

 Friedrich-Ebert-Schule Esslingen FES	Mikrocontroller	Name: Rahm Datum: 21.07.2022 7_2_MP3_Player_Steuerung.docx
	Steuerung des DFPlayer Mini-Moduls über RS232	7.2.1

Die Steuerung des MP3-Players kann entweder über die auf dem MP3-Player-Shield verbauten Taster oder über RS232-Kommandos erfolgen.

Bei Tastersteuerung benötigt das Shield lediglich die Spannungsversorgung vom Carrierboard. Mit dem + und – Taster werden mit kurzem Druck die Titel weitergeschaltet und bei langem Druck die Lautstärke geändert.

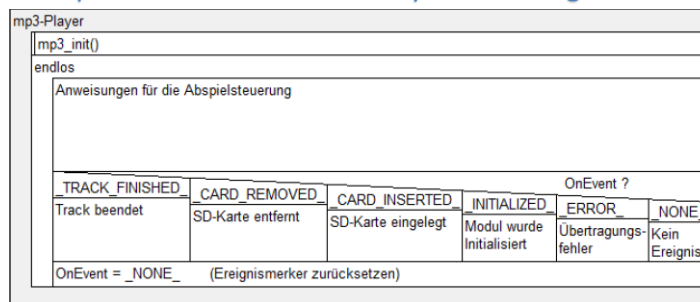
Für die Steuerung über RS232-Kommandos gibt es mehrere fertige Bibliotheken für Arduino und ebenso eine Implementation zur FA205-konformen AVR-Programmierung, die hier vorgestellt wird.

Für die Kommunikation ist ein serieller Empfangsinterrupt implementiert, der Rückgabewerte und spezielle Ereignisse zurückgibt.



Langer Druck: +/- Volume
Kurzer Druck: next/prev Track

Prinzipieller Ablauf einer MP3-Player-Steuerung



Die Variable *OnEvent* ist in der mp3player-Bibliothek definiert. Wenn eines der Ereignisse auftritt, wird diese Variable gesetzt und kann abgefragt werden. Danach muss *OnEvent* wieder zurückgesetzt werden.

Code Grundgerüst

```
#include "controller.h"
#include "mp3player.h"

void setup (void) // Initialisierungen
{
    mp3_init();
    mp3_setPlaybackSource(1); // Quelle: SD-Card
}

int main (void)
{
    setup();

    //mp3_playRandomTrackFromAll();

    while(1) // loop()
    {
        // hier Anweisungen für Abspielsteuerung

        // Ereignisauswertung
        switch (OnEvent)
        {
            case _TRACK_FINISHED_: break;
            case _CARD_REMOVED_: break;
            case _CARD_INSERTED_: break;
            case _INITIALIZED_: break;
            case _ERROR_: break;
            case _NONE_: break;
        }
        OnEvent = _NONE_;
    }
}
```

	Mikrocontroller	Name: Rahm Datum: 21.07.2022 7_2_MP3_Player_Steuerung.docx
	Steuerung des DFPlayer Mini-Moduls über RS232	7.2.2

Funktionsprototypen und Konstanten zur Steuerung des DFPlayer

```
// Initialisierung-, Status- und Info-Funktionen
void mp3_init(void);
void mp3_reset(void);
void mp3_sleep(void);
uint16_t mp3_getStatus(void);
uint8_t mp3_getPlaybackMode(void);
uint16_t mp3_getTotalTrackCount(void);
uint16_t mp3_getFolderTrackCount(void);
uint16_t mp3_getTotalFolderCount(void);
uint16_t mp3_getCurrentTrack(void);

// Lautstärke, EQ
void mp3_setVolume(uint8_t volume); // 0 ... 30
uint8_t mp3_getVolume(void);
void mp3_increaseVolume(void);
void mp3_decreaseVolume(void);
void mp3_setEq(uint8_t eq);
uint8_t mp3_getEq(void);

// Abspiel-Steuerung
void mp3_playGlobalTrack(uint16_t track);
void mp3_start(void);
void mp3_stop(void);
void mp3_pause(void);
void mp3_prevTrack(void);
void mp3_nextTrack(void);
void mp3_setRepeatPlay(uint8_t repeat); //1=start; 0=stop
void mp3_playRandomTrackFromAll(void);
void mp3_loopGlobalTrack(uint16_t track);
void mp3_playFolderTrack(uint8_t folder, uint8_t track);
void mp3_playMP3FolderTrack(uint16_t track);
void mp3_setPlaybackSource(uint8_t source); //1=SD-Card
void mp3_setPlaybackMode(uint8_t mode);
void mp3_playAdvertisement(uint16_t track);
void mp3_stopAdvertisement(void);
```

```
enum
{
    _PLAYBACK_REPEAT_ = 0,
    _PLAYBACK_FOLDER_REPEAT_,
    _PLAYBACK_SINGLE_REPEAT_,
    _PLAYBACK_RANDOM_,
} playback_mode;









enum
{
    _PLAYBACK_STOP_ = 0,
    _PLAYBACK_PLAY_,
    _PLAYBACK_PAUSE_
} play_mode;

enum EQ_MODE
{
    _EQ_NORMAL_ = 0,
    _EQ_POP_,
    _EQ_ROCK_,
    _EQ_JAZZ_,
    _EQ_CLASSIC_,
    _EQ_BASS_,
};
```

Die (enum) Variablen **play_mode** und **playback_mode** können für die Speicherung der Abspielart verwendet werden.

Mit EQ_MODE werden Klangpresets gewählt. Bsp:
`mp3_setEq(_EQ_JAZZ_);`

Vordefinierte Display-Symbole zur Abspielsteuerung

							
<code>_C_LS_</code>	<code>_C_STOP_</code>	<code>_C_PAUSE_</code>	<code>_C_PLAY_</code>	<code>_C_FWD_</code>	<code>_C_RWD_</code>	<code>_C_REPEAT_</code>	<code>_C_RANDOM_</code>

Anwendung: `lcd_setcursor(2,13); lcd_char(_C_STOP_);`

Verzeichnisstruktur auf der SD-Karte

Wichtig ist die Nummerierung.

Im Stammverzeichnis: xxxx

Ordnerernamen: xx

Tracks im Ordner: xxx

01 Dirty Deeds Done Dirt Cheap (ACDC)	001_Dirty Deeds Done Dirt Cheap.mp3
02 High Voltage (ACDC)	002_Love At First Feel.mp3
03 Deep Purple In Rock	003_Big Balls.mp3
04 Black Ice (ACDC)	004_Rocker.mp3
0001 Sultans Of Swing.mp3	005_Problem Child.mp3
0002 Eastbound Train.mp3	006_There's Gonna Be Some Rockin'.mp3
0003 Southbound Again.mp3	007_Ain't No Fun Waiting Round To Be A Millionaire.mp3
0004 Water Of Love.mp3	008_Ride On.mp3
0005 Whats The Matter With You Baby.mp3	009_Squealer.mp3
0006 Lions.mp3	
0007 Six Blade Knife.mp3	
0008 Down The Water Line.mp3	
0009 In The Gallery.mp3	
0010 Wild West End.mp3	

Datenblatt

https://www.robotshop.com/media/files/pdf2/rb-dfr-562_-_dfplayer_mini_mp3_player.pdf

Die Benennung der Bibliotheksfunktionen wurde weitgehend an die Mp3-Player Bibliothek von Makuna für Arduino angepasst:

<https://github.com/Makuna/DFMiniMp3/wiki/API-Reference>