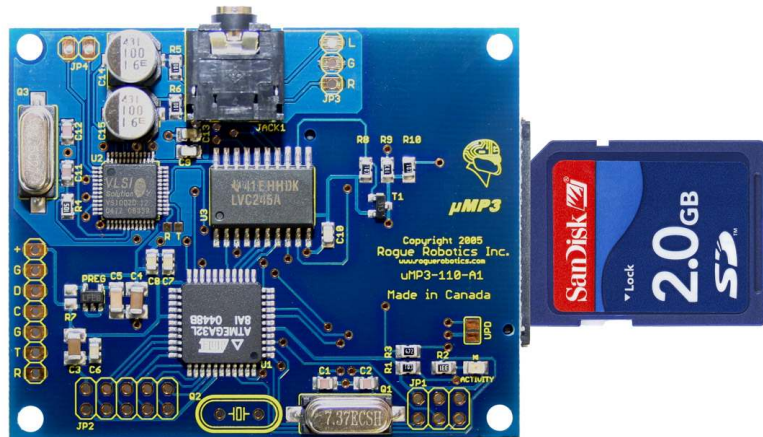


## µMP3 Playback Module Data Sheet



Version 1.00  
March 20<sup>th</sup>, 2005



**Contents**

**Version Information .....5**  
    Hardware Version .....5  
    Firmware Version.....5  
    Document Version .....5  
**General Description .....6**  
**Features .....7**  
**Specifications and Standards .....9**  
    Electrical .....9  
        Absolute Maximum Ratings .....9  
        DC Characteristics.....9  
    Temperature ..... 10  
    Communications ..... 11  
    File System ..... 11  
**Electrical Connections.....12**  
**Mechanical Drawing.....13**  
**Updating the Firmware .....14**  
**Digital Control of the μMP3 .....15**  
    8 Switch Mode ..... 15  
    7 Bit Parallel Mode..... 17  
    Hardware Busy Indicator..... 19  
**Communications Protocol.....20**  
    Description.....20  
    Command Format.....21  
        Command Listing Format .....21  
    General Commands.....22  
        Version/Serial Number.....22  
    Play Control Commands .....23  
        Play File.....23  
        Stop Playback.....24  
        Pause Playback.....24  
        Play Status .....25



Playback Information .....	26
Play Tone .....	27
Playback Speed Change .....	28
Settings.....	29
File Control Commands .....	31
Close File .....	31
Free Handle.....	31
Open File.....	32
Read File .....	33
Write File .....	34
Information .....	35
Make Directory .....	36
Erase File .....	36
Query Volume.....	37
List Directory .....	38
Device Information.....	39
Example Session .....	40
Error Responses.....	41
<b>Notices .....</b>	<b>43</b>
<b>Copyright Information.....</b>	<b>43</b>
<b>Trademarks.....</b>	<b>43</b>
<b>Life Support Policy.....</b>	<b>43</b>
<b>Contact and Support.....</b>	<b>43</b>

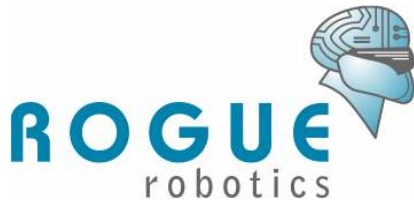


**Figures**

Figure 1 - μMP3 Electrical Connections .....12  
Figure 2 - Mechanical Drawing ..... 13

**Tables**

Table 1 - Absolute Maximum Ratings .....9  
Table 2 - DC Characteristics .....9  
Table 3 - 8 Switch Mode Pin Assignments..... 16  
Table 4 - 8 Switch Mode File Numbering ..... 16  
Table 5 - 8 Switch Mode File Number Offset Example ..... 17  
Table 6 - 7 Bit Parallel Mode Pin Assignments ..... 18  
Table 7 - 7 Bit Parallel Mode File Numbering ..... 19  
Table 8 - Settings and Values .....30  
Table 9 - Error Codes ..... 42



## **Version Information**

### **Hardware Version**

Current Version: **UMP3-110-A1**

Jan 22/2005 – UMP3-110-A1

- Initial Release

### **Firmware Version**

Current Version: **110.10**

Jan 22/2005 – 110.10

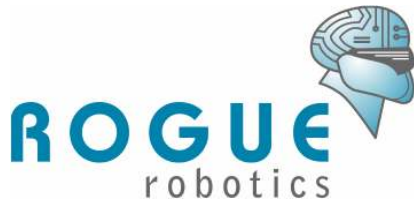
- Initial Release

### **Document Version**

Current Version: **1.00**

March 20/2005 – 1.00

- Initial Release



## **General Description**

The μMP3 (μMP3 or "micro MP3") module allows designers to integrate high quality MP3/PCM/WAV/IMA ADPCM playback into designs with an easy to use TTL serial protocol.

Sound files such as voice prompts, sound effects, and music can be played using the μMP3. MP3 playback is CBR (Constant Bit Rate) up to 48 KHz at 192 Kbps, or VBR (Variable Bit Rate) up to 48 KHz at 320 Kbps peak. MP3 files are stored on SD or MMC cards, in a format readable by your PC. PCM (WAV) files and IMA ADPCM files can also be played.

You can use the μMP3 for your data logging applications as well as playback of MP3 files. The μMP3 can create files and store data to MMC and SD cards which can then be removed and read from any standard MMC/SD card reader on a PC. The μMP3 will read SD or MMC cards in either FAT16 or FAT32 formats in any size commercially available.

Each module has a 1/8" stereo headphone jack for headsets (16 Ohms+) or for line-in connections. The μMP3 also has a multi-function 10-pin connector that allows direct operation of the module using simple logic triggers.

The firmware on the μMP3 is updateable via a bootloader, so you can take advantage of any new features or fixes that may become available for the μMP3 from the [Rogue Robotics](http://www.roguerobotics.com) website.

## **Features**

- Audio Playback
  - MP3
    - MPEG audio layer 3 decoder (ISO11172-3)
    - Supports MPEG 1 & 2, and 2.5 extensions, all their sample rates and bit rates, in mono and stereo
    - CBR 48 KHz at 192 Kbps maximum
    - VBR 48 KHz at 320 Kbps peak maximum
  - Configurable bass enhancement
  - MP3+V file support (vlsi.fi)
  - PCM
    - Supports rates up to 44100Hz stereo at 8 or 16 bits
  - IMA ADPCM
    - Supports rates up to 44100Hz (mono only)
  - Playback speed can be altered
  - Tones
- Audio output characteristics
  - 1/8" jack for audio output
  - 0.100" header
  - 1.8 Vpp output
  - 87 dB S/N Ratio
- SD card and MMC card compatible
  - SD card lock respected
- FAT16 or FAT32 formats supported
  - Long filename support
  - Up to 2 Terabytes accessible through FAT32
- Serial control interface
  - 5 Volt serial UART connection
  - Simple command interface
- 8 bit digital control
  - 10 pin DIL 0.100" header (8 bit data, 2 power)
  - 8 individual file control, falling edge triggered
  - 128 file control, 1 clock signal, falling edge triggered
  - Multiple banks
  - Uninterruptible setting available
  - Internal pull-ups provided
- Playback indicator
  - Selectable playback indicator
  - 30 mA maximum at 5 Volts
- Read and Write capability



- 512 byte block maximum transfer size
- Up to 4 files open at any time for read or write
- Simultaneous read/write access while playing MP3 files
- 5 Volt operation
  - 200 mA maximum (during card write)
  - 15 mA typical (during MP3 playback)
  - Onboard 3.3 Volt LDO Voltage regulator for card
- Bootloader for firmware updates



## Specifications and Standards

### Electrical

#### *Absolute Maximum Ratings*

Operating Temperature	-40 C to +100 C
Storage Temperature	-55 C to +140 C
Maximum Operating Voltage	+6.0V
Voltage on any input with respect to Ground	-1.0 V to +5.5V

**Table 1 - Absolute Maximum Ratings**

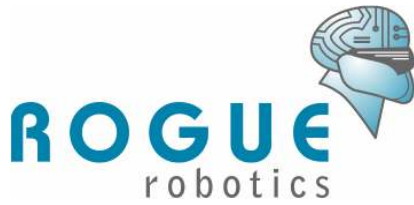
**NOTICE:**

Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to components of the module. This is a stress rating only and functional operation of the module at these or any other condition beyond those indicated is not implied. Exposure to absolute maximum rating conditions for extended periods may affect component and module reliability.

#### *DC Characteristics*

Parameter	Symbol	Min	Typ	Max
Power Supply Voltage	V+		5.0 V	5.5 V
Power Supply Current			30 mA	200 mA
Input Low Voltage (including Serial Rx)		-0.5 V		1.0V
Input High Voltage (including Serial Rx)		3.0 V		5.5 V
Output Low Voltage (Hardware Busy Indicator, Serial Tx)				0.7 V
Output High Voltage (Hardware Busy Indicator, Serial Tx)		4.0 V		
Input pin pull-up resistance		20 kOhms		100 kOhms

**Table 2 - DC Characteristics**



## **Temperature**

This version of the uMP3 is rated for commercial temperature ranges.

*Commercial Temperature Operating Range: 0°C to +70°C*

Industrial temperature ranges are special order devices.

*Industrial Temperature Operating Range: -40°C to +85°C*

## **Communications**

### *TTL Serial Port:*

- 9600, 19200, 38400, 57600, or 115200 bps (9600 bps default)
- 8 bits
- no parity
- 1 stop bit
- no echo

## **File System**

FAT16 and FAT32 only. (FAT12 is not supported)

File system MUST be formatted on a PC initially. The device does not format cards.

Long filenames are supported. Filenames created by the module are 8.3 filenames only (8 character name, 3 letter extension).

SD card locks are respected, so locked cards will not be writeable. Encryption features of SD cards are not implemented.

The number of files that can be created is limited by the file system (FAT16, FAT32).

Maximum of 4 files can be open at any time using file handles.

Maximum of 512 bytes per read/write command.

Files must be closed before removing cards for data reliability. If the Activity LED is on, you should not remove a card. When the LED is off, and you have closed all files, you can safely remove the MMC or SD card.

# Electrical Connections

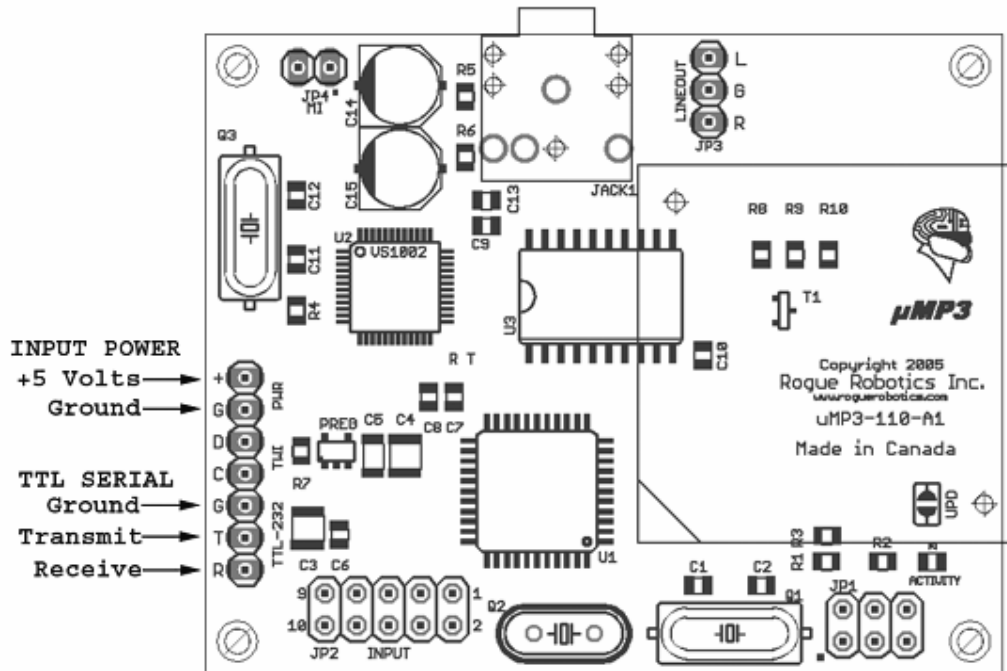
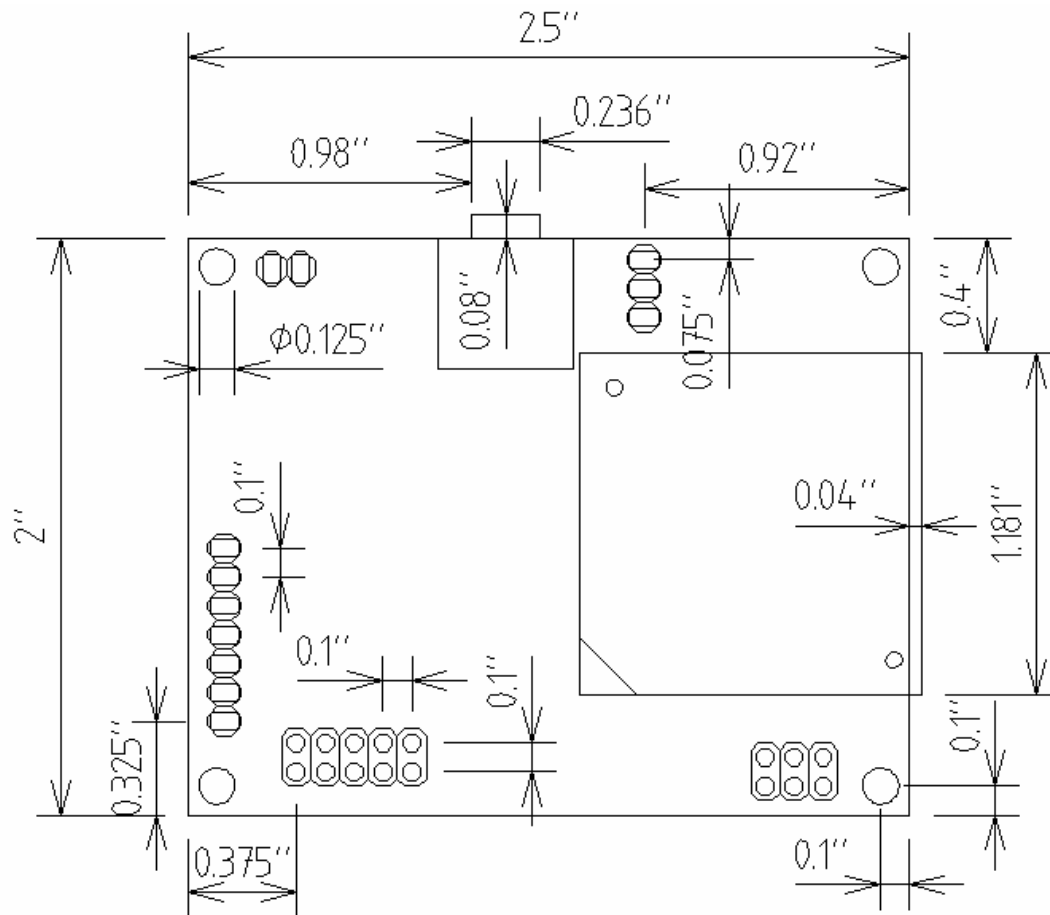


Figure 1 - μMP3 Electrical Connections

## Mechanical Drawing

- Units in inches.
- The MMC/SD connector is mounted on the back.
- DXF files or IGES files available on request.
- All dimensions are +/- 0.01"



**Figure 2 - Mechanical Drawing**

## **Updating the Firmware**

If there is a new firmware update for the μMP3, it may be downloaded from <http://www.roguerobotics.com/>. Once downloaded, it can be sent to the μMP3 using the “update” program. Both the firmware update and the update program will be packaged together in a Zip compressed file.

You will need to connect the μMP3 to a PC serial port through a TTL Level converter, such as a MAX232. **THE μMP3 WILL BE DAMAGED IF CONNECTED DIRECTLY TO A PC SERIAL PORT!**

To put the μMP3 into bootloader mode and download the firmware:

- Disconnect power to the μMP3, and remove any MMC/SD card from the μMP3.
- Use a small flat-blade screwdriver to bridge the “UPD” jumper, and continue to hold the screwdriver in place.
- Connect the power to μMP3. The Activity LED will stay illuminated (longer than 2 seconds)
- Start the update program from the command line (Start -> Run -> “cmd.exe” or “command.exe”):
  - “update ump1-11010.rfw -COM1” (if your serial port is something other than COM1, use “update ump1-11010.rfw -comN”, where comN is your com port)

The update program will show the progress and you will see the μMP3 Activity LED blink as the firmware is updated. Once complete, the μMP3 will reset and start normally.

When the μMP3 is put into bootloader mode, all settings for the **Settings** command are reset to default values (this is so that you can reset the μMP3 if an unknown value is put in any of the settings). You do not have to download anything to reset the values. Just simply put the μMP3 into bootloader mode (explained above), then remove the power to the μMP3. The values will be reset.

## Digital Control of the μMP3

The μMP3 has a multipurpose 8 bit interface for controlling the module without using the serial interface. This allows for very simplistic interfacing to the μMP3.

There are two different styles for the Input interface:

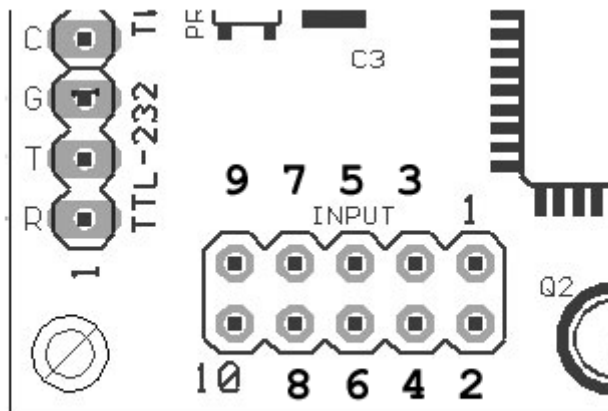
- 8 Switch Mode
- 7 Bit Parallel Mode

The files are played from the MMC/SD card only.

### 8 Switch Mode

The 8 Switch mode allows each of the 8 data bits of the Input interface to be used as a trigger for 8 different files.

Pin 1 plays a file named “/B0000.MP3”. Pin 2 plays a file named “/B0001.MP3” and so on.



Input Interface Pin	Description
1	Bit/Switch 0
2	Bit/Switch 1
3	Bit/Switch 2
4	Bit/Switch 3
5	Bit/Switch 4
6	Bit/Switch 5
7	Bit/Switch 6

8	Bit/Switch 7
9	Ground
10	V+ (5 Volts)

**Table 3 - 8 Switch Mode Pin Assignments**

To set the Input interface into 8 Switch Mode, you need to configure the μMP3 using the **Settings** “Input Style” command. The style value for 8 Switch Mode is “1”. Once the setting has been configured, it is stored in EEPROM, and does not have to be set again. You can reconfigure it at any time, or clear the settings to default values by putting the μMP3 into “update mode”.

Once in 8 Switch Mode, all pins of the Input interface are weakly pulled high internally. To trigger a switch, the switch must be pulled to ground. Once the switch is pulled to ground, the associated file with the switch is played.

<b>Input Interface Pin</b>	<b>Bit/Switch</b>	<b>Associated Filename</b>
1	0	/B0000.MP3
2	1	/B0001.MP3
3	2	/B0002.MP3
4	3	/B0003.MP3
5	4	/B0004.MP3
6	5	/B0005.MP3
7	6	/B0006.MP3
8	7	/B0007.MP3

**Table 4 - 8 Switch Mode File Numbering**

All files are stored in the root directory of the MMC or SD card.

The input pins are not debounced, so a capacitor/resistor debounce circuit may be needed if the input pins are connected directly to a switch.

Files are played when a switch is pulled low. Playback will be interrupted if another switch is pulled low before the end of playback. To make playback uninterrupted, you can use the **Settings** “Uninterrupted playback from Input” command to ensure each file is played to completion before a new switch can trigger playback.

You can use the **Settings** “Input File Number Offset” command to introduce an offset to the filenames associated with each switch. This allows you to create “banks” of files to associate with the switches. For example, if the “Input File Number Offset” value is set to 200, then the associated files are as follows:



Input Interface Pin	Bit/Switch	Associated Filename
1	0	/B0200.MP3
2	1	/B0201.MP3
3	2	/B0202.MP3
4	3	/B0203.MP3
5	4	/B0204.MP3
6	5	/B0205.MP3
7	6	/B0206.MP3
8	7	/B0207.MP3

**Table 5 - 8 Switch Mode File Number Offset Example**

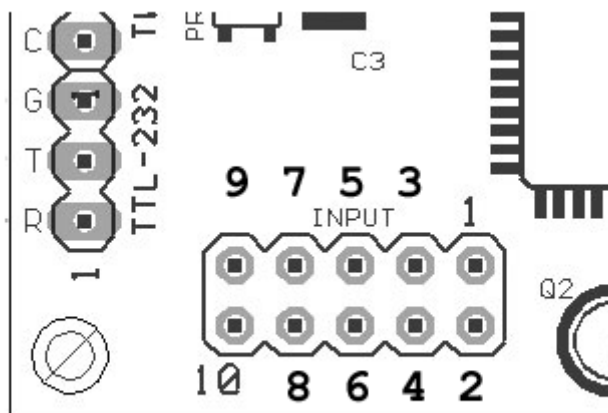
The offset can be set at any time using the serial interface.

**Note:** If the SD/MMC card has not been initialized before a switch is triggered, there can be a noticeable delay (during the FAT file system initialization). This will only happen once if the card has not been initialized.

## 7 Bit Parallel Mode

The 7 Bit Parallel mode provides a 7 bit file selection interface. Pins 1 through 7 (bits 0 through 6) will have the file number applied, and Pin 8 (bit 7) will trigger playback.

In this mode, a file number of 0 plays a file named “/N0000.MP3”. A file number of 1 plays a file named “/N0001.MP3” and so on.



<b>Input Interface Pin</b>	<b>Description</b>
1	Bit 0 (LSB)
2	Bit 1
3	Bit 2
4	Bit 3
5	Bit 4
6	Bit 5
7	Bit 6 (MSB)
8	Trigger (Active Low)
9	Ground
10	V+ (5 Volts)

**Table 6 - 7 Bit Parallel Mode Pin Assignments**

To set the Input interface into 7 Bit Parallel Mode, you need to configure the μMP3 using the **Settings** “Input Style” command. The style value for 7 Bit Parallel Mode is “2”. Once the setting has been configured, it is stored in EEPROM, and does not have to be set again. You can reconfigure it at any time, or clear the settings to default values by putting the μMP3 into “update mode”.

Once in 7 Bit Parallel Mode, all pins of the Input interface are weakly pulled high internally. The lower 7 pins of the Input interface (and pin 8, if desired) can be connected to a port on microcontroller. The lower 7 pins are read as non-inverted. That is, if you want to play file number “7”, the Input pins would be “0000111”. The trigger on pin 8 will start playback when its value goes from high to low (falling edge triggered).

The trigger pin is not debounced, so a capacitor/resistor debounce circuit may be needed if the trigger pin is connected directly to a switch.

<b>Value on Pins 1 through 7</b>	<b>Binary Value Bit 6 ... Bit 0</b>	<b>Hexadecimal Value</b>	<b>Associated Filename</b>
0	000 0000	00	/N0000.MP3
1	000 0001	01	/N0001.MP3
2	000 0010	02	/N0002.MP3
...	...	...	...
42	010 1010	2A	/N0042.MP3
43	010 1011	2B	/N0043.MP3
...	...	...	...
126	111 1110	7E	/N0126.MP3
127	111 1111	7F	/N0127.MP3

**Table 7 - 7 Bit Parallel Mode File Numbering**

Just the same as with the 8 Switch Mode, you can use the **Settings** “Input File Number Offset” command to introduce an offset to the filenames. This allows you to create “banks” of associated playback files.

## **Hardware Busy Indicator**

The μMP3 can indicate when it is busy playing an MP3 file. You must use the **Settings** “Hardware Busy Indicator” command to enable the indicator. Once set, the pin labeled “D” is configured as an active output. When a file is playing, the indicator is high (5 Volts). When the system is idle (no file playing), the indicator is low (0 Volts). If a file is paused, the indicator will remain high. If the playback is stopped, the indicator will go low.

## **Communications Protocol**

### **Description**

The Protocol for the μMP3 employs a simple but robust asynchronous serial control protocol. A command prompt ">" ("greater than" symbol, ASCII 62, HEX 0x3E) indicates that the μMP3 is ready to accept a command. A command can be sent, a response will be returned, and the command prompt will be sent again.

#### **Example**

```
V{cr}  
110.10 SN:UMP1-0000-1234>
```

If an error occurs while processing a command, an error is returned in the format `Enn`, listed in Table 1.

#### **Example:**

```
>PC F /MP3S/FIRSTMP3.MP3{cr}  
EF2>
```

### ***Important***

After a card is inserted, the card must be scanned the first time for file system information. For FAT16, this can take up to 10 seconds. On FAT32 cards, this never takes more than 2 seconds (although, if the card has just been formatted, the very first time it is used on the μMP3, it can take up to 20 seconds to initialize the file system). Always wait for the command prompt (">") before sending any commands.

## **Command Format**

*CC{sp}Parameter1{sp}Parameter2{sp}...{cr}*

Where:

“CC” is the command character(s) (commands listed below)

{sp} is a single space character (ASCII 32, HEX 0x20) [this space is necessary]

Parameter1, Parameter2, ... are parameters associated with the command

{cr} is a carriage return character (ASCII 13, HEX 0x0d)

## ***Command Listing Format***

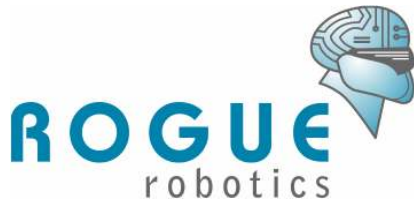
***CC Parameter1 [Parameter2 [Parameter3]]...***

“**CC**” is the command character(s).

“**Parameter1**” is the first parameter for the command. Often, this is a sub-command.

“**Parameter2**” is the second parameter for the command. If it is listed inside of square brackets [ ] then the parameter is optional.

Any parameters listed inside of square brackets [ ] are optional. Most commands that have optional parameters will require the previous parameter to be given.



## **General Commands**

### ***Version/Serial Number***

**V**

#### **Description**

The ***Version/Serial Number*** command will return the current firmware version and the μMP3 serial number. The format is `VVV.MM SN:UMP1-NNNN-NNNN`.

#### **Example**

```
>V{cr}
110.10 SN:UMM1-0000-0001>
```



## Play Control Commands

### ***Play File***

#### **PC F *path***

- *path* is the absolute path to the file. A properly formatted path must begin with a "/" and be absolute (that is, it must begin at the root directory). Sub-directories are separated with "/" (forward slash).
  - e.g. "/VOICE/F01/F42.MP3"

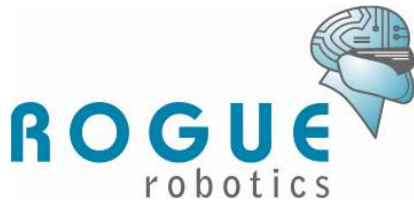
#### **Description**

This plays file from the given path. Note: the file MUST be a valid CBR or VBR MP3, MP3+V, or RIFF formatted PCM/IMA ADPCM file.

If you send another ***Play File*** command while a file is playing, it will stop the current file from playing, then immediately start the new file.

#### **Example**

```
>PC F /MP3S/FIRSTMP3.MP3{cr}
>
```



## ***Stop Playback***

**PC S**

### **Description**

Playback is stopped.

### **Example**

```
>PC S{cr}
>
```

## ***Pause Playback***

**PC P**

### **Description**

Playback is paused. If another ***Pause Playback*** command is issued, playback is resumed. The ***Play Status*** and ***Playback Information*** commands can be issued while playback is paused.

### **Example**

```
>PC P{cr}
>
```





## ***Play Status***

### **PC Z**

#### **Description**

This shows the current status of the μMP3 playback. You can use this command to monitor when the current file has finished playing.

The first item returned is the playback state. “S” indicates that the μMP3 is stopped, “P” indicates that it is playing, and “D” indicates that it is paused.

The second item is the current position in the file, given in seconds.

The third item is the current loop number. If this value is 0, the file is looped infinitely. To change the number of times a file is played, see the Loop Count in the **Settings** command.

#### **Example**

```
>PC Z {cr}
P 6 1>
```



## ***Playback Information***

### **PC I**

#### **Description**

This displays the information from the file currently playing. The current position in the file (in seconds), the sample rate, and the bit rate (varies if the file is a Variable Bit Rate file) are returned.

The first item returned is the current position in the file, given in seconds.

The second item (separated from the first item by a space) is the sample rate. Typical MP3 sample rates are: 32 KHz, 44 KHz, or 48 KHz (only the numerical value is returned).

The third item is the bit rate. Typical MP3 bit rates are 96kbps, 128kbps, 160 kbps, and 192kbps. This value will change if the file currently playing is a VBR file (only the numerical value is returned).

The fourth and final item indicates whether the file is a stereo or mono file. "S" indicates a stereo file, while "M" indicates a mono file.

#### **Example**

```
>PC I{cr}  
92 44 128 S>
```

## ***Play Tone***

### **PC T [tone]**

- optional *tone* is a value between 0 and 255
  - Tone Frequency = Base Frequency \* d/128
  - tone value = base frequency value + divider value (d)

<b>Hex Value</b>	<b>Decimal Value</b>	<b>Base Frequency</b>
0x00	0	44100 Hz
0x20	32	48000 Hz
0x40	64	32000 Hz
0x60	96	22050 Hz
0x80	128	24000 Hz
0xA0	160	16000 Hz
0xC0	192	11025 Hz
0xE0	224	12000 Hz

- the frequency divider value d (1 through 31) will give a range of dividers from 1/128 through 31/128 (a divider value of 0, will produce no sound).
- E.g. Base Frequency of 48000 Hz, and a divider value of 16 gives a Tone Frequency of  $48000 * 16/128 = 6000$  Hz – the tone value used in the command would be  $32 + 16 = 48$

### **Description**

This command plays a tone derived from the parameter. The tone is played indefinitely until the μMP3 receives either the ***Play Tone*** command again with no parameter, or the ***Stop Playback*** command. If a file is being played while the ***Play Tone*** command is received, it will resume playing if the ***Play Tone*** command is received with no parameter.

### **Example**

```
>PC T 65{cr}
>
```



## ***Playback Speed Change***

### **PC X [*speed*]**

- optional *speed* is a value between 90 and 250
  - represents the fractional speed increment for playback (percentage). i.e. a value of 90 will play the file in 90% of the time it takes to play normally, and likewise, a value of 120 will play the file in 120% of the time it takes to play normally.

### **Description**

This command changes the playback speed. If a value less than 100 is given, the file will play faster (and thus the pitch will be higher), and a value greater than 100 will play the file slower (with a lower pitch). If the *speed* parameter is not received, then the playback will return to normal. NOTE: the playback speed will return to normal after every file played.

### **Example**

```
>PC X 95{cr}  
>
```

## Settings

Several settings on the μMP3 control its behavior. Each Setting is changed using the following format. The list of settings and their values are shown in Table 1.

### ***ST s [newvalue]***

- *s* is the setting name, shown in the table below.
- *newvalue* is the new value to assign to the setting. If *newvalue* is not provided, then the current value for the setting is returned.

Setting	Name	Value	Description (default in bold)
D	Serial Bit Rate	<b>0</b>	<b>9600 bps</b>
		1	19200 bps
		2	38400 bps
		3	57600 bps
		4	115200 bps
V	Playback Volume	<b>0</b> to 254	Volume for playback. 0 = loudest 254 = Off (individual channel volume control set by using two values LEFT RIGHT e.g. "ST V 20 254" – mutes right channel)
O	Loop Count	0 to 254	This is number of times that every file is played. 0 = Infinite loops <b>1</b> = Play one time only 2 – 254 = number of times to play.
B	Bass Boost	<b>0</b> to 254	Audio enhanced Bass and Treble. <b>Value = Amplitude value * 32 + Low Limit Frequency value</b> Amplitude Enhancement Value = 0 to 15 (in 1dB steps)

			Lower Limit Frequency Value = 2 to 15 (in 10 Hz steps) e.g. "ST B 74" – 2 dB boost with Lower Limit Frequency of 100 Hz ( $2 * 32 + 10$ )
S	Input Style	<b>0</b>	No input style monitored
		1	8 switch mode – 8 individual switches corresponding to 8 files Falling edge triggered
		2	7 bit parallel mode – The lower 7 bits represent the file to play (0 through 127), and the MSB triggers the file Falling edge triggered (data is non-inverted)
T	Write Time-out	<b>0 to 254</b>	Time in 10ms increments (eg. 20 = 200ms) <b>0 = No time-out (waits indefinitely)</b>
N	Input File Number Offset	<b>0 to 9999</b>	For both Input Styles, this setting allows for an offset for which files are used for playback
U	Uninterrupted playback from Input	<b>0 or 1</b>	If set to 1, files started using the Input pins will not be interrupted by another input signal until playback is complete.
H	Hardware Busy Indicator	<b>0 or 1</b>	If set to 1, the "D" pin becomes a playback indicator.

**Table 8 - Settings and Values**

**Example**

```
>ST B{cr}
0>ST B 1{cr}
>ST B{cr}
1>
```

## File Control Commands

### ***Close File***

#### **FC C *fh***

- *fh* is a file handle (1 - 4)

#### **Description**

This closes an open file.

#### **Example**

```
>FC C 1{cr}
>
```

### ***Free Handle***

#### **FC F**

- *dev* is the device number (1 – 2)

#### **Description**

Returns the next available free handle (1 through 4). If no handles are available, 0 is returned. *For simple applications, this is not necessary to implement.*

#### **Example**

```
>FC F{cr}
1>
```

## ***Open File***

### ***FC O fh mode path***

- *fh* is a file handle (1 - 4)
- *mode* is the open mode for the file.
  - 'R' – Open in read mode. Data can be read non-sequentially (randomly). **The filename in the path must exist.**
  - 'W' – Open in write mode. This opens a new file for writing. Data is written to the file sequentially. **The filename in the path must NOT exist.**
  - 'A' – Open in append mode. This opens a new or existing file for writing. Data is written to the file sequentially. **If the filename in the path does not exist, it will be created.**
- *path* is the absolute path to the file. A properly formatted path must begin with a "/" and be absolute (that is, it must begin at the root directory). Sub-directories are separated with "/" (forward slash).
  - e.g. "/LOGS/2004/JANUARY/JAN3.LOG"

### **Description**

This will open a file on the card in one of three modes.

### **Example**

```
>FC O 1 R /LOGS/2004/JANUARY/JAN03.LOG{cr}  
>
```

### **Important**

All files created with the µMP3 will have their modification and creation dates set to 01/01/2004 00:00:00. The date is not updated.





## ***Read File***

### **FC R *fh* [*bytes* [*address*]]**

- *fh* is a file handle (1 - 4)
- *bytes* is the number of bytes to read. This parameter is optional. If it is not specified, up to 512 bytes will be returned.
- *address* is the address at which to start reading. The *bytes* parameter MUST be given for the address parameter to be used.

### **Description**

You can read up to 512 bytes at a time using the ***Read*** command. If the *bytes* parameter is larger than the number of bytes remaining in the file, then only the remaining bytes are returned. Use the ***Info*** command to find the current position in and the size of the file. If the ***Read*** command is successful, a single *{sp}* character is sent, followed by the data. If an error occurs, the first character returned is “E”, followed by an error code (see Table 2). Data is sent verbatim (no escape characters) from the card.

### **Example**

(the data file contains only two lines of information)

```
>FC R 1{cr}
 13:22:02 ADC1=4.9V
13:22:32 ADC1=4.9V
>FC R 1 18 0{cr}
 13:22:02 ADC1=4.9V>
```



## **Write File**

### **FC W *fh bytes***

- *fh* is a file handle (1 - 4)
- *bytes* is the number of bytes to be written. You must send this number of bytes to return to the command prompt.

### **Description**

You can write up to 512 bytes at a time with the **Write** command. If the bytes parameter is omitted, then 512 bytes will be expected on the incoming serial stream. Data is accepted and written to the card directly (there is no escape sequence).

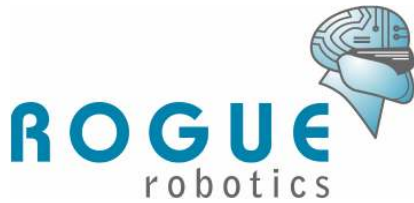
By default, there is no time-out for how long it takes to send the bytes to the µMP3. This means that the µMP3 will wait indefinitely for all the bytes to be sent (unless the power is removed, or a time-out value has been set using the **Settings** command).

If you assign a value to the write time-out setting using the **Settings** command, then the **Write** command will terminate, write the accepted bytes to the file, and return to the command prompt; no error will be returned.

If the file has been opened for append, the **Write** command will append all bytes to the end of the file.

### **Example**

```
>FC W 1 18{cr}  
13:22:02 ADC1=4.9V>
```



## ***Information***

### **FC I *fh***

- *fh* is a file handle (1 - 4)

### **Description**

The ***Information*** command returns the current file position and the current file size for a given file handle. The format is **position/filesize**. The two values are given in decimal format.

### **Example**

```
>FC I 1{cr}
19/37>
```

## ***Make Directory***

### **FC M *path***

- *path* is the absolute path to the directory to be created. A properly formatted path must begin with a "/" and be absolute (that is, it must begin at the root directory). Sub-directories are separated with "/" (forward slash).

### **Description**

The ***Make Directory*** command will create a directory.

### **Example**

To create a directory named "JANUARY" under the path "/LOGS/2004":

```
>FC M /LOGS/2004/JANUARY{cr}
>
```

## ***Erase File***

### **FC E *path***

- *path* is the absolute path to the file to be erased. A properly formatted path must begin with a "/" and be absolute (that is, it must begin at the root directory). Sub-directories are separated with "/" (forward slash).

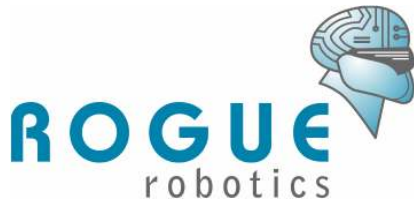
### **Description**

The ***Erase File*** command will erase a file.

### **Example**

To erase a file named "JAN03.LOG" under the path "/LOGS/2004/JANUARY":

```
>FC E /LOGS/2004/JANUARY/JAN03.LOG{cr}
>
```



## ***Query Volume***

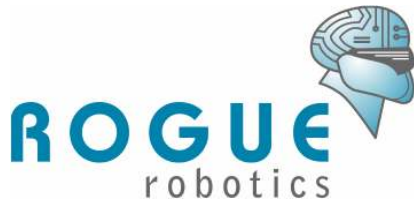
### **FC Q**

#### **Description**

The ***Query Volume*** command will return the free space and the total volume of the current memory card. The format is **freespace totalspace** (separated by a space). The two values are given in decimal format and in Kibibytes (i.e. 1024 bytes = 1 KiByte).

#### **Example**

```
>FC Q{cr}
51245 61525>
```



## List Directory

### FC L *path*

- *path* is the absolute path to the directory to be listed. A properly formatted path must begin with a "/" and be absolute (that is, it must begin at the root directory). Sub-directories are separated with "/" (forward slash).

### Description

The entire contents of the directory will be listed. This means that if the directory content is large, then the host must be prepared to deal with data. The format of the listing is:

D for "Directory" or number indicating the size of the file in bytes, then the filename terminated by a {cr}.

If the **List Directory** command is successful, a single {sp} character is sent, followed by the data. If an error occurs, the first character returned is "E", followed by an error code (see Table 2).

### Example

To list the contents under the path "/LOGS/2004":

```
>FC L /LOGS/2004{cr}  
  D JANUARY{cr}  
  D FEBRUARY{cr}  
  D MARCH{cr}  
 3429 YEARLY.TXT{cr}  
 1502 NOTES.TXT{cr}  
>
```



## ***Device Information***

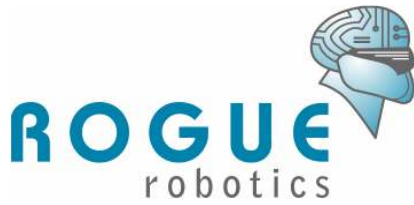
**FC D**

### **Description**

Lists all the devices, each size in KiBytes, and free space in KiBytes

### **Example**

```
>FC D{cr}
2 30150 23016>
```



## Example Session

This example session plays a file, checks the playback information, and opens a file for reading.

```
>PC F /FIRSTMP3.MP3{cr}
>PC I{cr}
92 44 128 S>O 1 R /LOGS/2004/JANUARY/JAN03.LOG{cr}
>R 1 18{cr}
13:22:02 ADC1=4.9V>C 1{cr}
>
```



## Error Responses

When a command produces an error, the response is **E<sub>nn</sub>**. The table of responses is given below.

Below is an example error response from an open command that attempts to open a file that does not exist.

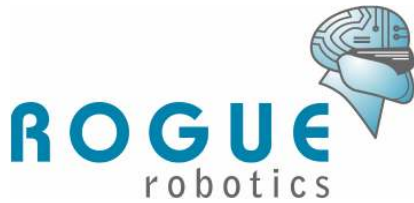
**Example:**

```
>PC F 2 /BAD/NOMP3.MP3{cr}
EF2>
```

Error Code	Description
E02	Buffer Overrun – Too many bytes were sent in the command. All command can be a maximum of 256 bytes (including the path).
E03	No Free Files – This is a response from the <b>Free File</b> command. There are no more open handles. You must close an open file handle before a new one can be opened.
E04	Unrecognized command.
E06	Command formatting error – this occurs if parameters are missing or invalid.
E07	End of file
E08	Card not inserted
E09	MMC/SD Reset failure
E0A	Card write protected
E0C	Device not available
E0F	Can not play from that device
EE6	Read-only file – a Read-Only file (file attributes) is trying to be opened for write or append.
EE7	Not a file – an invalid path.
EE8	Write Failure – There could be many reasons for this (damaged card, card removed WHILE writing, etc...)
EEA	No free space – There is no free space on the card.
EEB	File not open – The file handle specified has not been opened with the <b>Open</b> command.
EEC	Improper mode – A <b>Read</b> command was attempted while the file has been opened for writing, or vice-versa.
EED	Invalid <b>Open</b> mode – only 'R', 'W', and 'A' are acceptable open modes.
EF1	Handle in use – The specified handle is already being used.

<b>EF2</b>	File does not exist – The file in the path specified does not exist.
<b>EF4</b>	File already exists – A <b>Write</b> command was issued, and the file in the path already exists.
<b>EF5</b>	Path invalid – The path specified does not exist. Ensure that all directory names in the path exist.
<b>EF6</b>	Invalid handle – The handle specified is not valid.
<b>EFB</b>	Bad FSINFO Sector (FAT32 only)
<b>EFC</b>	Unsupported FAT version. Ensure the card is inserted correctly and that the card has been formatted to FAT16 or FAT32.
<b>EFD</b>	Unsupported Partition type
<b>EFE</b>	Bad Partition information
<b>EFF</b>	Unknown Error

**Table 9 - Error Codes**



## **Notices**

This document may contain technical inaccuracies or typographical errors. The information in this document may change without notice. Changes and improvements to the products may occur at any time.

This document is provided “as-is” and without warranty of any kind, either express or implied.

## **Copyright Information**

This document is copyright © 2005 Rogue Robotics Corporation Inc. All rights reserved.

Any person may view, copy, print and distribute this document or any portion of this document for informational purposes only as long as the copyright notice remains included.

## **Trademarks**

µMP3 is a trademark of Rogue Robotics Corporation Inc.

## **Life Support Policy**

Rogue Robotics Corporation Inc. products are not intended or authorized for use in any life support situation. These systems may include devices for sustaining life, surgical implant into or onto the body, or any other system whose failure to perform correctly could result in life support failure.

## **Contact and Support**

Contact:  
Rogue Robotics Corporation  
support@roguerobotics.com  
Toronto, ON Canada

Support:  
1 866 99 ROGUE (76483)  
PH: (416) 288-0600 (in Toronto)  
Fax: (416) 288-0608