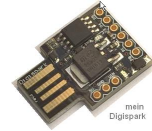


Als Bastelobjekt soll ein Bewegungsmelder einen Sound auslösen.

Es werden genutzt:

1. Ein PIR-Sensor wie HC-SR505^{*)}
2. Ein DFPlayer-Mini
3. Ein Arduino Uno oder ein DigiSpark
4. Ein Kleinlautsprecher mit max. 3W



Der DFPlayer-Mini ist an sich ein autarkes Gerät welches sich auch über eine serielle Schnittstelle steuern lässt. Auf einer MicroSD-Karte können Soundfiles als mp3, wav oder wmv's gespeichert werden. Die Files müssen sich in einem Ordner x:\mp3\ befinden. Zwecks Identifizierung wird den Titeln eine 4-stellige Nummer vorgesetzt. 0001Halloagin.mp3 , 0002xxx... usw..

^{*)} Mein PIR-Sensor nennt sich: Mini IR Pyroelectric Infrared PIR Motion Human Sensor..... hat einen OUT-Ausgang der mit einem Digital-Input des Arduino Uno klar kommt. Beim DigiSpark funktioniert dies auf dem Eingang PB2 des inhärenten Attiny85 nicht. Leider habe ich kein passendes Schaltbild vorliegen. Ich habe einen BS170 eingesetzt um dieses Problem zu beheben.

Von DF-Robot gibt es zum DFPlayer eine komfortable Software mit der man den Player komfortabel in einem Sketch steuern kann. Getestet mit einem Uno. Geplant ist aber der Einsatz eines DigiSparks.

Arduino UNO

Statt der DFPlayer-Software-Includes ist hier im Sketch die serielle Schnittstellensoftware zum DF-Player vorhanden. Im Sketch für den UNO:

```
#include <SoftwareSerial.h>
SoftwareSerial mySerial(10,11);
```

DigiSpark

Hier muss man statt der SoftwareSerial.h die SoftSerial.h von DigiSpark nutzen.

Möchte man die komfortable Lib von DF-Robot nutzen, so muss man hier mit einem einfachen Editor oder z.B. notepad++ in den Dateien

DFPlayer_Mini_Mp3.h und -.cpp ersetzen: "SoftwareSerial" durch "SoftSerial"

```
#include <SoftSerial.h> // von DigiSpark! Statt SoftwareSerial.h
#include <DFPlayer_Mini_Mp3_with_DigiSpark.h> // meine aktualisierte Version (s.o.)
SoftSerial mySerial(3, 0); // RX, TX (Rx brauche ich nicht)
```

Es geht weiter



Die Schaltung und der Sketch sind nun soweit fertig.

Die serielle Routine des „Unos“ habe ich nun auch im DigiSpark- Sketch eingebaut.

Ein Problem war die Signale des DFPlayers auf dem DigiSpark einzulesen.

Hier habe ich Pegelanpassungstechnisch rumgetrickst und die Levelshifter so gebaut, dass auch die Pegel in etwa am Digispark stimmen. Ich hoffe ich habe keinen I/O-Pin vergewaltigt ☺.

Ob das jetzt so korrekt ist?? -> Also Schaltung ohne Gewähr!

Mir fehlen auch die Grundlagenkenntnisse zum Attiny45 auf dem Digispark.

Gibt es da überhaupt interne Input- Pullups? Sind die Fuses für meinen Einsatz richtig gesetzt?

Fazit: Viel Ahnung – wenig Wissen ☺.

Da ich den USB- PIN3 genutzt habe muss der Spielzeit- Schalter offen sein, wenn über die UBS-Verbindung ein Sketch- Upload erfolgen soll.

Nach dem Aufspielen des Sketches sollte man die Schaltung auch nur mit 5V betreiben.

Ich habe hierzu in meinem aktiven USB-HUB an der obersten Buchse die Datenleitungen aufgetrennt.

So kann ich nun einfach umstecken.

Zum Sketch:

Da mein PIR-Sensor immer für ca. 20sec den OUT schaltet kann man im Sketch- Setup eine Delaytime von 25s vorsehen um ein sofortiges Starten beim Einstecken der Schaltung zu verhindern. Zu einer Verzögerung von ca. 6s führt ja immer beim DigiSpark die USB- „Lausch“-Zeit.

Über den Spielzeitschalter kann gewählt werden:

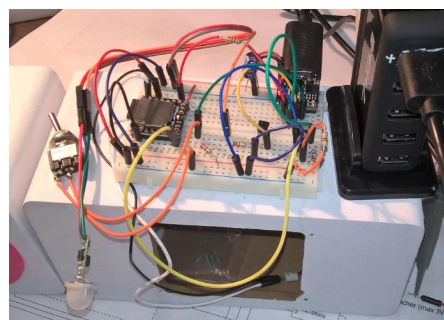
1. 30s Spielzeit (Schalter offen auch wenn Sketch- Upload erfolgen soll!))
Bei Aktivierung wird das Lied für die im Sketch definierte Zeit abgespielt.
Bei jeder neuen Aktivierung wird per Befehl play_next das nächste Lied gespielt.
2. Track (Schalter geschlossen)
Bei Aktivierung wird das Lied zu Ende gespielt. Kriterium ist hier das Busy- Signal des DFPlayers.
Bei jeder neuen Aktivierung wird per Befehl play_next das nächste Lied gespielt.

Der Schalter kann während des Betriebes umgelegt werden. Die vorherige Einstellung wird ggf. zu Ende geführt.

Mit den Tastern S2 und S3 (s. Schaltplan)
kann man je nach Betätigungslänge einstellen:

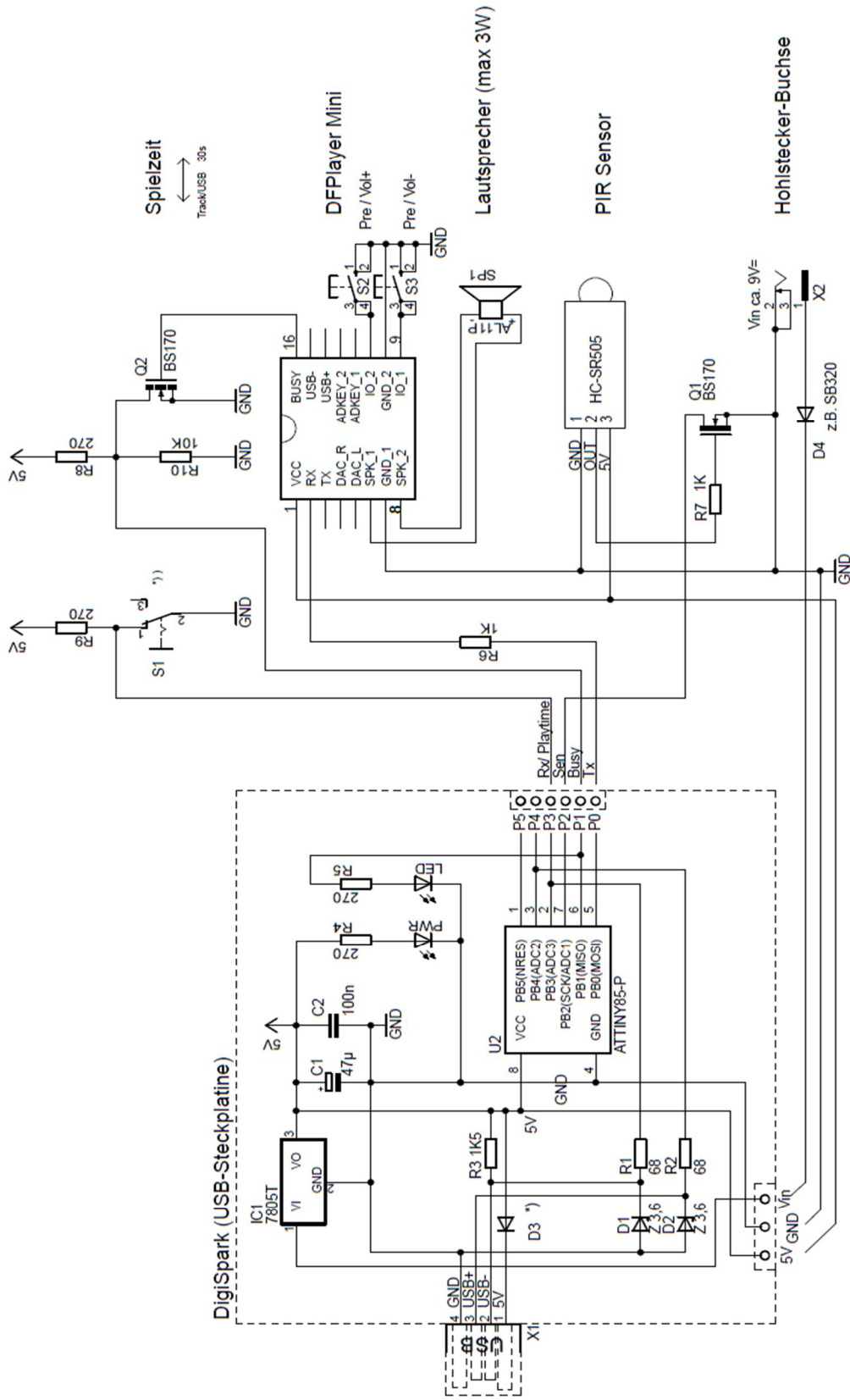
1. Kurz - Nächstes/Vorheriges Lied
2. Lang - Lautstärke + oder -

Testaufbau:



Manches Mal „brummt“ die ganze Schaltung und resettet sich selbst. Ich denke das hängt mit meiner Spannungsversorgung über die USB-Spannung zusammen.

Als nächstes werde ich das Ganze mal auf einer Rasterplatine aufbauen.



! Schaltung ohne Gewähr - siehe Beschreibung !

Title:	PIR-Sensor_DDFPlayer_DigiSpark
Document Number:	PIR-Sensor_DDFPlayer_DigiSpark.sch
Design:	Leopoldi
Date:	
REV:	1.1
Sheet	1/1

*) 1-2 Playtime = Track Time
1-3 Playtime 30s + Stellung bei USB-Upload !!!!

*) +5V AUTOSELECTOR- MBR0520LT (20V/0.5A) - SB320 / SR302 ?