Zwischenbericht AP Konzept

Projekt: Snake the Game  
Verantwortlich: Dominik Reindl  
Teammitglieder: Dominik Reindl, Diogo Abreu, Ioannis Christodoulakis  
Datum: 16.05.2017  
Revision: 01

Inhalt

[1 Konzept 2](#_Toc482715659)

[1.1 Hardware Definition 2](#_Toc482715660)

[1.2 Schnittstellen 3](#_Toc482715661)

[2 Spieldefinition / Bedienung 4](#_Toc482715662)

[3 Funktionsprinzip 4](#_Toc482715663)

[3.1 Vorstudie zur LED Matrix Ansteuerung 4](#_Toc482715664)

[4 Anhang 5](#_Toc482715665)

[4.1 Glossar 5](#_Toc482715666)

[4.2 Datenblätter 5](#_Toc482715667)

[5 Identifikation 5](#_Toc482715668)

# Konzept

## Hardware Definition

Wir haben uns nach dem Informieren auf folgende Hardware Elemente geeinigt.

|  |  |
| --- | --- |
| Anzeige: Eine LED Matrix 8 x 8, mit rot und grünen LED. | Eingabe: Einzelne Taster für Richtungswechsel (Pro Richtung eine Taste). |
| Ein / Ausschalter | Simple Sound effekte mit einem Piezo Buzzer |
| Logik wird auf ein Arduino gespeichert | Für die Pinerweiterung verwenden wir ein Shift Register |
| Vorwiderstände (LED / Piezo) | Potentiometer zur Lautsrärkenregelung |

## Schnittstellen

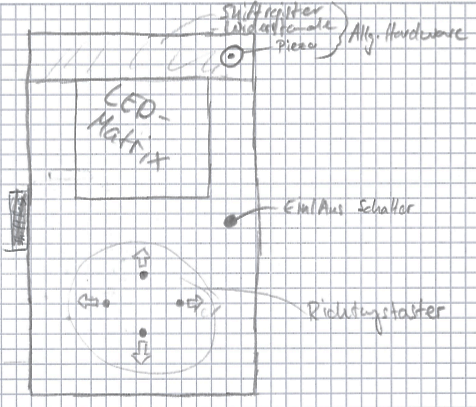
Wir wollen alle Schnittstellen vom Arduino zur Hardware auf einer Lochrasterplatine haben. Für das werden wir ein *Shield* herstellen. Dies muss dann ohne grossen Aufwand auf ein *Arduino-Board* gesteckt werden können, so dass alle notwendigen Verbindungen verknüpft sind.   
In der nachfolgenden Skizze ist ein erster Entwurf für unser *Shield* aufgezeichnet. Je nach dem wo die Pins zur Verbindung für das *Arduino* sein müssen, wird sich das Aussehen leicht anpassen.

Abbildung 1 Konzept vom Shield

Eine weitere wichtige Schnittstelle in unserem Projekt, ist die Kommunikation von der Spielelogik und dem Ausgeben der korrekten Informationen, welche für die Betreibung der LED Matrix zuständig ist. Da wir diese Spielelogik nicht selbst, sondern von einem Informatiker erstellen lassen, müssen wir die Schnittstelle klar definieren. So kamen wir auf folgende Definition:  
Wir haben 8 x 16 LEDs (8 Reihen mit jeweils 8 roten und 8 grünen LED) welche angesteuert werden. Somit wählten wir eine Datenstruktur welches ein zweidimensionales Array fordert. Dieses Array hat wie die LEDs eine Grösse von 8 x 16 Datenfelder. Jedes einzelne Datenfeld repräsentiert ein LED. Hat dieses Feld eine «1» somit soll das LED leuchten und bei einer «0» leuchtet es nicht.

# Spieldefinition / Bedienung

Die Steuerung der Schlange erfolgt durch *Cursor-Pfeiltasten*. Die Darstellung besteht aus einer LED Matrix mit roten und grünen LED’s.  
Das Ziel ist es mit der Schlange (grüne LED’s), möglichst viele Futterhappen (rote LED) aufzunehmen.  
Mit der Aufnahme jedes Futterhappens wird die Schlange etwas länger. Die angehäufte Futterhappen werden als Punktzahl beziffert.  
Zu beachten ist, dass Berührungen des Spielfeldrandes oder des eigenen Schwanzes zum Tod der Schlange führen und somit das Spiel beenden.

# Funktionsprinzip

## Vorstudie zur LED Matrix Ansteuerung

**Beschreibung:**Wir haben in unserem Konzept festgelegt, das wir die Anzeige auf einer LED Matrix (8 x 8) realisieren. Um dies realisieren zu können, müssen wir mit einem Schieberegister arbeiten. Mithilfe diesem Bauteils kann aus drei Eingängen 8 Ausgänge erstellt werden (Bei unserem Aufbau).  
In der Vorstudie haben wir uns damit befasst, ob es für uns möglich ist, mithilfe von Arduino und einem Schieberegister eine einzelne LED Reihe anzusteuern. Für die Vereinfachung haben wir den Versuch mit 4 anstatt 8 LEDs aufgebaut.  
Im ersten Teil schauten wir ob wir von der Softwareseite mit einem Array das Schieberegister so ansteuern können, dass das Muster des Arrays dem Muster bei den LEDs entspricht. Das heisst bei einem Array von [1, 0, 1, 0] sollte ebenfalls nur jedes zweite LED leuchten.  
Im zweiten Teil wollten wir die Geschwindigkeit herausfinden, in der die acht Reihen durchgeschalten werden müssen um kein Flackern zu erzeugen. Diese Überlegung ist nötig, da wir nur eine Reihe auf einmal leuchten lassen.

**Aufbau:**

|  |  |  |
| --- | --- | --- |
| * Arduino: - 8 = *Anst*. Shift Register - 9 = *Anst.* Storage Register - 10 = Anst. Data Pin | * R1 – R4 = 330Ω * T1 – T4 = BS270 * Shiftregister = 74HC595 |  |

**Erkenntnis:**Wir konnten ein Programm für das Arduino schreiben, welches das Muster des Array wie gewünscht auf den LED übertragbar ist.   
Beim Zeitlichen ansteuern der einzelnen Reihen haben wir uns überlegt das wir acht Reihen haben, welche nach einander angesteuert wird als kann dies mit einer einzelnen Reihe simuliert werden, indem sie nur einen achtel der Zeit angesteuert wird. Daraus haben wir gesehen das eine einzelne Reihe maximal 2ms angesteuert werden darf bevor die einzelnen Reihen ein Flackern aufweisen.

Zusätzlich haben wir bemerkt, da wir zu Beginn die Eingänge des Schieberegisters mit Tastern angesteuert haben, dass bei den Tastern ein sehr starkes prellen auftritt. An dieses Prellen müssen wir auch in Zukunft bei der Entwicklung denken.

# Anhang

## Glossar

**Arduino / Arduino-Board**Handelsware: Eine bestückte Leiterplatine mit mehreren Schnittstellen (Ein und Ausgänge). Diese kann beliebig via Computer programmiert werden, um flexibel Codes ausführen zu können

**Shield**  
Ein Aufbau auf einer Platine, welche alle nötigen Hardware besitzt. Dies kann als fertiges Element direkt (zum Beispiel bei einem Arduino) eingesetzt werden.

**Cursor-Pfeiltasten**Bei unserem Aufbau sind die Cursor Pfeiltasten simple Drucktaster. Wie sie unter «1.1 Hardware Definition» sehen.

**Anst.**Abkürzung für Ansteuerung.

## Datenblätter

Bs270 : http://pdf1.alldatasheet.com/datasheet-pdf/view/50805/FAIRCHILD/BS270.html  
74hc595: http://pdf1.alldatasheet.com/datasheet-pdf/view/15644/PHILIPS/74HC595.html

# Identifikation

Ort, Datum Dominik Reindl Diogo Abreu