

MKE_I2C_MP3 API reference

```
#include <MKE_I2C_MP3.h>
MKE_I2C_MP3 mp3;
```

The constructor takes no parameters. Configure the I2C address and bus in `begin()`.

Setup

```
mp3.begin(50);           // Default bus: Wire
mp3.begin(51, Wire1);    // A different I2C bus, when supported by the board
```

`address` is a 7-bit I2C address; the default is `50 (0x32)`. `begin()` calls `Wire.begin()` for the selected bus and should be called once in `setup()`.

Playback functions

These functions return the `Wire.endTransmission()` result: `0` means the I2C packet was sent successfully.

Function	Description
<code>play(), pause(), stop()</code>	Resume, pause, or stop playback.
<code>next(), previous()</code>	Select the next or previous track.
<code>setVolume(volume)</code>	Set volume from 0 to 30. Values above 30 are clamped.
<code>playTrack(track)</code>	Play a Root-area track by its number.
<code>playTrackInFolder(folder, track)</code>	Play a track in a logical numbered folder.
<code>playTrackInMF(track)</code>	Play a system-area MF track.
<code>playInsertTrack(track)</code>	Interrupt playback with a Root-area track.
<code>playInsertTrackInFolder(folder, track)</code>	Interrupt playback with a folder track.
<code>playInsertTrackInMF(track)</code>	Interrupt playback with an MF track.

`playTrackInFolder(2, 5)` sends protocol value `2005`. MF is a system area in the built-in Flash; do not modify its files.

Read functions

The following functions return `true` only after receiving all four response bytes over I2C:

Function	Result
<code>getVolume(uint8_t &volume)</code>	Current volume, 0 to 30.
<code>getStatus(Status &status)</code>	<code>STOPPED</code> (0), <code>PLAYING</code> (1), or <code>PAUSED</code> (2).
<code>getModuleId(uint8_t &moduleId)</code>	Module ID; JQ8900 returns 5.
<code>readModuleAddress(uint8_t &address)</code>	Current module I2C address.

Address configuration

```
mp3.setAdminMode(true);  
delay(10);  
mp3.changeModuleAddress(51);  
mp3.setAdminMode(false);
```

Enable Admin Mode before changing the module address. `setAddress(address)` only changes the address used by the library; use it when the module was changed by its button or by another program.