2025/11/28 00:55 1/8 Sound Servoplatine

Sound Servoplatine

++ Diese Seite wird derzeit überarbeitet ++

Über die Servo-Platine 510 können drei Soundmodule angesteuert werden.

Es werden dabei JQ6500 Module oder MP3-TF16-p/DFPlayer Mini Module verwendet. Dabei spielt es keine Rolle ob nur JQ6500 oder nur MP3-TF16-p/DFPlayer Mini oder diese beliebig gemischt, eingesetzt werden.

Die drei Module können auch parallel je einen Sound abspielen.

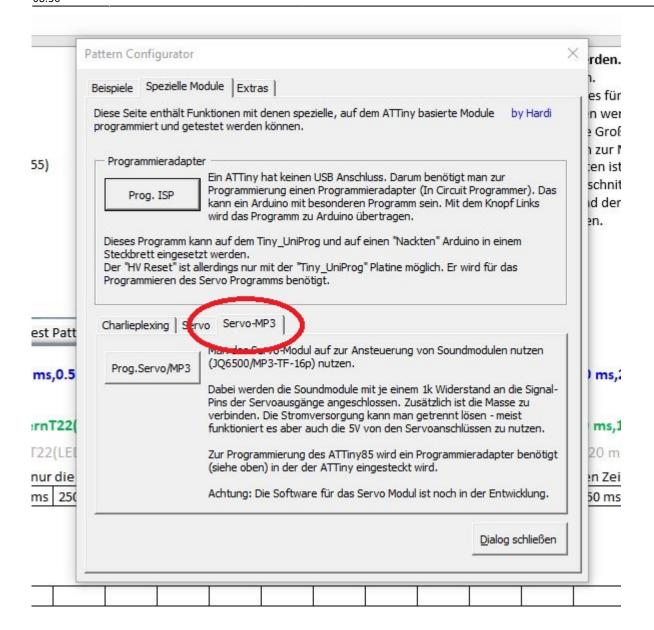
Das JQ6500 hat dabei den Vorteil das es bereits 2 MByte Speicher enthält, was für etliche Soundprojekte ausreichend sein dürfte.

Das MP3-TF-16p nutzt hingegen eine MicroSD-Karte mit bis zu 32 GByte und vereinfacht durch die mögliche Ordnerstruktur die Verwaltung der Sounddaten.

Vorbereitung Hardware

ATTiny 85

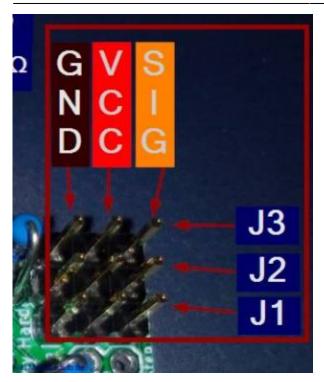
Als Servoplatine eignet sich jede 510DE-Platine, bestückt als Servoplatine. Die Lötjumper für den Servobetrieb SERVO, SERVO1, SERVO2, SERVO3 müssen geschlossen werden. Zunächst muss der ATTiny für die Servoplatine programmiert werden. Die Anleitung dazu Programmierung von ATTinys für Servo, Charlieplexing oder Sound ist im WIKI zu finden. Im Pattern-Configurator unter spezielle Module **Servo-MP3** auswählen.



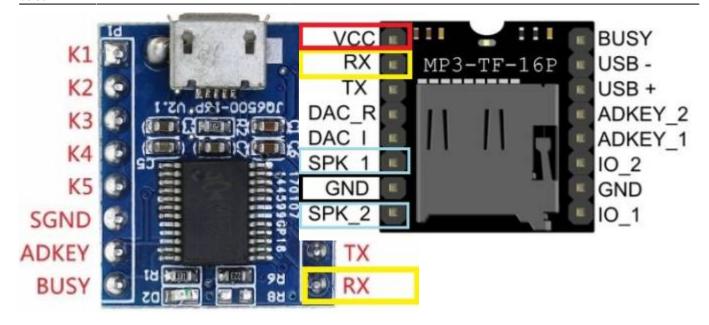
Das folgende Bild zeigt die Pins am Ausgang der Servo-Platine. Der SIG-J1 - Pin ist mit dem Eingang des ersten Soundmodules zu verbinden, entsprechend J2 mit Modul 2 und J3 mit Modul 3.

https://wiki.mobaledlib.de/ Printed on 2025/11/28 00:55

2025/11/28 00:55 3/8 Sound Servoplatine



JQ6500 und MP3-TF16-p/DFPlayer Mini



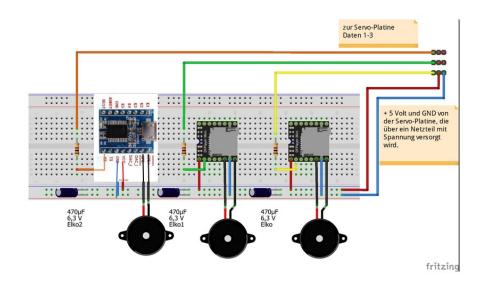
Verdrahtung

Die drei Ausgänge (SIG) der Servo-Platine werden nun über je einen 1 kOhm Widerstand mit den Eingängen (RX) der Sound-Module verbunden.

Die Versorgungsspannung sollte über eine Verteilerplatine mit angeschlossener stabiler Spannungsversorgung erfolgen, da die Leistung an der Hauptplatine zur Versorgung der Sound-Module nicht ausreicht.

Die 470µF Elektrolytkondensatoren dienen als Puffer für die recht hohen Einschaltströme der Soundmodule.

Eine separate 5V Spannungsversorgung ist ebenfalls möglich.



https://wiki.mobaledlib.de/ Printed on 2025/11/28 00:55

2025/11/28 00:55 5/8 Sound Servoplatine

Als weitere Möglichkeit bietet sich der Umbau einer unbestückten 501de_Soundplatine (MP3-TF-16p) an:

		by Hardi
Drahtbrücke (rot) an der Oberseite	Buchsenleisten: 2×8 Soundmodul	Widerstand 1kOhm an der Unterseite
R6 Widerstand 10hm	1×3 rot(VCC), schwarz(GND), gelb(SIG)	Jumper SJ1 schließen
Keramikkondensator 100nf	ACHTUNG-ServoAnschluss rot/schwarz vertauscht	
C6 Elko 470μF	1×2 Lautsprecher	

SD Karte

Die Verzeichnisstruktur der Sounddateien auf der SD-Karte ist relativ starr vorgeschrieben. Es können Sounddateien (wahlweise im mp3- oder wav-Format) in folgende Verzeichnisse der SD-Karte abgelegt werden:

- Wurzelverzeichnis (Dateinamen 4-stellig 0001.mp3 etc.)
- Verzeichnis ADVERT (Dateinamen 4-stellig)
- Verzeichnis mp3 (Dateinamen 4-stellig)
- Verzeichnisse 01 bis 32 (Dateinamen 3-Stellig 001.mp3 etc.)

Es werden nur die ersten Zeichen der Dateinamen ausgewertet. Der tatsächliche Dateiname kann länger sein. Damit sind Dateinamen der Art

0023Yesterday.mp3

möglich. Dieser Dateiname wird vom Soundmodul als "0023.mp3" behandelt. Das erleichtert den Umgang mit den Sounddateien deutlich und sollte ausgiebig genutzt werden.

Programm-Generator

Im Prog-Gen gibt es für den Servo-Sound folgende Befehle:

- ◀(i) ATTiny85	Soundmodule über ATTiny85	
Befehl an Soundmodul	Befehl an Soundmodul über Servoplatine	
Titel # abspielen	Track auf Soundmodul über Servoplatine abspielen	
Set Einstellungen	MP3-TF-16p einstellen	
▶ Titel aus Ordner abspielen	MP3-TF-16p, Track aus Ordner abspielen	
Pin MP3-Modul definieren	Anschluss für MP3-Modul auswählen	
Soundmodul definieren	Typ für angeschlossenes MP3-Modul einstellen	
# Titel # aus Hauptverzeichnis abspielen	Track # vom angegebenen Modul abspielen (Rootverzeichnis)	
Titel # aus mp3 abspielen	Track # aus mp3 auf Modul abspielen	

Beispiel der Soundsteuerung mit den Befehlen:

20		_		1	-	1	 -					_	-	_
99														
100	V		80	Rot				Soundmodul definieren	MP3_SET_TYPE(#LED, #InCh, 1, MP3_JQ6500)	1	C1-1	1	0	0
101	V		81	Rot			•	Soundmodul definieren	MP3_SET_TYPE(#LED, #InCh, 2, MP3_TF_16p)	1	^ C1-1	1	0	0
102	V		82	Rot			•	Soundmodul definieren	MP3 SET TYPE(#LED, #InCh, 3, MP3 TF 16p)	1	^ C1-1	1	0	0
103														
104	V		90	Rot			0	Pin MP3-Modul definieren	MP3 SELECT MODULE(#LED, #InCh, 1)	1	^ C1-1	1	0	0
105	V		91	Rot			0	Pin MP3-Modul definieren	MP3 SELECT MODULE(#LED, #InCh, 2)	1	^ C1-1	1	0	0
106	V		92	Rot			•	Pin MP3-Modul definieren	MP3 SELECT MODULE(#LED, #InCh, 3)	1	^ C1-1	1	0	0
107						1								
108 109	V		93	Rot			2	Befehl an Soundmodul	MP3 CMD(#LED, #InCh, MP3 DECREASE VOLUME)	1	^ C1-1	1	0	0
109	V		94	Rot			2	Befehl an Soundmodul	MP3 CMD(#LED, #InCh, MP3 INCREASE VOLUME)	1	^ C1-1	1	0	0
110	V	1	95	Rot			3	Titel # abspielen	MP3 TRACK(#LED, #Inch, MP3 PLAY TRACK, 1)	1	^ C1-1	1	0	0
111	V		96	Rot			3	Titel # abspielen	MP3 TRACK(#LED, #InCh, MP3 PLAY TRACK, 2)	1	^ C1-1	1	0	0
112								•						
113														-
114				1										$\overline{}$
115							-							-

In den Zeilen 100-102 wird der an die Sound-Platine angeschlossene Modultyp festgelegt. Diese Befehle müssen nur einmalig gesendet werden. Der ATTiny speichert die Einstellung. Eine Änderung ist nur dann notwendig, wenn Änderungen bei den angeschlossenen Modul-Typen vorgenommen werden. Im Beispiel

- Ch1 → JQ6500
- Ch2 → MP3-TF16-p
- Ch3 → MP3-TF16-p

Zeile 104 - 106: legt den Ausgang fest auf den der nächste Befehl gehen soll. Hier:

- Zeile 104 Ausgang 1, JQ6500.
- Zeile 105 Ausgang 2, MP3-TF16-p
- Zeile 106 Ausgang 3, MP3-TF16-p

Beispiel:

- DCC Befehl 90 und anschließend 95 spielt Track 1 vom JQ6500 ab.
- DCC Befehl 91 und anschließend 95 spielt Track 1 vom ersten MP3-TF16-p ab.
- DCC Befehl 92 und anschließend 95 spielt Track 1 vom zweiten MP3-TF16-p ab.
- DCC Befehl 91 und anschließend 93 verringert die Laustärke beim zweiten MP3-TF16-p.

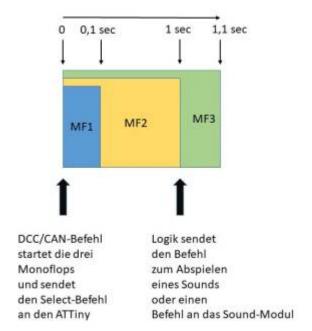
Anmerkungen:

- In der Macroauswahl wird der Hinweis "Sendet einen Befehl an ein MP3-TF-16p-Soundmodul welches über SERVO3 an einer Servoplatine angeschlossen ist" gegeben. Das trifft so nicht zu, da die Befehle an alle drei Ausgänge gesendet werden können und auch für beide Modultypen gelten. Einschränkungen gibt es natürlich beim JQ 6500, da nur fünf Tracks gespeichert werden können.
- Alle Befehle an diese ATTiny Soundplatine müssen über eine LED Adresse laufen, im Beispiel LED 1. Sollte es (ungewollt) eine Verschiebung geben mit dem Befehl "next LED -1" wieder auf die LED Adresse der Sound-Platine zurück gehen.
- Über den Kleinen Verteiler mit der Copy-Funktion, Einstellung über den Jumper, kann man parallel Test-LEDs anschließen und optisch die Funktion überprüfen.

https://wiki.mobaledlib.de/ Printed on 2025/11/28 00:55

2025/11/28 00:55 7/8 Sound Servoplatine

Steuerung über DCC/CAN-Befehle



Vor jedem Sound-Befehl muss das Modul ausgewählt werden auf dem sich die Sound-Datei befindet. So können Sound-Dateien in unterschiedlicher Reihenfolge von den drei Modulen abgespielt werden. Die Logik stellt sicher, dass zunächst über den ATTiny das Modul ausgewählt wird, der Befehl umgesetzt werden kann und dann, mit zeitlichem Verzug, der Track ausgewählt oder eine andere Funktion des Moduls aufgerufen wird.

Beispiel:

Im folgenden Beispiel wird bei Aufruf des:

- DCC-Befehl "3" der erste Sound des ersten Moduls (JP6500) abgerufen
- DCC-Befehl "4" der erste Sound des zweiten Moduls (DFPlayer Mini) abgerufen
- DCC-Befehl "5" der erste Sound des dritten Moduls (DFPlayer Mini) abgerufen

	Dialog	Z. Arduino schicken	Zeile einfüge	Lösche Zeilen Zeilen Zeilen Zeilen	Aus-oder Einblenden Lösche Tabelle Optionen Help					
1			3MC	AutoDetboard arduino:avr:nano:cpu=atmega328fullmempref progra						
	Aktiv Filter	oder Name	rp Star wer		Autiliei Autiliei –	Start LedNr			InCh	
105					<u> </u>	-	-			
106								\rightarrow	\rightarrow	
107				Modul 1 JQ6500				\rightarrow	\rightarrow	-
108	/	3 Ro	nt I	startet MonoFlop (MF) 1	Mono-Flop MonoFlop(MF11, #InCh, 0.1 Sek)			1	0	
109	V	MF11 Ro		sendet Select Befehl für Modul 1	Pin MP3-Modul definieren MP3 SELECT MODULE(#LED, #Inch, 1)	1	^ C1-1	1	0	0
110	V	3 Ro		startet MonoFlop (MF) 2	Mono-Flop MonoFlop(MF12, #Inch, 1 Sek)			1	0	
111	·	3 Ro	ot	startet MonoFlop (MF) 3	Mono-Flop MonoFlop(MF13, #InCh, 1.1 Sek)			1	0	
112	4	MF12		logische Verknüpfung der drei MFs	D Logische Verknüpfung Logic(MF12u13, NOT #InCh AND MF13)			1	0	
113	V	MF12u13 Ro	ot	spielt Track 1 vom JP6500 ab (Welding)	Titel # abspielen MP3 TRACK(#LED, #Inch, MP3 PLAY TRACK, 1)	1	^ C1-1	1	0	0
114										
115			100-100	Modul 2 DFPlayer Mini						
116	V	4 Ro		startet MF1	Mono-Flop MonoFlop(MF21, #InCh, 0.1 Sek)			1	0	4
117	1	MF21 Ro		sendet Select Befehl für Modul 2	Pin MP3-Modul definieren MP3_SELECT_MODULE(#LED, #InCh, 2)	1	^ C1-1	1	0	0
118	V	4 Ro		startet MF2	Mono-Flop MonoFlop(MF22, #InCh, 1 Sek)			1	0	
119	V	4 Ro	ot 📗	startet MF3	Mono-Flop MonoFlop(MF23, #InCh, 1.1 Sek)			1	0	
120	1	MF22		logische Verknüpfung der drei MFs	D Logische Verknüpfung Logic(MF22u23, NOT #InCh AND MF23)			1	0	
121	·	MF22u23 Ro	ot	spielt Track 1 aus Root vonn DFPlayer 1 ab	Titel # abspielen MP3_TRACK(#LED, #InCh, MP3_PLAY_TRACK, 1)	1	^ C1-1	1	0	0
122				L L IS DEDL. MILL				_	\rightarrow	
123 124	_	5 Ro		Modul3 DFPlayer Mini startet MF1	Mono-Flop MonoFlop(MF31, #InCh, 0.1 Sek)	-		1	0	
124	7	5 Ro MF31 Ro		sendet Select Befehl für Modul 3	Mono-Flop MonoFlop(MF31, #InCh, 0.1 Sek) Pin MP3-Modul definieren MP3 SELECT MODULE(#LED, #InCh, 3)	4	^ C1-1	1	0	0
126	7	5 Ro		startet MF2	Mono-Flop MonoFlop MonoFlop MF3_SELECT_MODULE(#LED, #Inch, 3)	-	C1-1	1	0	U
127	· ·	5 Ro		startet ME3	Mono-Flop MonoFlop(MF33, #Inch, 1.1 Sek)			1	0	
128	1	MF32	л	logische Verknüpfung der drei MFs	D Logische Verknüpfung Logisc(MF32u33, NOT #InCh AND MF33)	-		1	0	_
129	V	MF32u33 Ro	nt .	spielt Track 1 aus Root vonn DFPlayer 2 ab	Titel # abspielen MP3 TRACK(#LED, #InCh, MP3 PLAY TRACK, 1)	1	^ C1-1	1	0	0
130		52055		opion ridor ridor root form bit ridyor 2 db	- Man (Wally 11 2 1 bit 100 kg 2)			-	-	-
131	~			Welding Modul 1 JQ6500						
132				III A TOP IN THE REAL PROPERTY.						
133	V	MF12u13		Imported_Pattern (pc)	Muster Pattern_Configurator // Activation: N_Buttons1InCh_to_TmpVar1(#InCh, 1)PatternT26	2	1	1	0	0
134				Internal Control of the						

Über parallel angeschlossene Test-LEDs, Stichwort Mini-Verteiler, kann der Ablauf optisch sehr gut verfolgt werden. Andere/kürzere Zeitintervalle für die MonoFlops sind möglich und ggf. durch Tests zu ermitteln.

In Zeile 133 wird über die Variable MF12u13 zeitgleich mit dem dazu gehörigen Geräusch vom Sound-Modul 2 ein Schweißlicht ausgelöst. Geräuschlänge und Länge des Lichts können leicht durch Anpassung des Schweisslicht über den Pattern-Configurator angepasst werden.

Damit man den Schweißer nicht immer persönlich wecken muss, hier eine Lösung mit der Zufallsschaltung (Random-Funktion).

100	¥	4	<u> </u>	¥ ¥ ¥			~	-	-	4
			Modul 1 JQ6500							
1	3	AnAus 0		Zufallsschaltung 1 Ausgang	Random(WL1, #InCh, RM_NORMAL, 3 sec, 15 sec, 1 sec, 1 sec)			1	0	
V	WL1	Rot	startet MonoFlop (MF) 1	Mono-Flop	MonoFlop(MF11, #InCh, 0.1 Sek)			1	0	i
V	MF11	Rot	sendet Select Befehl für Modul 1	Pin MP3-Modul definieren	MP3 SELECT MODULE(#LED, #InCh, 1)	1	^ C1-1	1	0	•
1	WL1	Rot	startet MonoFlop (MF) 2	Mono-Flop	MonoFlop(MF12, #InCh, 1 Sek)			1	0	•
4	WL1	Rot	startet MonoFlop (MF) 3	Mono-Flop	MonoFlop(MF13, #InCh, 1.1 Sek)			1	0	Ī
1	MF12		logische Verknüpfung der drei MFs	D Logische Verknüpfung	Logic(MF12u13, NOT #InCh AND MF13)			1	0	Ī
1	MF12u13	Rot	spielt Track 1 vom JP6500 ab (Welding)	Titel # abspielen	MP3_TRACK(#LED, #InCh, MP3_PLAY_TRACK, 1)	1	^ C1-1	1	0	Ī
1	MF12u13		Imported_Pattern (pc)	Muster Pattern_Configurator	// Activation: N_Buttons1InCh_to_TmpVar1(#InCh, 1)PatternT26	2	1	1	0	
1				■ LED Nummer manipulieren	// Next_LED(-2)	3	-2	0	0	
			Modul 2 DFPlayer Mini							
~	4	Rot	startet MF1	Mono-Flop	MonoFlop(MF21, #InCh, 0.1 Sek)			1	0	Ī
V	MF21	Rot	sendet Select Befehl für Modul 2	 Pin MP3-Modul definieren 	MP3_SELECT_MODULE(#LED, #InCh, 2)	1	^ C1-1	1	0	Ī
V	4	Rot	startet MF2		MonoFlop(MF22, #InCh, 1 Sek)			1	0	Ī
1	4	Rot	startet MF3		MonoFlop(MF23, #InCh, 1.1 Sek)			1	0	ĺ
V	MF22		logische Verknüpfung der drei MFs	D Logische Verknüpfung	Logic(MF22u23, NOT #InCh AND MF23)			1	0	Ĺ
~	MF22u23	Rot	spielt Track 1 aus Root vonn DFPlayer 1 ab	Titel # abspielen	MP3_TRACK(#LED, #InCh, MP3_PLAY_TRACK, 1)	1	^ C1-1	1	0	_
			Modul3 DFPlayer Mini							
V	5	Rot	startet MF1	Mono-Flop	MonoFlop(MF31, #InCh, 0.1 Sek)			1	0	i
V	MF31	Rot	sendet Select Befehl für Modul 3	Pin MP3-Modul definieren	MP3_SELECT_MODULE(#LED, #InCh, 3)	1	^ C1-1	1	0	Ī
1	5	Rot	startet MF2		MonoFlop(MF32, #InCh, 1 Sek)			1	0	Ī
V	5	Rot	startet MF3		MonoFlop(MF33, #InCh, 1.1 Sek)			1	0	Ĺ
V	MF32		logische Verknüpfung der drei MFs	D Logische Verknüpfung	Logic(MF32u33, NOT #InCh AND MF33)			1	0	ĺ
V	MF32u33	Rot	spielt Track 1 aus Root vonn DFPlayer 2 ab	■ Titel # abspielen	MP3 TRACK(#LED, #Inch, MP3 PLAY TRACK, 1)	1	^ C1-1	1 1	0	

From:

https://wiki.mobaledlib.de/ - MobaLedLib Wiki

Permanent link:

https://wiki.mobaledlib.de/anleitungen/spezial/sound_servoplatine?rev=1727427395

Last update: **2024/09/27 08:56**



https://wiki.mobaledlib.de/
Printed on 2025/11/28 00:55