

Mikrocontroller-Modul für Flipperautomaten

Variante B, Version 2015

1 Einleitung

Durch ein Mikrocontroller-Modul kann der Flipperautomat mit folgenden Funktionen erweitert werden:

- Sound: Abspielen von Audiodateien
- Display: Anzeige von Punktestand und Lauftext sowie weiteren Informationen
- Spiellogik: Zählen von Punkten, Spielende nach definierbarer Anzahl Bällen, permanente Speicherung des Punkterekords (*Highscore*)

Das Abspielen von Audiodateien und die Aktionen der Spiellogik werden durch das Schliessen von Kontakten ausgelöst. Die konkreten Aktionen können für jeden Kontakt ohne Programmierkenntnisse konfiguriert werden. Es können bis zu zehn unterschiedliche Aktionen definiert werden.

Audio- und Konfigurationsdateien sowie der Punkterekord werden auf einer SD-Karte gespeichert.

2 Anschlüsse

2.1 Stromversorgung

Die Stromversorgung wird über das USB-Kabel sichergestellt.

2.2 Kontakte und Lautsprecher

Die Kontakte des Automaten und die Lautsprecher werden über eine 14-polige Schraubklemme angeschlossen.



Abbildung 1 Schraubklemme

Auf den Anschlüssen IN1 bis IN10 stehen Eingänge für den Anschluss von Kontaktschaltern zu Verfügung. Wenn einer dieser Pins durch Schliessen eines Schalters mit der Masse (GND) verbunden wird, löst dies eine oder mehrere konfigurierbare Aktionen aus.

An den Anschlüssen SPK+ und SPK- wird der Lautsprecher angeschlossen.

Anschluss	Bezeichnung	Bedeutung	Anschluss	Bezeichnung	Bedeutung
1	IN1	Eingang 1 (A)	8	IN8	Eingang 8
2	IN2	Eingang 2 (A)	9	IN9	Eingang 9
3	IN3	Eingang 3 (A)	10	-	–
4	IN4	Eingang 4 (A)	11	GND	Masse 0 V
5	IN5	Eingang 5 (A)	12	5V	Spannungsversorgung 5 V
6	IN6	Eingang 6 (A)	13	SPK-	Lautsprecher Masse 0 V
7	IN7	Eingang 7 (A)	14	SPK+	Lautsprecher

Tabelle 1 Anschlüsse Schraubklemme

2.3 Anschluss von Kontakten

Damit das Mikrocontroller-Modul auf einen Kontakt reagieren kann, müssen die Kontakte auf dem Spielfeld mit dem Modul verbunden werden. Dazu wird der Kontakt zwischen dem gewünschten Eingang und der Masse (0 V) geschaltet. Wenn mehrere Kontakte die gleiche Bedeutung haben, so können sie parallel am gleichen Eingang angeschlossen werden (siehe [Abbildung 2](#)).

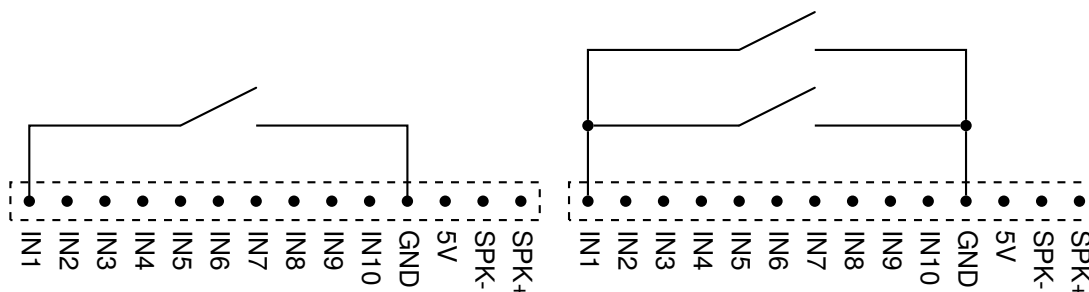


Abbildung 2 Anschluss von Kontakten

2.4 LED-Matrixanzeige

An das Steuerungsmodul können bis zu vier 32×8-LED-Matrizen von SURE Electronics angeschlossen werden. Diese Anzeigen werden über ein 16-poliges Flachbandkabel in Serie geschaltet. Die erste Matrix wird am Steuerungsmodul angeschlossen. Für das Kabel ist auf der Platine eine 2×8-polige Wannensteckleiste vorhanden (siehe [Abbildung 3](#)).

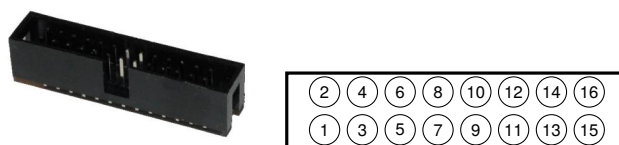


Abbildung 3 Wannensteckleiste mit Pinbezeichnung

Da alle Matrizen über den gleichen Anschluss angesteuert werden, wird die Matrix, mit welcher kommuniziert werden soll, über ein sogenanntes *chip select*-Signal ausgewählt. Die Pins 1 bis 4 dienen diesem Zweck.

Damit das funktioniert, muss auf jeder Matrix eine entsprechende, eindeutige Adresse eingestellt werden.

Zur Kommunikation mit der Matrix werden die Pins 5 und 7 benötigt.

Die Spannungsversorgung von 5 V und die Masse (GND) können auf mehreren Pins angeschlossen werden. Es spielt keine Rolle, welcher Pin gewählt wird.

Pin	Bezeichnung	Bedeutung	Pin	Bezeichnung	Bedeutung
1	CS2	Auswahl Matrix 2 (<i>chip select 2</i>)	9	OSC	(nicht benötigt)
2	CS3	Auswahl Matrix 3 (<i>chip select 3</i>)	10	SYNC	(nicht benötigt)
3	CS1	Auswahl Matrix 1 (<i>chip select 1</i>)	11	GND	Masse (0 V)
4	CS4	Auswahl Matrix 4 (<i>chip select 4</i>)	12	VCC	Spannungsversorgung 5 V
5	WR	Schreibtakt (<i>write clock</i>)	13	GND	Masse (0 V)
6	RD	Lesetakkt (<i>read clock</i> , nicht benötigt)	14	VCC	Spannungsversorgung 5 V
7	DATA	Datenleitung	15	GND	Masse (0 V)
8	GND	Masse (0 V)	16	VCC	Spannungsversorgung 5 V

Tabelle 2 Anschluss LED-Matrixanzeige

3 Konfiguration

Das Modul kann über die Speicherkarte konfiguriert werden.

3.1 Speicherkarte

Die SD-Speicherkarte muss mit dem **FAT16**-Dateisystem formatiert sein. Dateinamen müssen alle **klein** geschrieben werden, und dürfen höchstens acht Zeichen lang sein. Die Dateiendung darf aus maximal drei Zeichen bestehen.

3.2 Ereignisse

Das Mikrocontroller-Modul reagiert auf bestimmte Ereignisse, indem es Dateien mit einem entsprechenden Namen auf der Speicherkarte sucht. **Tabelle 3** zeigt die Zuordnung von Ereignissen zu Dateinamen.

Ereignis	Dateiname
Das Modul ist eingeschaltet worden.	init
Ein Eingang ist aktiviert worden.	in01 bis in10
Es ist ein neuer Highscore erreicht worden.	high
Das Spiel ist vorbei.	over

Tabelle 3 Ereignisse

3.3 Dateitypen

Für jedes Ereignis können mehrere Dateien mit unterschiedlicher Endung auf der Karte vorhanden sein (siehe **Tabelle 4**). Je nach Dateiendung wird eine andere Aktion durchgeführt.

Dateiendung	Aktion
.prg	Der Inhalt der Datei wird als Befehl ausgeführt.
.txt	Der Inhalt der Textdatei wird als Laufschrift angezeigt.
.wav	Die Audiodatei wird abgespielt.

Tabelle 4 Dateitypen

Beispielsweise wird die Audiodatei **init.wav** abgespielt, wenn der Automat eingeschaltet wird. Der Inhalt der Datei **high.txt** wird als Laufschrift angezeigt, wenn ein neuer Highscore erreicht wird. Die Befehle in der Datei **in05.prg** werden ausgeführt, wenn der Eingang 5 aktiviert wird.

3.4 Befehlsdateien (.prg)

Das Modul speichert den Punktestand und die Anzahl verfügbarer Bälle. Diese Zahlen können durch Befehle geändert werden. **Tabelle 5** zeigt alle verfügbaren Befehle und deren Auswirkung. Mit Ausnahme von **n** (neues Spiel) werden Befehle nur ausgeführt, wenn die Anzahl verfügbarer Bälle nicht Null ist.

Ein Befehl besteht aus einem Zeichen auf welches manchmal eine Zahl folgt. Beispielsweise erhöht der Befehl **+100** den Punktestand um 100 Punkte, der Befehl **n3** beginnt ein neues Spiel mit drei Bällen.

Befehl	Bedeutung
+ Zahl	Erhöht den Punktestand um die angegebene Zahl
* Zahl	Erhöhe den Punktestand langsam um die angegebene Zahl
n Zahl	Neues Spiel: Setzt den Punktestand auf Null und die Anzahl Bälle auf die angegebene Zahl
b	Ballverlust: Reduziert die Anzahl Bälle um eins. Bei Null Bällen wird das Spiel beendet und ein allfälliger Punktestandrekord gespeichert
x	Extraball: Erhöht die Anzahl Bälle um eins

Tabelle 5 Befehle

Mit Hilfe dieser Befehle lässt sich ein Flipperautomat mit vollständiger Spiellogik bauen. Dazu benötigt der Automat einen Sensor, welcher jeden Ballverlust erkennt sowie einen Schalter, um ein neues Spiel zu beginnen.

Alternativ dazu kann ein einfacher Automat gebaut werden, bei welchem das Spiel immer läuft. Dazu muss beim Einschalten des Automaten die Anzahl Bälle auf eine Zahl grösser Null gesetzt werden. Dass heisst, in der Datei `init.prg` muss der Befehl `n1` stehen.

3.5 Textdateien (.txt)

Wenn eine Aktion ausgelöst wird und eine entsprechende Textdatei vorhanden ist, wird der Inhalt der Datei als Lauftext angezeigt. Sobald der ganze Text über die Anzeige gelaufen ist, wird wieder der Punktestand angezeigt.

Der Lauftext kann höchstens 40 Zeichen lang sein.

3.6 Audiodateien (.wav)

Das Steuerungsmodul kann nur Audiodaten im PCM-Format (Puls-Code-Modulation) abspielen. Die Dateien müssen im WAV-Format (Waveform Audio File Format) mit einem Kanal (Mono), einer Bit-Tiefe von 8 und einer Abtastfrequenz von 16000 Hz vorliegen.

Mit dem frei verfügbaren Programm **Audacity** kann jede Audiodatei mit kleinem Aufwand in das geeignete Format umgewandelt werden:

1. Audiodatei öffnen.
2. Menüpunkt **Spuren / Stereosput in Mono umwandeln** auswählen.
3. Projektfrequenz auf 16000 Hz stellen (unten links).
4. Exportieren als **Andere unkomprimierte Dateien** mit Optionen:
 - Header: WAV (Microsoft)
 - Codec: Unsigned 8 bit PCM

4 Zusammenbau

4.1 Bauteile

Für die Realisierung des Moduls werden folgende Bauteile benötigt:

Bauteil	Lieferant	Preis	Anzahl	Total
Arduino Duemilanove	Play-Zone	25.00	1	25.00
Lötplatine 80×120 mm, Punktraster	Play-Zone	9.00	1	9.00
75HC4050 HEX Konverter IC	Conrad	0.60	1	0.60
IC-Sockel 2×8-Pol	Conrad	0.30	1	0.30
Wannenstiftleiste 2×8-Pol (Anschluss Display)	Conrad	0.50	1	0.50
Buchsenleiste 2×8-Pol (Anschluss SD-Kartenleser)	Conrad	0.60	1	0.60
Stiftleiste 1×6-Pol (Anschluss Arduino)	Conrad	0.30	2	0.60
Stiftleiste 1×8-Pol (Anschluss Arduino)	Conrad	0.35	2	0.70
SD-Kartenleser	Play-Zone	8.00	1	8.00
SD-Karte 2 GB	Conrad	6.00	1	6.00
USB-Netzteil	Conrad	11.00	1	11.00
USB-Kabel A-B 80 cm	Play-Zone	3.00	1	3.00
Schraubklemme 6-Pol (Anschluss Kontakte)	Conrad	0.85	2	1.70
Total				67.00

Tabelle 6 Bauteile Grundmodul

4.2 LED-Matrix-Anzeige

Für die Ansteuerung der LED-Matrix-Anzeige müssen die benötigten *chip select*-Pins, die zwei Datenübertragungsleitungen (**WR** und **DATA**) und die Stromversorgung (**VCC** und **GND**) mit dem Arduino verbunden werden. *Diagramm 4* zeigt, wie die Verbindungen realisiert werden müssen.

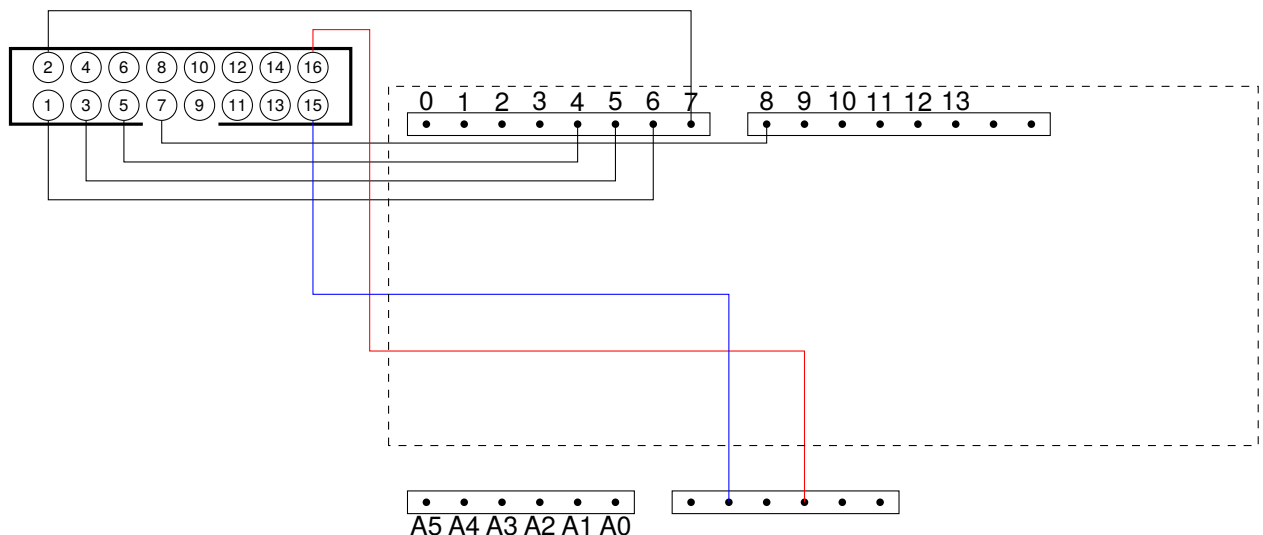


Abbildung 4 Anschluss LED-Matrix

4.3 SD-Kartenleser

Zum Lesen der Befehls- und Audiodateien von der SD-Karte wird ein SD-Kartenleser benötigt. SD-Karten werden über das Serial Peripheral Interface (SPI) angesteuert (siehe *Abbildung 5*). Diese Schnittstelle benötigt vier Datenleitungen.

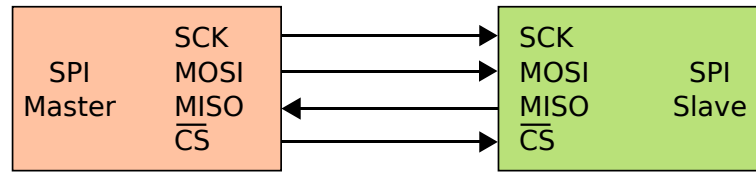


Abbildung 5 Serial Peripheral Interface (SPI)

SD-Karten werden mit 3.3 V betrieben, der Arduino jedoch mit 5 V. Deshalb muss die Spannung der drei Busleitungen **SCK**, **MOSI** und **\overline{CS}** heruntertransformiert werden. Dazu wird ein **74HC4050**-Chip eingesetzt. Dieser IC ist mit sechs Operationsverstärkern bestückt. **Abbildung 6** zeigt, wie der SD-Kartenleser mit dem 74HC4050-Chip und dem Arduino verdrahtet wird.

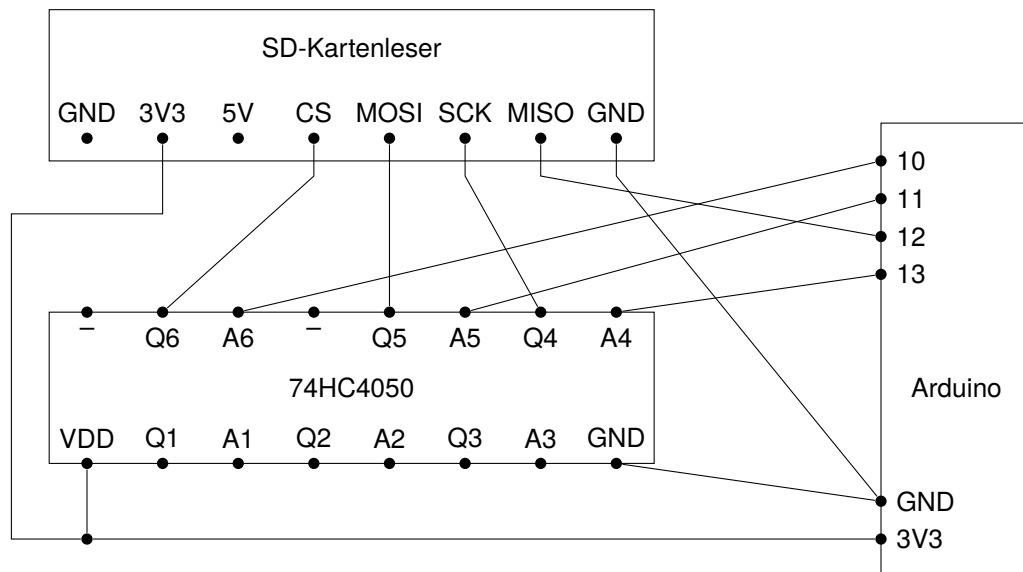


Abbildung 6 Anschluss SD-Kartenleser

4.4 Arduino

Die folgende Tabelle zeigt, wie die Anschlüsse des Arduino belegt sind:

Pin Arduino	Funktion	Anschluss an
A0	Eingang 6 (IN6)	Schraubklemme
A1	Eingang 5 (IN5)	Schraubklemme
A2	Eingang 4 (IN4)	Schraubklemme
A3	Eingang 3 (IN3)	Schraubklemme
A4	Eingang 2 (IN2)	Schraubklemme
A5	Eingang 1 (IN1)	Schraubklemme
0	Eingang 7	Schraubklemme
1	Eingang 8	Schraubklemme
2	Eingang 9	Schraubklemme
3	Eingang 10	Schraubklemme
4	LED-Matrix-Anzeige WR	Steckleiste 2×8
5	LED-Matrix-Anzeige CS1	Steckleiste 2×8
6	LED-Matrix-Anzeige CS2	Steckleiste 2×8
7	LED-Matrix-Anzeige CS3	Steckleiste 2×8
8	LED-Matrix-Anzeige DATA	Steckleiste 2×8
9	Audioausgang	Schraubklemme
10	\overline{CS} SD-Kartenleser	74HC4050
11	SPI MOSI (Master Out Slave In)	74HC4050
12	SPI MISO (Master In Slave Out)	SD-Kartenleser
13	SPI SCK (Serial Clock)	74HC4050

Tabelle 7 Anschluss des Arduino