Conways game of life – auf dem 8051

**Ein Projekt von:**

Thomas Pötzsch

Alexander Kimmig

Jennifer Hauß

(screenshot glider)

**Karlsruhe, den 14.06.2018**

Inhaltsverzeichnis

[Einleitung 3](#_Toc516130054)

[Motivation 3](#_Toc516130055)

[Aufgabenstellung 3](#_Toc516130056)

[Grundlagen 3](#_Toc516130057)

[Assembler 3](#_Toc516130058)

[Der 8051 Mikrocomputer 3](#_Toc516130059)

[Entwicklungsumgebung MCU-8051 DIE 3](#_Toc516130060)

[Konzept 3](#_Toc516130061)

[Analyse 3](#_Toc516130062)

[Programmentwurf 3](#_Toc516130063)

[Implementation 3](#_Toc516130064)

[Zusammenfassung 3](#_Toc516130065)

# Einleitung

## Motivation

Die Motivation dieses Projektes war es die systemnahe Programmierung mit Assembler kennenzulernen und uns mit den wichtigsten Befehlen vertraut zu machen. Ein grundlegendes Verständnis von Assembler kann auch sehr hilfreich sein, um die Funktionsweise abstrakter, höherer Programmiersprachen zu verstehen und Fehler nachvollziehen zu können.

Weiterhin können Compiler-Schritte durch die Programmierung in Assembler gespart werden und Overhead vermieden werden, was besonders bei Embedded Systems sehr wichtig ist.

## Aufgabenstellung

Die Aufgabenstellung war ein kleines, simples Programm in Assembler zu programmieren. Dabei durfte man sich selbst aussuchen, was für ein Programm man implementiert. Unsere Idee war es „Conway’s Game of life“ so nach zu programmieren, dass der sogenannte „Glider“ auf einer 8x8 LED Matrix erscheint und sich nach den Regeln des Spiels bewegt.

# Grundlagen

## Assembler

Eine Assemblersprache ist eine Programmiersprache, die zur systemnahen Programmierung eines bestimmten Computers ausgerichtet ist. Die Assemblerbefehle werden direkt in Maschinensprache (also Bytecode) umgewandelt. Damit ist es die erste Stufe ab der Maschinensprache, die für Menschen lesbar ist.

Der Quellcode eines Assemblerprogramms wird mit Hilfe einer Übersetzungssoftware (einem Assembler) in Maschinensprache übersetzt.

Der Befehlssatz der Assemblersprache ist abhängig vom Zielcomputer und somit nicht einheitlich.

https://de.wikipedia.org/wiki/Assemblersprache

## Der 8051 Mikrocomputer

Der 8051 hat 2 getrennte Adressbereiche für den Programmspeicher und für den Arbeitsspeicher, welche zusätzlich noch in internen und externen Bereich unterteilt sind.

Hier evt. Füllen

## Entwicklungsumgebung MCU-8051 DIE

Die IDE MCU-8051 ist kostenlos im Internet verfügbar und bietet Simulation des Programmablaufs inklusive der verschiedenen Speicherzustände während des Ablaufs. Dazu kann an Hardware simuliert werden:

* LED Panel
* LED Display
* LED Matrix
* Multiplexed LED Matrix
* Simple Keypad
* Matrix Keypad
* LCD Display
* Temperatursensor
* Etc.

# Konzept

## Analyse

Zunächst haben wir die vier Regeln des „Game of life“ analysiert und überlegt, wie diese umgesetzt werden können.

Die Regeln lauten:

* Eine tote Zelle mit genau drei lebenden Nachbarn wird in der Folgegeneration neu geboren.
* Lebende Zellen mit weniger als zwei lebenden Nachbarn sterben in der Folgegeneration an Einsamkeit.
* Eine lebende Zelle mit zwei oder drei lebenden Nachbarn bleibt in der Folgegeneration am Leben.
* Lebende Zellen mit mehr als drei lebenden Nachbarn sterben in der Folgegeneration an Überbevölkerung

Eine Zelle kann 8 Nachbarn haben (direkte Nachbarn links, rechts, oben und unten und die diagonalen Nachbarn). Es musste überlegt werden, wie diese herausgefunden und überprüft bzw. angesprochen werden konnten. Dies lösten wir mit einer Maske.

Die nächste Überlegung war, wie wir die einzelnen LEDs der Matrix ansprechen können und wie neuberechnen und neuzeichnen implementiert werden sollte. Hier entschieden wir uns eine zweite Matrix zu berechnen, die Ergebnisse darauf anzuwenden und wenn alle Regeln angewandt wurden die angezeigte Matrix durch die neue zu ersetzen.

Des Weiteren musste überlegt werden, was mit Zellen passiert, die über den Rand der LED Matrix „hinauslaufen“. Hier fiel schnell die Entscheidung einen Übergang zwischen den Rändern zu implementieren. Das heißt, dass die Zelle ganz rechts für unser Programm neben der Zelle ganz links liegen. Dasselbe Prinzip gilt für oben und unten. So gibt es für das Programm keine Ränder.

## Programmentwurf

# Implementation

# Zusammenfassung