



DHBW KARLSRUHE

PROJEKT SYSTEMNAHE PROGRAMMIERUNG

Metronom für Intel 8051

Felix Starke (Matr.:)

Lars Hübner (Matr.: 5485901)

betreut von
PROF. DR. RALPH LAUSEN

6. Juni 2018

Inhaltsverzeichnis

1	Einleitung	3
1.1	Motivation	3
1.2	Aufgabenstellung	3
2	Grundlagen	4
2.1	Assembler	4
2.2	Intel 8051 Mikrocontroller	4
2.3	Entwicklungsumgebung MCU-8051	5
3	Konzept	6
3.1	Eingaben, Ausgaben und Ports	6
3.2	Programmablauf	7
4	Besondere Schwierigkeiten	10
5	Zusammenfassung	11

1 Einleitung

1.1 Motivation

Ein Metronom beschreibt ein mechanisches oder elektronisches Gerät, welches zur regelmäßigen Tonerzeugung genutzt wird um musikalische Rhythmen zu unterstützen. Die Mehrzahl der neueren Metronome haben eine Funktion, bei der über ein regelmäßiges Klicken auf einen Button der entsprechende Klick-Takt übernommen wird.

Da Metronome häufig als kleine portable Geräte mit geringem Stromverbrauch umgesetzt werden, kann eine entsprechende Funktionalität auf einem Intel 8051 Mikrocontroller implementiert werden. Die Programmierung auf diesem Mikrocomputer geschieht in Assembler. Ziel dieses Projektes ist entsprechend die Implementierung einer solchen "Tap to Beat" Funktion in Assembler für den 8051.

1.2 Aufgabenstellung

Aus diesen Anforderungen ergibt sich die folgende Aufgabenstellung.

Das Programm muss eine Funktionalität implementieren, die das Aufnehmen des Taktes über eine Eingabehardware ermöglicht. Außerdem muss das Ausgeben des Taktes über eine passende Ausgabehardware implementiert werden. Das gesamte Programm lässt sich damit in zwei große Teile splitten, das Recording und das Anzeigen des Taktes.

Recording

Die Aufnahme des Taktes muss über eine Eingabehardware getätigt werden. In diesem Projekt wurde eine Keypad-Matrix mit Tasten als Eingabe gewählt. Über eine dieser Tasten soll nun der Takt gemessen werden. Hierzu wird jeweils die Zeitspanne zwischen zwei Tastendrücken mithilfe eines Timers gemessen. Die Messungen sollen im Anschluss aufsummiert und am Ende durch die Anzahl der Messungen geteilt werden, um den Mittelwert zu erhalten.

Display Metronom

Nachdem der durchschnittliche Takt gemessen und ermittelt wurde, gilt es diesen auf einem Ausgabemedium anzuzeigen. Anstelle einer Tonausgabe wird in dieser Implementierung eine LED-Matrix genutzt um den Takt zu visualisieren.

2 Grundlagen

2.1 Assembler

Assembler ist eine maschinenorientierte Programmiersprache, welche auf einem prozessorspezifischen Befehlssatz basiert. Die Sprache wird als Programmiersprache der zweiten Generation bezeichnet, da es nicht mehr notwendig ist, die Befehle als Binärcode zu schreiben. Statt dessen ist es möglich tatsächliche Befehle in Textform, wie zum Beispiel "ADD Op1, Op2" zu verwenden. Assembler wird anschließend mit einem Assembler in den tatsächlichen Maschinencode übersetzt, welchen dann vom Prozessor ausgeführt wird.

2.2 Intel 8051 Mikrocontroller

Der Intel 8051 Mikrocontroller ist einer der am weitesten verbreiteten Mikrocontroller. Der 8051 ist ein 8-bit Mikrocontroller und kommt in verschiedensten Gebieten zum Einsatz. Dazu gehören beispielsweise die Automobilindustrie oder Anwendungen in der Robotik oder Telekommunikation. Er verfügt unter anderem über 4 KB Programmspeicher, 128 Byte Datenspeicher und zwei 16-Bit Timern. Des Weiteren existieren noch Varianten des Prozessors, die über besondere Komponenten wie beispielsweise UART verfügen und den Mikrocontroller entsprechend noch leistungsfähiger machen.

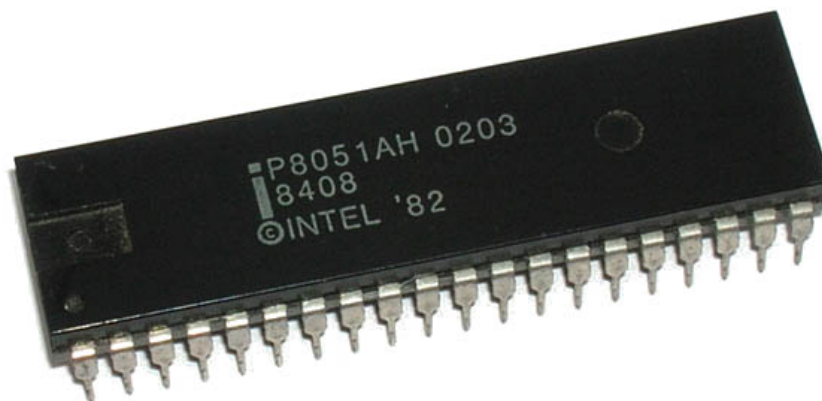


Abbildung 1: Intel 8051

Quelle: <http://www.ne555.at/2014/images/stories/UC/Intel-P8051AH.jpg>

3 Konzept

3.1 Eingaben, Ausgaben und Ports

Um dem Metronom den Takt vorzugeben, kommt eine 4x4 Keypad-Matrix (siehe Abbildung 3) zur Eingabe zum Einsatz. Diese ist an alle acht Pins von Port 0 angeschlossen. Dabei werden die Pins 0-3 für die vier Spalten der Matrix und Pins 4-7 für die Zeilen der Matrix verwendet. Um eine gedrückte Taste zu detektieren müssen zunächst die Pins 0-3 auf "1" gesetzt werden und die Pins 4-7 gelesen werden. So lässt sich herausfinden, ob überhaupt eine Taste gedrückt ist und welcher Spalte diese angehört. Im nächsten Takt können dann die Pins 4-7 auf "1" gesetzt werden und 0-3 auf "0". Anschließend kann die Zeile der gedrückten Taste ermittelt werden.

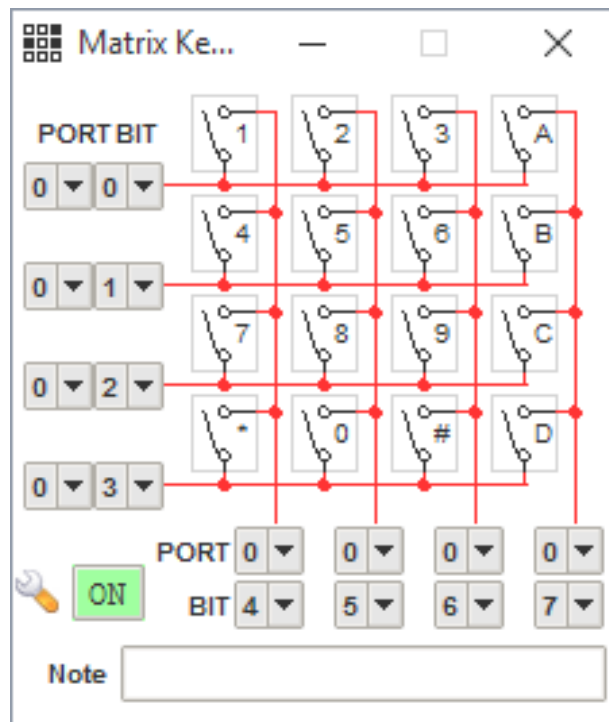


Abbildung 3: Keypad Matrix

Für die Ausgabe des ermittelten Taktes wird eine 8x8 LED-Matrix (siehe Abbildung 4) verwendet. Diese ist an die Ports 2 und 3 angeschlossen. Die acht Pins von Port 2 werden genutzt, um die Spalten der Matrix anzusteuern, die acht Pins von Port 3 werden für die Zeilen der LED-Matrix verwendet. Eine der 64 LED's leuchtet genau dann, wenn sowohl

der Pin der entsprechenden Zeile als auch der entsprechenden Spalte gesetzt sind.

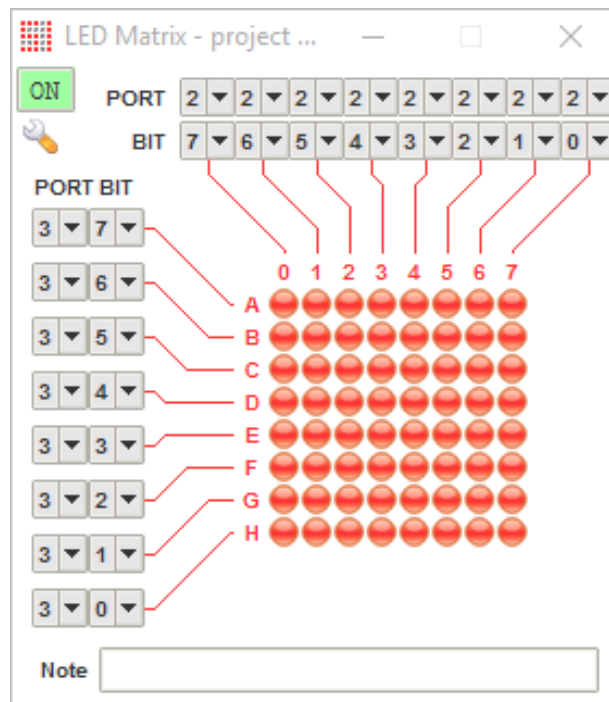


Abbildung 4: LED Matrix

3.2 Programmablauf

Das Programm besteht wie beschrieben aus zwei Teilen. Im ersten Schritt wird zunächst der Takt ermittelt um diesen dann im zweiten Teil des Programms zu visualisieren.

Programmteil 1 besteht damit aus einer Schleife, die solange Zeitmessungen durchführt bis 3 Messungen vorhanden sind. Diese Schleife startet mit dem ersten Tastendruck auf das Keypad. Eine Messung ermittelt die Zeitspanne zwischen zwei Tastendrücken.

Nachdem genug Messungen durchgeführt wurden, kann die Summe der Zeitspannen durch die Anzahl (hier 3) geteilt werden, um so den mittleren Takt zu ermitteln. Diese Mittelwertrechnung ist deshalb wichtig, da der Mensch seinen Takt beim Drücken der Taste nicht konstant halten kann und so geringen Ungenauigkeiten entstehen.

Durch die Mittelwertberechnung kann der Takt einfach gemittelt werden, um möglichst nah an den tatsächlichen Takt ran zu kommen. In Abbildung 5 ist der Programmablauf aus Programmteil 1 gezeigt.

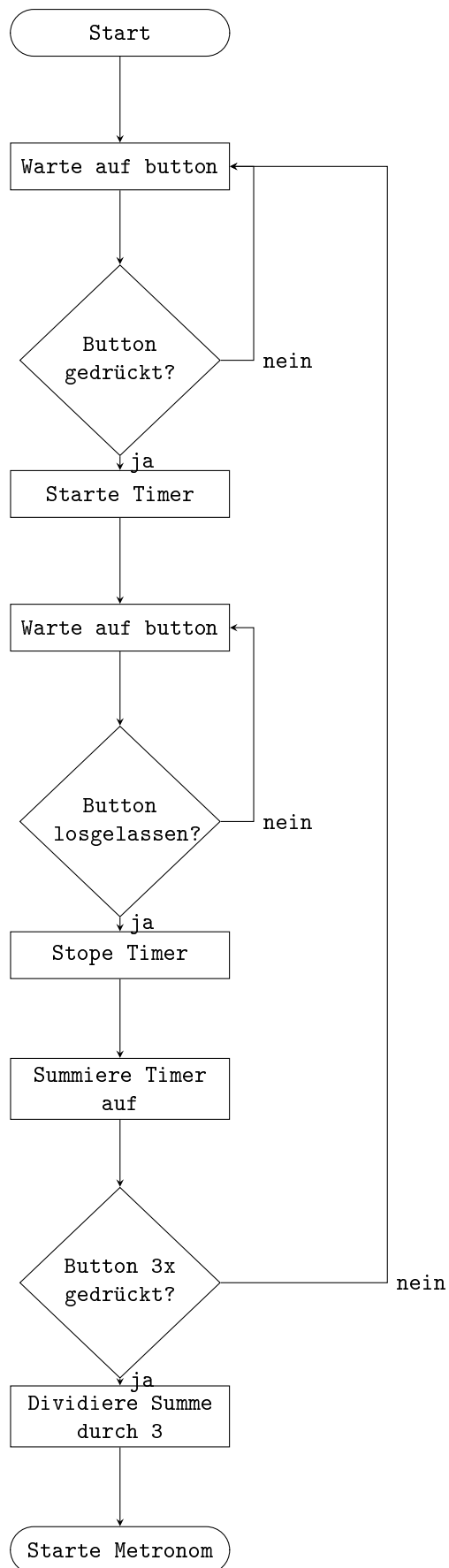


Abbildung 5: Programmablauf

In Programmteil 2 kann der ermittelte Durchschnittswert genutzt werden, um den Takt mit Hilfe eines Timers auf der LED-Matrix zu visualisieren. Zur Visualisierung blinkt abwechselnd die linke und die rechte Hälfte der LED-Matrix auf. Dieses hin und her wechseln der Seiten erinnert somit an ein gewöhnliches Metronom. Hierbei ist zu vermerken, dass die Routine zum Zeichnen auf der LED-Matrix stark vereinfacht wurde, um den Geschwindigkeitsanforderungen gerecht zu werden und das Metronom trotz Simulation und virtueller Hardware nutzbar zu machen. In Abschnitt 4 wird beschrieben, warum diese Schwierigkeit vorliegt und wie damit umgegangen wird.

Der Timerüberlauf wird zur Takterzeugung auf den in Teil 1 verwendeten Timerüberlauf gesetzt. Mit jedem Überlauf des Timers wird ein entsprechender Interrupt geworfen, welcher von einer Interruptroutine (siehe timer function im asm-code) gefangen wird und eine passende visuelle Änderung auf der LED-Matrix veranlasst.

Wie in Abbildung 6 gezeigt wird eine Art Bitmap eingeführt, welche als Indikator dafür dient, welche Spalten aufleuchten sollen. P2 (Port 2) ist mit der LED-Matrix verbunden und sorgt für die Ausgabe. Anschließend muss geprüft werden, ob bereits die in Teil 1 ermittelte Zeitspanne (in Anzahl Überläufen) abgelaufen ist. Um dies zu ermitteln wird der aktuelle tickCounter vom maxTickCounter (in Teil 1 ermittelt) abgezogen. Sobald das Ergebnis "0" erreicht muss der Zähler zurückgesetzt werden.

Beim Zurücksetzen wird zusätzlich mit "cpl A" die Bitmap zur Visualisierung bitweise invertiert um so die leuchtende Seite der LED-Matrix zu ändern.

```
1 timer:
2 mov P2, tickState
3 inc tickCounter
4
5 clr c
6 mov A, maxTickCount
7 subb A, tickCounter
8 jnz keep
9 call resetTickCounter
10 keep:
11 ret
12
13 resetTickCounter:
14 mov tickCounter, #00h
15 mov A, tickState
16 cpl A
17 mov tickState, A
18 ret
```

Abbildung 6: Interruptroutine

4 Besondere Schwierigkeiten

Besondere Schwierigkeiten ergaben sich aufgrund der hohen Laufzeiten durch die Entwicklungsumgebung. Vorallem die Nutzung virtueller Hardware verlangsamt den Ablauf des Programms stark. So vergehen, auf einem recht leistungsstarken Rechner, etwa 7 Sekunden, bis im Simulator der Entwicklungsumgebung eine Millisekunde verstrichen ist. Da gutes Timing für ein Metronom sehr wichtig ist, musste entschieden werden, wie mit der langsamen Laufzeit umgegangen werden sollte. Da das Metronom nutzbar sein sollte wurde das Timing nicht an der simulierten Laufzeit der Entwicklungsumgebung orientiert. Da weder die Eingabe zur Festlegung des Taktes noch die Ausgabe des Takts über die LED-Matrix an die Zeit der Simulation gebunden sind, ist es einfach möglich, beides zu synchronisieren und so umzusetzen, dass das Metronom nutzbar ist.

5 Zusammenfassung

Das Metronom konnte vollständig funktionsfähig umgesetzt werden und erfüllt damit unsere Aufgabenstellung. Die Umsetzung dieses Projektes war eine ideale Möglichkeit, die Vorlesungsinhalte in der Praxis anzuwenden und zu vertiefen. So konnte der Umgang mit Mikrocontrollern und der Assemblerprogrammierung gelernt und damit verbundene Schwierigkeiten erfahren werden. Der Einsatz der Entwicklungsumgebung mit Simulation bringt zwar ebenfalls Herausforderungen mit sich, ist aber dennoch eine sehr gute Wahl, da so schwerwiegendere Probleme, die bei der Verwendung von echter Hardware auftreten können, vermieden werden können.

Eidesstattliche Versicherung

Wir versichern hiermit, dass wir unser Projekt Systemnahe Programmierung mit dem Thema: *Metronom für Intel 8051* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Karlsruhe, 29. Juni 2018

Felix Starke, Lars Hübner