

PIC Simulator

Technische Dokumentation

1.1 Funktion des Dokuments 4

1.2 Zielgruppe 4

2 Programmübersicht 5

2.1 Menüleiste 5

2.1.1 Datei 5

2.1.2 Ausführen 5

2.1.3 Einstellungen 5

2.1.4 Hilfe 5

2.2 Initialisierung 5

2.3 Startseite 5

2.3.1 Statusleiste 5

2.3.2 Spezialfunktionsregister 5

2.3.3 Frequenz / Laufzeit 6

2.3.4 Steuerung 6

2.3.5 Frequenzgenerator 6

2.4 Auswahl Listing 6

3 Grundsätzliche Arbeitsweise eines Simulators 7

4 Vor- und Nachteile einer Simulation 8

4.1 Vorteile 8

4.2 Nachteile 8

5 Softwarebeschreibung 9

5.1 Struktur / Gliederung 9

5.1.1 Microcontroller 9

5.1.2 Model 10

5.1.3 Parser 10

5.1.4 User Interface 10

5.2 Beschreibung der Funktionen 10

5.2.1 Operations 10

5.2.2 Flags 14

5.2.3 Interrupts 14

5.2.4 TRIS-Register 14

5.2.5 Breakpoints 14

5.2.6 EEPROM 14

5.3 Programmiersprache 14

6 Fazit 15

7 Quellen 16

## Funktion des Dokuments

Dieses Dokument enthält die erforderlichen Informationen, um den PIC-Simulator zu betreiben und zu bedienen. Außerdem beschreibt es die grundsätzliche Arbeitsweise eines Simulators sowie die Vor- und Nachteile einer Simulation.

## Zielgruppe

Diese Bedienungsanleitung enthält die erforderlichen Informationen für den bestimmungsgemäßen Gebrauch der darin beschriebenen Produkte. Sie wendet sich an Herrn Lehmann

# Programmübersicht

Das Programm verfügt über eine Menüleiste und verschiedene Felder zum Bedienen der Software. In den folgenden Abschnitten werden die Menüleiste und die Bedienfelder sowie die zugehörigen Arbeitsschritte beschrieben.

## Menüleiste

Die Software beinhaltet eine Menüleiste mit folgenden Optionen:

### Datei

Öffnen

Ende

### Ausführen

Ausführen

Stop

Single Step

Reset

### Einstellungen

Frequenz

### Hilfe

Ruft die Dokumentation auf.

## Initialisierung

Alle Felder werden zunächst leer initialisiert, erst beim Laden der LST Datei werden alle Felder entsprechend befüllt

## Startseite

### Statusleiste

Die Statusleiste wird unterhalb der Steuerungselemente auf der Startseite angezeigt

### Spezialfunktionsregister

W-Reg

PC

PCLATH

Status

Option

Intcon

### Frequenz / Laufzeit

Zyklen

Laufzeit

Frequenz

Zyklusdauer

### Steuerung

Ausführen

Stop

Step

Reset

PWR Reset

### Frequenzgenerator

Aktiv

Frequenz

## Auswahl Listing

Über die Schaltfläche Datei öffnen können Lst-Dateien eingelesen werden

# Grundsätzliche Arbeitsweise eines Simulators

Unter einer Simulation versteht man im Allgemeinen die Anwendung eines Modells um daraus Einblicke auf das Verhalten eines Objektes der realen Welt zu gewinnen. Eine Simulation ermöglicht den Ersatz von realen Systemen durch beispielsweise einen Rechner. Dadurch wird bei Änderung des Modells für Alternative Lösungswege keine nötige Hardware oder ähnliches gefährdet.

# Vor- und Nachteile einer Simulation

## Vorteile

Genaues Betrachten und Dokumentieren des Objekts durch die Modellierer des Systems führt schon vor Durchführung der Simulation zu einem Verständnis und zu Erkenntnissen. Anpassungen der Bedingungen und experimentieren mit verschiedenen Faktoren ist in einer Simulation mit weniger Risiken und Kosten verbunden als bei einem realen System. Alternative Lösungsansätze können so ohne Gefährdung des Systems getestet werden.

## Nachteile

Eine Simulation bietet nur annähernde Lösungen.

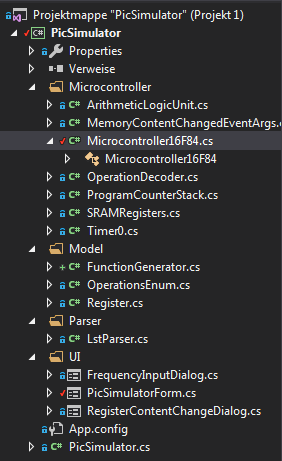
Das Modell des Systems, welches simuliert wird, ist nicht immer optimal, bei einem fehlerhaften Modell sind auch die Lösungsansätze und gewonnenen Erkenntnisse fehlerhaft.

Um dem vorzubeugen ist eine Modellvalidierung notwendig, welche Zeit kostet und bei nicht real existierenden Systemen nicht möglich ist.

# Softwarebeschreibung

## Struktur / Gliederung

Das Programm ist in vier Hauptmodule unterteilt, Microcontroller, Model, Parser und das User Interface.



### Microcontroller

ArithmeticLogicalUnit: Zuständig für die mathematischen und logischen Operationen.

MemoryContentChangedEventArgs: Zuständig für Speicherinhaltsänderungen

Microcontroller16F84: Hauptklasse der Anwendung

OperationDecoder: Decodiert die eingelesenen Befehlscodes und weist diese den entsprechenden Befehlen im OperationsEnum (siehe 5.3.2) zu.

ProgramCounterStack: Implementierung der Stackfunktionen also anlegen des Speichers und Push und Pop

SRAMRegisters: Hier warden die 68 RAM Reister angelegt

Timer0: Bei aktivem Timer werden verschiedene Methoden zum hochzählen und aktualisieren des Timers aufgerufen und je nach Prescaler Wert angepasst.

### Model

FunctionGenerator: Externer Taktgeber der ein Rechtecksignal produziert. Hierbei sind die Frequenz und der anzusteuernde Pin frei wählbar.

FunctionGeneratorEventArguments

OperationsEnum: Auflistung der Befehle geordnet nach Byte-orientierten File Register Operations, Bit-orientierten File Register Operations und Literal and Control Operations.

Register: Register-Klasse

### Parser

LstParser: Trennt die für den Simulator relevanten Informationen (z.B. Die Befehle und die Befehlsnummer) von den für den Simulator irrelevanten Informationen

(z.B. Kommentare) und bereitet diese zur Weiterverarbeitung vor.

### User Interface

FrequencyInputDialog: Dient zum Einstellen der Frequenz

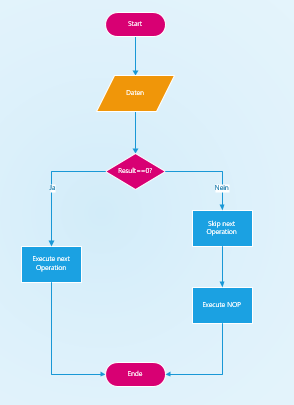
PicSimulatorForm: Das User Interface das der Endbenutzer sieht.

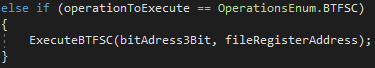
RegisterContentChangeDialog: Zum manuellen ändern des Registerinhalts.

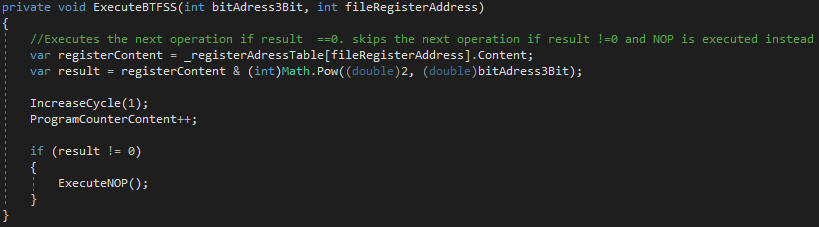
## Beschreibung der Funktionen

### Operations

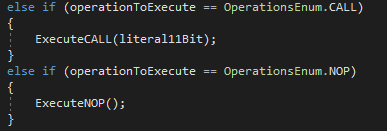
BTFSS

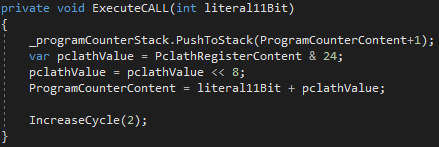




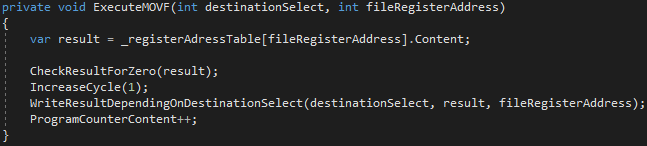
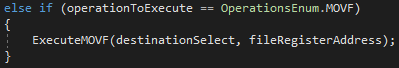


CALL

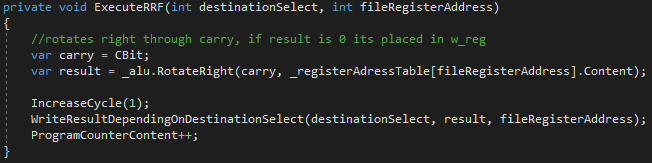
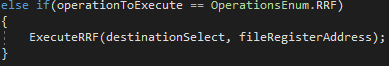




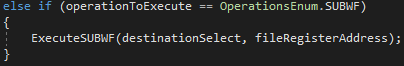
MOVF

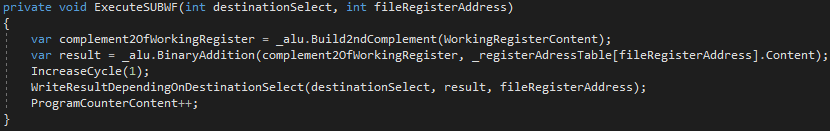


RRF

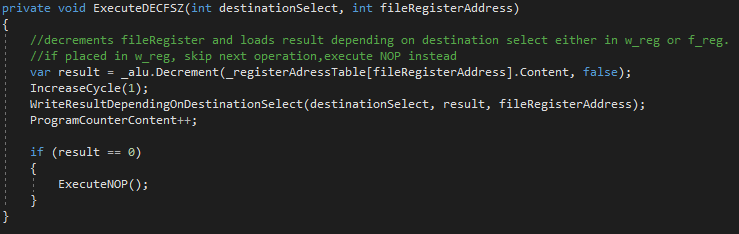
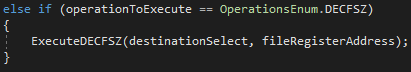


SUBWF

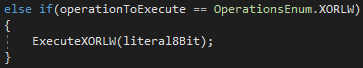


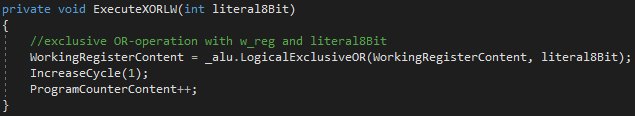


DECFSZ

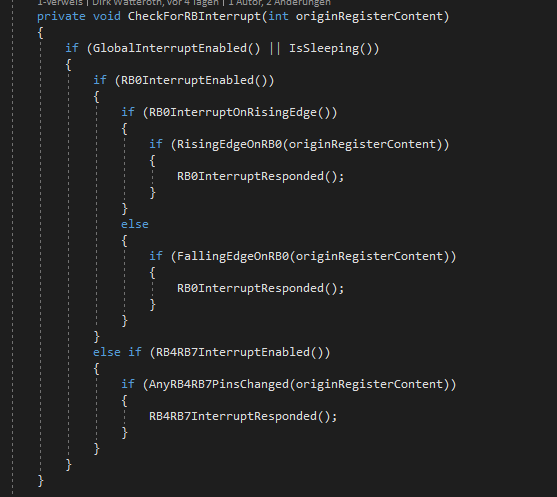


XORLW





### Interrupts



## Programmiersprache

Als Programmiersprache wurde c# gewählt, da hier die meisten Vorkenntnisse vorhanden waren. Außerdem ist das User Interface leicht zu realisieren. Auch die Hardwareansteuerung ist durch vorhandene Bibliotheken zum Öffnen von COM-Ports mit weniger Aufwand verbunden als in anderen Programmiersprachen.

# Fazit

Der Simulator funktioniert ordnungsgemäß und die geforderten Funktionen wurden implementiert. Die Wahl der Programmiersprache C# stellt keine Probleme dar. Die Vorgehensweise, Funktionen nach und nach zu implementieren um das nächste Listing funktionsfähig zu machen, ist empfehlenswert. Die Codequalität wurde durch objektorientierte Programmierung und durch zahlreiche Kommentare hochgehalten. Die Struktur des Quellcodes ist ebenfalls sehr übersichtlich.

# Quellen

Um den Simulator umzusetzen wurde nur das von Herrn Lehmann gelieferte Datenblatt „PIC16F8x“ benutzt.