

Sound Servoplatine

Über die Servo-Platine 510 können drei Soundmodule angesteuert werden. Es können JQ6500 Module oder MP3-TF16-p/DFPlayer Mini Module verwendet werden. 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.

ATTiny 85

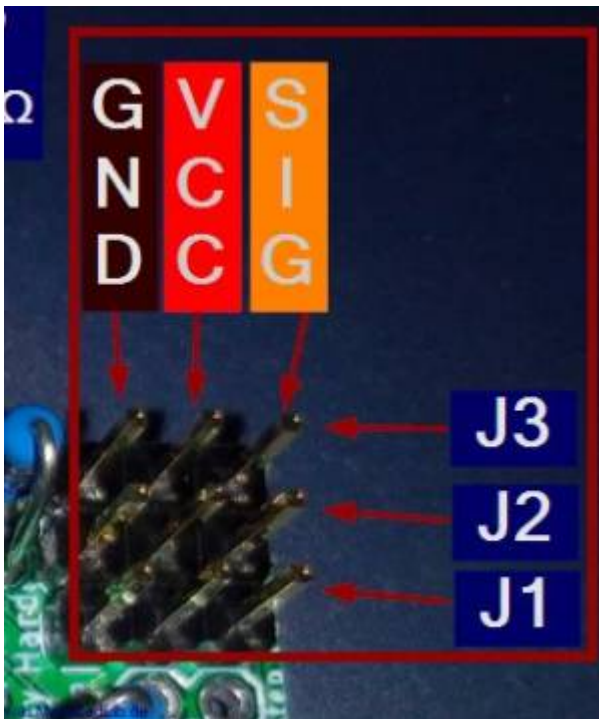
Zunächst muss der ATTiny für die Servoplatine programmiert werden.

Die Anleitung dazu [Programmierung von Attinys für Servo oder Charlieplexing](#) ist im WIKI zu finden.

Im Pattern-Configurator unter spezielle Module **Servo-MP3** auswählen.

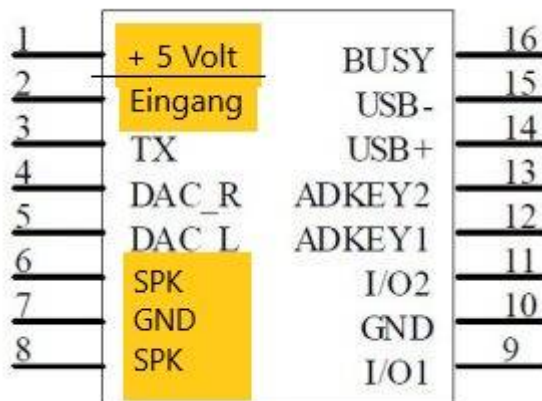
The screenshot shows a 'Pattern Configurator' dialog box with three tabs: 'Beispiele', 'Spezielle Module', and 'Extras'. The 'Spezielle Module' tab is active. The text inside the dialog reads: 'Diese Seite enthält Funktionen mit denen spezielle, auf dem ATTiny basierte Module programmiert und getestet werden können. by Hardi'. Below this, there are two sections. The first section is titled 'Programmieradapter' and contains a button labeled 'Prog. ISP'. The text below the button says: 'Ein ATTiny hat keinen USB Anschluss. Darum benötigt man zur Programmierung einen Programmieradapter (In Circuit Programmer). Das kann ein Arduino mit besonderen Programm sein. Mit dem Knopf Links wird das Programm zu Arduino übertragen. Dieses Programm kann auf dem Tiny_UniProg und auf einen "Nackten" Arduino in einem Steckbrett eingesetzt werden. Der "HV Reset" ist allerdings nur mit der "Tiny_UniProg" Platine möglich. Er wird für das Programmieren des Servo Programms benötigt.' The second section is titled 'Charlieplexing | Servo | Servo-MP3' and contains a button labeled 'Prog.Servo/MP3'. The text below the button says: 'Man kann das Servo-Modul auf zur Ansteuerung von Soundmodulen nutzen (JQ6500/MP3-TF-16p) nutzen. Dabei werden die Soundmodule mit je einem 1k Widerstand an die Signal-Pins der Servoausgänge angeschlossen. Zusätzlich ist die Masse zu verbinden. Die Stromversorgung kann man getrennt lösen - meist funktioniert es aber auch die 5V von den Servoanschlüssen zu nutzen. Zur Programmierung des ATTiny85 wird ein Programmieradapter benötigt (siehe oben) in der der ATTiny eingesteckt wird. Achtung: Die Software für das Servo Modul ist noch in der Entwicklung.' At the bottom right of the dialog is a button labeled 'Dialog schließen'.

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 Modul3.



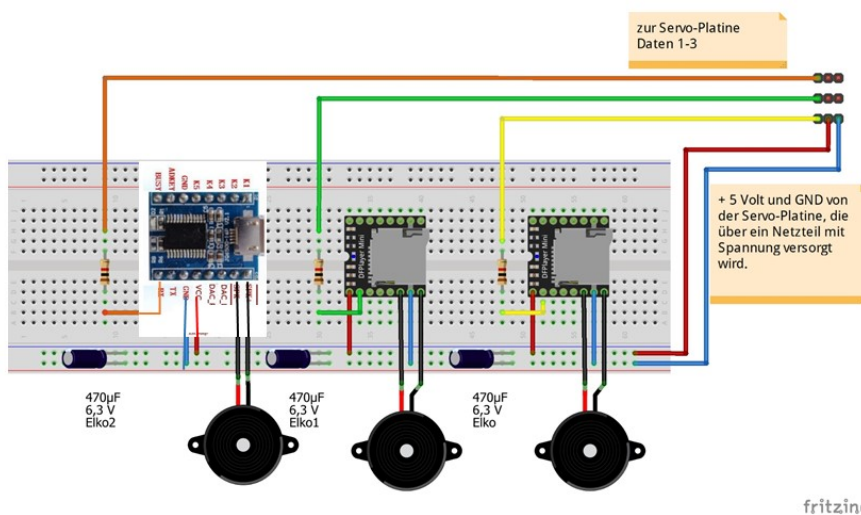
JQ6500 und MP3-TF16-p/DFPlayer Mini





Schaltung

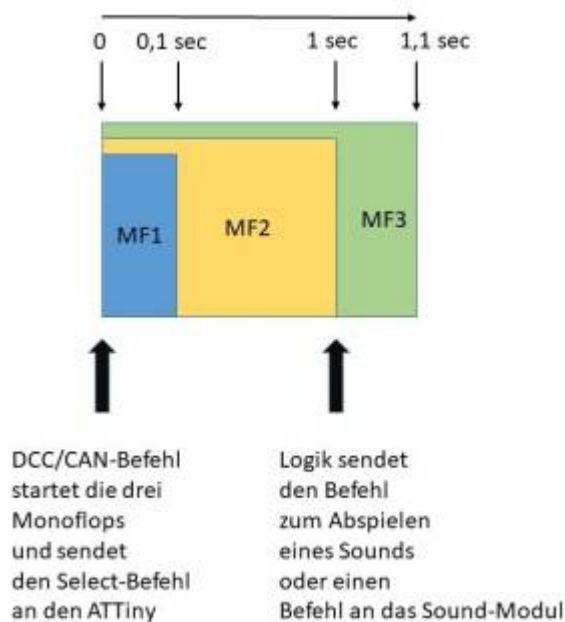
Die drei Ausgänge der Servo-Platine werden nun über je einen 1 kOhm Widerstand mit den Eingängen der Sound-Module verbunden. Die Versorgungsspannung sollte über eine Verteilerplatine mit angeschlossener stabiler Spannungsversorgung erfolgen, da die Leistung an der Hauptplatine zur Versorgung der Sound-Module nicht ausreicht. Die 470uF Elektrolytkondensatoren dienen als Puffer für die recht hohen Einschaltströme der Soundmodule. Eine separate 5V Spannungsversorgung ist ebenfalls möglich. Dann müssen GND und -5 Volt verbunden werden.



Programm-Generator

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

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

Aktiv	Filter	Adresse oder Name	Typ	Startwert	Beschreibung	Verteiler-Nummer	Stecker-Nummer	Leuch	Name	Beleuchtung, Sound, oder andere Effekte	Start LedNr	LEDs	InCh	Loc InCh	LED/ Sound Kanal
-COM6 AutoDet -board arduino-ser-nano-cpu-amega328old-avr-programmer-arduino-arduno															
-COM7 AutoDet -board arduino-ser-nano-cpu-amega328old-avr-programmer-arduino-arduno															
105					Modul 1 JQ6500										
108	✓	3	Rot		startet MonoFlo (MF) 1				! Mono-Flop	MonoFlo(MF11, #InCh, 0.1 Sek)					
109	✓	MF11	Rot		sendet Select Befehl für Modul 1				Pin MP3-Modul definieren	MP3_SELECT_MODULE(#LED, #InCh, 1)	1	^ C1-1			0
110	✓	3	Rot		startet MonoFlo (MF) 2				! Mono-Flop	MonoFlo(MF12, #InCh, 1 Sek)					
111	✓	3	Rot		startet MonoFlo (MF) 3				! Mono-Flop	MonoFlo(MF13, #InCh, 1.1 Sek)					
112	✓	MF12	Rot		logische Verknüpfung der drei MFs				! Logische Verknüpfung	Logic(MF12u13, NOT #InCh AND MF13)					
113	✓	MF12u13	Rot		spielt Track 1 vom JP6500 ab (Welding)				! Titel # abspielen	MP3_TRACK(#LED, #InCh, MP3_PLAY_TRACK, 1)	1	^ C1-1			0
114															
115					Modul 2 DFPlayer Mini										
116	✓	4	Rot		startet MF1				! Mono-Flop	MonoFlo(MF21, #InCh, 0.1 Sek)					
117	✓	MF21	Rot		sendet Select Befehl für Modul 2				Pin MP3-Modul definieren	MP3_SELECT_MODULE(#LED, #InCh, 2)	1	^ C1-1			0
118	✓	4	Rot		startet MF2				! Mono-Flop	MonoFlo(MF22, #InCh, 1 Sek)					
119	✓	4	Rot		startet MF3				! Mono-Flop	MonoFlo(MF23, #InCh, 1.1 Sek)					
120	✓	MF22	Rot		logische Verknüpfung der drei MFs				! Logische Verknüpfung	Logic(MF22u23, NOT #InCh AND MF23)					
121	✓	MF22u23	Rot		spielt Track 1 aus Root von DFPlayer 1 ab				! Titel # abspielen	MP3_TRACK(#LED, #InCh, MP3_PLAY_TRACK, 1)	1	^ C1-1			0
122															
123					Modul3 DFPlayer Mini										
124	✓	5	Rot		startet MF1				! Mono-Flop	MonoFlo(MF31, #InCh, 0.1 Sek)					
125	✓	MF31	Rot		sendet Select Befehl für Modul 3				Pin MP3-Modul definieren	MP3_SELECT_MODULE(#LED, #InCh, 3)	1	^ C1-1			0
126	✓	5	Rot		startet MF2				! Mono-Flop	MonoFlo(MF32, #InCh, 1 Sek)					
127	✓	5	Rot		startet MF3				! Mono-Flop	MonoFlo(MF33, #InCh, 1.1 Sek)					
128	✓	MF32	Rot		logische Verknüpfung der drei MFs				! Logische Verknüpfung	Logic(MF32u33, NOT #InCh AND MF33)					
129	✓	MF32u33	Rot		spielt Track 1 aus Root von DFPlayer 2 ab				! Titel # abspielen	MP3_TRACK(#LED, #InCh, MP3_PLAY_TRACK, 1)	1	^ C1-1			0
130															
131					Welding Modul 1 JQ6500										
132	✓	MF12u13	Rot		Imported_Pattern (pc)				Muster Pattern_Configurator	// Activation: N_ButtonsInCh_to_TapVar1(#InCh, 1)Pattern26	2	1			0
133															
134															

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 Schweißler nicht immer persönlich wecken muss, hier eine Lösung mit der Zufallsschaltung (Random-Funktion).

Aktiv	Filter	Adresse oder Name	Typ	Startwert	Beschreibung	Verteiler-Nummer	Stecker-Nummer	Leuch	Name	Beleuchtung, Sound, oder andere Effekte	Start LedNr	LEDs	InCh	Loc InCh	LED/ Sound Kanal
-COM6 AutoDet -board arduino-ser-nano-cpu-amega328old-avr-programmer-arduino-arduno															
-COM7 AutoDet -board arduino-ser-nano-cpu-amega328old-avr-programmer-arduino-arduno															
106					Modul 1 JQ6500										
108															
109	✓	3	AnAus	0					! Zufallsschaltung 1 Ausgang	Random(WL1, #InCh, RN_NORMAL, 3 sec, 15 sec, 1 sec, 1 sec)					1
110															
111	✓	WL1	Rot		startet MonoFlo (MF) 1				! Mono-Flop	MonoFlo(MF11, #InCh, 0.1 Sek)					
112	✓	MF11	Rot		sendet Select Befehl für Modul 1				Pin MP3-Modul definieren	MP3_SELECT_MODULE(#LED, #InCh, 1)	1	^ C1-1			0
113	✓	WL1	Rot		startet MonoFlo (MF) 2				! Mono-Flop	MonoFlo(MF12, #InCh, 1 Sek)					
114	✓	WL1	Rot		startet MonoFlo (MF) 3				! Mono-Flop	MonoFlo(MF13, #InCh, 1.1 Sek)					
115	✓	MF12	Rot		logische Verknüpfung der drei MFs				! Logische Verknüpfung	Logic(MF12u13, NOT #InCh AND MF13)					
116	✓	MF12u13	Rot		spielt Track 1 vom JP6500 ab (Welding)				! Titel # abspielen	MP3_TRACK(#LED, #InCh, MP3_PLAY_TRACK, 1)	1	^ C1-1			0
117	✓	MF12u13	Rot		Imported_Pattern (pc)				Muster Pattern_Configurator	// Activation: N_ButtonsInCh_to_TapVar1(#InCh, 1)Pattern26	2	1			0
118															
119	✓								! LED Nummer manipulieren	// Next_LED(-2)	3	-2			0
120															
121					Modul 2 DFPlayer Mini										
122	✓	4	Rot		startet MF1				! Mono-Flop	MonoFlo(MF21, #InCh, 0.1 Sek)					
123	✓	MF21	Rot		sendet Select Befehl für Modul 2				Pin MP3-Modul definieren	MP3_SELECT_MODULE(#LED, #InCh, 2)	1	^ C1-1			0
124	✓	4	Rot		startet MF2				! Mono-Flop	MonoFlo(MF22, #InCh, 1 Sek)					
125	✓	4	Rot		startet MF3				! Mono-Flop	MonoFlo(MF23, #InCh, 1.1 Sek)					
126	✓	MF22	Rot		logische Verknüpfung der drei MFs				! Logische Verknüpfung	Logic(MF22u23, NOT #InCh AND MF23)					
127	✓	MF22u23	Rot		spielt Track 1 aus Root von DFPlayer 1 ab				! Titel # abspielen	MP3_TRACK(#LED, #InCh, MP3_PLAY_TRACK, 1)	1	^ C1-1			0
128															
129					Modul3 DFPlayer Mini										
130	✓	5	Rot		startet MF1				! Mono-Flop	MonoFlo(MF31, #InCh, 0.1 Sek)					
131	✓	MF31	Rot		sendet Select Befehl für Modul 3				Pin MP3-Modul definieren	MP3_SELECT_MODULE(#LED, #InCh, 3)	1	^ C1-1			0
132	✓	5	Rot		startet MF2				! Mono-Flop	MonoFlo(MF32, #InCh, 1 Sek)					
133	✓	5	Rot		startet MF3				! Mono-Flop	MonoFlo(MF33, #InCh, 1.1 Sek)					
134	✓	MF32	Rot		logische Verknüpfung der drei MFs				! Logische Verknüpfung	Logic(MF32u33, NOT #InCh AND MF33)					
135	✓	MF32u33	Rot		spielt Track 1 aus Root von DFPlayer 2 ab				! Titel # abspielen	MP3_TRACK(#LED, #InCh, MP3_PLAY_TRACK, 1)	1	^ C1-1			0
136															
137															

From:
<https://wiki.mobaledlib.de/> - MobaLedLib Wiki

Permanent link:
https://wiki.mobaledlib.de/anleitungen/spezial/sound_servoplatine?rev=1639688482

Last update: 2021/12/16 22:01

