

Snake Projekt

Jan Scholz

16. März 2013

Inhaltsverzeichnis

1	Einführung	3
2	Hardware Entwicklung	4
2.1	Mikrocontroller	4
2.2	LED-Matrix	4
2.2.1	Grundidee	4
2.2.2	Eagle Schaltplan (LED-Matrix)	4
2.2.3	Eagle Layout (LED-Matrix)	6
2.2.4	Fertige LED-Matrix	6
2.2.5	Probleme mit der LED-Matrix	6
2.2.6	Technische Daten	7
2.3	Eingabegerät (Sega Gamepad)	8
2.3.1	Steuerung	8
2.4	Steuerschaltung	9
2.4.1	Testaufbau	9
2.4.2	Eagle Schaltplan (Steuerschaltung)	11
2.4.3	Eagle Layout (Steuerschaltung)	12
2.4.4	Fertige Steuerplatine	12
2.4.5	Technische Daten	13
3	Software Entwicklung	14
3.1	Ansteuerung der Matrix	14
3.2	Eingaben über das Sega Gamepad	16

1 Einführung

Aus dem einfachen Grund des Interesses an der Mikrocontroller Programmierung ist dieses kleine Projekt entstanden, in dem ich das Spiel Snake auf einem Mikrocontroller implementiere. Als Herzstück dient mir ein ATmega32, den ich noch rumliegen hatte. Damit die Schlange und das Futter visualisiert werden kann, habe ich nach einiger Recherche beschlossen, das es wohl das Günstigste wäre, mir einfach eine eigene LED-Matrix, bestehend aus LOW-Current LEDs, zu basteln. Dieses Vorhaben zeigte sich durch die vielen Bohrungen (504 Stück) in viel nerviger Arbeit. Aber es war eben das Günstigste was mir einfiel.

Um die Schlange zu steuern, erschien es mir am besten ein altes Sega Gamepad zu verwenden. Dieses hatte ich sowieso noch rumliegen und die Datenverarbeitung ist sehr einfach gehalten, aber dazu später mehr. Als letzte Hürde ging es nun um die Programmierung der Firmware, für welche vielleicht nicht immer die elegantesten Lösungen gefunden wurden, aber dazulernen tue ich immer gerne und denke das in dieser ersten Version noch viel Potenzial für Verbesserungen steckt.

Im folgenden Dokument werde ich auf die Realisierung und die Funktionsweise dieser ersten Version eingehen.

2 Hardware Entwicklung

2.1 Mikrocontroller

Als Mikrocontroller wird ein ATmega32 der Firma Atmel verwendet. Dieser Mikrocontroller besitzt genügend Programmspeicher (Flash-ROM) und es stehen auch die nötigen IO-Pins zur Verfügung um eine Eingaben und Ausgaben zu realisieren.

2.2 LED-Matrix

2.2.1 Grundidee

Da für diese Spielereien einfach nicht das nötige Kleingeld zur Verfügung stand und ich mit einem Eigenbau bei der Matrix auch variabler in der Ansteuerung bin, habe ich mir eine eigene 8x16 LED Matrix gebaut. Für dieses Vorhaben wurde ein Umgebauter Gesichtsbräuner als Belichtungsgerät verwendet und ein paar Schüsseln und ein Bunsenbrenner zum Ätzen verwendet. Die Bohrungen habe ich mit einem 0,8mm Bohrer gebohrt. Dabei kam ich sogar mit einem einzigen aus, ohne ihn abzubrechen. Um Pins zu sparen, verwende ich ein Multiplexverfahren. In diesem werden nacheinander die Spalten 0 bis 7 (Kathoden der LEDs je Spalte sind verbunden) über einen BC107 NPN-Transistor gegen Masse geschaltet. Die Anoden werden durch 16 Ausgänge über Vorwiderstände angesprochen. Wenn nun je nach Spalte die gewünschten LEDs geschaltet werden und die Spalten schnell genug nacheinander aktiviert werden, können beliebige Muster erzeugt werden.

2.2.2 Eagle Schaltplan (LED-Matrix)

Im folgenden ist der Eagle Schaltplan (Abb.2.1) zu sehen, welcher für die LED-Matrix entworfen wurde.

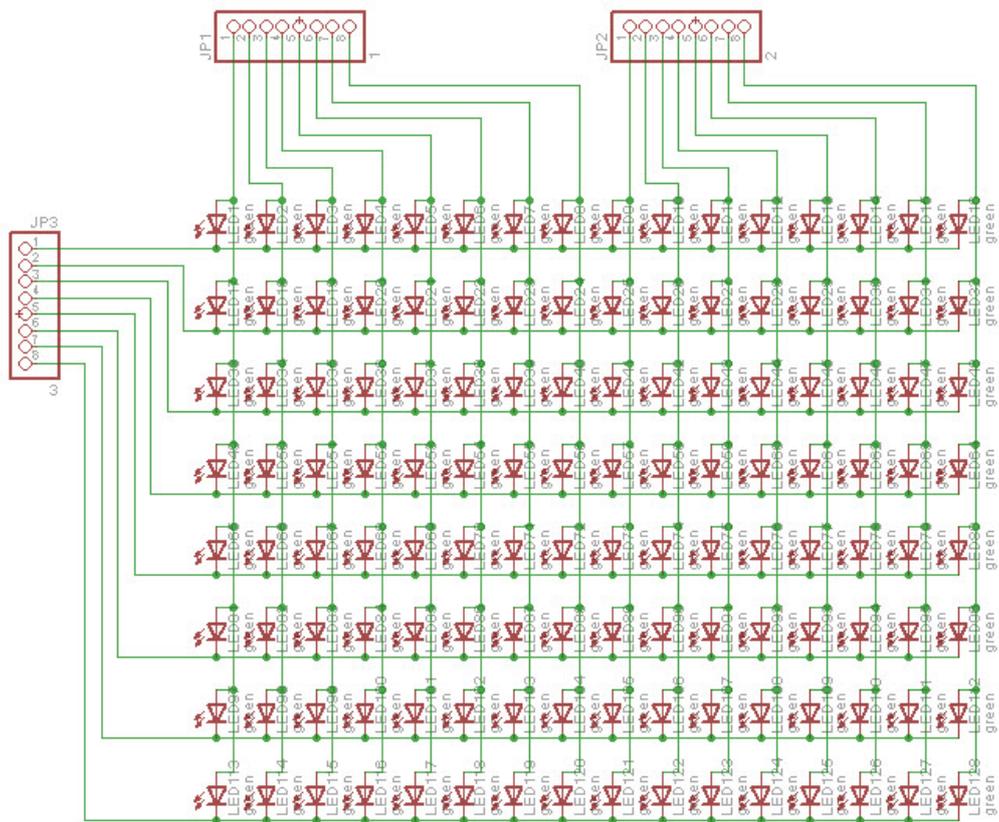


Abbildung 2.1: Eagle Schaltplan der LED-Matrix

2.2.3 Eagle Layout (LED-Matrix)

Nachdem der Schalplan entworfen ist, wird das Layout erstellt, welches in Abbildung 2.2 zu sehen ist.

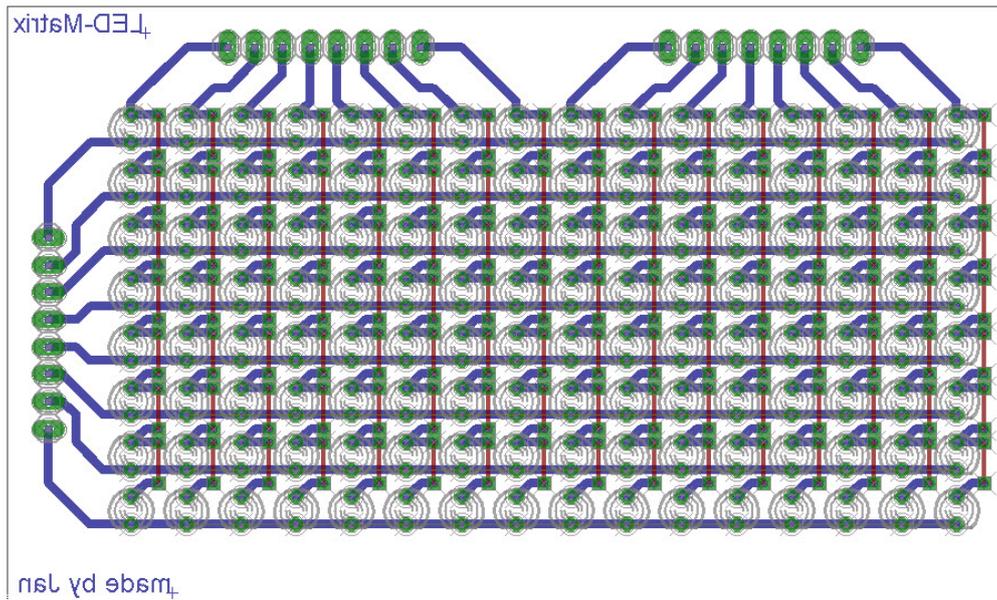


Abbildung 2.2: Eagle Layout der LED-Matrix

2.2.4 Fertige LED-Matrix

Nach einiger Arbeit funktioniert die Matrix nun. Und kann mit 19 Pins über den Mikrocontroller gesteuert werden. Die Matrix von beiden Seiten ist in Abb.2.3 zu sehen.

2.2.5 Probleme mit der LED-Matrix

Der Nachteil der sich durch das Multiplex-Verfahren zeigt ist, dass die LEDs durch das Periodische ein- und ausschalten nie die volle Helligkeit erreichen, so dass die bei hellem Zimmerlicht teilweise schlecht zu sehen sind.

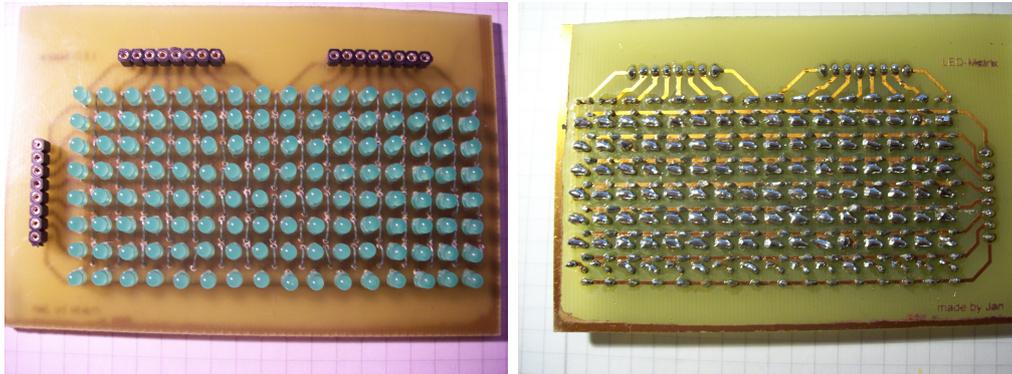


Abbildung 2.3: Fertig erstellte LED-Matrix von beiden Seiten

2.2.6 Technische Daten

- Platine:
 - Layer: 1
 - Leiterbahndicke: 0,4064mm
 - Bohrungen: 0,8mm
 - Anzahl der Bohrungen: 504 Stk.
- LEDs:
 - Typ: Low Current (2mA)
 - Farbe: grün
 - Durchmesser: 3mm
 - Anzahl: 128
- Verbindungen zur Steuerschaltung:
 - Typ: Buchsenleiste
 - Pole: 8
 - Anzahl: 3
- Sonstiges:
 - Benötigte IO-Pins: 24

2.3 Eingabegerät (Sega Gamepad)

2.3.1 Steuerung

Die Steuerung der Schlange erfolgt über ein Sega Gamepad. Die Vorteile sind die einfache Datenverarbeitung, da die Daten Parallel anliegen und der Anschluss welcher über einen D-Sub Male Connector erfolgen kann und das die Steuerspannung mit 5V erfolgt. Die Signale werden im Gamepad auf eine einfache Art erzeugt. Als Quelle für die Informationen habe ich die sehr übersichtliche Erklärung von der Internetseite [http://segaretro.org/Control_Pad_\(Mega_Drive\)](http://segaretro.org/Control_Pad_(Mega_Drive)) genutzt.

Das Gamepad nutzt den Chip 74HC157, welcher ein 8-zu-4 Multiplexer ist. Mit einem Selectsignal wird periodisch zwischen den verschiedenen Eingängen (Bedientaster des Gamepads) umgeschaltet. In der Tabelle 2.1 sind die Pins des D-Sub connectors aufgeführt.

Pin	Beschreibung
1	Taste hoch
2	Taste runter
3	Pin 4 vom Chip. Ausgang 1Y
4	Pin 7 vom Chip. Ausgang 2Y
5	+5V
6	Pin 9 vom Chip. Ausgang 3Y
7	Select
8	GND
9	Pin 12 vom Chip. Ausgang 4Y

Tabelle 2.1: Pins des D-Sub Connectors des Gamepads

Aus den gegebenen Anschlüssen ergeben sich, auf die Taster des Gamepads übertragen, die Muster aus Tabelle 2.2 und 2.3.

Für genauere Informationen empfehle ich die auch oben schon genannte Seite: [http://segaretro.org/Control_Pad_\(Mega_Drive\)](http://segaretro.org/Control_Pad_(Mega_Drive)).

Aus den beiden Tabellen ist ersichtlich, das die Schaltung auch ohne das Toggeln des Select-Signals auskommt. Im weiteren Projekt wird also mit der Tabelle 2.3 gearbeitet. Hier sind alle Richtungstasten vorhanden und für das navigieren durch ein Menü können die Tasten B und C verwendet werden. Mit den obigen Informationen kann nun also eine Steuerschaltung entworfen werden.

Pin	Beschreibung
1	Taste hoch
2	Taste runter
3	—
4	—
5	+5V
6	Taste A
7	SELECT
8	GND
9	Taste Start

Tabelle 2.2: Sel=0

Pin	Beschreibung
1	Taste hoch
2	Taste runter
3	Taster links
4	Taste rechts
5	+5V
6	Taste B
7	SELECT
8	GND
9	Taste C

Tabelle 2.3: Sel=1

2.4 Steuerschaltung

2.4.1 Testaufbau

Um die Steuerschaltung zu testen, bevor ein endgültiges Layout erstellt wird, wurde sie vorher auf einem Steckbrett aufgebaut. So kann die Elektrische Funktion mit der fertigen LED-Matrix getestet werden. Des weiteren wurde mit diesem Aufbau die Firmware geschrieben, welche die hier beschriebene erste Version darstellt. Das Bild 2.4 veranschaulicht den Aufbau.

Nach dem eine erste Version der Firmware geschrieben ist (hierauf wird genauer in Kap.3 eingegangen) und die vollständige Funktion des Hardwareaufbaus gewährleistet werden kann, wird mit dem entwickeln der Steuerplatine begonnen.

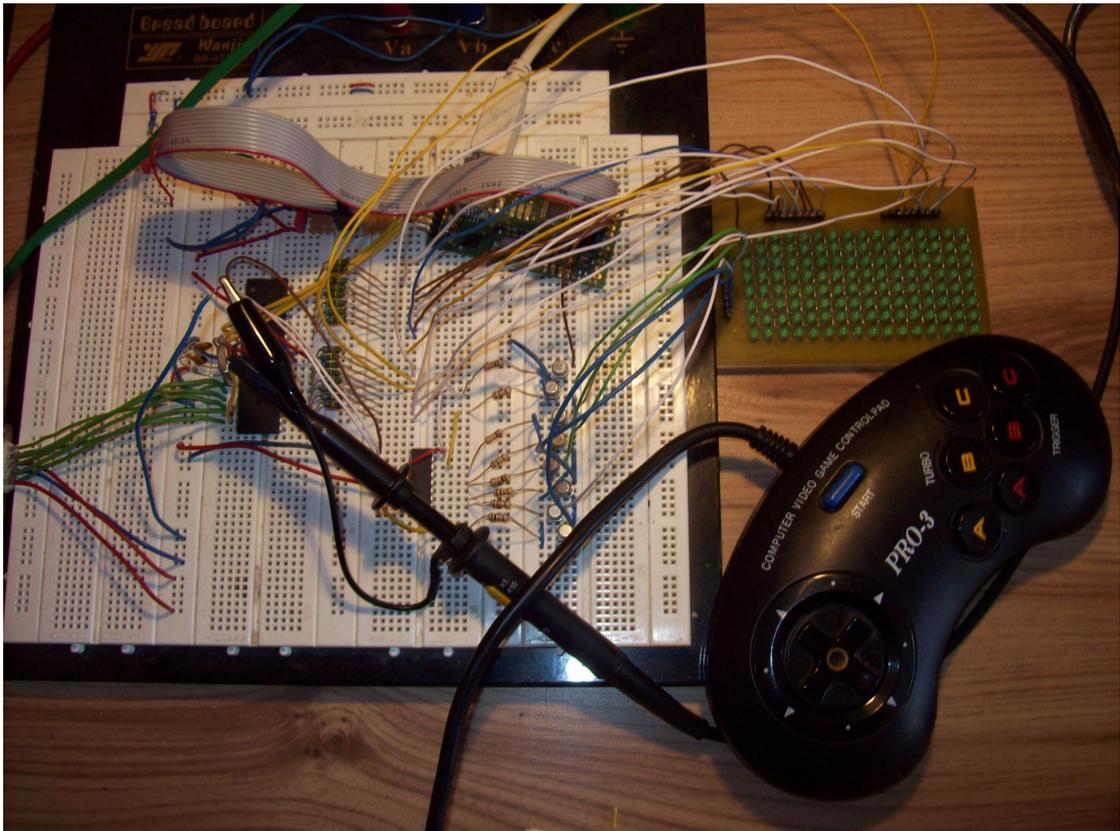


Abbildung 2.4: Testschaltung auf dem Steckbrett

2.4.2 Eagle Schaltplan (Steuerschaltung)

In Abb.2.5 ist der entworfene Schaltplan zu sehen.

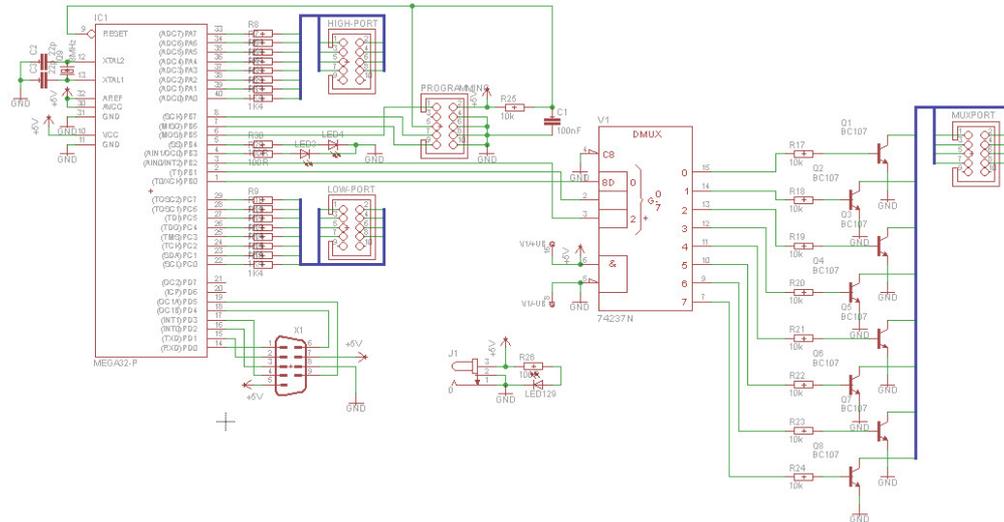


Abbildung 2.5: Schaltplan der Steuerplatine

2.4.3 Eagle Layout (Steuerschaltung)

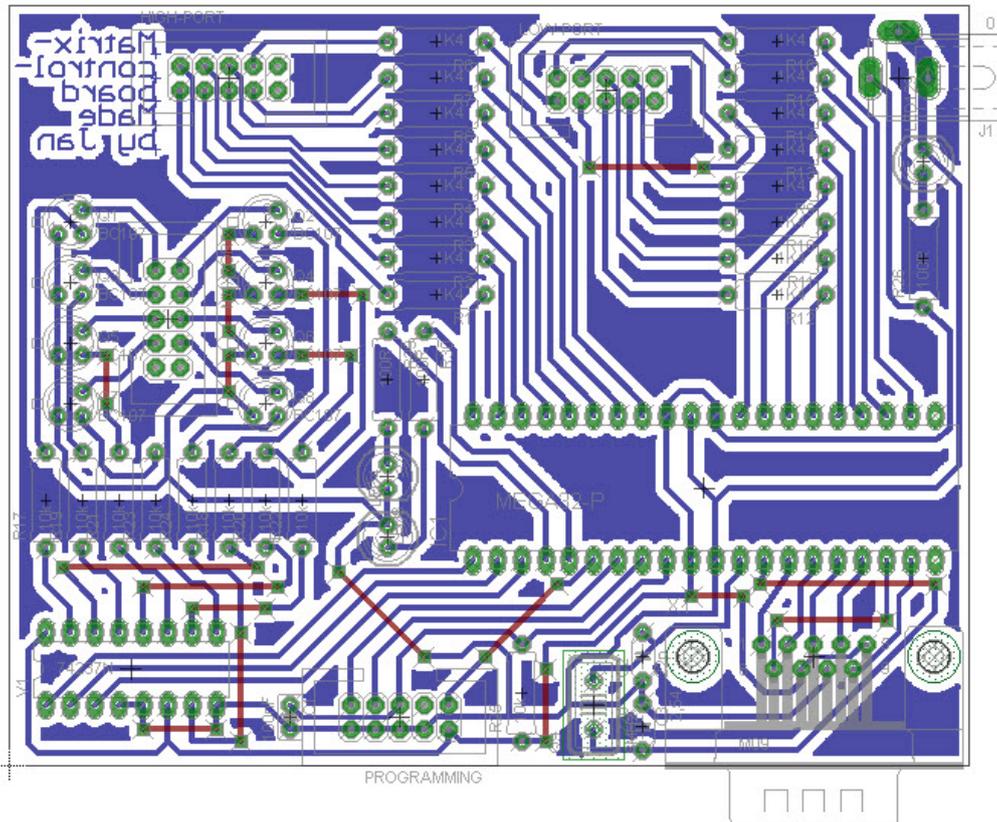


Abbildung 2.6: Layout der Steuerplatine

2.4.4 Fertige Steuerplatine

Nachdem, auf Grund von einigen Drehern in Anschlüssen, die Platine mehrmals überarbeitet werden musste, ist auf der Abb.2.7 endlich die fertige Platine von beiden Seiten zu sehen.

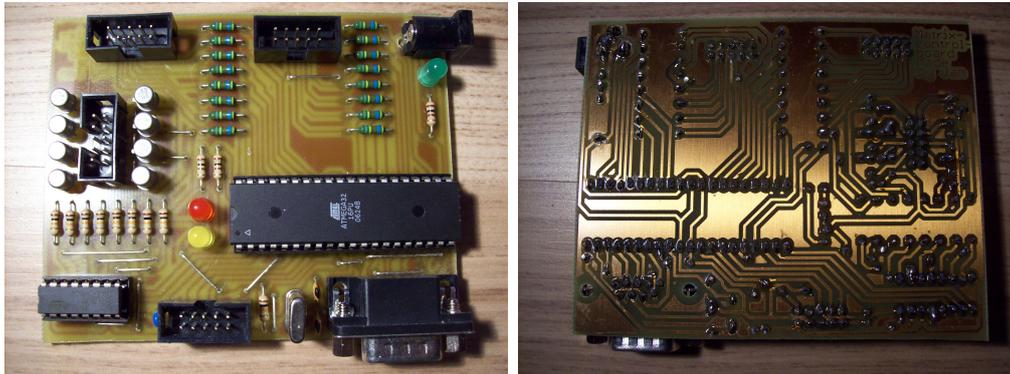


Abbildung 2.7: Fertig erstellte LED-Matrix von beiden Seiten

2.4.5 Technische Daten

- Mikrocontroller:
 - Typ: ATmega32
 - Benötigte IO-Pins: 24
 - Takt: 16MHz
- Transistoren (BC107):
 - Typ: NPN
 - I_{CEmax} :
- Multiplexer-IC:
 - Typ: M74HC237
- Anschluss Sega-Gamepad:
 - Typ: D-Sub 9-Pol
 - Ausführung: Male-Connector
- Spannungsversorgung:
 - Typ: Anschluss für 5V Netzteil

3 Software Entwicklung

3.1 Ansteuerung der Matrix

Um die Matrix anzusteuern, wird wie bereits aus Punkt 2.2 bekannt, ein Zeilen-Multiplexverfahren verwendet. Dazu wird der IC M74HC237 verwendet. Dieser IC schaltet auf Hochzählen des Eingangswertes je nach Wert einen Eingang durch. Für die Spalten werden zwei Ports des Mikrocontrollers verwendet. Für die Realisierung wird ein Timerinterrupt genutzt. Bei jedem Durchlauf des Interrupts wird der Eingangswert um 1 erhöht. Mit dem Wert, der jeweils aktuell ist, werde die gewünschten LEDs eingeschaltet, so das später das Bild entsteht. Für die LEDs jeder Spalte wird eine short-Variable (16-Bit) genutzt um die Werte zu speichern. Im Interrupt werden die LEDs der aktuellen Zeile eingeschaltet. Abb.3.1 soll diesen Ablauf veranschaulichen.

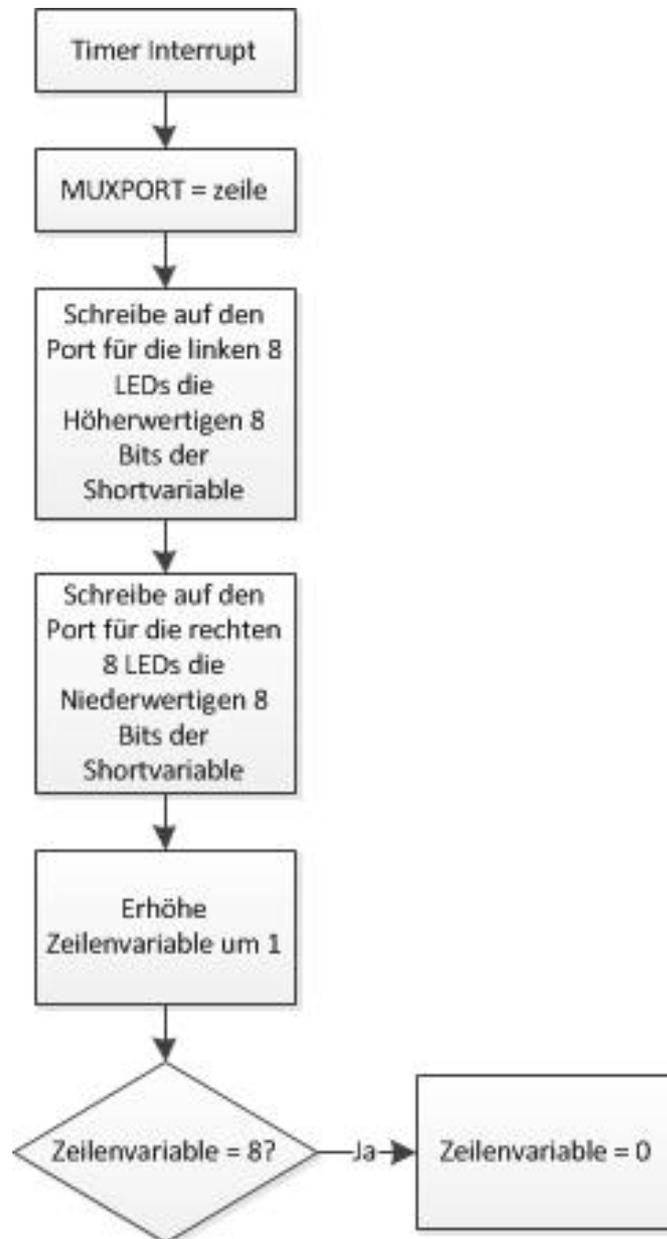


Abbildung 3.1: Ablauf des Timerinterrupts zur Matrixansteuerung

3.2 Eingaben über das Sega Gamepad

Um eine Datenverarbeitung bezüglich des Gamepads zu realisieren sind einige Überlegungen nötig. Wie im Kapitel 2.3 beschrieben liegen die Daten Parallel an, was bedeutet das die Möglichkeit eines Interrupteingangs ohne zusätzliche Hardwarebeschaltung wegfällt.

Um eine Verarbeitung zu realisieren und ein Polling in der Main zu umgehen, wurde eine Variante gewählt, in der über ein Timerinterrupt die genutzten Pins gepollt werden. Dies geschieht mit einer Frequenz von $250kHz$.

Nachdem festgestellt wurde das die Tasten nur sehr eigenwillig funktionierten, wurde in Messungen (siehe Bild 3.2) klar, das dieses Gamepad prellt.

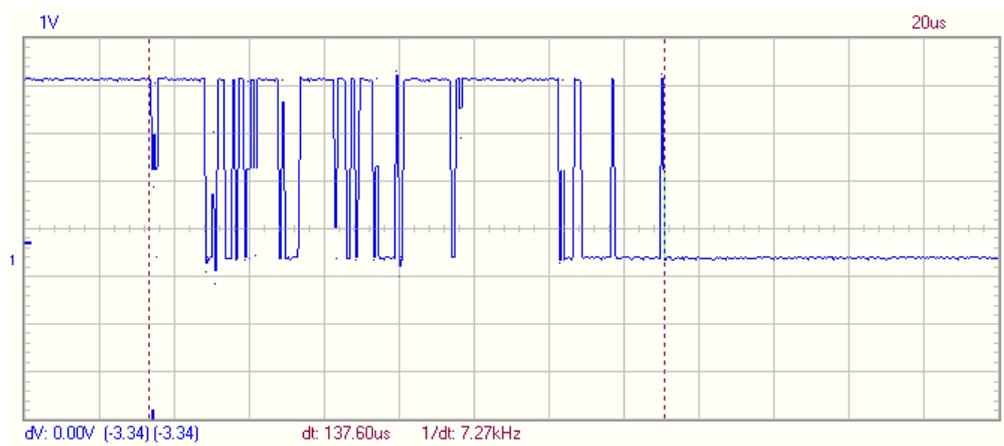


Abbildung 3.2: Prellen auf drücken von Button C

Um eine Software-Entprellung zu realisieren, wurde eine Methode implementiert, die auf detektieren eines Tastendruck eine Zeit lang immer wieder überprüft, ob der Taster immer noch gedrückt ist. Dies geschieht im Interrupt des Timer-Tastenpollings. Wenn der Timerwert 1000 erreicht ist und während der gesamten Zeit keine Pegeländerung detektiert würde, ist davon auszugehen, dass das Prellen vorbei ist und ein sicherer Zustand anliegt. In diesem Fall ist liegt der Zustand seit $4ms$ fest an $((1/250kHz) * 1000)$. Erst dann wird ausgewertet welcher Button gedrückt wurde. Folgendes Ablaufdiagramm (Abb.3.3) soll die Timer Interrupt-prozedur veranschaulichen.

Nachdem nun die Möglichkeit besteht die Matrix anzusteuern und die prellenden Taster vom Gamepad auszulesen. Muss nun das eigentliche Spiel Programmiert werden. Da die Schlange ein zusammenhängendes Objekt ist, welches sich über

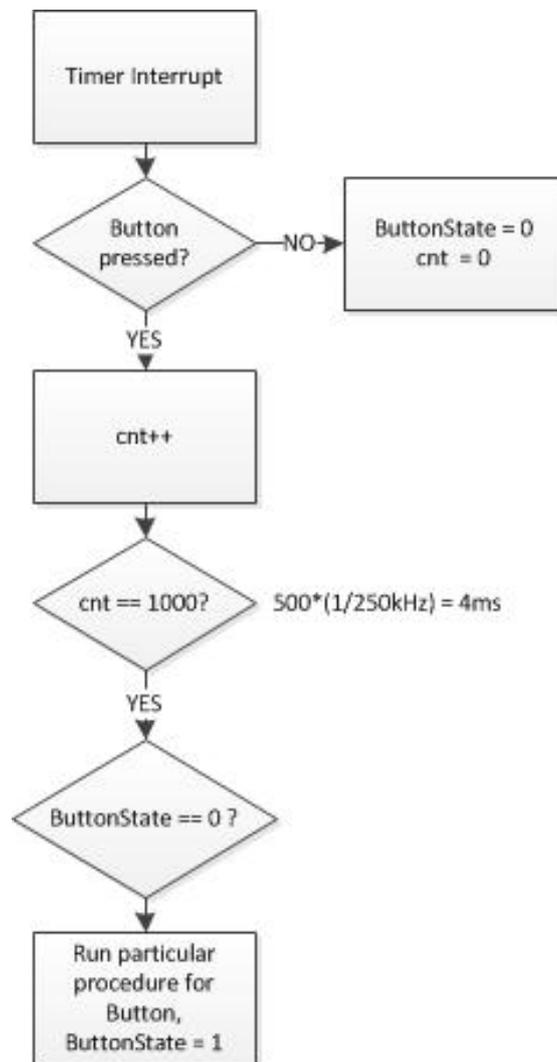


Abbildung 3.3: Timer Interruptprozedur zum entprellen

die Matrix bewegen soll, bietet es sich an mit einer verketteten Liste zu arbeiten. Es wird in jedem Listenelement die X- und Y-Koordinate gespeichert. Bei jedem Schritt in die Richtung die das Gamepad vorgibt, wird, wenn keine Kirsche gegessen wurde ein Element in Bewegungsrichtung angehängt und das letzte Element gelöscht. Nach einigen zusätzlichen abfragen, die Prüfen ob die Schlange sich selber, bzw. die Wand beißt, ist es nun Möglich mit dem Sega-Gamepad Snake zu Spielen.