

DFplayer Mini

Der DFPlayer Mini Player ist ein kleines und günstiges Sound-Modul welches MP3- und WAV-Dateien abspielen kann. Auch Werbeeinblendungen sind möglich.

Dieses Modul ist bei den einschlägigen Versendern (und daher auch in Unterschiedlichen Ausführungen) um wenig Geld erhältlich.

Das Modul verfügt über einen direkten 8 Ohm Mono Lautsprecheranschluss mit einer Ausgangsleistung von 3W.

Das Stereosignal steht an einem weiteren Anschluss zur Verfügung. Dort kann ein Verstärker oder Kopfhörer angeschlossen werden.

Der Verstärker sollte mittels Kondensatoren abgeblockt werden.

Zum steuern des Moduls steht eine serielle Schnittstelle (TTL, 9600, 8, N, 1) zur Verfügung.

Eine Befehlssequenz besteht aus zehn Byte (alle unterstützten Kommandos (Befehlsbyte) werden weiter unten zusammengefasst erklärt).

0x7E Startbyte 0x7E

0xFF Versionsbyte

0x06 Anzahl der Bytes beginnend mit dem Versionsbyte bis einschließlich dem LSB des Parameters, immer 0x06

CMD Befehlsbyte

FB Feedback, 0x01: Gibt Feedback zurück, 0x00: kein Feedback

PMSB Höherwertiges Byte des Parameters

PLSB Niederwertiges Byte des Parameters

CMSB Höherwertiges Byte der Prüfsumme

CLSB Niederwertiges Byte der Prüfsumme

0xEF Endbyte 0xEF

Die Prüfsumme berechnet sich aus der folgenden Formel:

$0 - \text{Versionsbyte} - \text{Anzahl der Bytes} - \text{Commandbyte} - \text{Feedback} - \text{Parameter_MSB} - \text{Parameter_LSB}$

Wenn dann das Ergebnis im Hexadezimalformat vorliegt ergeben die rechten vier Stellen der Hexadezimalzahl die Prüfsumme.

Aus diesen nunmehr vier Stellen ergeben die linken zwei Stellen (ein Byte) das Checksum_MSB und die rechten zwei Stellen (ein Byte) das Checksum_LSB.

Grundsätzlich kann auch ohne Prüfsumme gearbeitet werden (die Befehlszeichenkette wäre dann nur acht Byte lang) dies ist aber nur dann ratsam wenn ein Baudratenquarz verwendet wird!

Da die hier weiter unten vorgestellten Befehle die Prüfsummen ohnehin selbst ermitteln, kann das vernachlässigt werden.

Die wesentlichen Produkteigenschaften sind:

- Unterstützt MP3- und WAV-Decodierung.
- Unterstützt das Dateisystem FAT16 und FAT32.
- 24-Bit-DAC-Ausgabe und unterstützt den dynamischen Bereich 90 dB und SNR 85 dB.
- Unterstützt den AD-Key-Steuerungsmodus und den seriellen UART-RS232-Steuerungsmodus.
- Unterstützt maximal 32 GB Micro SD-Karte und 32 GB USB-Flash-Laufwerk.
- Unterstützt maximal 3000 Audiodateien im Stammverzeichnis des Speichermediums.
- Unterstützt maximal 9999 Audiodateien im Ordner **MP3**
- Unterstützt maximal 9999 Audiodateien im Ordner **ADVERT** (unterbrechende Werbeeinblendungen)
- Unterstützt maximal 99 Ordner und jeder Ordner kann 255 Audiodateien speichern.
- Unterstützt die zufällige Wiedergabe der Dateien.
- Eingebauter 3-Watt-Verstärker, der einen 3-Watt-Lautsprecher direkt ansteuern kann.
- Einstellbar Lautstärke (30 Stufen).
- Einstellbarer Equalizer (6 Stufen).

Technische Parameter:

Unterstützt 1172-3 und ISO13813-3 Layer3 Audiodecoding

MP3 Audio Format: Unterstützt verschiedene Samplingraten (8/11.025/12/16/22.05/24/32/44.1/48 kHz)

Unterstützt verschiedene Equalizermodis (Normal, Pop, Rock, Jazz, Classic, Bass)

USB Port:	Standard USB 2.0
UART Port:	Standardisierte serielle Schnittstelle mit TTL-Pegel (Parameter: 9600, 8, N, 1)
Betriebsspannung:	3.3 bis 5.0Vdc, typisch 4.2Vdc
Nennstrom:	<15mA (ohne USB-Flashdrive)
Temperaturbereich:	-40° bis +80°
Feuchtigkeit:	5% ~95%

Regeln für die Dateinamen:

Audiodateien, die direkt im Hauptverzeichnis des Speichermediums (SD-Karte oder USB-Flash) gespeichert sind (maximal 3000 Dateien), müssen in 0001.mp3 / 0001.wav, 0002.mp3 / 0002.wav, 0003.mp3 / 0003.wav, 3000.mp3 / 3000.wav usw. umbenannt werden. Nach dem vierstelligen numerischen Wert kann weiterer Text entsprechend dem verwendeten FAT-Format verwendet werden (z.B. 0001_Testdatei.MP3)

Grundsätzlich unterstützt das Modul auch verschiedene Unterverzeichnisse **\00** bis **\99** mit dreistelligen Dateinamen (001.MP3 bis 255.MP3), diese werden aber in dieser Beschreibung nicht näher behandelt!

In den Ordnern **\MP3** und/oder **\ADVERT** können jeweils bis zu 9999 Dateien gespeichert werden. Die oben genannten Namensregeln gelten aber auch hier!

HINWEIS:

Es ist wichtig zu wissen, dass in der Folge beim Absetzen eines Befehls keine tatsächlichen Dateinamen übergeben werden sondern eine Platznummer welche sich aber auf die tatsächliche physikalische Position in der Dateizuordnungstabelle (FAT) des Mediums bezieht und eigentlich nichts mit dem Dateinamen zu tun hat! Von daher ist es von essentieller Bedeutung in welcher Reihenfolge die Dateien auf das Speichermedium übertragen werden.

Es kann daher notwendig sein, die Dateizuordnungstabelle mit entsprechenden Tools zu sortieren (z.B. FATsorter o.ä.). Üblicherweise erstellt man sich am PC einen entsprechenden Ordner, legt in diesen Ordner die gewünschten Dateien und Verzeichnisse an, benennt die Dateien so um, dass die gewünschte Reihenfolge entsteht und kopiert schließlich alle Dateien und Verzeichnisse in einem Schwung auf die SD-Karte oder den USB-Stick (für Windows: STRG-A im Quellordner, STRG_V im Zielordner).

Das Grundschaltbild des Moduls:

[DFP_GS.JPG](#)

Grundsätzlich kann das Modul auch nur mit Tasten also ganz ohne Microcontroller) bedient werden, dies wird in dieser Beschreibung aber nicht behandelt.

Mehr dazu kann dem Datenblatt (von welchem es viele verschiedene mehr oder weniger umfangreiche im Internet zu finden gibt) entnommen werden.

Wie auf dem Bild zu erkennen, wird das Modul mit 5Vdc betrieben. Daher ist im Stromversorgungszweig die Diode D2 (1N4004) eingefügt um die Betriebsspannung etwas zu senken.

Die Diode D1 (1N4148) war notwendig, weil das Modul Störgeräusche beim Abspielen entwickelte.

Der im Datenblatt vorgesehene Vorwiderstand verhinderte die Störgeräusche übrigens auch nicht.

Die im Internet vielfach empfohlene eigene Stromversorgung für das Modul verhinderte diese Störgeräusche ebenfalls nicht und stellt oft ohnehin keine Option dar.

Der Busy-Pin führt während dem abspielen einer Sounddatei Low-Pegel und kann so weiter ausgewertet werden.

Steuerung des Moduls mittels Steuerbefehle:

Bevor die Befehle der Bibliothek erklärt werden, wird hier eine Zusammenfassung aller möglichen Befehle zusammengefasst.

Es wird darauf hingewiesen, dass dies eine Zusammenstellung aller Befehle aus diversen Datenblättern zu verschiedenen Modulversionen ist (an diese Stelle möchte ich mich noch einmal für seine Mitarbeit bei **mac5150** hier aus dem Forum bedanken).

Daher kann es sein, dass nicht jeder Befehl bei jedem Modul funktioniert oder plausible Daten zurück gibt.

Befehle:

01 Spielt die nächste Datei

02 Spielt die vorherige Datei

03 Spielt eine Datei aus dem Hauptverzeichnis (1-3000)

- 04 Lautstärke um 1 erhöhen
- 05 Lautstärke um 1 verringern
- 06 Lautstärke setzen (0-30)
- 07 Equalizer setzen (0-5)
- 08 Spiele eine Datei wiederholend aus dem Hauptverzeichnis (1-3000)
- 09 Setzt die zu verwendende Datenquelle fest (0-4)
- 0A Das Modul wird in den Standby-Modus versetzt
- 0B Das Modul wird in den Normal-Modus versetzt (aufgeweckt)
- 0C Setzt das Modul zurück und gibt den Status zurück
- 0D Spielt einen pausierten Track weiter oder startet den aktuellen Track
- 0E Pausiert die gerade abgespielte Datei
- 0F Spielt eine Datei aus dem Ordner 01 - 99
- 10 Schaltet den Verstärker ein oder aus und setzt den Verstärkungsfaktor
- 11 Schaltet das wiederholende Abspielen aller Dateien ein oder aus
- 12 Spielt eine Datei aus dem Ordner MP3 ab (1-9999)
- 13 Unterbricht die laufende Datei, spielt eine Werbeunterbrechung aus ADVERT und setzt das Abspielen fort (1-9999)
- 14 Spielt eine Datei aus einem "großen" Ordner ab (1-15)
- 15 Beendet das abspielen einer ADVERT-Datei und setzt danach das abspielen der zuvor unterbrochenen Datei wieder fort
- 16 Beendet das abspielen der laufenden Datei
- 17 Wiederholt die Dateien aus dem Ordner (1-99)
- 18 Spielt alle Dateien des Mediums zufällig ab
- 19 Spielt eine laufende Datei wiederholend ab
- 1A Schalte den Verstärker stumm

Abfragen:

- 3F Gibt das aktuelle Speichermedium zurück (1-4)
- 42 Gibt den aktuellen Status zurück
- 43 Gibt den aktuellen Lautstärkepegel zurück
- 44 Gibt den aktuellen Equalizer-Modus zurück
- 45 Gibt den aktuellen Abspiel-Modus zurück
- 46 Gibt die aktuelle Softwareversion zurück
- 47 Gibt die Anzahl der Dateien vom Medium USB-Speicherstick zurück
- 48 Gibt die Anzahl der Dateien vom Medium SD-Karte zurück
- 49 Gibt die Anzahl der Dateien vom Medium Flash zurück
- 4A Keep On, dieser Befehl wird in einigen Datenblättern erwähnt, die genaue Funktion ist allerdings unbekannt
- 4B Gibt die momentan gespielte Datei vom Medium SD-Karte zurück
- 4C Gibt die momentan gespielte Datei vom Medium USB-Speicherstick zurück
- 4D Gibt die momentan gespielte Datei vom Medium Flash zurück
- 4E Gibt die Anzahl der Dateien im abgefragten Ordner zurück
- 4F Gibt die Anzahl der Ordner auf dem Medium zurück

Unabhängige Statusmeldungen:

- 3A Speichermedium wurde angeschlossen: 1=USB-Stick, 2=CD-Card, 4=PC-Mode
- 3B Speichermedium wurde entfernt: 1=USB-Stick, 2=CD-Card, 4=PC-Mode
- 3C Datei wurde beendet auf USB-Speicherstick
- 3D Datei wurde beendet auf SD-Karte
- 3E Datei wurde beendet auf Flash-Speicher

Systemmeldungen:

- 40 Fehlermeldungen, 1=nicht bereit, 2=Sleepmode, 3=Fehler RS232, 4K=Checksumerror, 5=ungültige Dateinummer, 6=Datei nicht gefunden, 7=Inter-cut error, 8=SD-Card error, 10=in Sleepmode
- 41 Feedback, Modul bestätigt Datenempfang

Um das Modul mit einfachen Befehlen aus BasCom heraus zu steuern, wurden dafür alle bekannten Befehle in BasCom-Code umgesetzt.

Dazu einige Erläuterungen:

Wenn BasCom bezüglich Unterprogrammen und Funktionen noch nach der alten Methode verwendet wird (Standardeinstellung), müssen diese Befehle noch im Code deklariert werden (mit den Befehlen **Declare Sub...**, **Declare Function...**), der zusätzliche Code kann dann an geeigneter Stelle hinter dem eigentlichen Programm eingefügt werden. Nach der neuen Methode (**Config Submode = New**) ist dies nicht mehr erforderlich, allerdings muss der zusätzliche Code vor dem eigentlichen Programm bzw. vor dessen ersten Aufruf eingefügt werden.

Folgende Befehle werden aktuell unterstützt:

Hinweis: Allfällige übergebene Parameter werden nicht auf deren Korrektheit überprüft!

- **MP3_Init**
Setzt das Modul zurück und gibt den Status zurück
Rückgabe:
0 = Nicht bereit, kein Speichermedium vorhanden
1 = USB-Speicherstick
2 = SD-Karte
3 = USB-Speicherstick und SD-Karte
4 = PC
- **MP3_Play pFile**
Spielt die Datei *pFile* [/i]aus dem Hauptverzeichnis ab (*[i]pFile* liegt im Bereich zwischen 1 und 3000 entsprechend der FAT-Reihenfolge).
- **MP3_PlayFromMP3 pFile**
Spielt die Datei *pFile* aus dem Ordner \MP3 ab (*pFile* liegt im Bereich zwischen 1 und 9999).
- **MP3_PlayAdvert pFile**
Unterbricht das Abspielen der aktuellen Datei, spielt die Datei *pFile* (liegt im Bereich zwischen 1 und 9999 entsprechend der FAT-Reihenfolge) aus dem Ordner ADVERT ab und setzt danach das abspielen der unterbrochenen Datei wieder fort (Werbeunterbrechung oder Einblendung).
- **MP3_PlayLoop pFile**
Spielt die Datei *pFile* Datei als Dauerschleife ab (*pFile* liegt im Bereich zwischen 1 und 3000 entsprechend der FAT-Reihenfolge).
- **MP3_PlayAllLoop pValue**
Schaltet das wiederholende Abspielen aller Dateien ein oder aus, *ppValue* =0 stoppt das abspielen, *pValue* <>0 startet das abspielen
- **MP3_Prev**
Spielt die vorhergehende Datei entsprechend der FAT-Reihenfolge ab.
- **MP3_Next**
Spielt die nächste Datei entsprechend der FAT-Reihenfolge ab.
- **MP3_Resume**
Spielt einen pausierten Track weiter oder startet den aktuellen Track welcher mittels *MP3_Play(Tracknummer)* vorher aktiviert wurde.
- **MP3_PlayRandom**
Spielt alle Dateien des Speichermediums zufällig ab.
Hinweis: Es wird immer mit Track 1 begonnen, die Titel danach sind dann jedes mal zufällig!
- **MP3_Pause**
Pausiert eine Datei, mittels *MP3_PlayResume* wird ein derartig gestoppter Track wieder fortgesetzt.
- **MP3_Stop**
Beendet das abspielen der laufenden Datei.
- **MP3_StopAdvise**
Beendet die mit *MP3_PlayAdvise* gestartete Datei sofort und setzt nach dem beenden der Datei die zuvor unterbrochene Datei wieder fort.
- **MP3_VolumeDecr**
Verringert die Lautstärke um 1.
- **MP3_VolumeIncr**
Erhöht die Lautstärke um 1.
- **MP3_SetVolume pVolume**
Setzt die Lautstärke auf den übergebenen Wert *pVolume* (*pVolume* liegt im Bereich 0 bis 30).
- **MP3_SetEqualizer pEQ**
Setzt den Equalizer auf den übergebenen Wert *pEQ*. Der einzustellende Equalizerwert liegt zwischen 0 und 5.
0=Normal, 1=Pop, 2=Rock, 3=Jazz, 4=Classic, 5=Bass.
- **MP3_SetSource pSource**
Setzt die zu verwendete Datenquelle *pSource* fest.
pSource liegt im Bereich von 0 bis 4 und entspricht folgenden Medien:
0 = USB -Speicherstick, 1 = SD-Karte, 2 = Aux (nicht verwendet!), 3 = Sleep-Modus, 4 = Flash Speicher.
- **MP3_Standby**
Das Modul wird in den Standby-Modus versetzt.
HINWEIS: Einige Module können mit dem Befehle "MP3_WakeUp" nicht wieder "aufgeweckt" werden. Diese Module müssen kurz von der Stromversorgung getrennt werden!
- **MP3_Wakeup**

Das Modul wird in den Normal-Modus versetzt (aufgeweckt).

HINWEIS: Dieser Befehl wird nicht von allen Modulen unterstützt!

Je nach Modul wird eventuell gar nichts oder auch eine Fehlermeldung zurückgegeben!

Eine mögliche Rückgabe ist zum Beispiel (nicht bei allen Modulen: 7E FF 06 40 00 00 01 FE BA EF.

- **MP3_PlayFromFolder *pFolder*, *pFile***
Spielt eine Datei aus dem Ordner 01 - 99 (Dateiformat im Folder : 123.MP3) ab.
Der Bereich für *pFolder* liegt zwischen 1 bis 99 und für *pFile* zwischen 1 bis 255.
- **MP3_SetAmpGain *pAmp*, *pGain***
Schaltet den Verstärker ein oder aus und setzt den Verstärkungsfaktor
pAmp schaltet bei 0 den Verstärker aus und bei 1 ein.
pGain liegt im Bereich zwischen 0 und 31 und setzt den Verstärkungsfaktor.
- **MP3_PlayFromBigFolder *pFolder*, *pFile***
Spielt eine Datei aus einem "großen" Ordner (Dateiformat im Ordner: 1234.MP3) ab.
Der Bereich für *pFolder* liegt zwischen 1 und 15 und der Bereich für *pFile* zwischen 1 und 3000.
- **MP3_PlayRepeatFolder *pFolder***
Wiederholt die Dateien aus dem Ordner 01 - 99, *pFolder* liegt zwischen 1 und 99.
- **MP3_PlayLoopCurrent *pValue***
Spielt eine laufende Datei wiederholend ab.
Bei *pValue* = 0 ist wird die Wiederholung eingeschaltet, bei *pValue* = 1 wird wieder ausgeschaltet.
Hinweis: Dies ist kein Schreibfehler!
- **MP3_Mute *pValue***
Schaltet den Verstärker stumm.
Bei *pValue* = 0 wird der Mute-Modus deaktiviert (DAC wird aktiviert) und
bei *pValue* = 1 wird der Mute-Modus aktiviert (DAC wird deaktiviert).
- **MP3_GetPlayMode**
Gibt den aktuellen Abspiel-Modus zurück.
Rückgabe: 0 = Normal, 2 = Loop Track
- **MP3_GetVersion**
Gibt die aktuelle Softwareversion zurück.
Die Rückgabe ist abhängig vom verwendeten Modul und liegt zwischen 0 bis 255.
- **MP3_KeepOn**
Dieser Befehl wird in einigen Datenblättern erwähnt, die genaue Funktion ist allerdings unbekannt.
Eine mögliche Rückgabe liegt zwischen 0-65535.
- **MP3_GetCurrentFileCard**
Gibt die momentan gespielte Datei vom Medium SD-Karte zurück
Rückgabe: Dateinummer entsprechend der FAT-Reihenfolge.
- **MP3_GetCurrentFileUSB**
Gibt die momentan gespielte Datei vom Medium USB-Stick zurück
Rückgabe: Dateinummer entsprechend der FAT-Reihenfolge.
- **MP3_[b]GetCurrentFileFlash**
Gibt die momentan gespielte Datei vom Medium Flash zurück
Rückgabe: Dateinummer entsprechend der FAT-Reihenfolge.
- **MP3_GetFilesInFolder *pFolder***
Gibt die Anzahl der Dateien im abgefragten Ordner zurück.
pFolder liegt im Bereich von 1 bis 99 entsprechend dem abzufragenden Ordner.
Rückgabe: Die Anzahl der Dateien im angefragten Ordner oder 0 (Null) wenn der Ordner nicht existiert oder auch leer ist.
- **MP3_GetCurrentDevice**
Gibt das aktuelle Speichermedium zurück.
Rückgabe: 1=USB-Flash, 2=SD-Card, 3=USB-Flash und SD-Card, 4=PC.
- **MP3_GetStatus**
Gibt den aktuellen Status zurück
Rückgabe: 10=USB-Flash ist gestoppt, 11=USB-Flash ist im Abspielmodus, 12=USB-Flash ist im Pausenmodus,
20=SD-Card ist gestoppt, 21=SD-Card ist im Abspielmodus, 22=SD-Card ist im Pausenmodus,
30=Modul ist im Sleep-Modus.
- **MP3_GetVolume**
Gibt den aktuellen Lautstärkepegel zurück.
Rückgabe: 0 bis 30

- **MP3_GetEqualizer**
Gibt den aktuellen Equalizertyp zurück,
Rückgabe: 0=Normal, 1=Pop, 2=Rock, 3=Jazz, 4=Classic, 5=Bass.
- **MP3_GetCountUSB**
Gibt die Anzahl der Dateien vom Speichermedium USB-Flash zurück.
Rückgabe: Anzahl der Dateien.
- **MP3_GetCountCard**
Gibt die Anzahl der Dateien vom Speichermedium CD-Card zurück.
Rückgabe: Anzahl der Dateien.
- **MP3_GetCountFlash**
Gibt die Anzahl der Dateien vom Speichermedium Flash zurück.
Rückgabe: Anzahl der Dateien.
- **MP3_GetFolders**
Gibt die Anzahl der Ordner auf dem Medium zurück.
Rückgabe: Anzahl der Ordner.
Hinweis: Auch das Hauptverzeichnis ist ein Ordner und wird daher hinzu gezählt!
Das Resultat hat daher der Mindestwert von 1.

Hier der Quellcode der Bibliothek:

Software für den Modultest:

Für den Modultest gibt es hier eine Software welche auf verschiedenen Microcontrollern ablaufen kann. Diese kleine selbst ablaufende Demo für den DFPlayer mini kommt ursprünglich von **mac5150** hier aus dem Forum. Die Demo wurde etwas adaptiert und optisch verbessert, zum Betrieb ist noch ein Speichermedium (SD-Karte mit der unten beschriebenen Struktur) nötig:.

Diese Testapplikation sollte grundsätzlich mit den folgenden Microcontrollern laufen:

ATmega8, ATmega8A, ATmega328P, ATmega16, ATmega32, ATmega64 und ATmega1284P und kann in der Applikation einfach konfiguriert werden.

Getestet im 3.3V Betrieb mit einem ATmega328P, LCD2004 I2C, DFPlayer mini sowie im 5V Betrieb mit einem ATmega1284p, LCD2004 4bit-Mode, DFPlayer mini.

Je nach verwendetem Prozessor, dessen Taktfrequenz und des verwendeten LCD Displays sind noch geringfügige Anpassungen und Codeänderungen notwendig die wegen der Dokumentation im Code einfach durchgeführt werden können!

Quellcode

1. Verzeichnis- und Dateistruktur für das Medium zum Testen von DFplayerMini-Modulen
2. \01
4. \02
5. \20
6. \21
7. \ADVERT
8. \MP3
9. \0001.mp3
10. \0002.mp3
11. \0003.mp3
12. \0004.mp3
13. \0005.mp3
14. \01\0001.mp3
16. \01\0002.mp3
17. \01\0003.mp3
18. \01\0004.mp3
19. \01\0005.mp3
20. \01\0006.mp3
21. \01\0007.mp3
22. \01\0008.mp3
23. \01\0009.mp3
24. \01\0010.mp3

26. \02\0001.mp3
27. \02\0002.mp3
28. \02\0003.mp3
29. \02\0004.mp3
30. \02\0005.mp3
31. \02\0006.mp3
32. \02\0007.mp3
33. \02\0008.mp3
34. \02\0009.mp3
35. \02\0010.mp3
- ~~36.~~ \20\001.mp3
38. \20\002.mp3
39. \20\003.mp3
40. \20\004.mp3
41. \20\005.mp3
42. \20\006.mp3
43. \20\007.mp3
44. \20\008.mp3
45. \20\009.mp3
46. \20\010.mp3
- ~~47.~~ \21\001.mp3
49. \21\002.mp3
50. \21\003.mp3
51. \21\004.mp3
52. \21\005.mp3
53. \21\006.mp3
54. \21\007.mp3
55. \21\008.mp3
56. \21\009.mp3
57. \21\010.mp3
58. \ADVERT\0001.mp3
60. \ADVERT\0002.mp3
62. \MP3\0001.mp3
63. \MP3\0002.mp3
64. \MP3\0003.mp3
65. \MP3\0004.mp3
66. \MP3\0005.mp3
67. \MP3\0006.mp3
68. \MP3\0007.mp3
69. \MP3\0008.mp3
70. \MP3\0009.mp3
71. \MP3\0010.mp3

Alles anzeigen

Einen Diskussionsbeitrag zu diesem Beitrag gibt es [HIER](#).

Dateien zum herunterladen:

Die Test- und Demosoftware: [DEMO_DFplayerMini.zip](#)

Die Bibliothek für den DFplayerMini: [LIB_DFplayerMini.zip](#)

Das Image für den Testdatenträger: [/attach](#)