

# PicSimulator Dokumentation

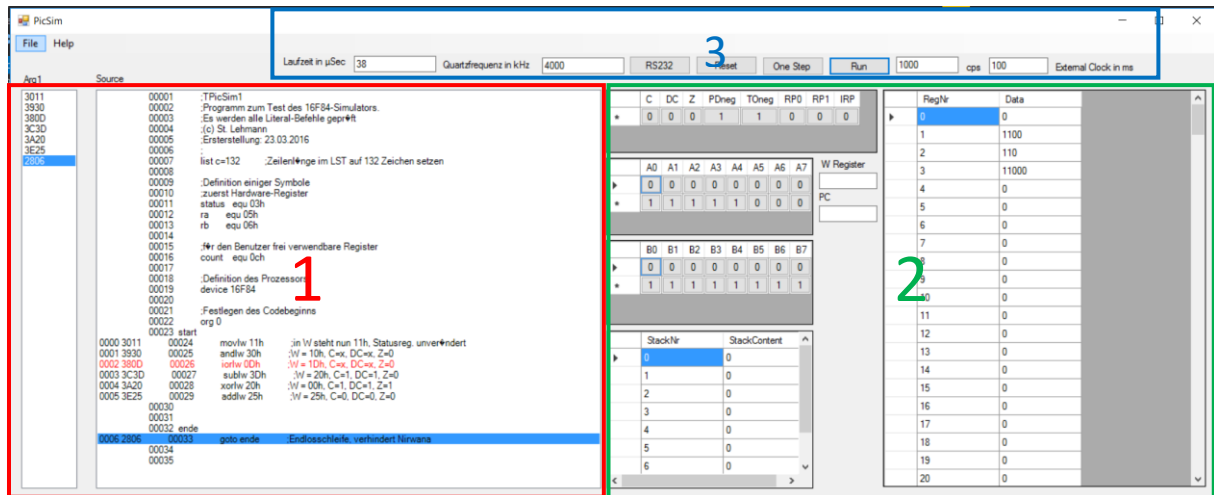
von Damian P. und Emil B.

## Inhaltsverzeichnis

Simulator .....	2
Was ist ein Simulator.....	<b>Fehler! Textmarke nicht definiert.</b>
Vorteile eines Simulators .....	<b>Fehler! Textmarke nicht definiert.</b>
Benutzeroberfläche .....	2
Aufbau des Programmes .....	4
Funktionen.....	4
Code einlesen .....	4
Befehl bearbeiten.....	4
Breakpoints.....	4
Interrupts.....	4
RS232.....	4
Befehle.....	4

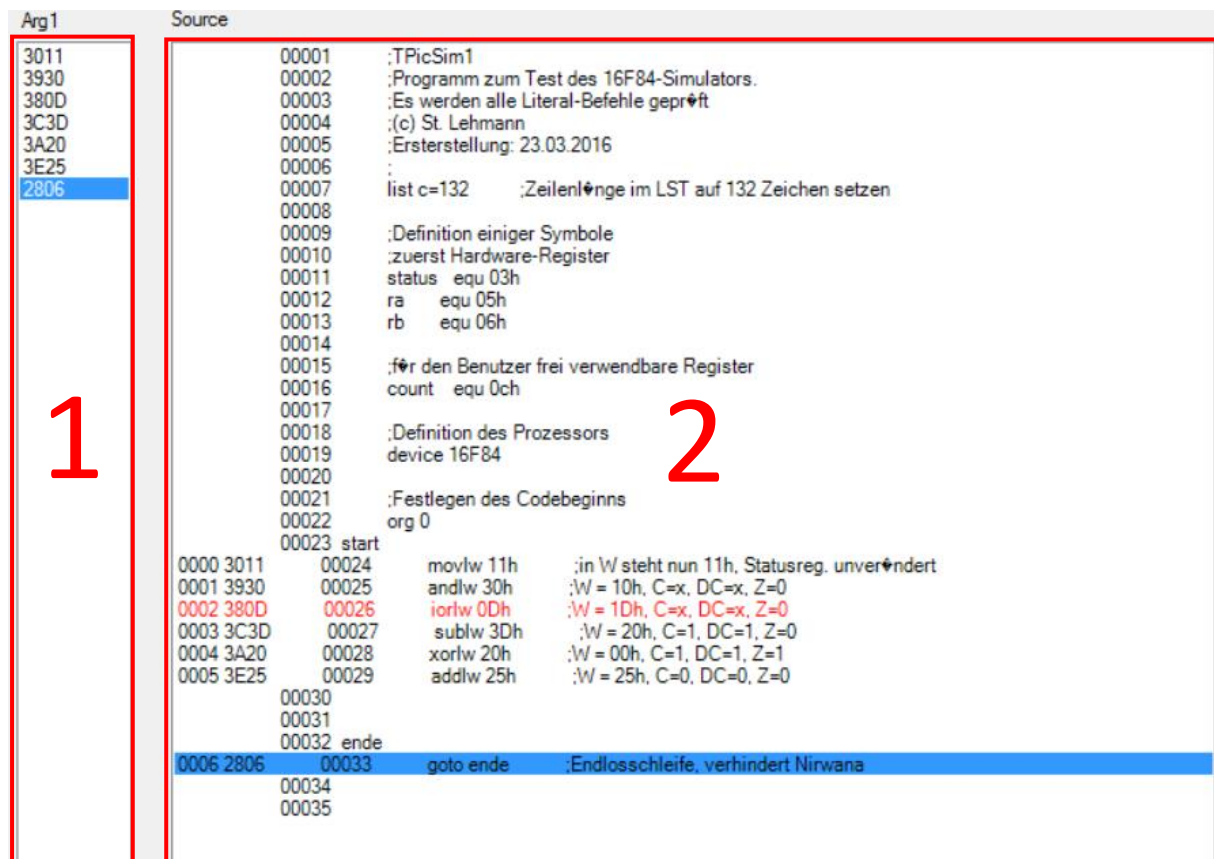
# Simulator

## Benutzeroberfläche



Die Benutzeroberfläche ist in 3 Bereiche unterteilt auf welche im Folgenden genauer eingegangen werden soll.

Der **erste Bereich** beinhaltet die Programminformationen. In 1 kann man die verschiedenen Befehle in ihrer Hexadezimalen Darstellung erkennen und in 2 ist der erweiterte Programmcode dargestellt. Außerdem erkennt man durch die blaue Markierung welcher Befehl der nächste ist welcher ausgeführt wird. Durch einen Doppelklick in 2 auf eine beliebige Zeile wird dort ein Breakpoint gesetzt, was man an der roten Einfärbung erkennen kann.



Im **zweiten Bereich** werden die Register und der Stack dargestellt. Im ersten Feld wird das Statusregister angezeigt und in den Feldern 1 und 2 werden Port A und B und darunter die Passenden Tris Werte angezeigt. Die Werte von Port A und B lassen sich mit einem Klick auf diese ändern. In Feld 4 wird der Stack angezeigt mit der NR und dem passenden Inhalt. In Feld 5 werden die Werte vom W Register und des PCs dargestellt und in Feld 6 sind alle Register in Binärschreibweise vorhanden.

The screenshot displays the second area of the simulator, which is divided into several sections for monitoring hardware components:

- Status Register (Feld 1):** A row of buttons for C, DC, Z, PDneg, TOneg, RP0, RP1, and IRP. The values are 0, 0, 0, 1, 1, 0, 0, 0 respectively.
- Port A and B (Feld 2):** Two rows of buttons for A0-A7 and B0-B7. The values for A are 0, 0, 0, 0, 0, 0, 0, 0 and for B are 1, 1, 1, 1, 1, 1, 1, 1.
- Stack (Feld 4):** A table with columns 'StackNr' and 'StackContent'. The stack is empty, with all values being 0.
- W Register and PC (Feld 5):** Two input fields for the W Register and PC, both currently empty.
- Registers (Feld 6):** A large table with columns 'RegNr' and 'Data'. It lists registers 0 through 20. Register 0 contains 0, 1 contains 0, 2 contains 110, 3 contains 11000, and all other registers contain 0.

Im **dritten Bereich** sind die Interaktionsmöglichkeiten dargestellt welche für den Benutzer vorhanden sind. Im Feld 1 kann man die Quarzfrequenz festlegen und dort wird die Laufzeit basierend auf der Quarzfrequenz angezeigt. Feld 2 beinhaltet verschiedenen Buttons um ein Programm zu starten, es Schritt für Schritt durch zu gehen, es zu resetten oder die Schnittstelle RS232 anzusprechen. Im Feld 3 kann man die Geschwindigkeit einstellen mit welcher der Simulator Befehle ausführen soll und in Feld 4 kann man die Externe Clock einstellen.

The screenshot displays the third area of the simulator, which contains controls for running the simulation:

- Laufzeit in µSec (Feld 1):** A text input field with the value 10.
- Quarzfrequenz in kHz (Feld 2):** A text input field with the value 4000.
- Buttons (Feld 3):** A row of buttons: RS232, Reset, One Step, and Run. The 'One Step' button is highlighted.
- Speed (Feld 4):** A text input field with the value 1000, followed by the unit 'cps'.
- External Clock (Feld 5):** A text input field with the value 100, followed by the unit 'ms'.

# Aufbau des Programmes

Das Programm ist in verschiedenen Bestandteile unterteilt. Auf die genauere Funktionsweise soll im Anschluss eingegangen werden.

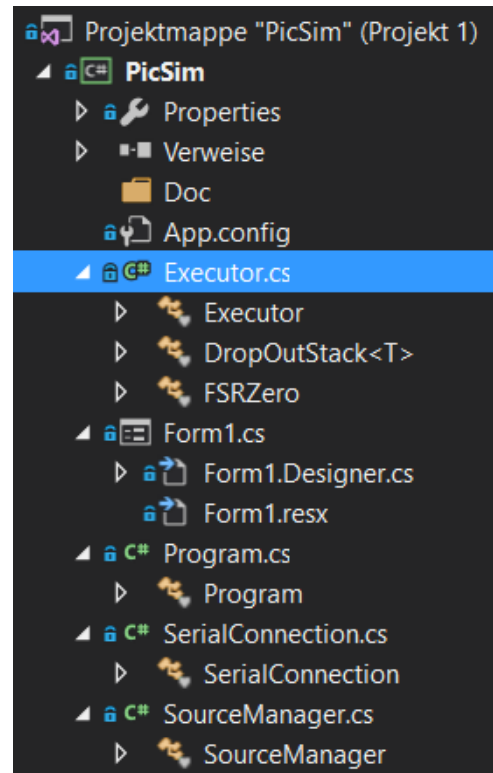
So ist die Form1 und Program Klasse dafür Zuständig die Benutzeroberfläche dar zu stellen.

Die SourceManager Klasse ist zuständig den Programmcode eines PIC Programmes ein zu lesen und diesen in ein Array um zu wandeln.

Die Executor Klasse wiederum beinhaltet die Logik des PIC Simulators, da hier alle Befehle und Routinen abgebildet sind.

Und die SerialConnection Klasse ist zuständig für die Verbindung mit der RS232 Schnittstelle und das senden/empfangen der Daten über diese.

Das Programm wurde in C# in der Entwicklungsumgebung VisualStudio erstellt.



## Funktionen

### Code einlesen

### Befehl bearbeiten

### Breakpoints

### Interrupts

### RS232

## Befehle

