

# **4-Gewinnt auf einem Mikrocomputer der 8051-Famile**

von

**Heidinger, Matthis, Riesinger, Stephan**

Kurs TINF17B1

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Motivation . . . . .	3
1.2	Aufgabenstellung . . . . .	3
<b>2</b>	<b>Grundlagen</b>	<b>4</b>
2.1	Assembler . . . . .	4
2.2	Der 8051 Mikrocomputer . . . . .	4
2.3	Entwicklungsumgebung MCU-8051 IDE . . . . .	5
<b>3</b>	<b>Konzept</b>	<b>6</b>
3.1	Analyse . . . . .	6
3.2	Programmentwurf . . . . .	6
<b>4</b>	<b>Implementation</b>	<b>8</b>
<b>5</b>	<b>Zusammenfassung</b>	<b>9</b>

# 1 Einleitung

## 1.1 Motivation

Mit dem 8051 ist ein relativ einfacher Einstieg in die Assemblerprogrammierung möglich. Assembler ist sehr maschinennah und bietet aus diesem Grund nicht die Annehmlichkeiten, die beispielsweise objektorientierte Programmiersprachen wie Java mit sich bringen.

An einem Simulator lässt sich gefahrlos erproben, wie mit Pointern und sehr begrenztem Speicherplatz umzugehen ist.

Durch die andere Herangehensweise an ansonsten vertraute Programmierstrukturen wie zum Beispiel Vergleichen, die nun mit Jump-Befehlen und negativen Vergleichen implementiert werden müssen, wird das logische Denkvermögen geschult.

## 1.2 Aufgabenstellung

Es soll ein Spiel nach dem bekannten Spielkonzept von „4-Gewinnt“ in Assembler auf dem 8051 entwickelt und implementiert werden.

Dafür wird für die Visualisierung der Spielfläche eine entsprechende Hardware gewählt, auf der die Spielstände der beiden Spieler angezeigt werden (Output). Die Auswahl der Spalte, in der ein „Spielstein“ platziert werden soll, muss ebenfalls über eine entsprechende Hardware gelöst werden (Input).

Die beiden Spieler müssen damit in der Lage sein, abwechselnd sogenannte Spielsteine in selbst ausgewählte Spalten zu werfen, welche dann am oberen Ende des Stapels angefügt werden. Hat ein Spieler es geschafft, 4 seiner Spielsteine hintereinander in eine Reihe oder Spalte zu bringen, hat er gewonnen.

Dabei soll sicher gestellt sein, dass nicht mehr Spielsteine in eine Spalte geschmissen werden als Platz ist. Auch wichtig ist, dass die Spielsteine der beiden Spieler klar voneinander zu unterscheiden sein müssen, damit es nicht zu Verwechslungen kommen kann.

## 2 Grundlagen

### 2.1 Assembler

Assembler ist toll!

### 2.2 Der 8051 Mikrocomputer

Die ersten Mikroprozessoren der 8051-Reihe wurden im Jahr 1980 von Intel entwickelt. Es handelt sich dabei um einen direkten Nachfolger der 8048-Familie. Die 8051-Familie erfreute sich extrem großer Beliebtheit. So wurden über 250 Familienmitglieder von verschiedensten Herstellern wie Philips, Siemens, AMD, OKI und weiteren gebaut und produziert. Der Höhepunkt der Beliebtheit des 8051 war das Jahr 1995, in welchem diese Mikroprozessorfamilie einen Marktanteil von bis zu 30 Prozent erreichte und täglich mehr als eine Million Prozessoren hergestellt wurden.

Auch technisch war der 8051 zu damaligen Zeiten hochmodern, was man folgenden Eckdaten entnehmen kann:

- 1,2 - 18MHz Taktrate, oft werden 12MHz verwendet
- 4 kByte ROM
- 128 Byte RAM
- 4 8-Bit Eingabe- und Ausgabeports
- 2 16-Bit Zähler beziehungsweise Zeitgeber
- Eine USART-Schnittstelle
- 5 Interruptquellen
- Bei einer Taktrate von 12MHz laufen

- 58% der Befehle in  $1\mu s$
- 40% der Befehle in  $2\mu s$
- 2% der Befehle in  $4\mu s$

ab. Die langsamsten Befehle sind beispielsweise Multiplikation und Division.

## 2.3 Entwicklungsumgebung MCU-8051 IDE

**LANGSAM!!!**

# 3 Konzept

Wir machen ein 4-Gewinnt

## 3.1 Analyse

Was sollen wir denn Analysieren?

## 3.2 Programmentwurf

Der Programmfluss, wie er in Abbildung 3.1 dargestellt ist, beschreibt die Hauptschleife des Programmes. Nach einer Initialisierung, in welcher benötigte Speicher zurückgesetzt werden und der Timer gestartet wird, startet die Hauptschleife. In ihr wird wiederholt abgefragt, ob aktuell eine Eingabe getätigt werden kann. Falls ja wird diese eingelesen und ausgewertet. Anschließend wird überprüft, ob der aktuelle Spieler gewonnen hat, indem er 4 Steine nebeneinander gesetzt hat. Ist dies der Fall endet das Spiel. Andernfalls wird der aktive Spieler gewechselt und die Schleife wird erneut ausgeführt.

Zusätzlich zu dieser Hauptschleife wird über ein Timerinterrupt regelmäßig eine Routine ausgeführt, welche das aktuelle Spielbrett ausgibt. Dabei werden die vom ersten Spieler gesetzten Steine auf der LED-Matrix durchgängig beleuchtet, während die Steine des zweiten Spielers blinken.

**Hinweis:** Die Wiederholrate, mit welcher die LED-Matrix aktualisiert wird ist zu groß, um den Blinkeffekt sehen zu können. Sie musste allerdings für die Entwicklung im Simulator so stark erhöht werden, da der Simulator deutlich langsamer als die tatsächliche Hardware ist.

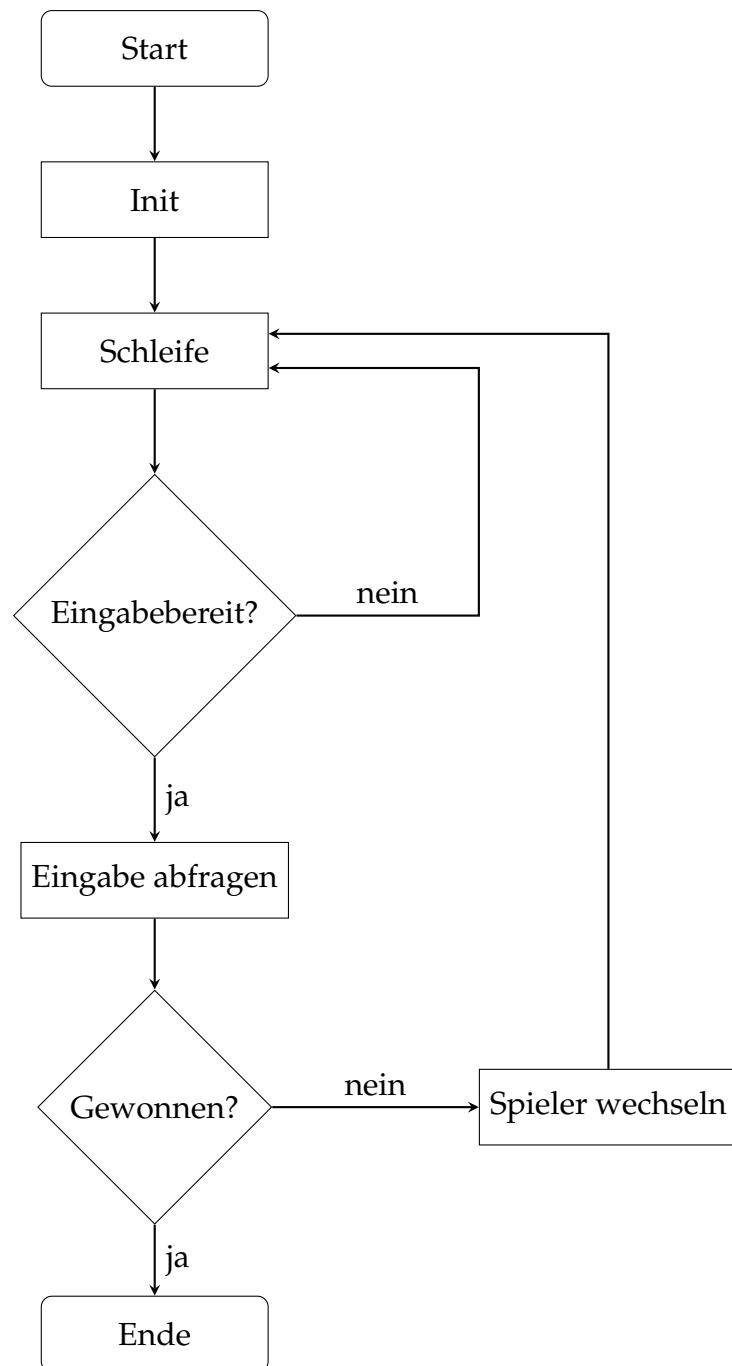


Abbildung 3.1: Programmfluss des Spieles

# 4 Implementation

In Assembler



# 5 Zusammenfassung

War gut, nochmal machen!